สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตรวจสอบลายนิ้วมือสำหรับระบบรักษาความปลอดภัย

Fingerprint checking tool for the security system

โดย

นายศุภโชค วีระชวนะศักดิ์ 47010880

นายสุรชาติ พิชยศ 47010892

อาจารย์ที่ปรึกษา

รศ. เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร.สุวิพล สิทธิชีวภาค

ปพ
ส8339
2550

เลขหมู่...........

เลขทะเบียน 82912

วัน,เดือน,ปี 29 ก.ค. 2551

b 11957037

i

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตรวจสอบลายนิ้วมือสำหรับระบบรักษาความปลอดภัย

The fingerprint checking tool for the security system

ผู้จัดทำ

1. นายสุภโชค วีระชวนะศักดิ์ 47010880
2. นายสุรชาติ พิชยศ 47010892

.... อาจารย์ที่ปรึกษา

(รศ.เกรียงไกร วงศ์โรจนภรณ์)

.... อาจารย์ที่ปรึกษา

(รศ.ดร.สุวิพล สิทธิชีวภาค)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานที่ 502434 **เครื่องตรวจสอบลายนิ้วมือสำหรับระบบรักษาความปลอดภัย**

Fingerprint checking tool for the security system

โดย นายสุภโชค วีระชวนะศักดิ์ 47010880

นายสุรชาติ พิชยศ 47010892

อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิทธิชีวภาค

รศ.เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอการรักษาความปลอดภัยด้วยการนำเครื่องตรวจสอบจากลายนิ้วมือของแต่ละบุคคลมาประยุกต์ใช้งานที่สามารถทำงานได้โดยลำพังคือไม่ต้องควบคุมด้วยคอมพิวเตอร์ โดยใช้เครื่องแสกนลายนิ้วมือและไมโครคอนโทรลเลอร์ทำการประมวลผลเปรียบเทียบลายนิ้วมือและควบคุมอุปกรณ์ในวงจร และยังมีการป้อนข้อมูลโดยใช้คีย์แพด (Keypad) และแสดงผลผ่านจอแอลซีดี (LCD) และนำไปประยุกต์ใช้ในการป้องกันการโจรกรรม

Abstract

This project presents safety preservation by checking the fingerprint of each personal and can be applied working alone without controlled by the computer. The fingerprint scanner and microcontroller process the data compared with the fingerprint and control the equipment in the circuit. The data is entered by using a keypad and display on the LCD and then is applied in robbery prevention.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

1.บทนำ

1.1 รายละเอียดโดยย่อของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 1

1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงาน 2

2.ทฤษฎีหรือหลักการ

2.1 การตรวจสอบลายนิ้วมือ 3

2.1.1 ไบโอเมตริกซ์ 3

2.1.1.1 เพื่อบ่งชี้ความเป็นตัวจริง (Verification) 4

2.1.1.2 เพื่อระบุว่าบุคคลนั้นๆเป็นใคร (Identification) 4

2.1.1.3 การลงทะเบียน (Enrollment) 4

2.1.2 หลักพื้นฐานในการพัฒนาเทคโนโลยีเพื่อระบุตัวบุคคล 5

2.1.3 ความแม่นยำของไอดีไบโอเมตริกซ์ 5

2.1.3.1 False Rejection Rate (FRR) 5

2.1.3.2 False Acceptance Rate (FAR) 5

2.1.3.3 Equal Error Rate (ERR) 5

2.1.4 ระบบเครื่องสแกนลายนิ้วมือ 6

2.1.5 ความรู้เบื้องต้นเกี่ยวกับลายนิ้วมือ 6

2.1.5.1 แบบพับหวาย (Loop) 8

2.1.5.2 แบบก้นหอย (Whorl) 10

2.1.5.3 แบบโค้ง (Arch) 10

2.1.5.4 แบบผสม (Mixed figure / Composite) 11

2.2 BioPrima OEM2000P-44B0 Fingerprint Identification Board 12

2.2.1 โครงสร้าง 12

2.2.2 คุณสมบัติของเครื่องสแกนลายนิ้วมือ 13

2.2.2.1 ลักษณะเด่น 13

2.2.2.2 คุณสมบัติทางเทคนิค 13

2.2.2.3 โครงสร้าง 14

2.2.2.4 ขนาดของ Optical sensor 14

2.2.2.5 RS-232C 15

2.2.2.6 DIPSW 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.3 โปรโตคอล (Protocol) ในการสื่อสาร 16

2.2.3.1 โหมดคำสั่ง 16

2.2.3.2 โหมดรับ 17

2.2.3.3 โหมดส่ง 18

2.2.4 โครงสร้าง แพ็คเกต 19

2.2.4.1 Command Packet (คำสั่ง) 19

2.2.4.2 Command Response Packet (การตอบสนองคำสั่ง) 20

2.2.4.3 Packet Result (ผลลัพธ์) 20

2.2.4.4 Block Data Packet (บล็อกข้อมูล) 21

2.3 PIC (Peripheral Interface Controller) 22

2.3.1 PORTA ใน PIC16F87X 23

2.3.2 PORTB ใน PIC16F87X 25

2.3.3 PORTC ใน PIC16F87X 27

2.3.4 PORTD และ PORTE ใน PIC16F87X 28

2.4 พอร์ตอนุกรมมาตรฐาน RS-232 30

2.4.1 RS-232 serial connecter pin assignment 30

2.4.2 รูปแบบของการเชื่อมต่อ 33

2.4.3 ลักษณะสัญญาณของอนุกรม RS-232 36

3.การคำนวณและการสร้าง

3.1องค์ประกอบโดยรวมของระบบ 38

3.2 หลักการทำงานของฮาร์แวร์ 38

3.2.1 ไมโครคอลโทรลเลอร์ 39

3.2.3 วงจร DS275 39

3.2.4 LCD 40

3.2.5 Keypad 41

3.3 หลักการทำงานของซอร์ฟแวร์ 45

3.3.1 การลงทะเบียน 46

3.3.2 การเข้าใช้งานระบบ 46

3.3.3 การลบข้อมูลผู้ใช้จากระบบ 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4. การทดลองและผลการทดลอง

4.1 การตรวจสอบการทำงานของบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์ 48

4.2 การทำการเขียนโปรแกรมไมโครคอนโทรลเลอร์ PIC16F877 50

5. บทวิจารณ์และบทสรุป

5.1 ปัญหาที่พบในการทำโครงงานนี้ 57

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 ลายนิ้วมือ 6

รูปที่ 2.2 ส่วนประกอบของลายนิ้วมือ 8

รูปที่ 2.3 แสดงลักษณะของกลุ่มพับหวายเอียงขวา 8

รูปที่ 2.4 แสดงลักษณะของกลุ่มพับหวายเอียงซ้าย 8

รูปที่ 2.5 แสดงลักษณะรูปแบบของกลุ่มมัดหวายคู่ 9

รูปที่ 2.6 แสดงลักษณะของ Right pocket loop 9

รูปที่ 2.7 แสดงลักษณะของ Left pocket loop 9

รูปที่ 2.8 แสดงลักษณะรูปแบบของกลุ่มก้นหอย 10

รูปที่ 2.9 แสดงลักษณะรูปแบบของกลุ่มเส้นโค้งราบ(Arch) 10

รูปที่ 2.10 แสดงลักษณะรูปแบบเส้นโค้งกระโจม(Tented Arch) 11

รูปที่ 2.11 แสดงลักษณะรูปแบบของกลุ่ม Mixed figure 11

รูปที่ 2.12 โมดูล OEM2000P-44B0 12

รูปที่ 2.13 Optical sensor 12

รูปที่ 2.14 โครงสร้าง 14

รูปที่ 2.15 ขนาดของ Optical sensor 14

รูปที่ 2.16 DIPSW 15

รูปที่ 2.17 PIC (Peripheral Interface Controller) 22

รูปที่ 2.18 บล็อกไดอะแกรม ขา RA3:RA0 และ RA 24

รูปที่ 2.19 บล็อกไดอะแกรม ขา RA4/TOCKI 24

รูปที่ 2.20 บล็อกไดอะแกรม ขา RB3:RB0 25

รูปที่ 2.21 บล็อกไดอะแกรม ขา RB7:RB4 26

รูปที่ 2.22 บล็อกไดอะแกรม PORTC RC<0:2> RC<5:7> 27

รูปที่ 2.23 บล็อกไดอะแกรม PORTC RC <3:4> 28

รูปที่ 2.24 บล็อกไดอะแกรม PORTD (ในโหมด พอร์ท I/O) 29

รูปที่ 2.25 บล็อกไดอะแกรม PORTE (ในโหมด พอร์ท I/O) 29

รูปที่ 2.26 RS-232 คอนเน็คเตอร์แบบ DB9 30

รูปที่ 2.27 RS-232 คอนเน็คเตอร์แบบ DB25 31

รูปที่ 2.28 RS-232 คอนเน็คเตอร์แบบ DCE Modify modular jack 31

รูปที่ 2.29 RS-232 คอนเน็คเตอร์แบบ DB9 ตัวผู้ที่อยู่ด้านหลังคอมพิวเตอร์ 32

รูปที่ 2.30 ลักษณะการเชื่อมต่อแบบเต็มรูปแบบ 33

รูปที่ 2.31 ลักษณะการเชื่อมต่อแบบ Partial Handshaking 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.32 ลักษณะการเชื่อมต่อแบบ Loop Back Handshaking 34

รูปที่ 2.33 ลักษณะการเชื่อมต่อแบบง่าย 35

รูปที่ 2.34 ลักษณะรูปสัญญาณของอนุกรม RS-232 36

รูปที่ 2.35 ลักษณะสัญญาณของข้อมูลที่มีความสำพันธ์กับแรงดันไฟฟ้า 37

รูปที่ 3.1 ภาพโดยรวมของระบบ 39

รูปที่ 3.2 ไมโครคอลโทรลเลอร์ (PIC16F877) 39

รูปที่ 3.3 การต่อ DS275 40

รูปที่ 3.4 วงจร LCD ขนาด 2x8 ที่ต่อใช้งาน 40

รูปที่ 3.5 Keypad แบบ 4x3 ที่ต่อใช้งาน 41

รูปที่ 3.6 วงจรที่ออกแบบที่ใช้ในโครงงานนี้ 42

รูปที่ 3.7 Flowchart การทำงานของวงจร 43

รูปที่ 3.8 Flowchart การทำงานของวงจร (ต่อ) 44

รูปที่ 4.1 วัดสัญญาณที่ออกจากบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์ 48

รูปที่ 4.2 วัดสัญญาณที่ออกมาจาก DS275 49

รูปที่ 4.3 วัดสัญญาณเปรียบเทียบระหว่างสัญญาณที่ออกจาก DS275 และ บอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์ 49

รูปที่ 4.4 หน้าจอแอลซีดีแสดงข้อความ “START” 50

รูปที่ 4.5 หน้าจอแอลซีดีแสดงข้อความ “Press Finger” 50

รูปที่ 4.6 หน้าจอแอลซีดีแสดงข้อความ “No Data” 50

รูปที่ 4.7 หน้าจอแอลซีดีแสดงข้อความ “Select Menu” 51

รูปที่ 4.8 หน้าจอแอลซีดีแสดงข้อความ “Master Register” 51

รูปที่ 4.9 หน้าจอแอลซีดีแสดงข้อความ “Input ID Number” 51

รูปที่ 4.10 หน้าจอแอลซีดีแสดงข้อความ “001” 52

รูปที่ 4.11 หน้าจอแอลซีดีแสดงข้อความ “002” 52

รูปที่ 4.12 หน้าจอแอลซีดีแสดงข้อความ “003” 52

รูปที่ 4.13 หน้าจอแอลซีดีแสดงข้อความ “Press Finger 1” 53

รูปที่ 4.14 หน้าจอแอลซีดีแสดงข้อความ “Pres Finger 2” 53

รูปที่ 4.15 หน้าจอแอลซีดีแสดงข้อความ “Fail” 53

รูปที่ 4.16 หน้าจอแอลซีดีแสดงข้อความ “Register OK” 54

รูปที่ 4.17 หน้าจอแอลซีดีแสดงข้อความ“No Finger” 54

รูปที่ 4.18 หน้าจอแอลซีดีแสดงข้อความ“Delete Member” 54

รูปที่ 4.19 หน้าจอแอลซีดีแสดงข้อความ “Input ID Number” 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.20 หน้าจอแอลซีดีแสดงข้อความ“Confirm Delete” 55
รูปที่ 4.21 หน้าจอแอลซีดีแสดงข้อความ “Delete OK” 55
รูปที่ 4.22 หน้าจอแอลซีดีแสดงข้อความ “Fail” 56
รูปที่ 4.23 หน้าจอแอลซีดีแสดงข้อความ “Delete All” 56
รูปที่ 4.24 หน้าจอแอลซีดีแสดงข้อความ “Delete All OK” 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 คุณสมบัติทางเทคนิค	13
ตารางที่ 2.2 ไอซี RS-232C	15
ตารางที่ 2.3 Command packet	19
ตารางที่ 2.4 Command Response Packet	20
ตารางที่ 2.5 Packet Result	21
ตารางที่ 2.6 Block Data Packet	21
ตารางที่ 2.7 รายละเอียดขาสัญญาณของพอร์ตอนุกรม RS-232	32
ตารางที่ 2.8 หน้าที่การทำงานของการเชื่อมต่อแบบเต็มรูปแบบ	33
ตารางที่ 2.9 หน้าที่การทำงานของการเชื่อมต่อแบบ Partial Handshaking	34
ตารางที่ 2.10 หน้าที่การทำงานของการเชื่อมต่อแบบ Loop Back Handshaking	35
ตารางที่ 2.11 หน้าที่การทำงานของการเชื่อมต่อแบบง่าย	35
ตารางที่ 2.12 ความสัมพันธ์ของ Logic และระดับแรงดัน	36
ตารางที่ 3.1 เปรียบเทียบแรงดันไฟฟ้าระหว่างสัญญาณ TTL กับ RS 232	40

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้เป็นการนำเสนอการรักษาความปลอดภัยด้วยการนำเครื่องตรวจสอบจากลายนิ้วมือของแต่ละบุคคลมาประยุกต์ใช้งานที่สามารถทำงานได้โดยลำพังคือไม่ต้องควบคุมด้วยคอมพิวเตอร์ โดยใช้เครื่องแสกนลายนิ้วมือและไมโครคอนโทรลเลอร์ทำการประมวลผลเปรียบเทียบลายนิ้วมือและควบคุมอุปกรณ์ในวงจร และยังมีการป้อนข้อมูลโดยใช้คีย์แพด (Keypad) และแสดงผลผ่านจอแอลซีดี (LCD) และนำไปประยุกต์ใช้ในการป้องกันการโจรกรรม

1.1 รายละเอียดโดยย่อของโครงงาน

โครงงานนี้เป็นโครงงานเครื่องรักษาความปลอดภัยโดยการสแกนลายนิ้วมือที่สามารถทำงานได้โดยลำพัง โดยใช้ OEM2000P-4460 เป็นเซ็นเซอร์ในการรับภาพลายนิ้วมือและไมโครคอนโทรลเลอร์ PIC (PIC16F877-20/P) ทำการประมวลผลเปรียบเทียบลายนิ้วมือและควบคุมอุปกรณ์ในวงจร และยังมีการป้อนรหัสผ่านคีย์แพด (Keypad) และแสดงผลผ่านจอแอลซีดี (LCD) โดยมีสัญญาณเสียงและแอลอีดี (LED) แสดงสถานะ การทำงานของอุปกรณ์ ซึ่งได้เขียนโปรแกรมควบคุมการทำงานทั้งหมดด้วยภาษาซี และได้นำไปประยุกต์ใช้ในการป้องกันการโจรกรรม

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างเครื่องแสกนลายนิ้วมือที่สามารถทำงานได้จริงโดยลำพัง
2. เพื่อนำเครื่องแสกนลายนิ้วมือไปใช้ในระบบการรักษาความปลอดภัย
3. เพื่อพัฒนาเทคโนโลยีในระบบรักษาความปลอดภัย

1.3 ขอบเขตของโครงงาน

สร้างเครื่องรักษาความปลอดภัยโดยการแสกนลายนิ้วมือที่สามารถทำงานได้โดยลำพัง โดยมีตั้งแต่ ขั้นตอนการแสกนลายนิ้วมือ ขั้นตอนการประมวลผล ขั้นตอนการเปรียบเทียบลายนิ้วมือ และยังมีการป้อนข้อมูลโดยใช้คีย์แพด (Keypad) และแสดงผลผ่านจอแอลซีดี (LCD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงานนี้

เนื่องจากในปัจจุบันการรักษาความปลอดภัยเป็นเรื่องสำคัญในชีวิตประจำวัน และลายนิ้วมือก็เป็นที่นิยมนำมาใช้ในการรักษาความปลอดภัย เพราะสะดวกในการตรวจสอบและมีความถูกต้องแม่นยำ จึงหวังว่าเครื่องรักษาความปลอดภัยโดยการแสกนลายนิ้วมือที่สร้างขึ้นนี้จะสามารถนำมาประยุกต์ใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

หลังจากที่ทราบถึงรายละเอียดโดยย่อของโครงงาน, วัตถุประสงค์, ขอบเขต, และประโยชน์ที่คาดว่าจะได้รับจากโครงงานนี้ ในบทที่ 1 แล้ว ต่อไปในบทที่ 2 นี้ จะเป็นทฤษฎีหรือหลักการที่เราได้ทำการศึกษาค้นคว้าเพื่อนำมาใช้ในการทำโครงงานนี้

2.1 การตรวจสอบลายนิ้วมือ

2.1.1 ไบโอเมตริกซ์ (Biometrics)

ไบโอเมตริกซ์ คือ การใช้ลักษณะทางกายภาพ หรือลักษณะทางพฤติกรรม ที่เป็นลักษณะเฉพาะของแต่ละคน ในการระบุตัวบุคคลโดยอัตโนมัติลักษณะทางกายภาพที่ใช้เป็นเกณฑ์ได้เช่น ลายนิ้วมือ, ใบหน้า, มือ, นิ้ว, หู, เรตินา และ Iris ภายในดวงตา เป็นต้น ส่วนลักษณะทางพฤติกรรมที่เป็นลักษณะเฉพาะของแต่ละบุคคล เช่น เสียงพูด, การพิมพ์, การเดิน เป็นต้น โดยใช้สิ่งที่ใช้เป็นเกณฑ์ได้นั้น คือ จะต้องเป็นสิ่งที่มีการเปลี่ยนแปลงน้อยตามกาลเวลา และที่สำคัญจะต้องเป็นสิ่งที่เฉพาะไม่เหมือนกันในแต่ละบุคคล

คำว่า Biometricsนี้มาจากภาษากรีกสองคำคือ Bio (Life) ซึ่งหมายถึงชีวิตหรือความมีชีวิต กับคำว่า Metro (Measure) หมายถึงการวัดหรือมาตรวัด ซึ่งเมื่อรวมกันกับไบโอเมตริกซ์ จึงหมายถึงวิธีการหรือเทคนิคในการตรวจสอบแยกแยะสิ่งมีชีวิตโดยวัดจากคุณลักษณะของสิ่งมีชีวิตนั้นๆ คนเราสามารถที่จะแยกแยะเพื่อนออกจากคู่อริได้โดยการพิจารณาจากใบหน้า แต่ในกรณีของแฝดเหมือน (Identical Twins) สมองของคนเราที่ว่าแน่นักก็อาจจะต้องยอมแพ้เหมือนกัน คำจำกัดความที่ใช้โดยทั่วไปสำหรับไบโอเมตริกซ์ก็คือ เทคนิคอัตโนมัติต่างๆในการตรวจวัดคุณลักษณะทางกายภาพ (Physical Characteristics) พฤติกรรม (Behavior) ตลอดจนร่องรอยอื่นๆ ในชีวิตประจำวัน (Personal Traits) ของบุคคลที่มีชีวิต แล้วนำมาเปรียบเทียบกับคุณลักษณะนั้นๆ ที่ได้มีการบันทึกไว้ก่อนหน้านี้ในฐานข้อมูล เพื่อวัตถุประสงค์ในการแยกแยะ (Recognizing) บุคคลนั้นจากบุคคลอื่น สำหรับไบโอเมตริกซ์แล้วคุณลักษณะต่างๆ ไม่ว่าจะเป็นคุณลักษณะทางกายภาพ ทางพฤติกรรม หรือร่องรอย อื่นๆ ในชีวิตประจำวันของแต่ละบุคคลจะต้องสามารถที่จะวัดในเชิงปริมาณได้ คุณลักษณะทางกายภาพนั้นส่วนใหญ่จะไม่แปรเปลี่ยนไปตามกาลเวลา ในขณะที่คุณลักษณะทางพฤติกรรมหรือร่องรอยในชีวิตประจำวันอาจมีการเปลี่ยนแปลงไปตามกาลเวลาตามการเรียนรู้ของเจ้าของได้ ด้วยเหตุนี้ไบโอเมตริกซ์ที่ใช้บอกคุณลักษณะทางกายภาพมาเป็นตัววัดจึงได้รับความเชื่อถือมากกว่า ตัวอย่างคุณลักษณะทางกายภาพที่ได้นำมาใช้เป็นไอดีไบโอเมตริกซ์อย่างแพร่หลาย ได้แก่ ลายนิ้วมือ ม่านตา เส้นเลือดที่ผนังลูกตาดำ รูปพรรณสัณฐานของมือ โครงสร้างใบหน้า นอกจากนี้ยังมีคุณลักษณะทางกายภาพอื่นๆอีกที่ยังอยู่ระหว่างการศึกษาความเป็นไปได้ในการนำมาใช้เป็นไอดี และพัฒนามาเป็นทางเลือกใหม่ให้กับเทคโนโลยีไบโอเมตริกซ์ เช่น ลักษณะรอยย่นของข้อนิ้ว (Knuckle Creases) คลื่นสมอง (Acoustic Head Resonance) หรือแม้แต่กลิ่นตัว (Body Odors) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลไบโอเมตริกซ์จะต้องเก็บบันทึกจากบุคคลที่ยังมีชีวิตอยู่เท่านั้น โดยอุปกรณ์เก็บข้อมูลไบโอเมตริกซ์จะมีระบบตรวจเช็คความมีชีวิตของบุคคลด้วย ดังนั้นการใช้เทปบันทึกเสียงแทนการพูดด้วยเสียงจริงจะไม่สามารถหลอกลวงเครื่องแยกแยะเสียงพูดของคนตามหลักการไบโอเมตริกซ์ได้ หรือฉากหนึ่งในภาพยนตร์เรื่อง Demolition Man ที่มีการควักลูกตาผู้คุมนักโทษจิ้มติดปลายดินสอตบตาเครื่องอ่านเรตินา เพื่อหลบหนีออกจากเรือนจำนั้น ได้รับการยืนยันจากบริษัทผู้ผลิตอุปกรณ์นี้ (ซึ่งเป็นสปอนเซอร์ให้กับภาพยนตร์เรื่องดังกล่าวด้วย) ว่าจริงๆแล้วการกระทำเช่นนี้ไม่อาจหลอกลวงอุปกรณ์นี้ได้ เนื่องจากเครื่องอ่านเรตินามีคุณสมบัติในการตรวจเช็คความมีชีวิตของเส้นเลือดดำที่อยู่ในผนังด้านในของลูกตาได้ คุณสมบัติข้อนี้เองทำให้เทคโนโลยีไบโอเมตริกซ์ แตกต่างออกไปจากศาสตร์ทางด้านการชันสูตรศพ (Forensic Science) ไบโอเมตริกซ์ถูกนำมาใช้เพื่อวัตถุประสงค์ในการทำความรู้จัก หรือแยกแยะตัวบุคคล จากวัตถุประสงค์นี้เองสามารถแบ่งการใช้งานออกได้เป็นสองโอกาสคือ

2.1.1.1 เพื่อบ่งชี้ความเป็นตัวจริง (Verification) การใช้งานในกรณีนี้จะมีการเปรียบเทียบข้อมูลไบโอเมตริกซ์ที่เก็บได้ใหม่ (ณ จุดใช้งาน) กับข้อมูลของบุคคลนั้นที่ได้เคยลงทะเบียนไว้ เพื่อพิสูจน์ว่าบุคคลที่มากล่าวอ้างนี้เป็นตัวจริง ซึ่งถือว่าเป็นการเปรียบเทียบแบบ One-to-One

2.1.1.2 เพื่อระบุว่าบุคคลนั้นๆเป็นใคร (Identification) กรณีต้องเทียบข้อมูลไบโอเมตริกซ์ที่เก็บใหม่ ณ จุดใช้งาน กับข้อมูลไบโอเมตริกซ์ทั้งหมดที่มีอยู่ในฐานข้อมูล เพื่อพิสูจน์ว่าบุคคลที่เป็นเจ้าของข้อมูลที่เก็บ ณ จุดใช้งานนั้นเป็นใครในฐานข้อมูล เป็นการเปรียบเทียบแบบ One-to-Many

2.1.1.3 การลงทะเบียน (Enrollment) คือ ลักษณะการทำงานที่ทำการสแกนลายนิ้วมือ เพื่อบันทึกเป็นฐานข้อมูลลงไปในหน่วยความจำภายใน

เทคโนโลยีไบโอเมตริกซ์ จะใช้เทคนิคการทำงานแบบ อัตโนมัติ ซึ่งต้องมีขั้นตอนมาตรฐานดังนี้

1. เก็บตัวอย่างคุณลักษณะที่ต้องการวัด เช่น สแกนลายนิ้วมือออกมาเป็นภาพถ่ายลายนิ้วมือ
2. เก็บข้อมูลไบโอเมตริกซ์จากตัวอย่างที่สแกนได้ในข้อ 1 เช่น เก็บข้อมูลเชิงปริมาณจากภาพถ่ายลายนิ้วมือด้วยการคำนวณโดยใช้อัลกอริธึมเฉพาะ
3. เปรียบเทียบข้อมูลเชิงปริมาณที่วัดได้จากข้อ 2 กับข้อมูลที่ได้บันทึกเอาไว้ก่อนหน้านี้ ซึ่งอาจบันทึกไว้ในฐานข้อมูลกลาง หรือบันทึกไว้บนบัตรอัจฉริยะ
4. พิจารณาผลการเปรียบเทียบว่าถูกต้องตรงกันหรือไม่
5. ตัดสินว่าบุคคลนี้เป็นใคร (Identification) หรือเป็นตัวจริงตามที่มีการกล่าวอ้าง (Verification) หรือไม่

ก่อนที่ระบบจะสามารถระบุบุคคล และบ่งชี้ความเป็นตัวจริงของแต่ละบุคคลนั้น ระบบจะทำการร้องขอบางสิ่งบางอย่างเพื่อจะเปรียบเทียบกับตัวมัน ดังนั้นประวัติและข้อมูลส่วนตัวทางด้านชีววิทยาจะถูกเก็บไว้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 หลักพื้นฐานในการพัฒนาเทคโนโลยีเพื่อระบุตัวบุคคล

พัฒนาการทางเทคโนโลยีในการระบุตัวบุคคลนั้นพัฒนามาจากหลักพื้นฐาน 3 หลัก ได้แก่

1) สิ่งที่รู้ (Something You Know) หมายถึง วิธีการยืนยันตัวบุคคลซึ่งเฉพาะบุคคลที่ตรวจสอบและบุคคลที่ถูกตรวจสอบเท่านั้นที่รู้เกี่ยวกับสิ่งที่ใช้ตรวจสอบ เช่น การใช้รหัสผ่าน (Password) รหัสประจำตัว (Personal Identification Number: PIN)

2) สิ่งมี (Something You Have) หมายถึง การที่บุคคลซึ่งถูกตรวจสอบมีสิ่งซึ่งใช้ยืนยันตัวบุคคล เช่น บัตรประจำตัวพนักงานในการบันทึกเวลาเข้า-ออก ในการทำงานแต่ละวัน บัตรสมาร์ทการ์ด (Smart Card) หรือบัตรแถบแม่เหล็ก (Magnetic Card) เป็นต้น

3) สิ่งที่เป็น (Something You Are) หมายถึง การใช้ลักษณะเฉพาะของตัวบุคคลในการตรวจสอบและพิสูจน์ตัวบุคคลสิ่งที่ใช้ตรวจสอบมักเป็นเทคโนโลยีชีวภาพ เช่น ลายนิ้วมือ (Fingerprints) ม่านตา (Iris) เสียง (Voice Prints) เป็นต้น

2.1.3 ความแม่นยำของไอดีไบโอเมตริกซ์

ความแม่นยำของไอดีไบโอเมตริกซ์ นอกจากจะพิจารณาจากประเภทของเทคโนโลยีที่ใช้แล้ว ยังพิจารณาได้จากตัววัดความแม่นยำทางเทคนิค ซึ่งผู้ผลิตอุปกรณ์จะต้องแจ้งให้ผู้ใช้ทราบขณะที่ทำการติดตั้งระบบ เพื่อการปรับแต่งสภาพการใช้งาน ประสิทธิภาพหรือความแม่นยำของอุปกรณ์ไบเมตริกซ์ จะวัดได้จากร้อยละของความผิดพลาดที่เกิดขึ้นขณะทำการตรวจสอบไอดี ดังต่อไปนี้

2.1.3.1 False Rejection Rate (FRR) หรืออัตราการปฏิเสธตัวจริง ค่า FRR จะบอกถึงร้อยละของความผิดพลาดที่เกิดขึ้นเมื่อตัวจริงถูกปฏิเสธ หากค่า FRR ของระบบต่ำแสดงว่าอัตราการปฏิเสธต่ำ นั่นคือ ไม่ว่าตัวจริงหรือตัวปลอมก็มีโอกาส ได้รับอนุญาตให้ผ่านเข้าสู่ระบบได้ง่าย ระบบจะหละหลวม ฉะนั้นหากระบบต้องการความปลอดภัยค่อนข้างสูงก็ควรจะปรับแต่งให้ FRRมีค่าสูงขึ้น อย่างไรก็ตาม ข้อดีของการปรับให้ FRR มีค่าต่ำก็คือ ตัวจริงจะไม่ต้องหงุดหงิดกับการถูกปฏิเสธซ้ำแล้วซ้ำเล่าโดยอัตโนมัติ

2.1.3.2 False Acceptance Rate (FAR) หรืออัตราการยอมรับตัวปลอม เป็นร้อยละของความผิดพลาดที่เกิดขึ้นจากการที่ตัวปลอมได้รับการยอมรับ ค่า FAR ที่ต่ำจะหมายถึงอัตราการยอมปล่อยให้ผ่านเข้าสู่ระบบต่ำ ไม่ว่าตัวจริงหรือตัวปลอมก็มีโอกาสถูกปฏิเสธสูง ระบบจะเข้มงวดมาก ฉะนั้นหากต้องการให้ระบบมีความเป็นมิตรกับผู้ใช้ (User friendliness) ก็ควรจะปรับให้ FAR มีค่าสูง อย่างไรก็ตาม การปรับแต่งให้ FAR มีค่าต่ำจะช่วยป้องกันความเสียหายอันเนื่องมาจากการยอมรับตัวปลอมให้ผ่านเข้าสู่ระบบได้โดยง่าย

2.1.3.3 Equal Error Rate (ERR) คืออัตราความผิดพลาดที่ได้รอมชอมด้วยการปรับแต่งให้ FRR มีค่าใกล้เคียงกับ FAR ระบบที่มีค่า ERR ต่ำจะเป็นที่ต้องการมากกว่าระบบที่มีค่า ERR สูงกว่า ในบางครั้งจะเรียก ERR ว่า Crossover Rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัติ หากเราต้องการให้ตัวจริงถูกปฏิเสธน้อย คือให้ค่า FRR ต่ำ เราจะต้องปล่อยให้ตัวปลอมได้รับการยอมรับมากขึ้น คือค่า FAR สูงขึ้น และในทางกลับกัน หากเราต้องการให้ตัวปลอมหลุดเข้ามาในระบบได้ยาก หรือFAR ต่ำ เราก็ต้องแลกด้วยการปล่อยให้ตัวจริงต้องหงุดหงิดกับการถูกปฏิเสธบ่อยครั้ง หรือคือให้ FRR สูง นั่นคือ ตัววัดทั้งสองนี้จะแปรผกผันซึ่งกันละกัน ฉะนั้นทางสายกลางที่ใช้ในการพิจารณาก็คือ ดูจากค่า ERR ซึ่งเป็นค่าร้อยละที่ได้จากการรอมซอมระหว่างตัวสองตัวนี้

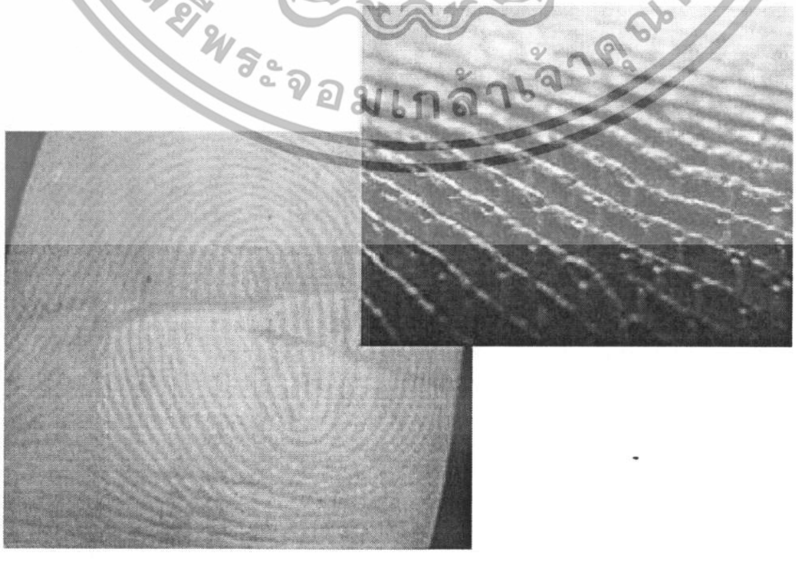
2.1.4 ระบบเครื่องสแกนลายนิ้วมือ

เครื่องสแกนลายนิ้วมือ (Fingerprint Scan) เป็นระบบที่ทำงานด้วยหลักเทคโนโลยีชีวภาค ใช้ลักษณะเฉพาะตัวของคนเราในการพิสูจน์ตัวบุคคลว่าคนคนนั้นคือใคร มีการนำไปใช้ควบคู่กับเทคโนโลยีสมัยใหม่หลายอย่าง อาทิ ระบบลงเวลาทำงานด้วยการสแกนลายนิ้วมือ ระบบควบคุมการเข้า-ออกอาคาร ระบบเปิด-ปิดประตูด้วยการสแกนลายนิ้วมือ เป็นต้น

การนำระบบสแกนลายนิ้วมือไปใช้ที่เห็นได้ชัดในระบบบันทึกเวลา ซึ่งปัจจุบันวิธีบันทึกเวลาได้แก่ ระบบเซ็นชื่อ นาฬิกาตอกบัตร เครื่องรูดบัตรแบบบาร์โค้ด แถบแม่เหล็กไร้สัมผัส และเครื่องสแกนลายนิ้วมือ

2.1.5 ความรู้เบื้องต้นเกี่ยวกับลายนิ้วมือ

ลายนิ้วมือของแต่ละคนเริ่มปรากฏขึ้นตั้งแต่เป็นตัวอ่อนอายุ 3 ถึง 4 เดือนในครรภ์มารดาซึ่งเป็นผิวหนังส่วนที่มีร่อง (Furrow) และมีสัน (Ridge) เอาไว้ใช้สำหรับอำนวยความสะดวกในการหยิบจับสิ่งของ สันและร่องที่ปรากฏนี้มีคุณลักษณะที่สำคัญสองประการ คือ ไม่มีการเปลี่ยนแปลงรูปแบบตามการเวลา (แต่อาจเปลี่ยนขนาดได้) และการมีรูปแบบเฉพาะในแต่ละคน



รูปที่ 2.1 ลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

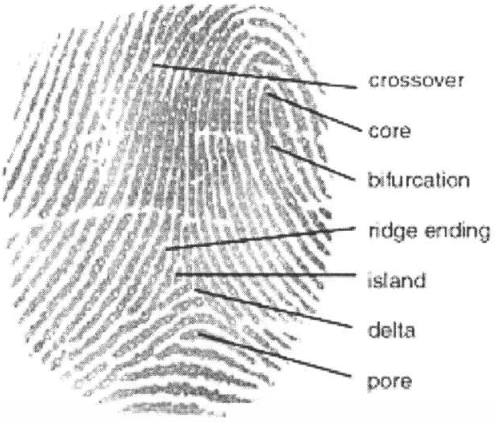
ลายนิ้วมือไม่เปลี่ยนแปลงรูปแบบ (Permanence) ตั้งแต่แรกเกิดจนกระทั่งวันที่เสียชีวิต ลายนิ้วมือจะสลายไปพร้อมกับร่างกาย แต่อาจเปลี่ยนแปลงขนาดได้ตามร่างกาย เหมือนกับการที่เราวาดรูปไว้บนรูปโป่ง ซึ่งไม่ว่าลูกโป่งจะเล็กหรือถูกเป่าให้พองใหญ่อย่างไรก็ยังเป็นรูปเหมือนเดิมแต่มีขนาดใหญ่ขึ้นเท่านั้น

การที่นิ้วมือมีรูปแบบเฉพาะในแต่ละคน(Individuality) เป็นคุณสมบัติที่สำคัญอีกอย่างหนึ่งของลายนิ้วมือนั้น ตั้งแต่เริ่มการใช้เก็บและเปรียบเทียบลายนิ้วมือโดยใช้วิธีสมัยใหม่ ซึ่งมีมาร้อยกว่าปีแล้วยังไม่ตรวจพบว่ามีการเหมือนกันของลายนิ้วมือ อีกทั้งถ้าจะอธิบายด้วยหลักการทางคณิตศาสตร์และวิทยาศาสตร์ก็มีการศึกษาของ Sir Francis Galton (1892) ซึ่งได้ประมาณไว้ว่า โอกาสที่คนสองคนจะมีลายนิ้วมือเหมือนกันนั้นมีความน่าจะเป็นอยู่ที่ 1/64,000,000,000 ซึ่งเป็นการประเมินค่าโดยใช้การแบ่งรายละเอียดรูปแบบของลายนิ้วมือออกเป็นส่วนๆและหาความน่าจะเป็นของการซ้ำกันของแต่ละส่วนนั้นแล้วนำความน่าเป็นของแต่ละส่วนมาคูณกันเพื่อหาความน่าจะเป็นทั้งหมด ท่าน Sir Francis Galton นี้เป็นผู้ที่เริ่มทำการวิจัยอย่างจริงจังกับลายนิ้วมือและถือว่าเป็นบุคคลแรกที่ศึกษาถึงการใช้ลายนิ้วมือในการระบุตัวบุคคล เป็นบุคคลแรกที่ทำการพิสูจน์ว่าลายนิ้วมือของแต่ละคนมีลักษณะเฉพาะ (Individuality) และไม่มีการเปลี่ยนแปลงรูปแบบ (Permanence) อีกทั้งยังเป็นผู้ที่กำหนดและแบ่งแยกประเภทของรูปแบบลายนิ้วมือที่ใช้กันอยู่จนถึงปัจจุบันนี้

ลายนิ้วมือของแต่ละคนนั้นมีลักษณะเฉพาะมากจนกระทั่งแม้แต่คู่แฝดแท้ (Identical Twin) ก็ยังมีลายนิ้วมือที่แตกต่างกัน (แต่มีรูปแบบ DNA เหมือนกัน) อย่างไรก็ตามรูปแบบของลายนิ้วมือนั้นมีลักษณะความคล้ายกันของคนภายในครอบครัว หรือพูดได้อีกอย่างหนึ่งว่ารูปแบบของลายนิ้วมือมีการถ่ายทอดกันทางพันธุกรรม

เมื่อเราหงายมือขึ้นและสังเกตไปที่นิ้วมือแต่ละนิ้วจะพบว่านิ้วนั้นมีลักษณะเป็นลายอยู่ เมื่อมองใกล้ๆจะเห็นว่าลายนิ้วมือทีมีลักษณะประกอบไปด้วย 2 ส่วนใหญ่ คือ ส่วนที่เป็นลายนูน(Ridges) และร่อง (Valleys or Furrows) วิ่งวนเป็นลักษณะส่วนโค้งไปรอบๆช่วงปลายนิ้วมือสลับกันไประหว่างร่องและลายนิ้วนูนทำให้เกิดเป็นลายขึ้นมา ลายดังกล่าวนี้จะมีส่วนของจุดที่เป็นจุดเฉพาะ (Singular Point) คือ จุดกึ่งกลาง (Core) และสันดอน (Delta) รวมทั้งลักษณะสำคัญ (Minutiae) ซึ่งก็คือ จุดปลายของลายนูน (Ridge Ending) และจุดแยกของลายนูน (Ridge Bifurcation) รูปแบบของลายนิ้วมือนี้จะมีอยู่หลายรูปแบบด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



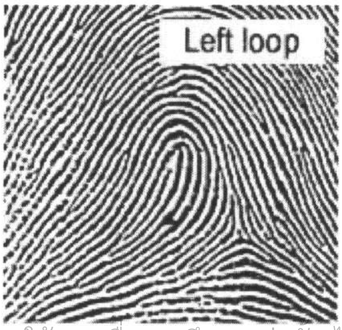
รูปที่ 2.2 ส่วนประกอบของลายนิ้วมือ

เมื่อศึกษารูปแบบของลายนิ้วมือตามลักษณะการวางตัวของสันที่นูนขึ้นอย่างคร่าวๆ โดยไม่คำนึงถึงส่วนรายละเอียดที่ลึกลงไป พบว่าสามารถจำแนกรูปแบบลายนิ้วมือออกได้เป็น 4 แบบหลักๆ ได้แก่

2.1.5.1 แบบมัดหวาย (Loop)

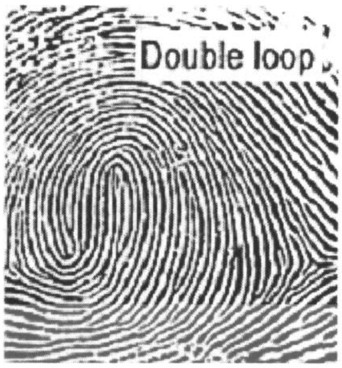
ประชากรประมาณ 60-70% มีลายนิ้วมือแบบนี้ซึ่งแบ่งออกเป็น 5 แบบย่อยดังนี้

รูปที่ 2.3 แสดงลักษณะของกลุ่มมัดหวายเอียงขวา



รูปที่ 2.4 แสดงลักษณะของกลุ่มมัดหวายเอียงซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงลักษณะรูปแบบของกลุ่มมัดหวายคู่

รูปที่ 2.6 แสดงลักษณะของ Right pocket loop

รูปที่ 2.7 แสดงลักษณะของ Left pocket loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

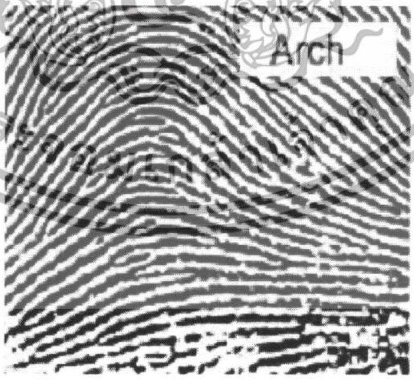
2.1.5.2 แบบก้นหอย (Whorl)

ประชากร 25-35% จะมีลายนิ้วมือแบบก้นหอย ลักษณะการวิ่งของลายนูนมีรูปแบบเป็นลักษณะเส้นโค้งเป็นรูปวงกลมจากวงใหญ่ค่อยๆเล็กลงไปเรื่อยๆจนเหลือวงกลมเล็กที่สุด

รูปที่ 2.8 แสดงลักษณะรูปแบบของกลุ่มก้นหอย

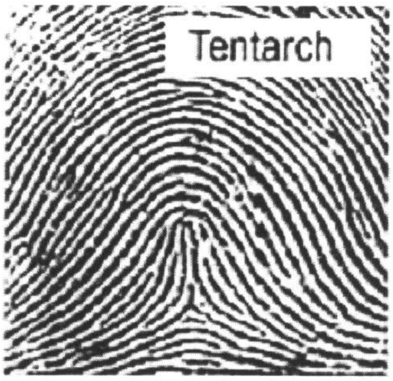
2.1.5.3 แบบโค้ง (Arch)

ประชากร 5% มีลายนิ้วมือแบบนี้ ซึ่งแบ่งออกเป็น 2 แบบย่อย โดยกลุ่มเส้นโค้งทั้งสองจะแตกต่างกันที่ลักษณะความสูงของส่วนโค้ง ดังนี้



รูปที่ 2.9 แสดงลักษณะรูปแบบของกลุ่มเส้นโค้งราบ(Arch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงลักษณะรูปแบบเส้นโค้งกระโจม(Tented Arch)

2.1.5.4 แบบผสม (Mixed figure / Composite)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่จัดเข้าเป็นลายนิ้วมือชนิดใดโดยเฉพาะ ประกอบด้วยลายนิ้วมือ 2 แบบมาผสมกัน และมีสันดอน 2 สันดอน หรือมากกว่า เช่นกรณีที่ไม่สามารถเข้ากับลายนิ้วมือกลุ่มที่กล่าวมาแล้วข้างต้นไม่ได้เลย โดยมีความยุ่งเหยิงและเป็นรูปแบบที่ไม่แน่นอน

รูปที่ 2.11 แสดงลักษณะรูปแบบของกลุ่ม Mixed figure

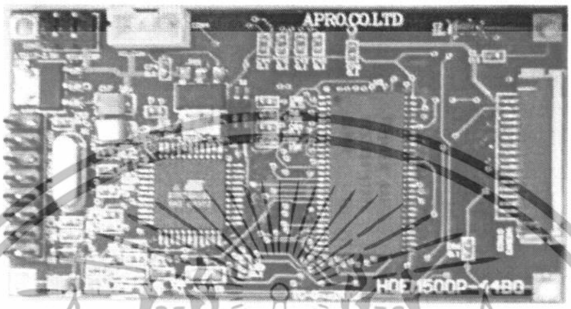
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 BioPrima OEM2000P-44B0 Fingerprint Identification Board

2.2.1 โครงสร้าง

OEM2000P-44B0 ประกอบด้วย fingerprint processor และ optical input sensor เป็น โมดูล High-Speed Fingerprint identification OEM, และประกอบด้วย CPU, Flash memory, SDRAM



รูปที่ 2.12 โมดูล OEM2000P-44B0

รูปที่ 2.13 Optical sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 คุณสมบัติของเครื่องสแกนลายนิ้วมือ

2.2.2.1 ลักษณะเด่น

- ใช้เทคโนโลยีการจับคู่ลายพิมพ์นิ้วมือด้วยความเร็วสูง
- ตัวตรวจจับสัญญาณแบบ Optical
- การจับคู่แบบ 1:1 และ 1: N
- อ่าน/เขียน ข้อมูลลายพิมพ์นิ้วมืออย่างละเอียด
- มีหน่วยควบคุมตามความเหมาะสมของอัตราส่งผ่าน
- รับและส่งผ่านคำสั่งผ่านการสื่อสารแบบขนานตามมาตรฐาน RS-232C

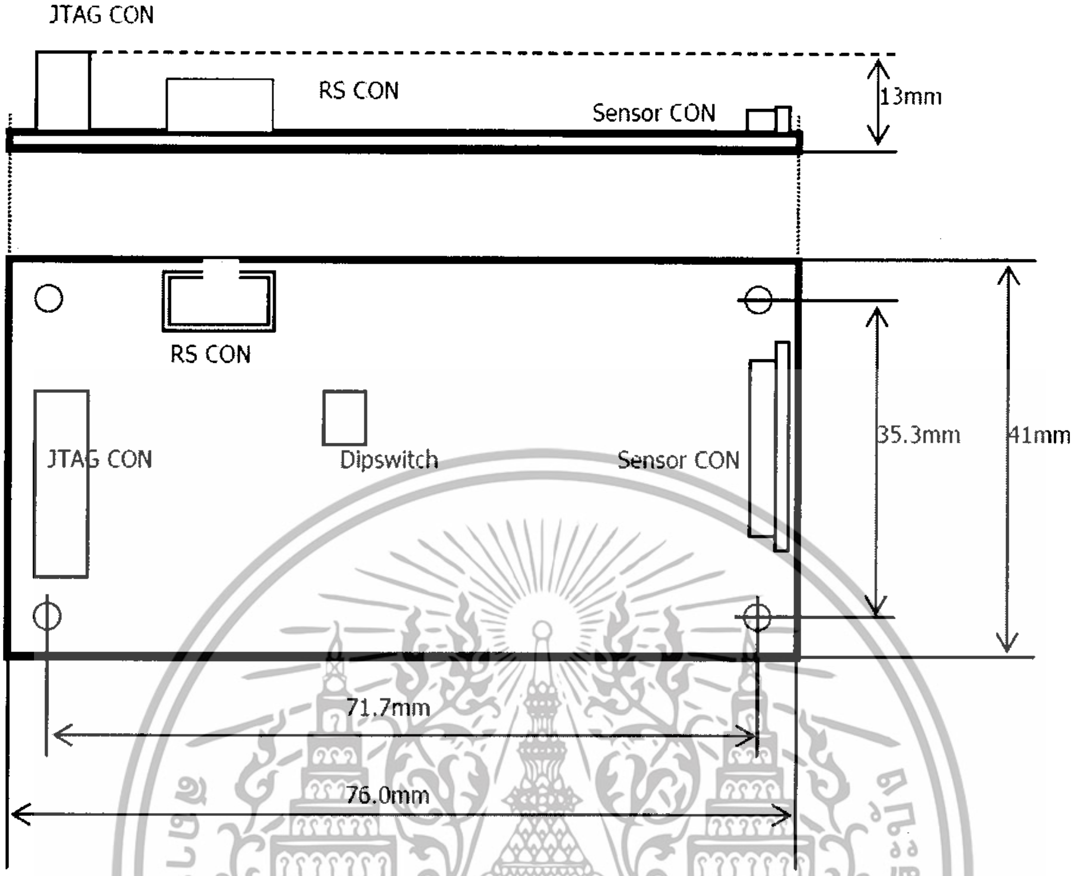
2.2.2.2 คุณสมบัติทางเทคนิค

ตารางที่ 2.1 คุณสมบัติทางเทคนิค

<table>
<tr><th colspan="2">หัวข้อ</th><th>รายละเอียด</th></tr>
<tr><td colspan="2">โหมดการตั้งค่า</td><td>RS232C(3 เส้น)
Baud rate 9600/19200/38400/115200bps
Parity = ไม่มี
Stop Bit = 1
Flow Control = ไม่มี</td></tr>
<tr><td colspan="2">ขนาดของข้อมูลลายพิมพ์นิ้วมือที่ลงทะเบียน</td><td>1404 Byte</td></tr>
<tr><td colspan="2">ความจุลายพิมพ์นิ้วมือสูงสุด</td><td>2000 ลายนิ้วมือ</td></tr>
<tr><td colspan="2">โหมดการจับคู่</td><td>1:1 , 1:N</td></tr>
<tr><td colspan="2">โหมดการลงทะเบียน</td><td>3 ครั้ง</td></tr>
<tr><td colspan="2">ปริมาณอัตราการส่งผ่าน</td><td>1 ~ 5</td></tr>
<tr><td colspan="2">FAR</td><td>< 0.0001% (ในกรณีอัตราการส่งผ่านเท่ากับ 3)</td></tr>
<tr><td colspan="2">FRR</td><td>< 0.01% (ในกรณีอัตราการส่งผ่านเท่ากับ 3)</td></tr>
<tr><td colspan="2">ความเร็วในการลงทะเบียน</td><td>< 3 วินาที</td></tr>
<tr><td colspan="2">ความเร็วในการตรวจสอบ</td><td>< 1 วินาที</td></tr>
<tr><td colspan="2">แรงดันไฟฟ้า</td><td>DC 5 โวลต์</td></tr>
<tr><td colspan="2">กระแสไฟฟ้า</td><td>200 มิลลิแอมแปร์</td></tr>
<tr><td rowspan="2">สภาวะการทำงาน</td><td>อุณหภูมิ</td><td>0 องศาเซลเซียล ~ +60 องศาเซลเซียล</td></tr>
<tr><td>ความชื้น</td><td>20% ~ 80 %</td></tr>
<tr><td rowspan="2">สภาวะการทำงานตอนบันทึกผล</td><td>อุณหภูมิ</td><td>0 องศาเซลเซียล ~ +70 องศาเซลเซียล</td></tr>
<tr><td>ความชื้น</td><td>10 % ~ 80 %</td></tr>
<tr><td colspan="2">ขนาดของบอร์ด</td><td>76 x 41 x 12 มิลลิเมตร</td></tr>
</table>

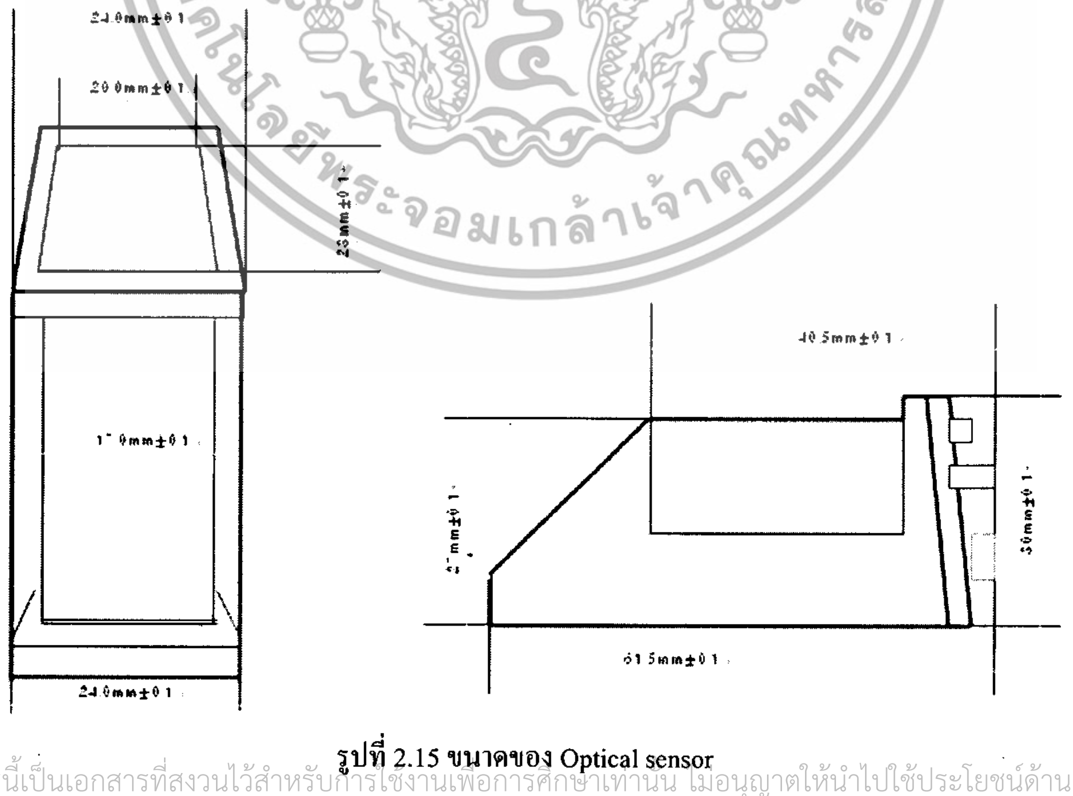
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.3 โครงสร้าง



รูปที่ 2.14 โครงสร้าง

2.2.2.4 ขนาดของ Optical sensor



รูปที่ 2.15 ขนาดของ Optical sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.5 RS-232C

ตารางที่ 2.2 ไอซี RS-232C

หัวข้อ	ตัวเลขของขาไอซี	รายละเอียด
+5V	1	แรงดันไฟฟ้า +5 โวลต์
GND	2	Ground
TXD0	3	การส่งผ่านข้อมูลของ RS232C สำหรับ serial port 0
RXD0	4	การรับข้อมูลของ RS232C สำหรับ serial port 0

2.2.2.6 DIPSW

รูปที่ 2.16 DIPSW

SW1 = ON: พอร์ตการตั้งค่า (ทำงาน)

= OFF: ปิด

Baud = 115200

Data Bit = 8

Parity = ไม่มี

Stop = 1

SW2 = ON: flash ข้อมูล (ทำงาน)

= OFF: ปิด

DIPSW การตั้งค่าในการใช้:

SW1 = ON

SW2 = ON

วิธีการตั้งค่าเริ่มต้น:

SW1 = ON

SW2 = ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

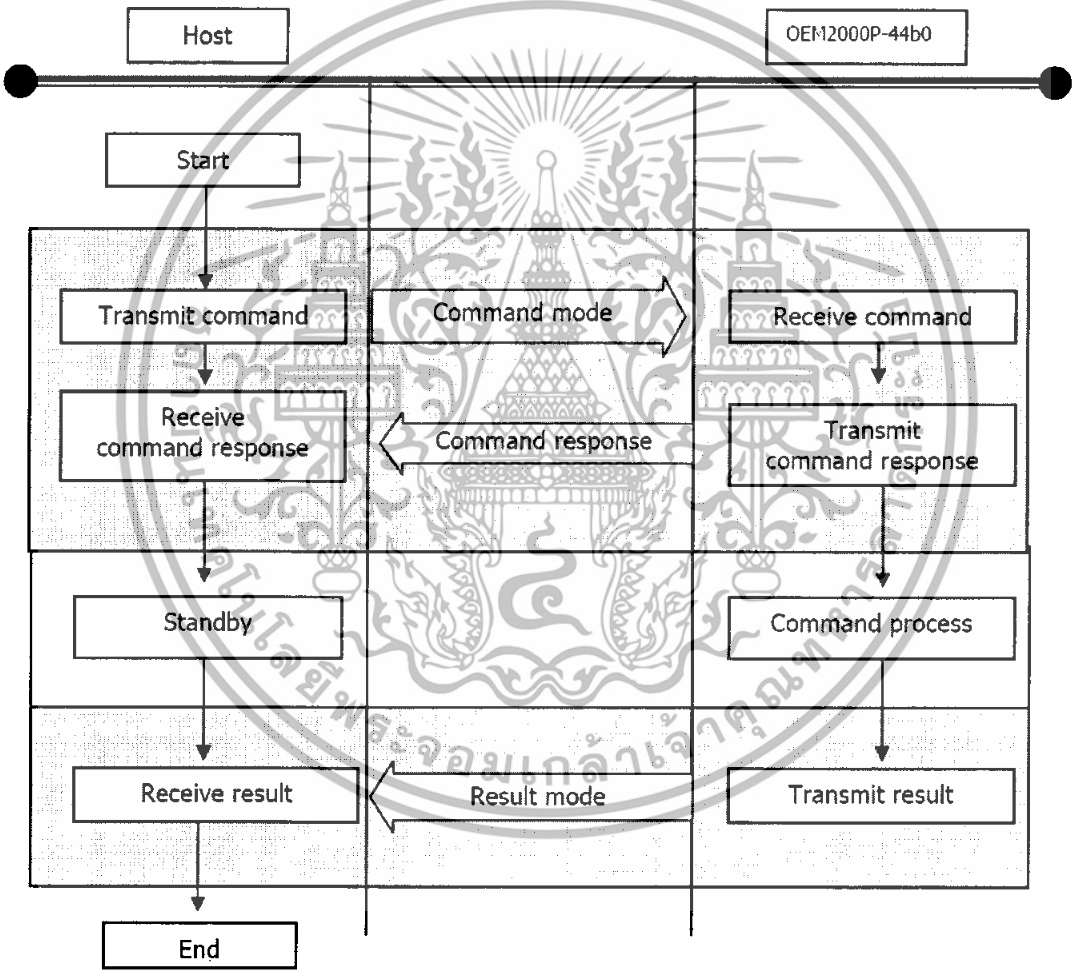
2.2.3 โปรโตคอล (Protocol) ในการสื่อสาร

โปรโตคอลในการสื่อสาร ของ RS232C มี 3 ชนิด

1. โหมดคำสั่ง (Command Mode)
2. โหมดรับ (Receive Mode)
3. โหมดส่ง (Send Mode)

2.2.3.1 โหมดคำสั่ง

คำสั่งเท่านั้นที่ถูกส่งผ่านและรับ ไม่ใช่การส่งผ่านและการรับของบล็อกข้อมูล

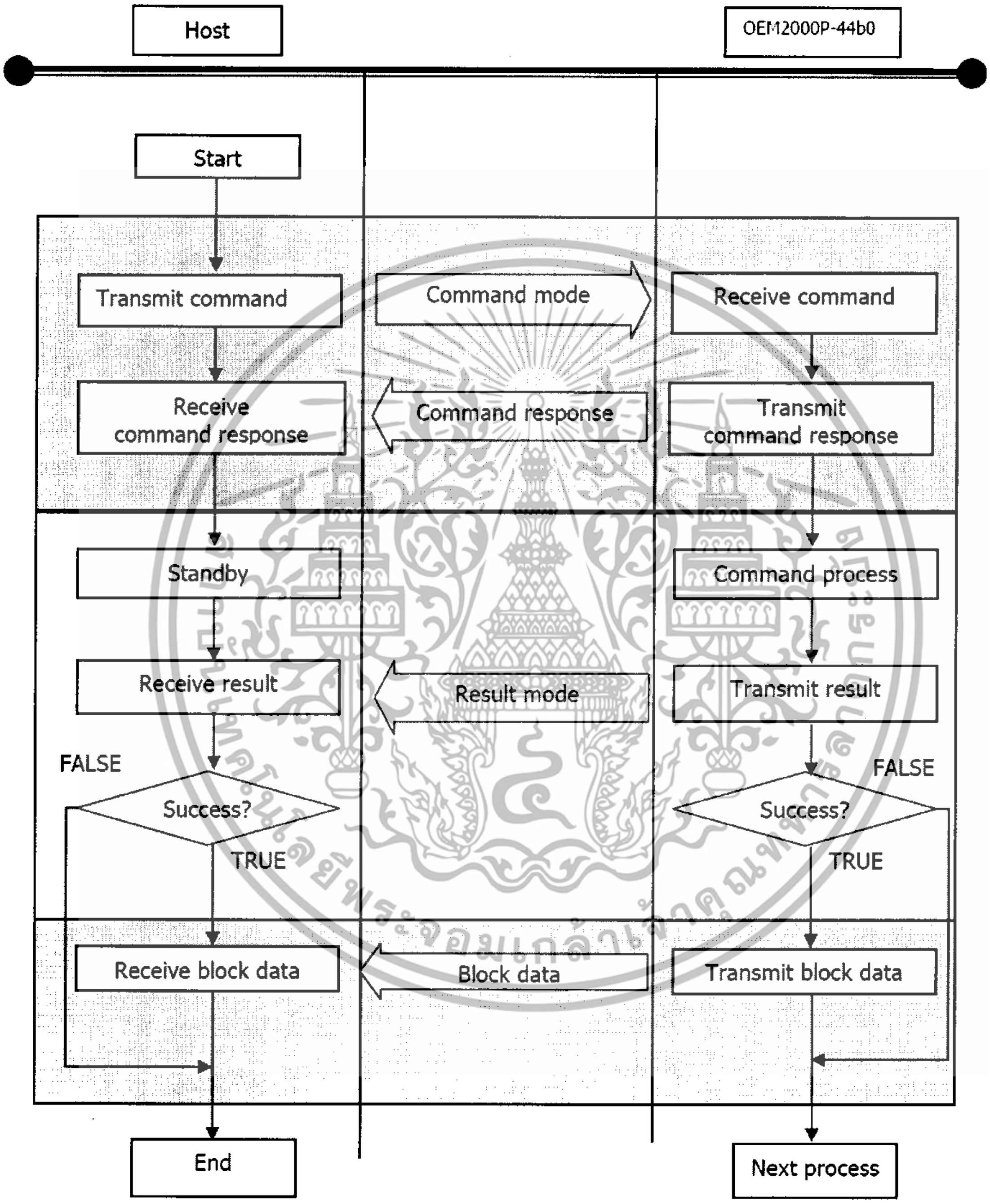


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.3.2 โหมดรับ

บล็อกข้อมูลถูกส่งผ่านจาก OEM2000P-44B0 ถึง โฮส (Host) เช่นเดียวกับผลของคำสั่งปฏิบัติการ

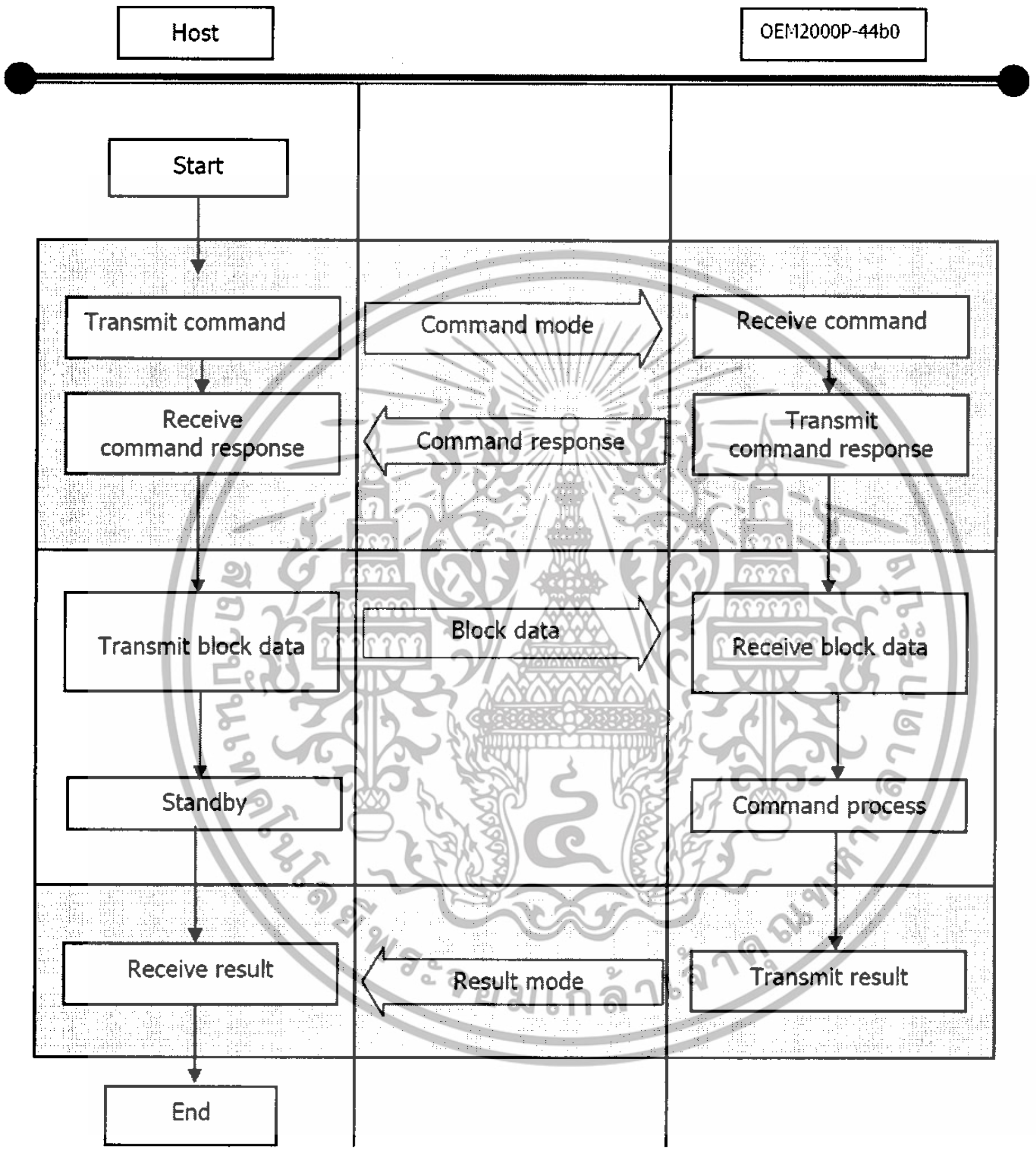


82912

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.3 โหมดส่ง

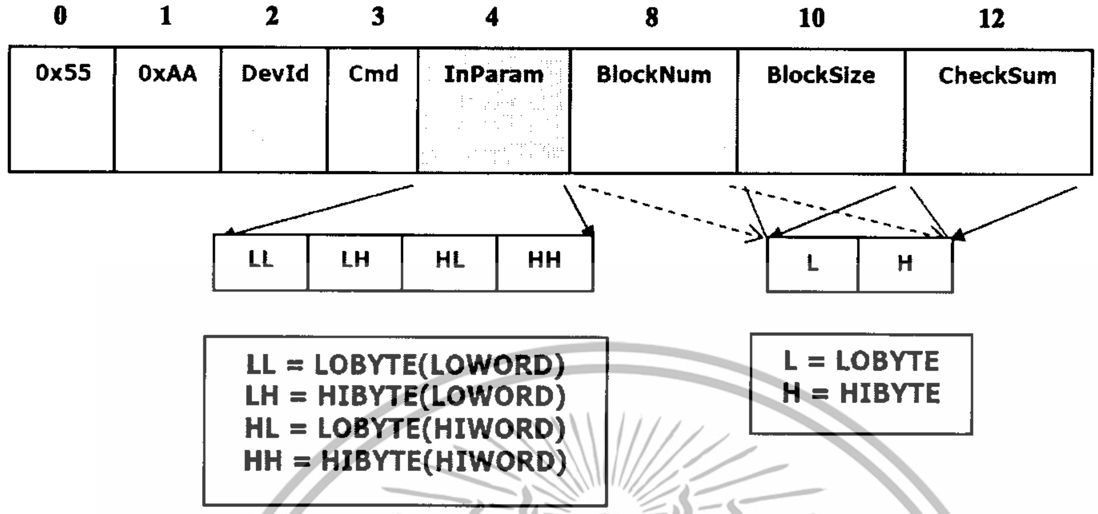
OEM2000P-44B0 รับบล็อกข้อมูลในการส่งผ่านจาก โฮส และทำการสั่งการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 โครงสร้าง แพ็คเกต

2.2.4.1 Command Packet (คำสั่ง)

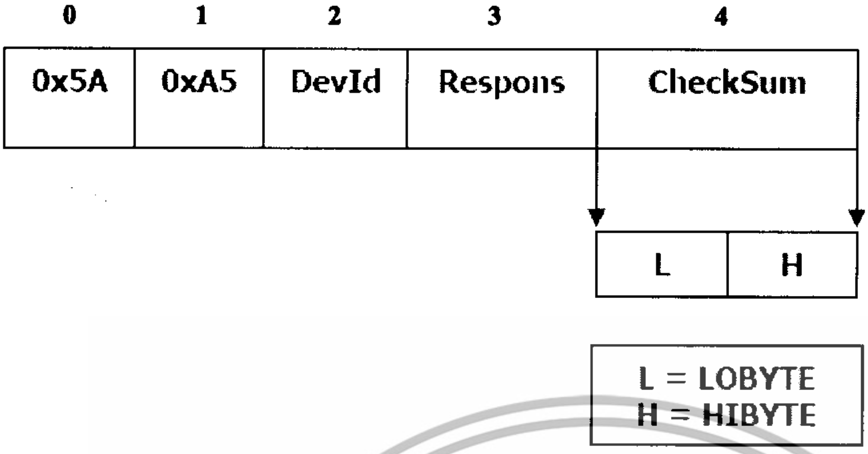


ตารางที่ 2.3 Command packet

Offset	หัวข้อ	ชนิด	คำอธิบาย
0	0x55	ไบท์	เริ่มต้น Code 1
1	0xAA	ไบท์	เริ่มต้น Code 2
2	DevID	ไบท์	รหัสหมายเลข (0x01-0xff) ปกติกำหนดเป็น 0x01
3	Cmd	ไบท์	รหัสคำสั่ง
4	InParam	DWORD	อินพุท
8	BlockNum	เวิร์ด	หมายเลขบล็อก ถูกใช้เมื่อมีการส่งผ่านบล็อกข้อมูล โหมดรับและโหมดส่งใช้คำสั่งนี้
10	BlockSize	เวิร์ด	ขนาดบล็อก ถูกใช้เมื่อมีการส่งผ่านบล็อกข้อมูล โหมดรับและโหมดส่งใช้คำสั่งนี้
12	CheckSum	เวิร์ด	Check Sum OFFSET[0] + … + OFFSET[11] =CheckSum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

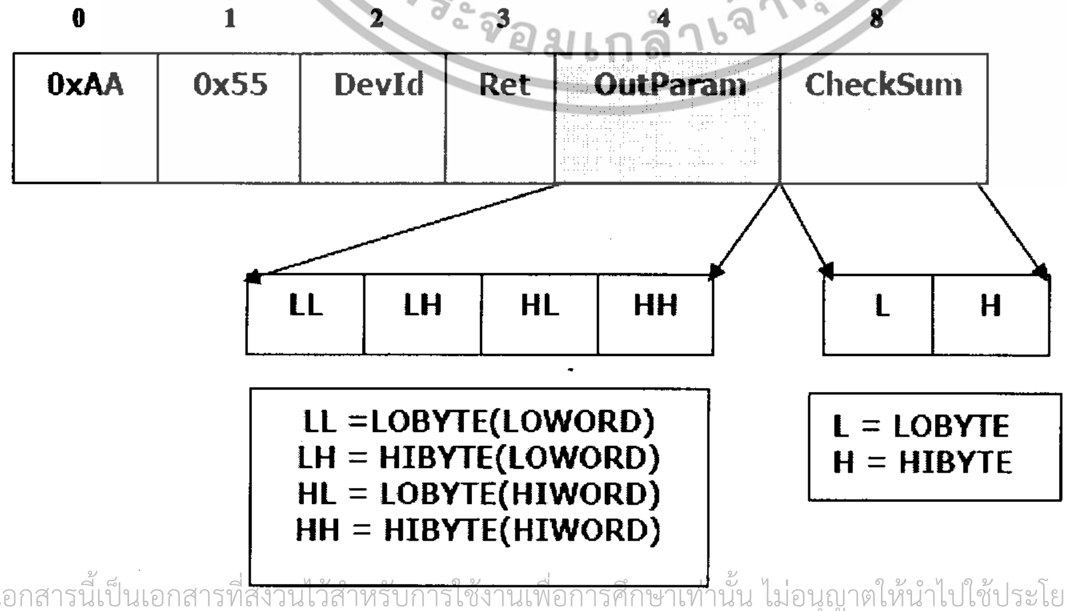
2.2.4.2 Command Response Packet (การตอบสนองคำสั่ง)



ตารางที่ 2.4 Command Response Packet

Offset	หัวข้อ	ชนิด	คำอธิบาย
0	0x5A	ไบท์	เริ่มต้น Code 1
1	0xA5	ไบท์	เริ่มต้น Code 2
2	DevID	ไบท์	รหัสหมายเลข (0x01-0xff) ปกติกำหนดเป็น 0x01
3	Respons	ไบท์	Response =0x02: ผลตอบสนองผิดพลาด(NAK) Response =0x03: ผลตอบสนองถูกต้อง (ACK)
4	CheckSum	เวิร์ด	Check Sum OFFSET[0] + … + OFFSET[3] =CheckSum

2.2.4.3 Packet Result (ผลลัพธ์)

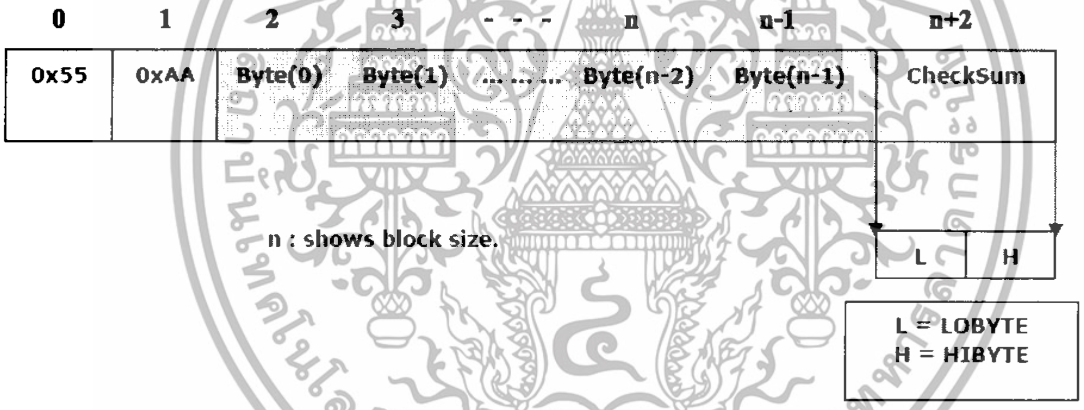


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 Packet Result

Offset	หัวข้อ	ชนิด	คำอธิบาย
0	0xAA	ไบท์	เริ่มต้น Code 1
1	0x55	ไบท์	เริ่มต้น Code 2
2	DevID	ไบท์	รหัสหมายเลข (0x01-0xff) ปกติกำหนดเป็น 0x01
3	Ret	ไบท์	รหัส Result 0x01:การกระทำคำสั่งถูกต้อง 0x00:การกระทำคำสั่งผิดพลาด
4	Output	DWORD	เอาท์พุท
8	CheckSum	เวิร์ด	Check Sum OFFSET[0] + … + OFFSET[7]=CheckSum

2.2.4.4 Block Data Packet (บล็อกข้อมูล)



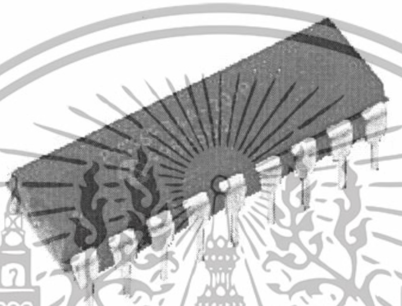
ตารางที่ 2.6 Block Data Packet

Offset	หัวข้อ	ชนิด	คำอธิบาย
0	0x55	ไบท์	เริ่มต้น Code 1
1	0xAA	ไบท์	เริ่มต้น Code 2
2	BYTE[0]	ไบท์	ข้อมูลที่ 1 (ไบท์)
3	BYTE[1]	ไบท์	ข้อมูลที่ 2 (ไบท์)
: :	: :	: :	: :
n + 1	BYTE [n - 1]	ไบท์	n (ขนาดบล็อก) หน่วยเป็นไบท์
n +2	CheckSum	เวิร์ด	Check Sum OFFSET[0] + … + OFFSET[n+1] =CheckSum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 PIC (Peripheral Interface Controller)

PIC คือ microcontroller อีกตระกูลหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่งแนวความคิด ของ microcontroller ตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัวของมันไม่ว่าจะเป็น PROGRAM MEMROY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำซึ่งทำให้เหมือนกับ CPU ตัวหนึ่ง



รูปที่ 2.17 PIC (Peripheral Interface Controller)

ความเร็วของ PIC ภาคของความถี่สัญญาณนาฬิกา ปัจจุบันสามารถทำสัญญาณนาฬิกาได้ที่ 20 MHz ซึ่งทำให้หนึ่งคำสั่งของ PIC ใช้เวลาเพียง 0.25 uSec แต่อย่างไรก็ตามได้มีบริษัทอื่นได้ซื้อลิขสิทธิ์ PIC จาก microchip และได้สร้าง chip ที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

หน่วยความจำของ PIC ในอดีตหน่วยความจำของ PIC จะค่อนข้างน้อย คืออยู่ระหว่าง 512 words ถึง 4K words แต่ในปัจจุบัน บริษัท microchip ซึ่งเป็นเจ้าของPIC ได้พัฒนาจนทำให้ memory ของ PIC มีขนาดเป็นหลายสิบกิโลไบต์ และมีทีท่าว่าจะขยายได้ใหญ่ขึ้นเรื่อยๆในเรื่องของการนับขนาดของหน่วยความจำของ PIC จะนับไม่เหมือนปกติ โดยที่ หนึ่งคำสั่งของ PIC จะมีขนาด 14 bits ดังนั้นเราจะเรียกว่า 1 word ของ PIC จะมีขนาด 14 bits เช่น PIC16F84A ระบุว่ามีหน่วยความจำ 1 K (ซึ่งหมายถึง 1 Kword ถ้าคำนวณให้เป็นแบบ 1 byte = 8 bit จะได้ว่า 1 x 1,024 x 14 = 14,336 bits ดังนั้นก็คือ 14,336/(8 x 1,024) = 1.75K bytes นั่นเอง

สถาปัตยกรรมของ PIC ตอนนี้มี 3 สายหลักๆ สมัยก่อนมีแค่สอง คือขึ้นต้นด้วย 16xxx, 17xxx และใหม่ล่าสุดคือ 18xxx ถ้าพูดถึงคุณสมบัติที่เหนือกว่าเรียงจากน้อยสุดไปมากสุดก็คือ 16 -> 17 -> 18 โดยคำสั่ง assembly ของ 17 มีน้อยกว่า 18 แต่จะมีมากกว่า 16 ทำให้เขียนโปรแกรมได้ง่ายกว่า ราคาก็จะสูงกว่าด้วย แต่ที่เป็นที่นิยมก็คือตระกูล 16xxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปแนวความคิดสถาปัตยกรรมของ PIC

PIC จะยึดถือการออกแบบที่ว่ารวมทุกอย่างไว้ใน chip ตัวเดียวโดยไม่ต้องต่ออุปกรณ์ใดๆ เพิ่มเติม ผลที่ตามมาก็คือแผ่นวงจรจะมีขนาดเล็ก และอุปกรณ์ที่ใช้จะไม่มาก บางงานอาจจะใช้แค่ PIC เพียงตัวเดียวโดยไม่ต้องใช้ chip อื่นมาเพิ่มเติมเลย นี่คือคุณสมบัติพิเศษของ PIC ซึ่งปัจจุบันหลายบริษัทที่ผลิต microcontroller ก็เริ่มที่จะมาใช้ตามแนวทางนี้ แต่ทุกอย่างย่อมมีข้อเสีย เนื่องจากแนวความคิดที่จะรวมทุกอย่างไว้ใน chip เดียว ทำให้ program memory และ data memory ไม่สามารถขยายโดยใช้กับ memory ภายนอกได้ (ในทางทฤษฎีเท่านั้น แต่ในทางปฏิบัติสามารถทำได้แต่ต้องใช้เทคนิค ซึ่งไม่นิยม) PIC จึงเหมาะสำหรับรับงานเล็กๆ ไม่ใช่งานใหญ่ๆ ที่ต้องใช้การคำนวณ และ memory เยอะๆ

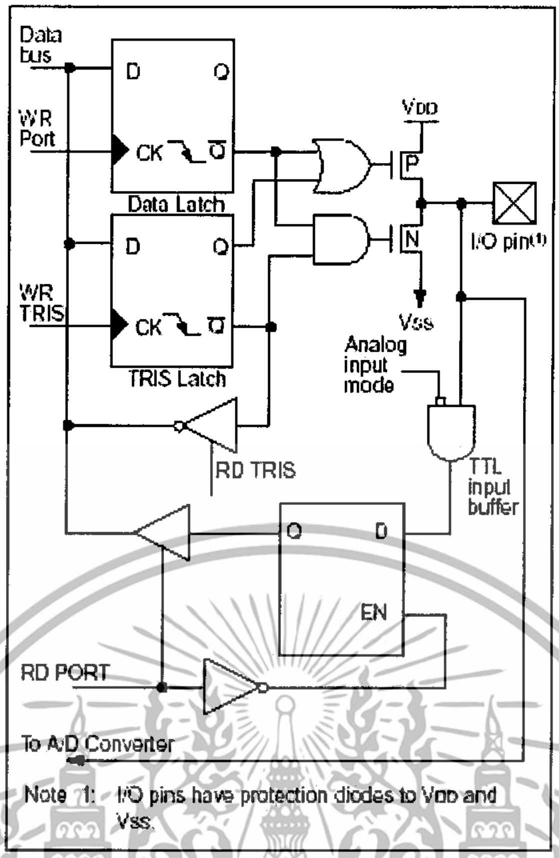
2.3.1 PORTA ใน PIC16F87X

I/O ports บางตัวของ PIC เป็นแบบ multiplexed ซึ่ง อาจเป็นทั้ง I/O หรือ peripheral features (เช่น A/D, Serial, IIC) ซึ่งเมื่อ ขาเหล่านี้ใช้งานในส่วน peripheral ก็จะไม่สามารถใช้งานในลักษณะของ I/O ได้ PORTA และ TRISA register

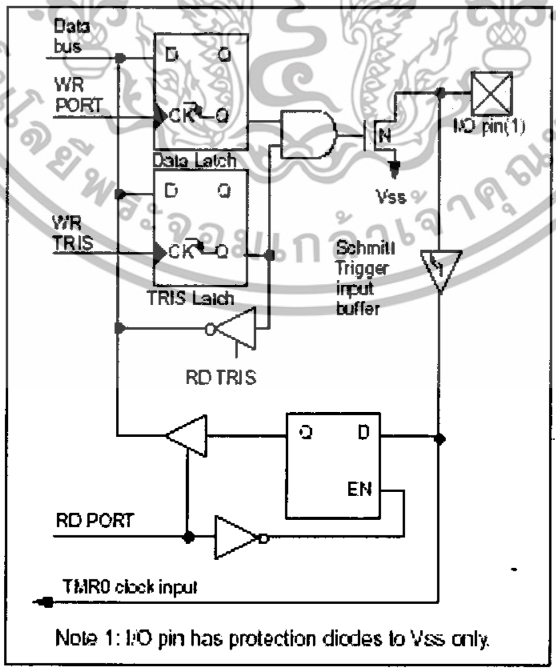
PORTA มีขนาด 6 bit ซึ่งเป็น port ที่เป็นได้ทั้ง Input และ Output โดยต้องเลือกแบบใดแบบหนึ่ง สามารถเลือกได้จาก register ที่มีชื่อว่า TRISA ซึ่งถ้า TRISA bit ถูก set เป็น ‘1’ PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็ทำงานเป็น input (ทำให้ port นั้นอยู่ในสถานะ hi-impedance) ส่วนถ้า TRISA bit ถูก set เป็น ‘0’ PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็จะทำงานเป็น output (port จะอยู่ในสถานะ output latch) การอ่านค่า PORTA register คือการอ่านค่าสถานะของ ขา PORTA ในขณะนั้น ส่วนการเขียนค่าไปยัง PORTA คือการเขียนไปยัง latch ของ port ลักษณะการเขียนจะเป็นแบบ read-modify-write operations ซึ่งหมายความว่า ในการเขียนไปยัง port จะเริ่มด้วยการ อ่านค่า port นั้นมาก่อนแล้วทำการเปลี่ยนแปลงค่า จากนั้นก็ทำการเขียนกลับไปยัง port latch อีกครั้งหนึ่ง

ขา RA4 จะ multiplexed กับ Timer0 module clock input ซึ่งจะเรียกรวมๆ ว่า RA4/T0CKI โดยที่ขา RA4/T0CKI จะเป็นลักษณะ Input แบบ Schmitt Trigger และ Output แบบ open drain. Port RA ทั้งหมด จะมี TTL input level และ มี output แบบ full CMOS drivers ส่วน PORTA ขาอื่นๆ จะ multiplex กับ analog inputs และ Vref ของ A/D input ซึ่งการกำหนดการทำงานของแต่ละขา สามารถเลือกได้โดย clear หรือ set control bits ใน ADCON1 register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 บล็อกไดอะแกรม ขา RA3:RA0 และ RA



รูปที่ 2.19 บล็อกไดอะแกรม ขา RA4/TOCKI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขณะที่เกิด Power-on Reset ขาเหล่านี้จะถูก config ให้เป็น analog input และจะอ่านค่าได้เป็น '0' TRISA register มีหน้าที่ควบคุมว่าขา PORTA ใดจะเป็น input/output ในกรณีที่ใช้ PORTA เป็น analog input TRISA register จะต้องถูก set

ตัวอย่างการ INITIALIZING PORTA

BCF STATUS, RP0

CLRF PORTA ; ทำการ clear output data ของ PORTA

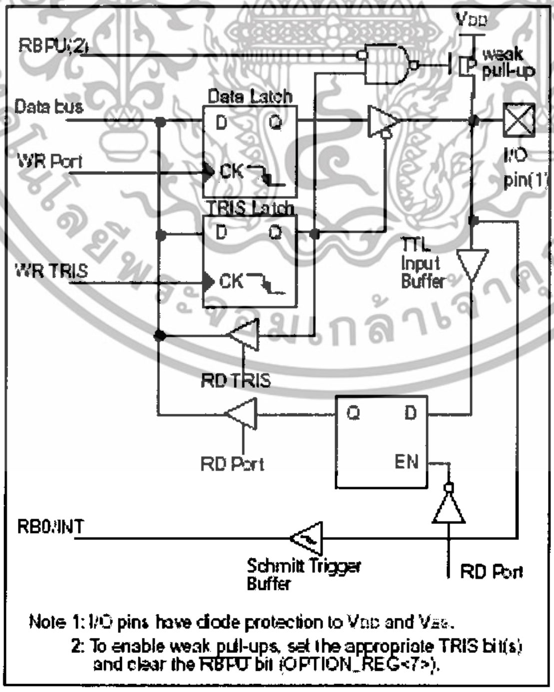
BSF STATUS, RP0 ; เลือก BANK1

MOVLW 0xCF ; ใส่ค่าคงที่ลงใน W register

MOVWF TRISA ; กำหนดให้ PORTA 0-3 เป็น input , กำหนดให้ PORTA 4-5 เป็น output

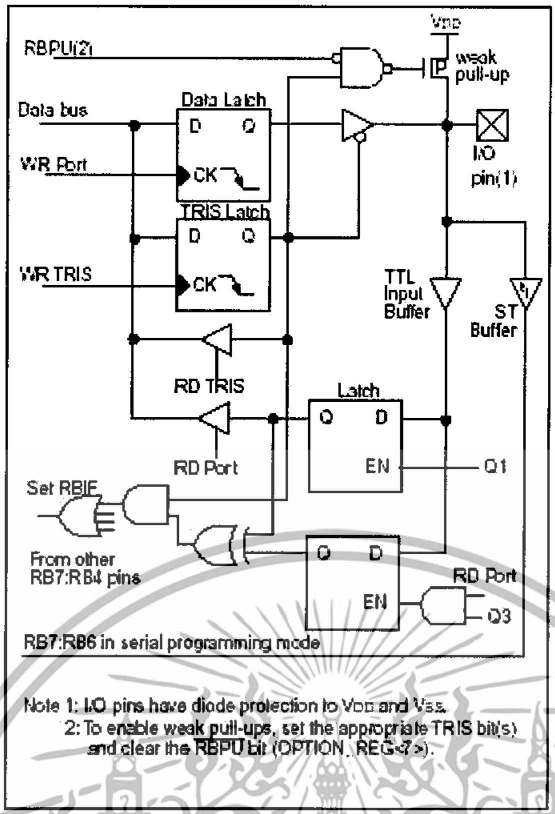
2.3.2 PORTB ใน PIC16F87X

PORTB เป็นลักษณะแบบ Port แบบสองทิศทาง ซึ่ง register ที่จะเป็นตัวกำหนดว่า port ใดจะเป็นแบบ input/output จะถูกกำหนดโดย TRISB register ถ้า set TRISB bit ใด (=1) PORTB ที่บิตนั้นก็จะเป็น input ถ้า clear TRISB bit ใด (=0) PORTB ที่บิตนั้นก็จะเป็น output ขาสามขาของ PORTB จะ multiplexed กับ Low Voltage Programming function ซึ่งได้แก่ RB3/PGM, RB6/PGC และ RB7/PGD



รูปที่ 2.20 บล็อกไดอะแกรม ขา RB3:RB0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 บล็อกไดอะแกรม ขา RB7:RB4

การ INITIALIZING PORTB

PORTB แต่ละ port จะมี weak pull up อยู่ภายใน (ถ้าต้องการ pull up แข็งๆต้องต่อวงจรภายนอก) เราสามารถกำหนดว่าจะใช้ pull up ภายในหรือไม่จากการ set หรือ clear RBPU\ (OPTION register บิต 7) โดยถ้าเรา clear RBPU\ จะหมายถึงเราทำการ disable pull-up ภายใน และถ้าเรากำหนดให้ PORTB เป็น OUTPUT แล้ว pull up จะถูก disable โดยอัตโนมัติ

สำหรับ PORTB นั้น ขา RB4-RB7 จะมี feature เพิ่มเติมก็คือ การกำหนดให้เกิด Interrupt เมื่อเกิดการเปลี่ยนแปลงของสถานะของสัญญาณไฟฟ้าที่ขา RB4-RB7 (โดยถ้าขาใดขาหนึ่งเกิดเปลี่ยนสถานะก็จะทำให้เกิด RB Port Change Interrupt ขึ้น ซึ่งทำให้ RBIF (INTCON.0) fag ถูก set) โดยที่ Interrupt ประเภทนี้สามารถทำการ “wake” microcontroller จากสถานะ sleep mode ได้

RBIF flag จะถูก Clear ได้ 2 กรณี คือ

1.ทำการอ่านหรือเขียน PORTB

2.ทำการ clear RBIF flag โดยตรง

หากเราใช้ interrupt on PORTB change แล้วไม่ควรจะ enable pull-up ของ PORTB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 PORTC ใน PIC16F87X

PORTC เป็นลักษณะแบบ Port แบบสองทิศทาง ซึ่ง register ที่จะเป็นตัวกำหนดว่า port ใดจะเป็นแบบ input/output จะถูกกำหนดโดย TRISC register ถ้า set TRISC bit ใด (=1) PORTB ที่บิตนั้นก็จะเป็น input ถ้า clear TRISC bit ใด (=0) PORTC ที่บิตนั้นก็จะเป็น output ที่ PORTC จะมีคุณสมบัติเพิ่มเติม เช่น IIC, UART, SPI, PWM, CAPTURE ขึ้นอยู่กับการเลือกใช้งาน โดยเมื่อเราทำการ enable คุณสมบัติเพิ่มเติมต่างๆ ที่ PORTC เราต้องระวังในเรื่องของการตั้งค่า TRISC ของแต่ละขาของ PORTC เพราะในการ enable คุณสมบัติบางตัวที่อยู่ที่ PORTC (เช่น UART) ตัวมันเองก็จะทำการเปลี่ยน bit TRISC โดยอัตโนมัติ ดังนั้นไม่ควรที่จะตั้งค่า TRISC โดยตรงกับขาใดของ PORTC ที่ทำการ enable คุณสมบัติเพิ่มเติม

INITIALIZING PORTC

BCF STATUS, RP0; ทำการ Initialize PORTC โดยทำการ clear output data latch

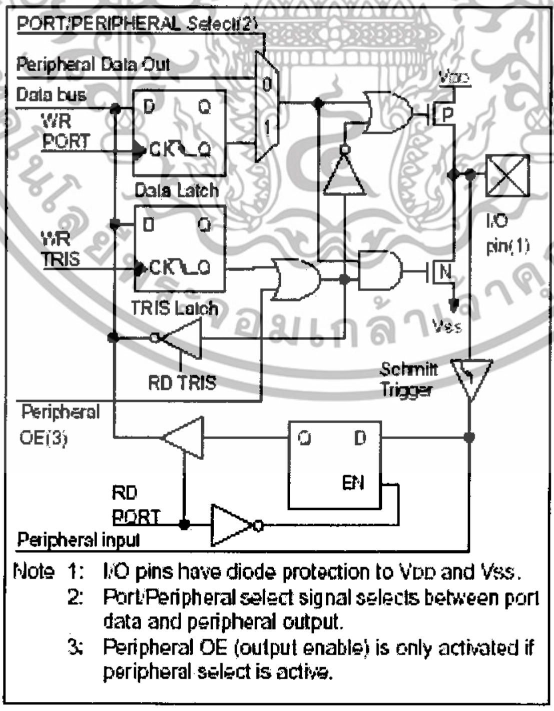
CLRF PORTC

BSF STATUS, RP0; ทำการเลือกไปยัง Bank1

MOVLW 0xCF; โหลดค่าที่ต้องการ set

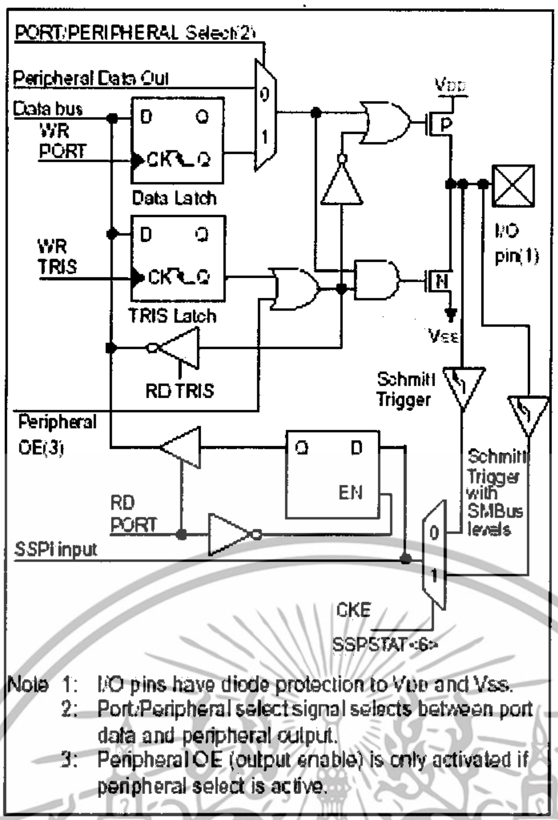
MOVWF TRISB ; Set PORTC0-3 เป็น inputs, Set PORTC4-5 เป็น outputs, Set PORTC6-7 เป็น inputs

PORTC แต่ละ port จะ Schmitt Trigger input buffers อยู่ภายในแต่ละขา



รูปที่ 2.22 บล็อกไดอะแกรม PORTC RC<0:2> RC<5:7>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 บล็อกไดอะแกรม PORTC RC <3:4>

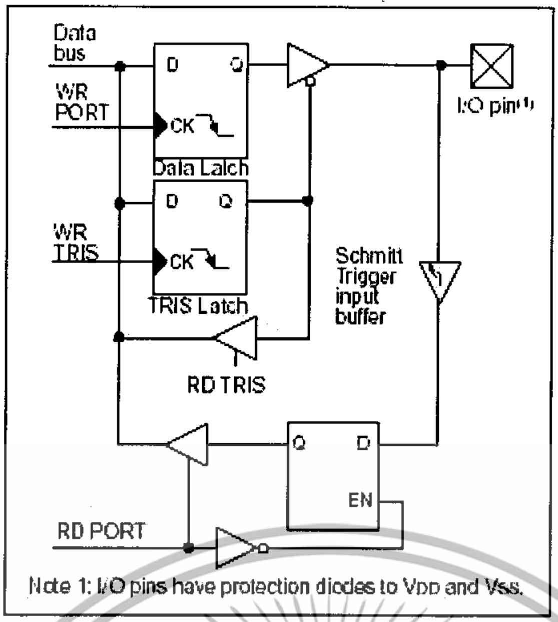
ลักษณะโครงสร้างของ PORT จะแบ่งเป็น 2 กลุ่มคือ PORTC0-2, 5-7 และอีกกลุ่มหนึ่งก็คือ PORTC3-4

2.3.4 PORTD และ PORTE ใน PIC16F87X

สำหรับ PORTD และ PORTE นั้นจะไม่มีอยู่ใน pic ในตระกูลนี้ที่มีขนาดขา 28 ขา

PORTD จะเป็น port ขนาด 8 bits ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของ port ให้เป็น input หรือ output ได้โดยอิสระจากกัน PORTD สามารถที่จะทำตัวเป็น parallel slave port ได้อีกด้วย โดยทำได้โดยการ set PSPMODE bit (TRISE<4>) ซึ่งใน mode นี้ buffer ภายในจะกลายเป็นแบบ TTL

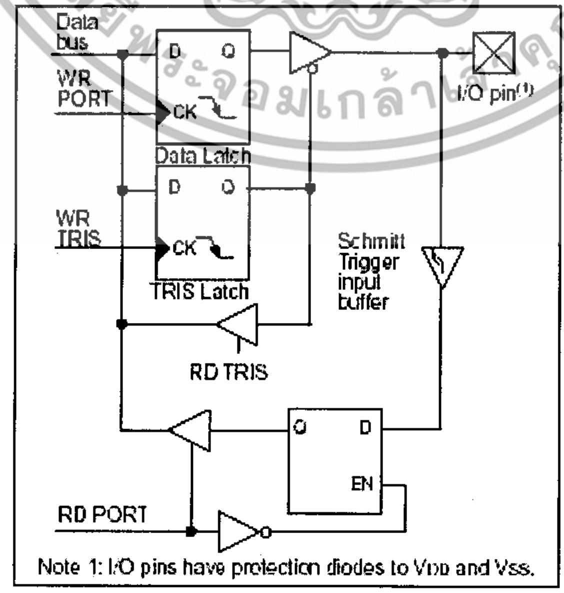
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 บล็อกไดอะแกรม PORTD (ในโหมด พอร์ท I/O)

PORTE จะมีทั้งหมด 3 ขา คือ RE/(RD\)/AN5, RE1/(WR\)/AN6 และ RE2(CS\)/AN7 ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของ port ให้เป็น input หรือ output I/O PORT สามารถกลายเป็น control input สำหรับ microprocessor port เมื่อ ทำการ set PSPMODE (TRISE<4>) bit ข้อควรระวังเมื่อยู่ในโหมดนี้ก็คือ ต้องตรวจดูให้ดีว่า TRISE ตั้งแต่บิต 0-2 ถูก set และต้องแน่ใจว่า ADCON1 ถูก set ให้อยู่ใน mode digital I/O ซึ่งใน mode นี้ input buffer จะเป็น TTL

PORTE จะมีลักษณะคือ จะ multiplex กับ analog inputs โดยเมื่อ PORTE ถูก set เป็น analog inputs แล้ว ขาเหล่านี้เมื่อทำการอ่านค่าจะมีค่าเป็น 0 ส่วน TRISE ซึ่งเป็น control register นั้นจะต้อง set ให้เป็น input เมื่อ set ให้อยู่ใน mode analog input



รูปที่ 2.25 บล็อกไดอะแกรม PORTE (ในโหมด พอร์ท I/O)

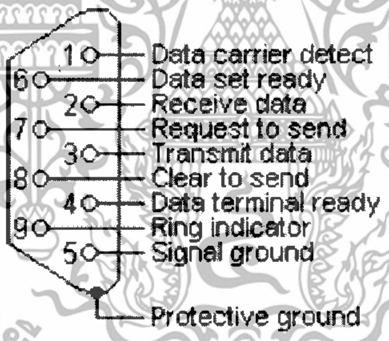
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 พอร์ตอนุกรมมาตรฐาน RS-232

ปัจจุบันการเชื่อมต่อคอมพิวเตอร์ด้วย RS-232 นั้นได้รับความนิยมสูง เพราะมีความง่ายดายไม่ยุ่งยากในการใช้งาน ในบทนี้จะกล่าวถึงรายละเอียดของการเชื่อมต่อคอมพิวเตอร์ด้วยการสื่อสารแบบอนุกรมมาตรฐาน RS-232

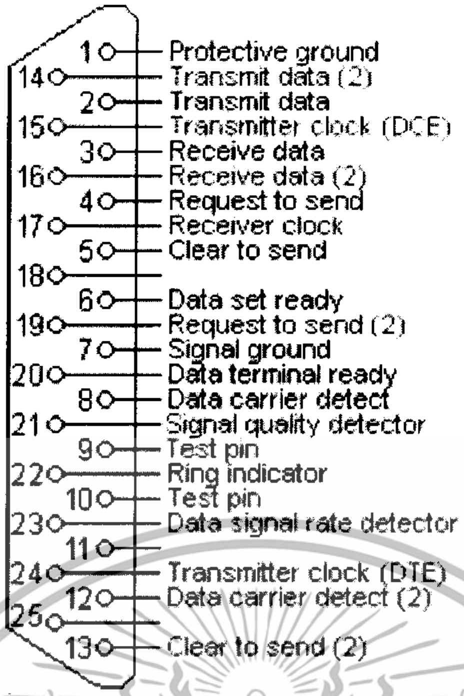
2.4.1 RS-232 serial connecter pin assignment

RS-232 คอนเน็คเตอร์ (Connecter) แต่เดิมนั้นใช้คอนเน็คเตอร์แบบ 25 ขาหรือเรียกว่าคอนเน็คเตอร์แบบ DB25 ออกแบบมาสำหรับการสื่อสารแบบอนุกรม RS-232 ถึง 2 ช่องการสื่อสาร (Channel) โดยมช่องการสื่อสารช่องหนึ่งที่มีระบบตรวจสอบความถูกต้อง (Handshake) ติดมาด้วย และมีคอมพิวเตอร์บางรุ่นที่มีระบบตรวจสอบความถูกต้องทั้ง 2 ช่องการสื่อสารเลย ส่วนอีกช่องการสื่อสารเป็นการสื่อสารแบบโมเด็ม (Telebit Modem) ซึ่งสามารถเช็คสถานะของโมเด็มว่าพร้อมหรือไม่พร้อมในการสื่อสารสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป (Personal Computer) ในปัจจุบัน RS-232 คอนเน็คเตอร์จะเป็นแบบเล็ก 9 ขาหรือเรียกว่า DB9 ดังแสดงในรูปที่ 1 – 3 โดยขาสีดำจะเป็นการสื่อสารแบบโมเด็มส่วนสีแดงจะเป็นการสื่อสารแบบมีระบบตรวจสอบความถูกต้อง ขา Protective Ground ใน DB25 จะเป็นส่วนเปลือกของคอนเน็คเตอร์แบบ DB9 เพื่อเป็นการป้องกันสัญญาณลบกวน

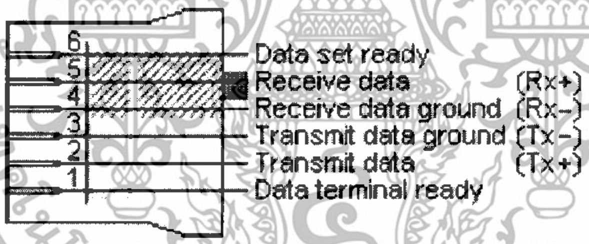


รูปที่ 2.26 RS-232 คอนเน็คเตอร์แบบ DB9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



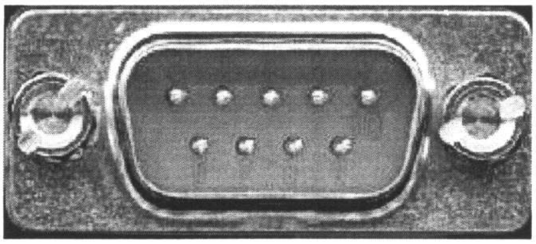
รูปที่ 2.27 RS-232 คอนเน็คเตอร์แบบ DB25



รูปที่ 2.28 RS-232 คอนเน็คเตอร์แบบ DCE Modify modular jack

นอกจากนั้นยังมีคอนเน็คเตอร์อีกแบบหนึ่งเรียกว่า DCE Modify modular jack ดังรูปที่ 2.3 การสื่อสารแบบอนุกรมมาตรฐาน RS-232 มีระดับแรงดันที่เป็นบวกและลบ (ไม่เป็น 0) ในการเชื่อมต่อนั้นจะต้องใช้กับอุปกรณ์เฉพาะที่มีระดับสัญญาณเหมือนกันเท่านั้น กำหนดให้อุปกรณ์ต้นทางคือ **DTE**: data terminal equipment เช่น คอมพิวเตอร์, ปริ้นเตอร์ เป็นต้น และอุปกรณ์ปลายทางคือ **DCE**: data communication equipment เช่นโมเด็ม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 RS-232 คอนเน็คเตอร์แบบ DB9 ตัวผู้ที่อยู่ด้านหลังคอมพิวเตอร์

สำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป (Personal Computer) ในปัจจุบันจะเป็น RS-232 คอนเน็คเตอร์แบบเล็ก DB9 ที่หลังคอมพิวเตอร์จะเป็นตัวผู้ ซึ่งมีรายละเอียดของขาดังตารางที่ 2.1

ตารางที่ 2.7 รายละเอียดขาสัญญาณของพอร์ตอนุกรม RS-232

DB25	DB9	Signal	Name	Direction (In/Out)
2	3	TD	Transmit Data	Output
3	2	RD	Receive Data	Input
4	7	RTS	Request to Send	Output
5	8	CTS	Clear to Send	Input
6	6	DSR	Data Set Ready	Input
7	5	SG	Signal Ground	Ground
8	1	CD	Carrier Detect	Input
20	4	DTR	Data Terminal Ready	Output
22	9	RI	Ring Indicator	Input

โดยมีรายละเอียดหน้าที่การทำงานดังต่อไปนี้

- TD: Transmit Data ใช้สำหรับส่งข้อมูลออก
- RD: Receive Data ใช้สำหรับรับข้อมูลเข้า
- RTS: Request to Send บอกให้ปลายทางเตรียมรับข้อมูล
- CTS: Clear to Send สำหรับตรวจสอบความพร้อมของอุปกรณ์ปลายทางที่จะรับข้อมูล
- DSR: Data Set Ready ใช้ตรวจสอบการเชื่อมต่อ
- SG: Signal Ground กราวด์
- CD: Carrier Detect จะ Active เมื่อมีสัญญาณ Carrier จากโมเด็ม
- DTR: Data Terminal Ready ใช้บอกอุปกรณ์ปลายทางว่าต้องการติดต่อ
- RI: Ring Indicator จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

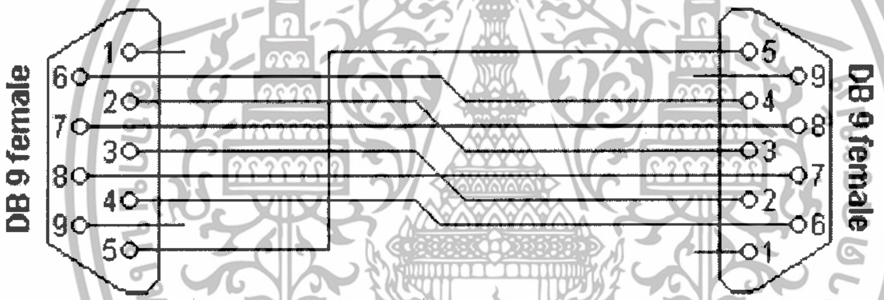
2.4.2 รูปแบบของการเชื่อมต่อ

สำหรับ RS-232 ที่เป็นการสื่อสารแบบโมเด็มนั้น สามารถที่จะประยุกต์การเชื่อมต่อระหว่างอุปกรณ์ 2 ตัวโดยใช้สาย RS-232 ซึ่งสามารถรองรับรูปแบบการเชื่อมต่อมากมาย เช่น สามารถใช้สายสัญญาณเพียง 3 เส้น คือ RD, TD และ SG ก็เพียงพอสำหรับการเชื่อมต่อแล้ว โดยรูปแบบการเชื่อมต่อนั้นขึ้นอยู่กับ Software ที่ใช้ ซึ่งบาง Software อาจจะไม่จำเป็นต้องมีการตรวจสอบความผิดพลาดเลย หรือการสื่อสารระหว่างคอมพิวเตอร์ (Windows 95, 98, ME) การเชื่อมต่อแบบ Loop Back Handshaking นั้นเป็นแบบที่เหมาะสมที่สุด

การเชื่อมต่อระหว่างอุปกรณ์ 2 ตัวนั้นมีหลากลายรูปแบบ เช่น Loop Back Handshaking เหมาะสาหรับการเชื่อมต่อคอมพิวเตอร์ หรือ แบบเต็มรูปแบบเหมาะสำหรับระบบใหญ่ 2 ระบบ เป็นต้น ในการเรียนรู้ในบทนี้เราจะใช้การเชื่อมต่อแบบง่ายในการสื่อสารข้อมูล ซึ่งรูปแบบการเชื่อมต่อที่ไปรับความนิยมมีดังนี้

-Full Handshaking

เป็นการใช้งาน RS-232 แบบเต็มรูปแบบ



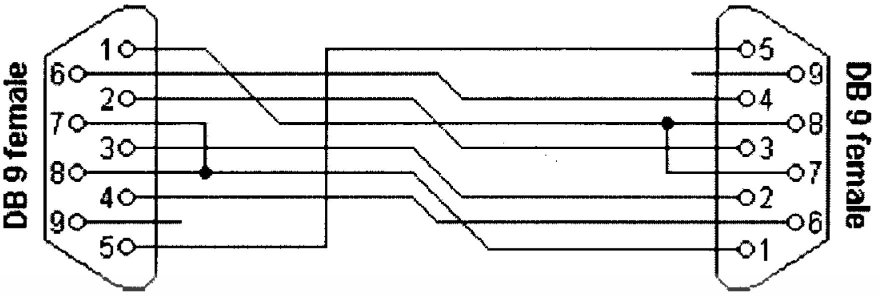
รูปที่ 2.30 ลักษณะการเชื่อมต่อแบบเต็มรูปแบบ

ตารางที่ 2.8 หน้าที่การทำงานของการเชื่อมต่อแบบเต็มรูปแบบ

Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
4	6	DTR → DSR
5	5	Signal ground
6	4	DSR ← DTR
7	8	RTS → CTS
8	7	CTS ← RTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Partial Handshaking

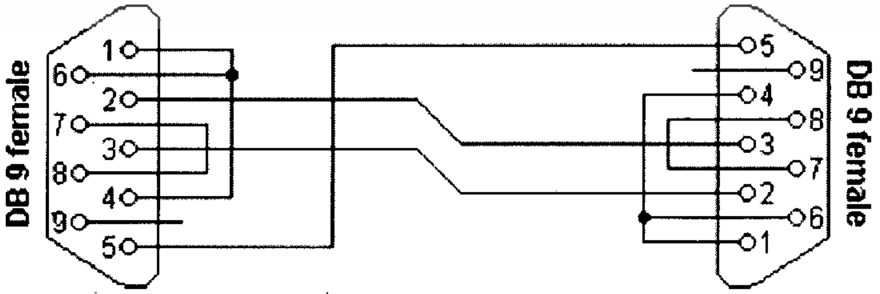


รูปที่ 2.31 ลักษณะการเชื่อมต่อแบบ Partial Handshaking

ตารางที่ 2.9 หน้าที่การทำงานของการเชื่อมต่อแบบ Partial Handshaking

Connector 1	Connector 2	Function
1	7 + 8	RTS2 → CTS2 + CD1
2	3	Rx ← Tx
3	2	Tx →Rx
4	6	DTR → DSR
5	5	Signal ground
6	4	DSR ← DTR
7 + 8	1	RTS1 → CTS1 + CD2

-Loop Back Handshaking



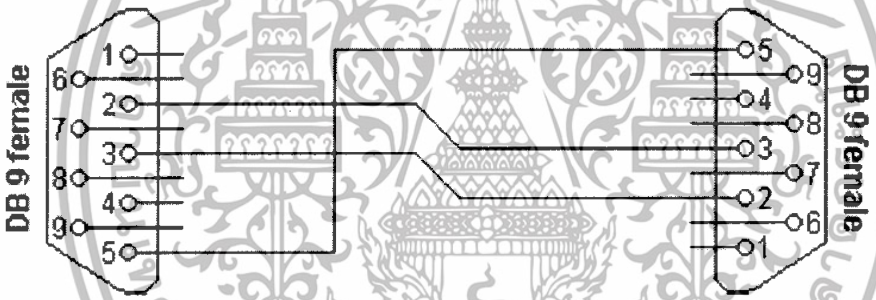
รูปที่ 2.32 ลักษณะการเชื่อมต่อแบบ Loop Back Handshaking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 หน้าที่การทำงานของการเชื่อมต่อแบบ Loop Back Handshaking

Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
5	5	Signal ground
1 + 4 + 6	-	DTR → CD + DTR
-	1 + 4 + 6	DTR → CD + DTR
7 + 8	-	RTS → CTS
-	7 + 8	RTS →CTS

-Simple



รูปที่ 2.33 ลักษณะการเชื่อมต่อแบบง่าย

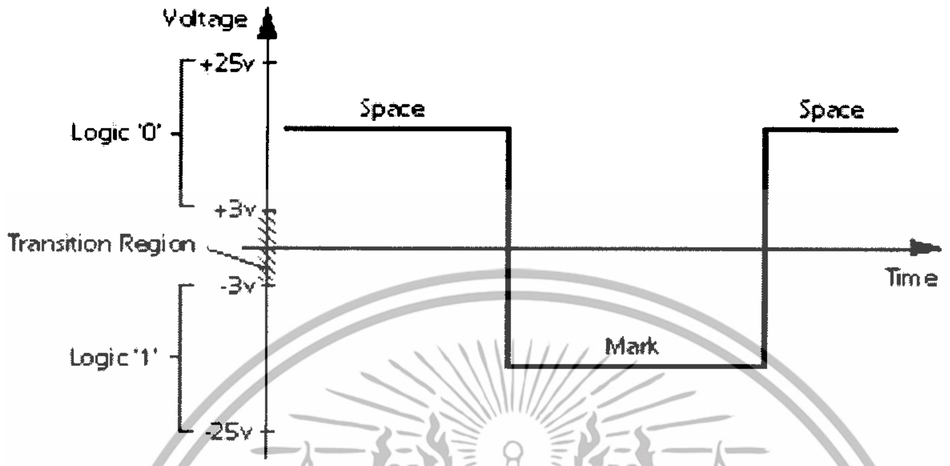
ตารางที่ 2.11 หน้าที่การทำงานของการเชื่อมต่อแบบง่าย

Connector 1	Connector 2	Function
2	3	Rx ← Tx
3	2	Tx → Rx
5	5	Signal ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ลักษณะสัญญาณของอนุกรม RS-232

การสื่อสารแบบอนุกรมมาตรฐาน RS-232 มีระดับแรงดันที่เป็นบวกและลบ (ไม่เป็น 0) ดังแสดงในรูปที่ 2.9 ในการเชื่อมต่อนั้นจะต้องใช้กับอุปกรณ์เฉพาะที่มีระดับสัญญาณเหมือนกันเท่านั้น



รูปที่ 2.34 ลักษณะรูปสัญญาณของอนุกรม RS-232

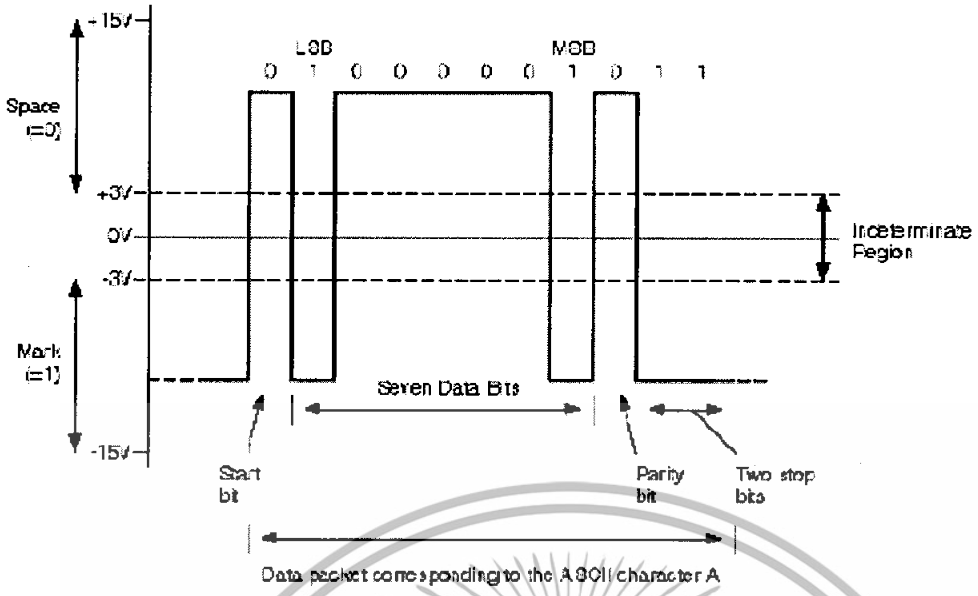
ตารางที่ 2.12 ความสัมพันธ์ของ Logic และระดับแรงดัน

Logic	Voltage
"0"	+3 to +25
"1"	-3 to -25
Unknown	+3 to -3

เนื่องจากสัญญาณรับกวนคือสัญญาณที่ไม่พึงประสงค์ในระบบ โดยทั่วไปสัญญาณรบกวนจะมีระดับแรงดันที่น้อย เพราะฉะนั้น RS-232 จึงเพิ่มระดับแรงดันให้มากกว่าระดับแรงดันของสัญญาณรบกวน เพื่อจุดประสงค์ในการกำจัดสัญญาณรบกวนนั่นเอง

สำหรับสัญญาณของข้อมูลนั้นจะเป็นลักษณะข้อมูลแบบอนุกรมทีละบิต โดยแต่ละบิตจะแสดงตามระยะเวลาที่กำหนด ซึ่งเป็นตัวกำหนดอัตราความเร็วในการส่งข้อมูล (Baud Rate) คือ จำนวนบิตต่อวินาที เช่น 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 256000 ตามมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 ลักษณะสัญญาณของข้อมูลที่มีความสำพันธ์กับแรงดันไฟฟ้า

รายละเอียดของสัญญาณแต่ละบิตมีดังต่อไปนี้

- Start Bit เป็นบิตเริ่มต้นของชุดบิตข้อมูล
- Data Bit เป็นชุดบิตข้อมูล มีขนาด 7 หรือ 8 บิต
- Parity Bit เป็นบิตที่ใช้ตรวจสอบความถูกต้องของข้อมูล
    - o Even Parity เป็น "1" เมื่อจำนวนบิต "1" เป็นเลขคู่
    - o Odd Parity เป็น "1" เมื่อจำนวนบิต "1" เป็นเลขคี่
    - o None จะไม่มีการตรวจสอบความถูกต้อง
- Stop Bit เป็นบิตปิดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 องค์ประกอบโดยรวมของระบบ

ประกอบด้วย 5 ส่วนหลักดังนี้

1.) บอร์ดควบคุมและโมดูลฟิงเกอร์พรินต์เซนเซอร์

2.) ไมโครคอนโทรลเลอร์ PIC16F877

3.) จอแอลซีดี

4.) รีเลย์

5) คีย์แพด

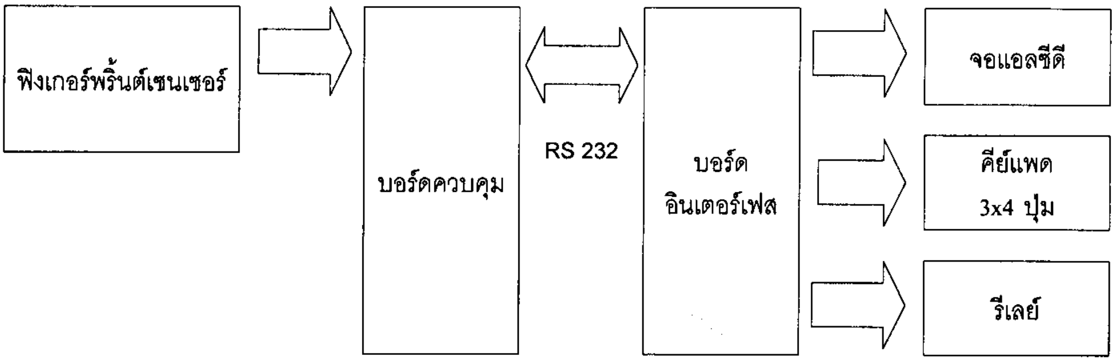
3.2 หลักการทำงานของฮาร์ดแวร์

การออกแบบโครงงานนี้ส่วนของฮาร์ดแวร์และบอร์ดอินเตอร์เฟส มีหน้าที่หลักคือการติดต่อกับบอร์ดควบคุมและโมดูลฟิงเกอร์พรินต์เซนเซอร์และควบคุมการทำงานของจอแอลซีดี บัซเซอร์ รีเลย์ ตลอดจนรับคำสั่งที่ผู้ใช้ป้อนเข้าทางคีย์แพด โดยหัวใจสำคัญที่ควบคุมการทำงานคือไมโครคอนโทรลเลอร์ PIC16F877

จากวงจรจะเห็นได้ว่า PIC16F877 ทำงานเป็นจุดศูนย์กลางที่จะควบคุมการทำงานทั้งหมดบนบอร์ดอินเตอร์เฟส โดยแต่ละพอร์ตของ PIC16F877 ที่ถูกใช้งานอยู่ มีดังนี้ พอร์ต RA3 ต่อกับขาเบสของทรานซิสเตอร์ เพื่อขับรีเลย์ให้ทำงาน โดยมี R(1k โอห์ม) เป็นตัวจำกัดกระแสให้กับพอร์ต และมี R(10 โอห์ม) เป็นตัวจำกัดกระแสให้ขดลวดของรีเลย์ เมื่อนำไปใช้งานเราสามารถใช้รีเลย์ ควบคุมอุปกรณ์ต่อพ่วง เช่น ชุดสลักกลอนประตูไฟฟ้า, สัญญาณเตือนภัย เพื่อป้องกันหรือส่งสัญญาณเตือนเมื่อมีการบุกรุกได้, พอร์ต RE0 ต่อกับขาเบสของทรานซิสเตอร์เพื่อขับการทำงานให้ บัซเซอร์ โดยมี R(1k โอห์ม) ต่อเป็นตัวต้านทานจำกัดกระแสให้กับพอร์ตเช่นกัน และมี R(10 โอห์ม) เป็นตัวจำกัดกระแสให้กับบัซเซอร์ด้วย ในขณะที่ใช้งานบัซเซอร์จะทำหน้าที่ส่งเสียงเตือนเมื่อเกิดเหตุการณ์ต่างๆ เช่น เมื่อการตรวจสอบผิดพลาด, เมื่ออนุญาตให้ผ่านการตรวจสอบได้ เป็นต้น, พอร์ต RB0-RB6 ของ PIC16F877 ต่ออยู่กับคีย์แพดขนาด 3*4 ปุ่ม โดยมี R-Pack(10k โอห์ม) ต่อพลูอัพให้กับทุกขา, พอร์ต RA0-RA2 และ พอร์ต RD0-RD7 ต่ออยู่กับจอแอลซีดีขนาด 8 ตัวอักษร 2 บรรทัด และพอร์ต RC7,RC6 (หรือ RxD, TxD ตามลำดับ) ต่ออยู่กับ DS275 ซึ่งเป็นไอซีสำหรับการอินเตอร์เฟสแบบ RS-232C

ในระหว่างทำงาน PIC16F877 จะสื่อสารกับโมดูลฟิงเกอร์พรินต์เซนเซอร์โดยผ่านตัวกลางคือบอร์ดควบคุม ซึ่งมีหน้าที่ควบคุมการทำงานต่างๆและจัดการกับข้อมูลลายนิ้วมือที่ได้จากเซนเซอร์ ทั้งในส่วนของการรู้จำลายนิ้วมือและการจัดเก็บข้อมูลในรูปของเท็มเพลตลงยังหน่วยความจำแบบแฟลชที่อยู่บนบอร์ดเอง การสื่อสารข้อมูลระหว่าง PIC16F877 กับบอร์ดควบคุม ใช้มาตรฐานทางไฟฟ้าแบบ RS-232C เป็นหลัก โดยมี DS275 เป็นตัวกลางในการอินเตอร์เฟส

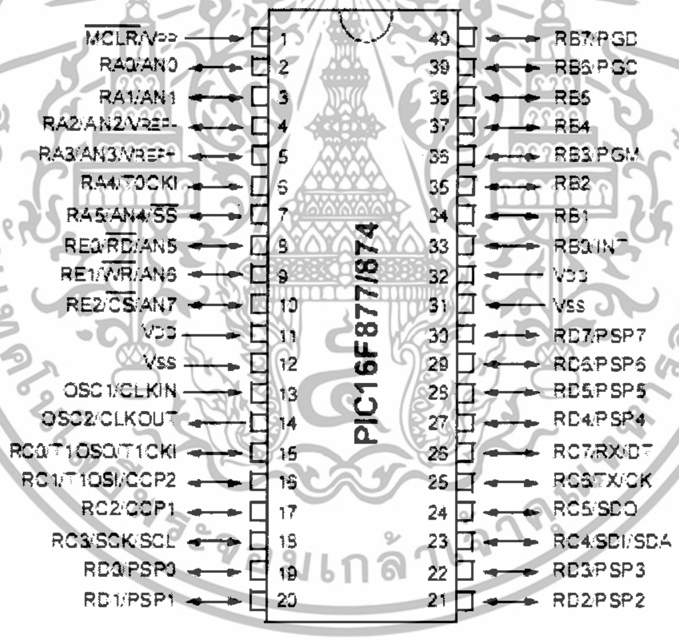
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ภาพโดยรวมของระบบ

3.2.1 ไมโครคอลโทรเลอร์

ในการทดลองนี้ใช้ไมโครคอลโทรลเลอร์ PIC16F877 เป็นส่วนประมวลผลหลักของทั้งวงจร ไมโครคอลโทรลเลอร์ PIC16F877 ซึ่งควบคุมการทำงานทั้งวงจร ทำงานที่ความถี่ 20 MHz



รูปที่ 3.2 ไมโครคอลโทรลเลอร์ (PIC16F877)

3.2.3 วงจร DS275

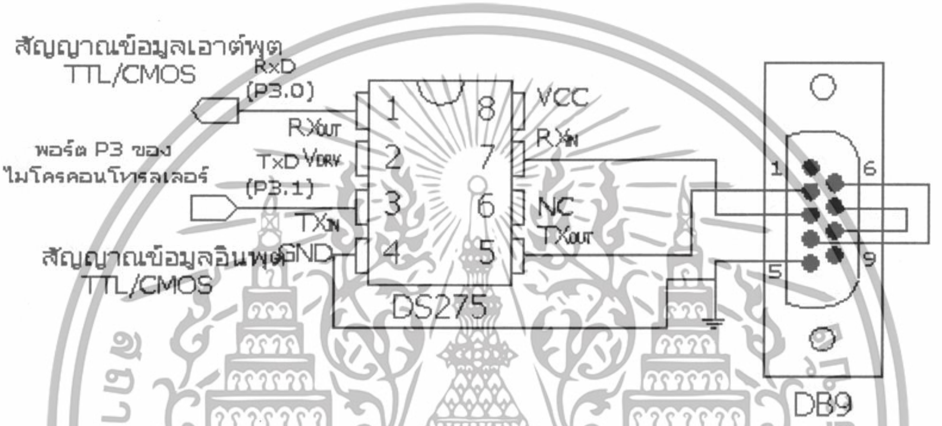
การเชื่อมต่อระหว่างเครื่องแสกนลายนิ้วมือกับไมโครคอนโทรลเลอร์ (PIC16F877) ต้องทำการเชื่อมต่อกันโดยผ่านพอร์ตสื่อสารอนุกรม เนื่องจากสัญญาณจากเครื่องแสกนลายนิ้วมือนั้นไม่สามารถเชื่อมต่อกับกับไมโครคอนโทรลเลอร์ (PIC16F877) ได้โดยตรง ดังนั้นในการเชื่อมต่อจึงต้องทำการเปลี่ยนระดับสัญญาณที่ได้จากเครื่องแสกนลายนิ้วมือให้อยู่ในมาตรฐานการสื่อสารพอร์ตอนุกรมก่อน ซึ่งใช้วงจรอินเตอร์เฟสโดยใช้ไอซี DS275 ในการเชื่อมต่อ ซึ่งวงจร DS275 นี้จะทำหน้าที่แปลงสัญญาณจากเครื่องแสกนลายนิ้วมือซึ่งเป็นสัญญาณ RS232ให้เป็น TTL/CMOS เพื่อส่งไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(PIC16F877) และขณะเดียวกันก็แปลงสัญญาณ TTL/CMOS จากไมโครคอนโทรลเลอร์ (PIC16F877) ให้เป็นสัญญาณ RS 232 เพื่อส่งกลับไปยังเครื่องแสกนลายนิ้วมือ

ตารางที่ 3.1 เปรียบเทียบแรงดันไฟฟ้าระหว่างสัญญาณ TTL กับ RS 232

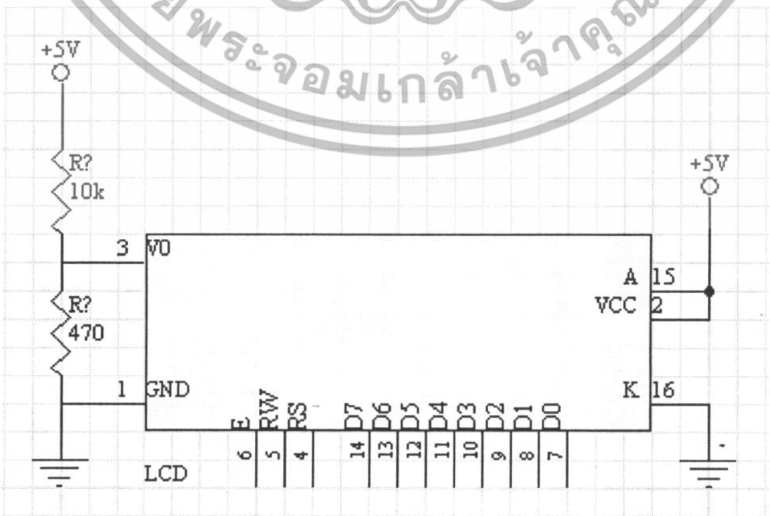
TTL	RS 232 (V)
0	+3 to +25
1	-3 to -25



รูปที่ 3.3 การต่อ DS275

3.2.4 LCD

ในโครงงานนี้ใช้ LCD ขนาด 2x8 คือแสดงผล LCD แบบ 2 บรรทัด บรรทัดละ 8 ตัวอักษร

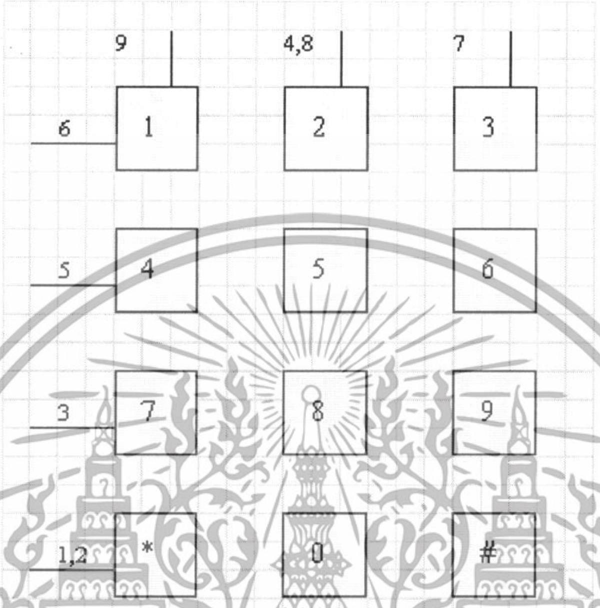


รูปที่ 3.4 วงจร LCD ขนาด 2x8 ที่ต่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

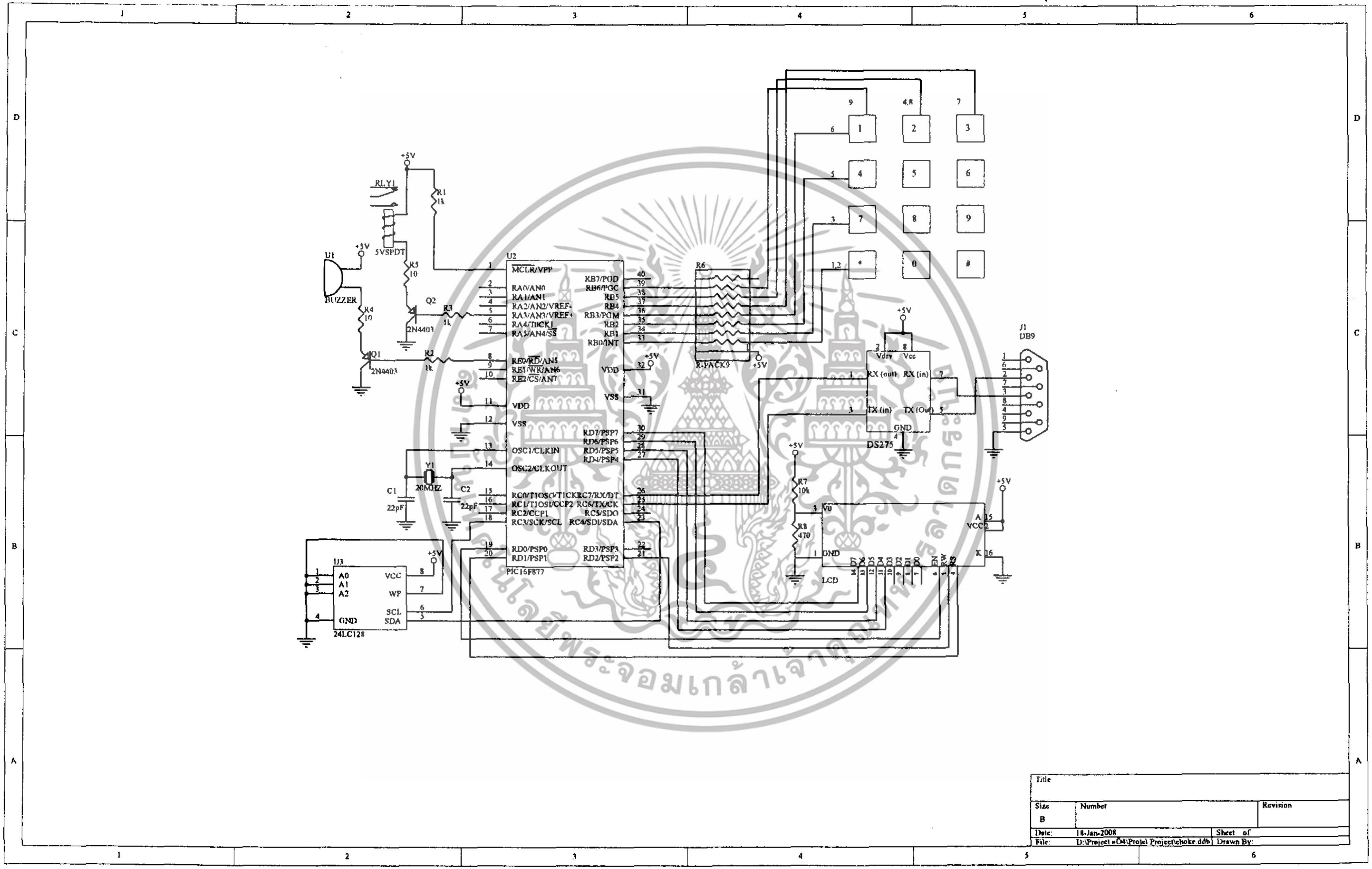
3.2.5 Keypad

ในโครงงานนี้ใช้ Keypad แบบ 4x3

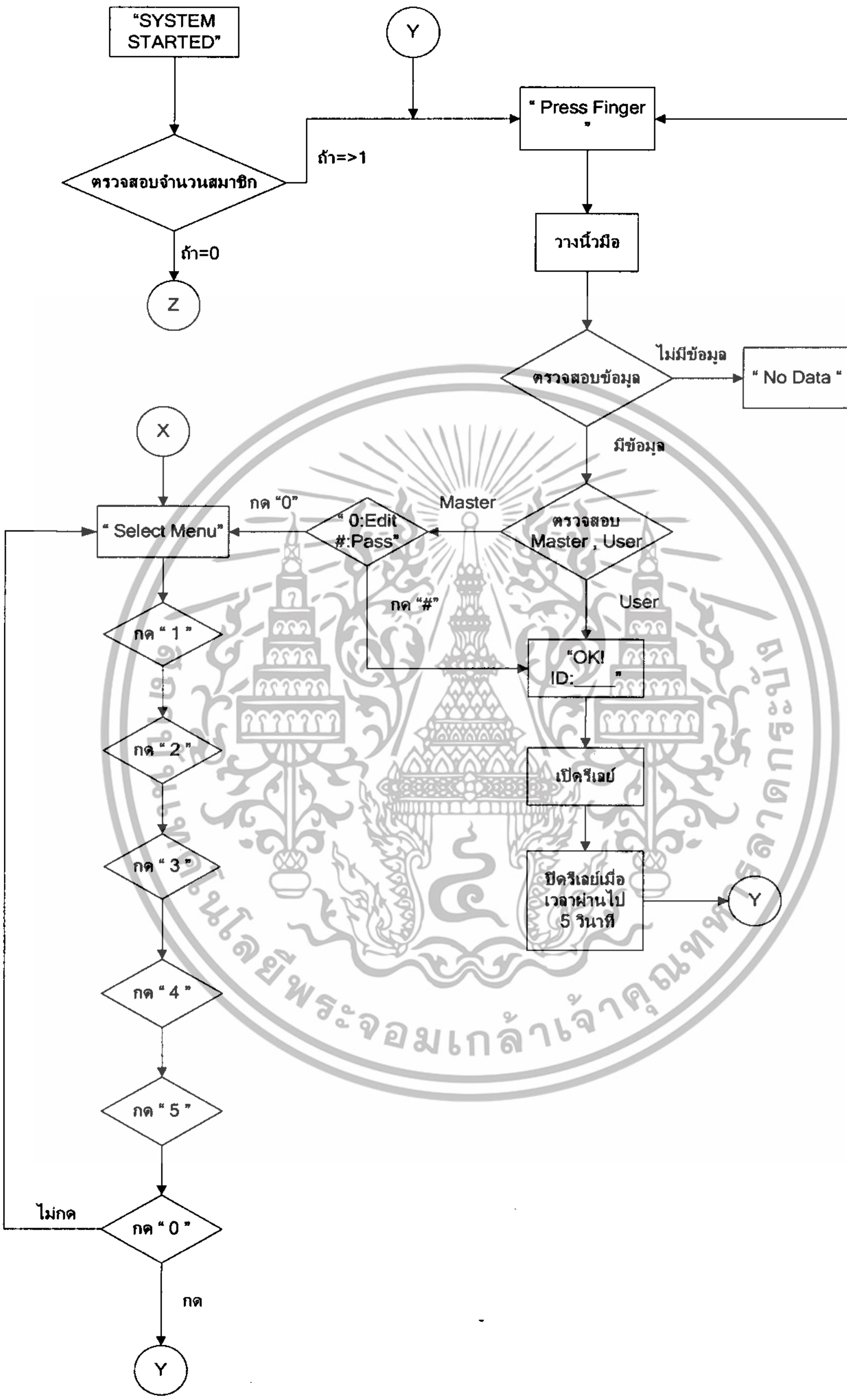


รูปที่ 3.5 Keypad แบบ 4x3 ที่ต่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

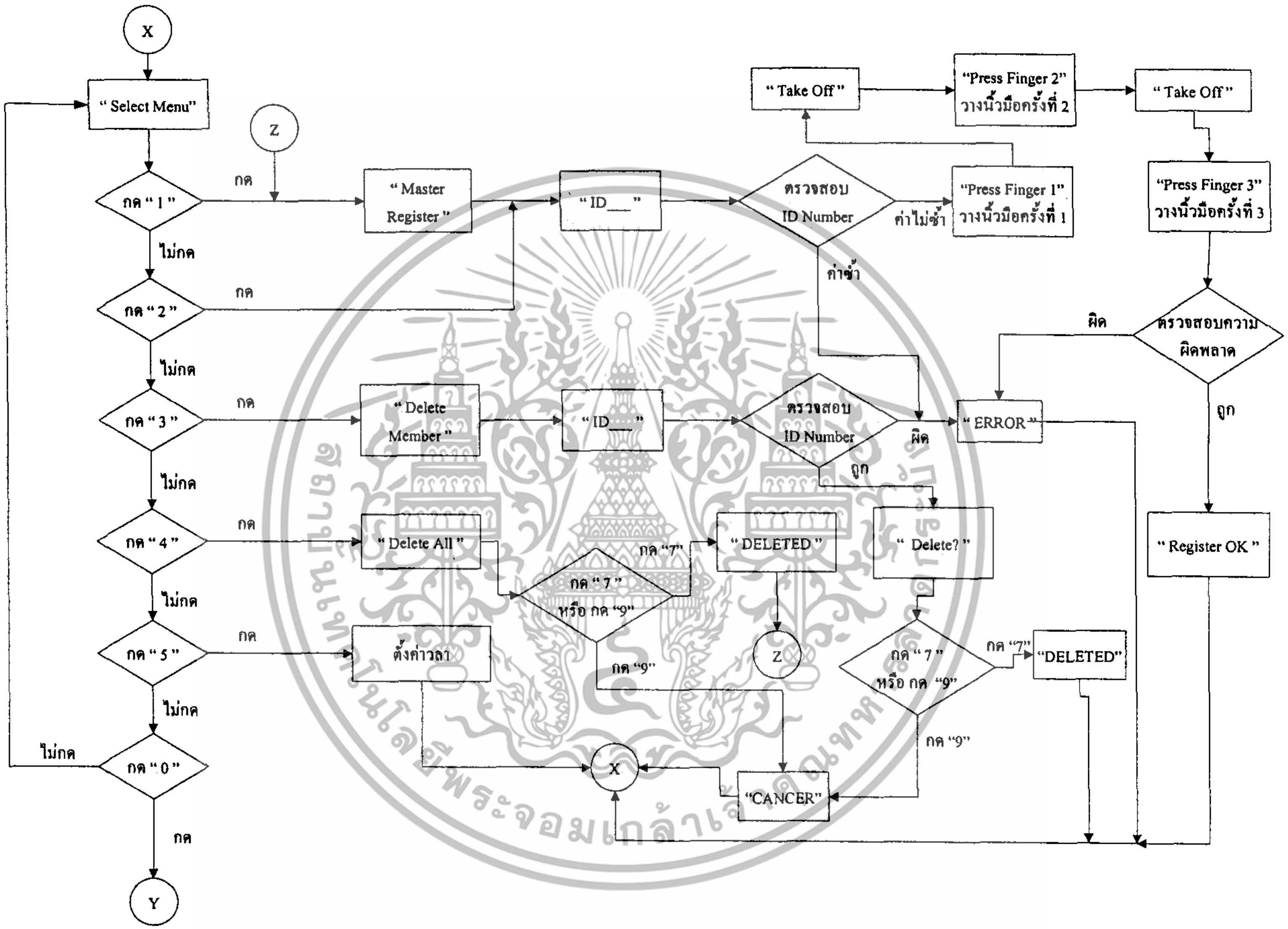

U1
BUZZER
+5V
R4
10
Q1
2N4403
RLY1
5VSPDT
R1
1k
R5
10
Q2
2N4403
R3
1k
R2
1k
U2
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
VDD
VSS
OSC1/CLKIN
OSC2/CLKOUT
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RD0/PSP0
RD1/PSP1
RB7/PGD
RB6/PGC
RB5
RB4
RB3/PGM
RB2
RB1
RB0/INT
RD7/PSP7
RD6/PSP6
RD5/PSP5
RD4/PSP4
RC7/RX/DT
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
RD3/PSP3
RD2/PSP2
PIC16F877
Y1
20MHZ
C1
22pF
C2
22pF
U3
A0
A1
A2
GND
VCC
WP
SCL
SDA
24LC128
R6
R-PACK9
1
2
3
4
5
6
7
8
9
*
0
#
Vdrv
Vcc
RX (out)
RX (in)
TX (in)
TX (Out)
GND
DS275
J1
DB9
R7
10k
R8
470
V0
GND
A
VCC2
K
LCD
D7
D6
D5
D4
D3
D2
D1
D0
EN
RW
RS
Title
Size
B
Number
Revision
Date:
18-Jan-2008
Sheet of
File:
D:\Project ปี4\Protel Project\choke.ddb
Drawn By:

รูปที่ 3.6 วงจรที่ออกแบบที่ใช้ในโครงงานนี้**Flowchart การทำงานของวงจร**



รูปที่ 3.7 Flowchart การทำงานของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Flowchart การทำงานของวงจร (ต่อ)

3.3 หลักการทำงานของซอร์ฟแวร์

หน้าที่การทำงานของซอร์ฟแวร์ โดยหลักๆก็คือ การติดต่อกับอุปกรณ์ คีย์แพดและจอแอลซีดี ตลอดจนการสื่อสารกับบอร์ดควบคุม เพื่อควบคุมการทำงานฟิงเกอร์พรินต์เซนเซอร์และจัดการกับข้อมูลของผู้ใช้แต่ละราย โดยการทำงานจะแบ่งออกเป็น 4 หมวดใหญ่ๆ คือ การลงทะเบียนผู้ใช้, การยืนยันตัวผู้ใช้กับเครื่อง, การลบผู้ใช้ออกจากระบบ และการดูข้อมูลปัจจุบันค่าต่างๆของเครื่อง

สำหรับหน้าที่การทำงานของแต่ละปุ่มบนคีย์แพดมีดังนี้

- ปุ่มหมายเลข 1 ใช้ลงทะเบียนสำหรับผู้ที่ใช้ที่เป็นมาสเตอร์ (Master)
- ปุ่มหมายเลข 2 ใช้ลงทะเบียนสำหรับผู้ใช้ทั่วไป (User)
- ปุ่มหมายเลข 3 ใช้ลบข้อมูลออกเป็นรายบุคคล
- ปุ่มหมายเลข 4 ใช้ลบข้อมูลทั้งหมดที่เคยบันทึกไว้ออกจากฐานข้อมูล
- ปุ่มหมายเลข 5 ใช้ตั้งเวลา
- ปุ่มหมายเลข 6 ใช้ลบข้อมูลใน EEPROM
- ปุ่มหมายเลข 7 ใช้เรียกดูข้อมูลใน EEPROM , เลือก YES
- ปุ่มหมายเลข 9 ใช้เลือก NO
- ปุ่มหมายเลข 0 ใช้ออกจากระบบ , เลือก EDIT
- ปุ่ม * ใช้เลื่อนไปทางซ้าย
- ปุ่ม # ใช้เลื่อนไปทางขวา , เลือก PASS

สำหรับผู้ใช้งานคนแรก (Master 1) โดยหน้าจอแอลซีดีจะแสดงข้อความ “Master Register” ต่อจากนั้นหน้าจอแอลซีดีจะแสดงข้อความ “Input ID Number” ซึ่งหมายความว่าเครื่องต้องการให้ป้อนค่าหมายเลข 4 หลัก ซึ่งก็คือหมายเลขประจำตัวผู้ใช้นั่นเอง ต่อจากนั้นระบบจะทำการตรวจสอบค่า ID Number ถ้าค่าซ้ำหน้าจอแอลซีดีจะสแดงข้อความ “Fail” หลังจากนั้นหน้าจอแอลซีดีก็จะกลับมาแสดงข้อความ “Select Menu” อีกครั้งหนึ่ง เพื่อรอรับการทำงานต่อไป แต่ถ้าค่าไม่ซ้ำ หน้าจอแอลซีดีจะแสดงข้อความ “Press Finger 1” ซึ่งเป็นการวางนิ้วมือครั้งที่ 1 บนฟิงเกอร์พรินต์เซนเซอร์โดยให้วางค้างไว้ประมาณ 1-2 นาที แล้วยกนิ้วมือขึ้นจากเซนเซอร์ ต่อจากนั้นจอแอลซีดี ก็จะแสดงข้อความ “Press Finger 2” เพื่อให้วางนิ้วมือที่วางลงไปก่อนหน้านี้ที่ฟิงเกอร์พรินต์เซนเซอร์ในลักษณะเดิมอีกครั้ง เครื่องจะทำการตรวจสอบความผิดพลาด ถ้าไม่ผิดพลาดจอแอลซีดีจะแสดงข้อความ “Register OK” แต่ถ้าผิดพลาดจอแอลซีดีจะแสดงข้อความ “No Finger”ขึ้นแทน หลังจากนั้นหน้าจอแอลซีดีจะแสดงข้อความ “Press Finger 1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การลงทะเบียน

ผู้เป็นมาสเตอร์คนแรกวางลายนิ้วมือบนฟิงเกอร์พรินต์เซนเซอร์ จากนั้นระบบจะทำการตรวจสอบข้อมูล จากนั้นให้กดปุ่ม "0" หน้าจอแอลซีดีจะแสดงข้อความ "Select Menu"

- กดปุ่มหมายเลข 1 หากต้องการลงทะเบียนผู้ใช้ที่เป็นมาสเตอร์
- กดปุ่มหมายเลข 2 หากต้องการลงทะเบียนผู้ใช้ทั่วไป

หลังจากนั้นหน้าจอแอลซีดีจะแสดงข้อความ "Input ID Number" ซึ่งหมายความว่าเครื่องต้องการให้ป้อนค่าหมายเลข 4 หลัก ซึ่งก็คือหมายเลขประจำตัวผู้ใช้นั่นเอง ต่อจากนั้นระบบจะทำการตรวจสอบค่า ID Number ถ้าค่าซ้ำหน้าจอแอลซีดีจะสแดงข้อความ "Fail" หลังจากนั้นหน้าจอแอลซีดีก็จะกลับมาแสดงข้อความ "Select Menu" อีกครั้งหนึ่ง เพื่อรอรับการทำงานต่อไป แต่ถ้าค่าไม่ซ้ำ หน้าจอแอลซีดีจะแสดงข้อความ "Press Finger 1" ซึ่งเป็นการวางนิ้วมือครั้งที่ 1 บนฟิงเกอร์พรินต์เซนเซอร์โดยให้วางค้างไว้ประมาณ 1-2 นาที แล้วยกนิ้วมือขึ้นจากเซนเซอร์ ต่อจากนั้นจอแอลซีดี ก็จะแสดงข้อความ "Press Finger 2" เพื่อให้วางนิ้วมือที่วางลงไปก่อนหน้านี้ที่ฟิงเกอร์พรินต์เซนเซอร์ในลักษณะเดิมอีกครั้ง เครื่องจะทำการตรวจสอบความผิดพลาด ถ้าไม่ผิดพลาดจอแอลซีดีจะแสดงข้อความ "Register OK" แต่ถ้าผิดพลาดจอแอลซีดีจะแสดงข้อความ "Fail" ขึ้นแทน หลังจากลงทะเบียนเรียบร้อยแล้ว หน้าจอแอลซีดีจะกลับมาแสดงข้อความ "Select Menu" เพื่อรอรับการทำงานต่อไป

3.3.2 การเข้าใช้งานระบบ

หน้าจอแอลซีดีจะแสดงข้อความ "Press Finger" ให้วางนิ้วมือ (นิ้วที่ได้ลงทะเบียนเอาไว้) ลงบนผิวหน้าของเซนเซอร์โดยให้วางค้างไว้ประมาณ 1-2 นาทีแล้วจึงยกนิ้วมือออก ระบบจะทำการตรวจสอบข้อมูล ว่ามีข้อมูลในฐานข้อมูลหรือไม่ ถ้าไม่มีข้อมูลในฐาน หน้าจอแอลซีดีก็จะแสดงข้อความ "No Data" แต่ถ้ามีข้อมูลในฐานข้อมูล ระบบก็จะทำการตรวจสอบอีกครั้งหนึ่งว่าเป็น มาสเตอร์ หรือ ผู้ใช้ทั่วไป ถ้าเป็นผู้ใช้ทั่วไป ระบบจะทำการเปิดรีเลย์ ส่วนผู้ที่เป็นมาสเตอร์ ก็จะให้เลือกกด "0" หรือไม่กด ถ้ากด "0" หน้าจอแอลซีดีจะแสดงข้อความ "Select Menu" เพื่อรอรับการทำงานต่อไป แต่ถ้าไม่กด "0" ระบบจะทำการเปิดรีเลย์

3.3.3 การลบข้อมูลผู้ใช้จากระบบ

สำหรับวิธีการลบผู้ใช้สามารถทำโดยกดปุ่มหมายเลข 3 สำหรับการลบข้อมูลออกทีละรายการ หน้าจอแอลซีดีจะแสดงข้อความ "Delete Member" จากนั้นหน้าจอแอลซีดีจะแสดงข้อความ "Input ID Number" ซึ่งหมายความว่าเครื่องต้องการให้ป้อนค่าหมายเลข 4 หลัก ซึ่งก็คือหมายเลขประจำตัวผู้ใช้ที่ต้องการลบนั่นเอง ต่อจากนั้นระบบจะทำการตรวจสอบค่า ID Number ถ้าไม่ถูกต้องหน้าจอแอลซีดีก็จะแสดงข้อความ "Fail" หน้าจอแอลซีดีจะกลับมาแสดงข้อความ "Select Menu" เพื่อรอรับการทำงานต่อไป แต่ถ้าถูกต้องหน้าจอแอลซีดีก็จะแสดงข้อความ "Confirm Delete" ให้กดหมายเลข "7" เพื่อทำการยืนยัน จากนั้นหน้าจอแอลซีดีก็จะแสดงข้อความ "Delete OK" แต่ถ้าไม่ต้องการลบก็ให้กดหมายเลข "9" หน้าจอแอลซีดีก็จะกลับมาแสดงข้อความ "Select Menu" อีกครั้งหนึ่ง เพื่อรอรับการทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือกดปุ่มหมายเลข 4 เพื่อทำการลบข้อมูลทั้งหมดที่เคยบันทึกไว้ออกจากฐานข้อมูล หน้าจอแอลซีดีจะแสดงข้อความ “Delete All” ให้กดหมายเลข “7” เพื่อทำการยืนยัน จากนั้นหน้าจอแอลซีดีก็จะแสดงข้อความ “Delete All OK” ถ้าไม่ต้องการลบให้ก็ให้กดหมายเลข “9” หน้าจอแอลซีดีก็จะแสดงข้อความ “Master Register” คือการกลับไปเริ่มต้นใช้เครื่องฟิงเกอร์พรินต์ใหม่อีกครั้งหนึ่ง

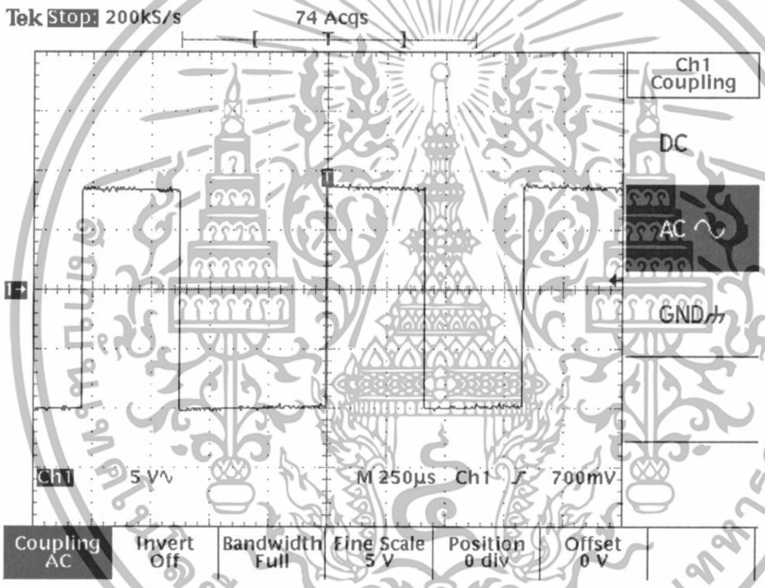
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ขั้นตอนแรกทำการต่อบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์กับวงจรอินเตอร์เฟส DS275 วัดดูรูปสัญญาณว่าเป็นสัญญาณ RS 232 หรือเป็นสัญญาณ TTL/CMOS

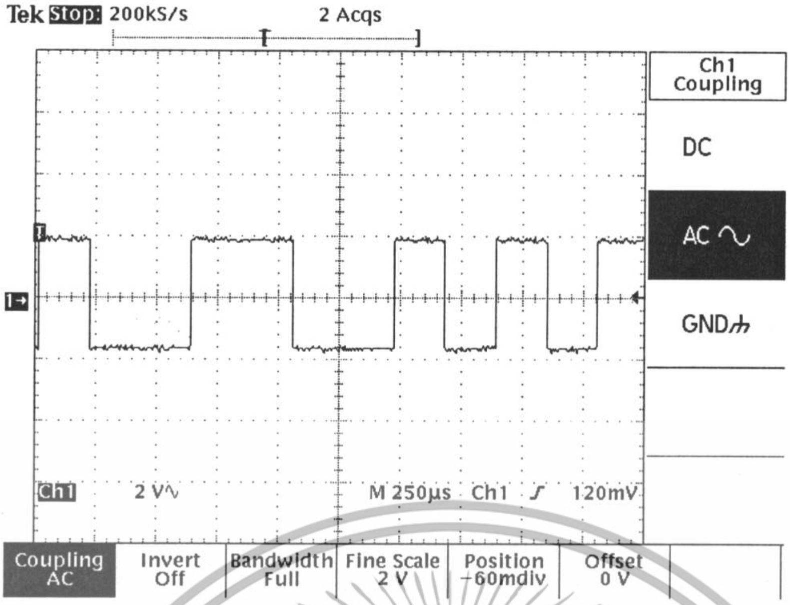
4.1 การตรวจสอบการทำงานของบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์

เราทำการตรวจสอบสัญญาณที่ส่งออกมาจากบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์และ สัญญาณที่ออกมาจาก DS275

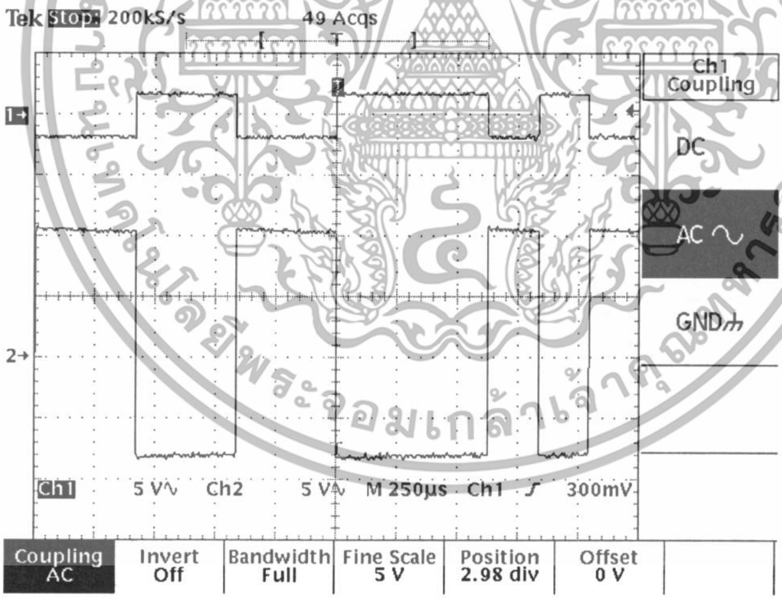


รูปที่ 4.1 วัดสัญญาณที่ออกจากบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 วัดสัญญาณที่ออกมาจาก DS275

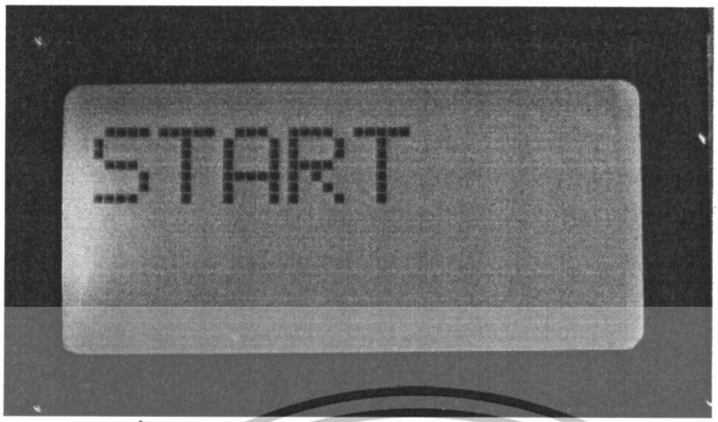


รูปที่ 4.3 วัดสัญญาณเปรียบเทียบระหว่าง
สัญญาณที่ออกจาก DS275 และ บอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์

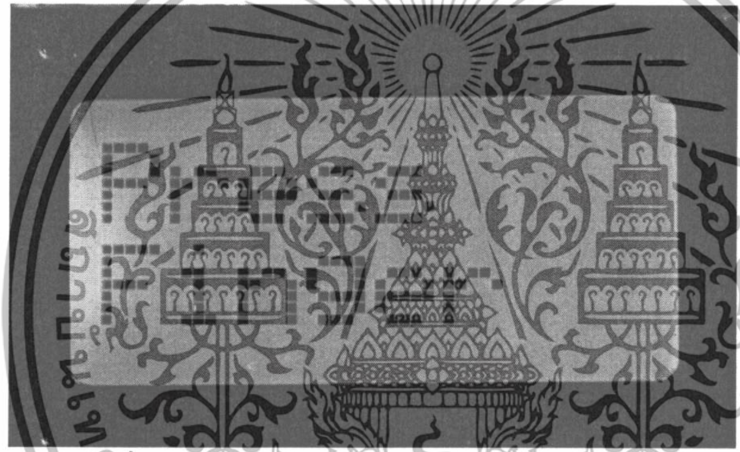
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำการเขียนโปรแกรมไมโครคอนโทรลเลอร์ PIC16F877

1.เริ่มต้นการทำงานหน้าจอแอลซีดีจะแสดงข้อความ

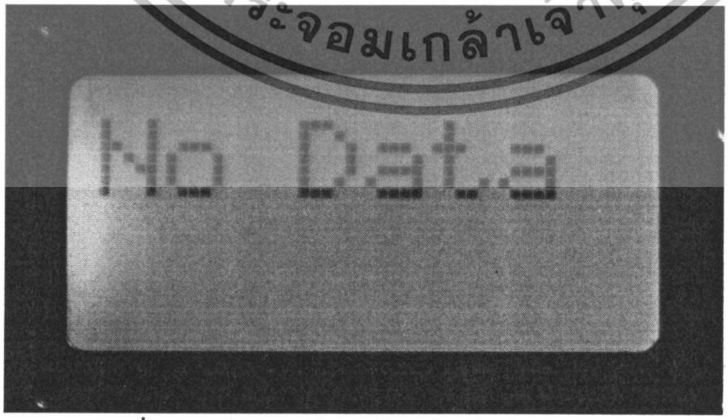


รูปที่ 4.4 หน้าจอแอลซีดีแสดงข้อความ “START”



รูปที่ 4.5 หน้าจอแอลซีดีแสดงข้อความ “Press Finger”

2.หลังจากวางนิ้วมือระบบจะตรวจสอบข้อมูล ถ้าไม่มีข้อมูลหน้าจอแอลซีดีจะแสดงข้อความ



รูปที่ 4.6 หน้าจอแอลซีดีแสดงข้อความ “No Data”

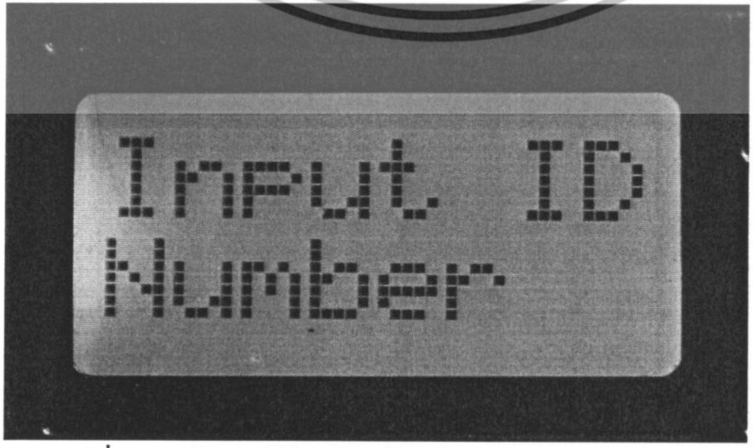
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นระบบจะตรวจสอบว่าเป็น Master หรือ User ถ้าเป็น Master และกดศูนย์หน้าจอแอลซีดีจะแสดงข้อความ

รูปที่ 4.7 หน้าจอแอลซีดีแสดงข้อความ “Select Menu”

4. ถ้ากด “1” หน้าจอแอลซีดีจะแสดงข้อความ

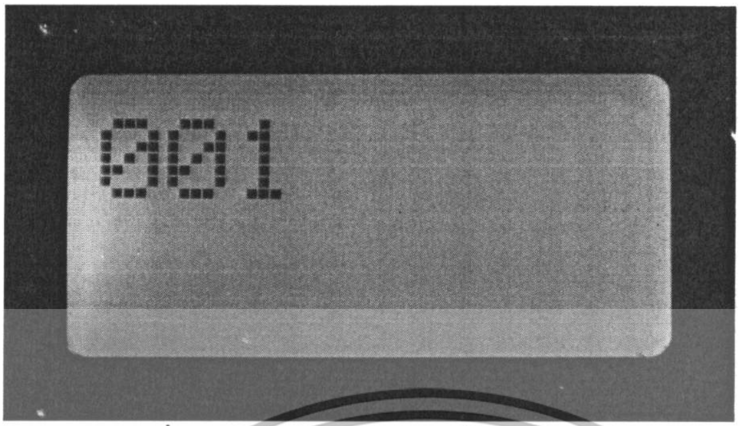
รูปที่ 4.8 หน้าจอแอลซีดีแสดงข้อความ “Master Register”



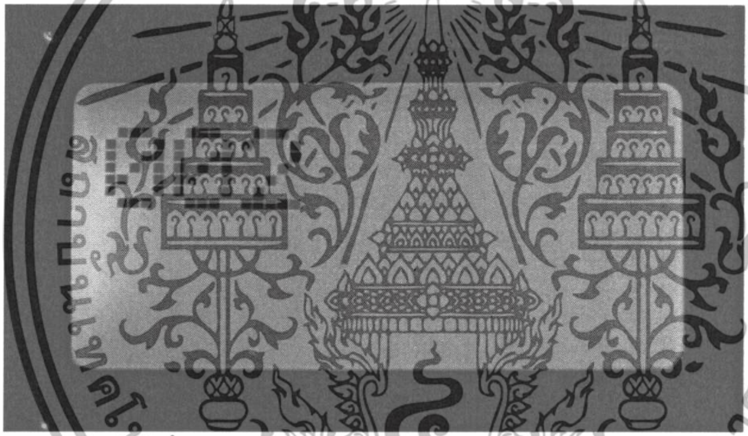
รูปที่ 4.9 หน้าจอแอลซีดีแสดงข้อความ “Input ID Number”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

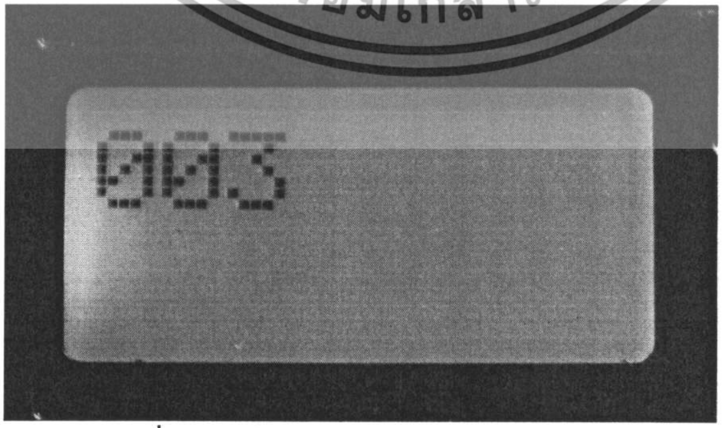
ตัวอย่างเมื่อกด ID Number



รูปที่ 4.10 หน้าจอแอลซีดีแสดงข้อความ “001”



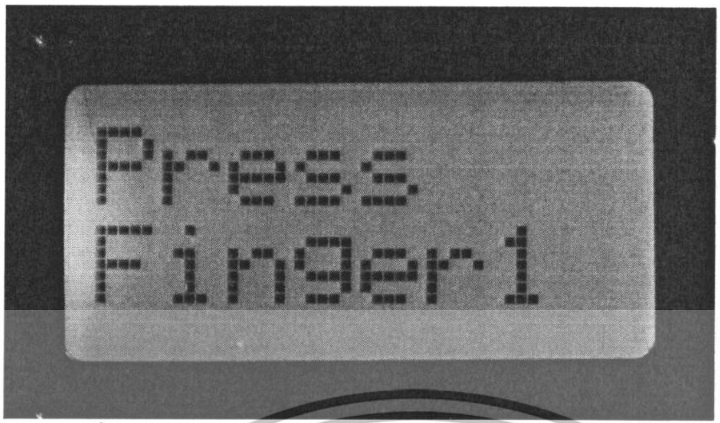
รูปที่ 4.11 หน้าจอแอลซีดีแสดงข้อความ “002”



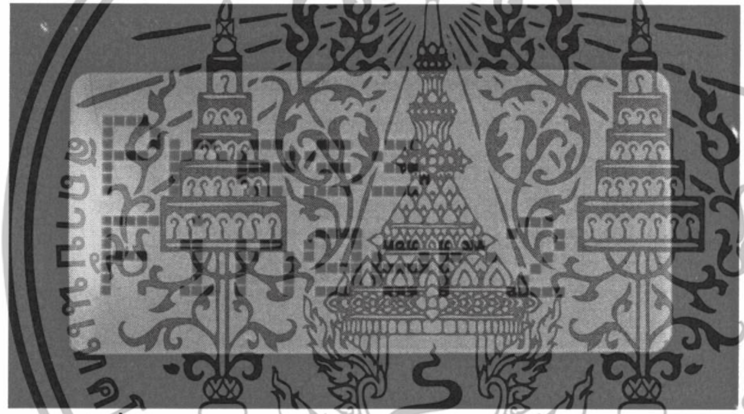
รูปที่ 4.12 หน้าจอแอลซีดีแสดงข้อความ “003”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ระบบจะตรวจสอบ ID Number ถ้าค่าไม่ซ้ำหน้าจอแอลซีดีจะแสดงข้อความ

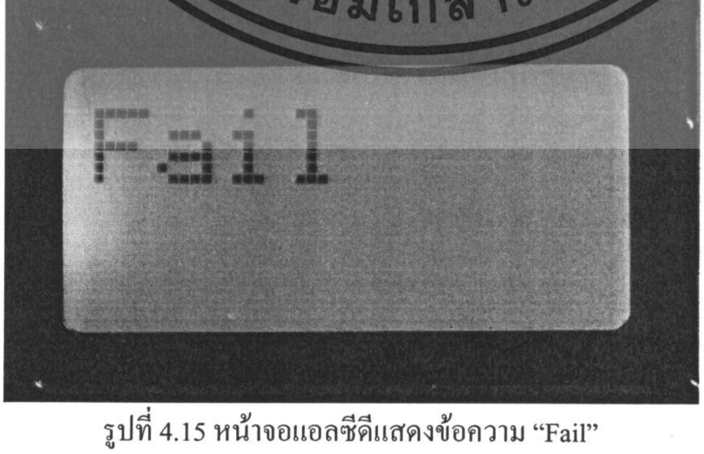


รูปที่ 4.13 หน้าจอแอลซีดีแสดงข้อความ “Press Finger 1”



รูปที่ 4.14 หน้าจอแอลซีดีแสดงข้อความ “Pres Finger 2”

แต่ถ้าค่าซ้ำหน้าจอแอลซีดีจะแสดงข้อความ



รูปที่ 4.15 หน้าจอแอลซีดีแสดงข้อความ “Fail”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

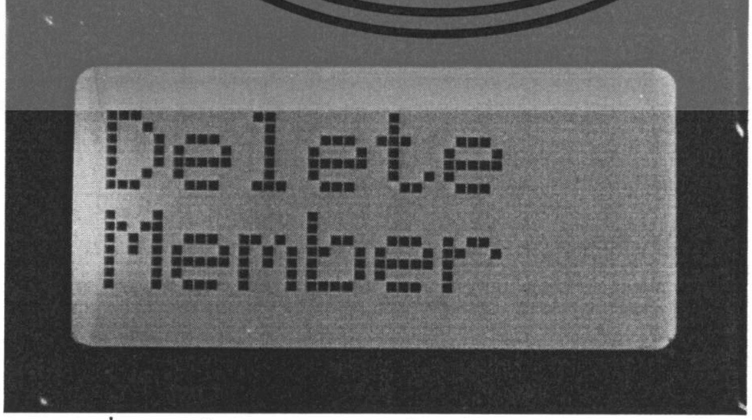
6.ระบบจะตรวจสอบความผิดพลาด ถ้าไม่ผิดพลาดหน้าจอแอลซีดีจะแสดงข้อความ

รูปที่ 4.16 หน้าจอแอลซีดีแสดงข้อความ “Register OK”

แต่ถ้าผิดพลาดหน้าจอแอลซีดีจะแสดงข้อความ

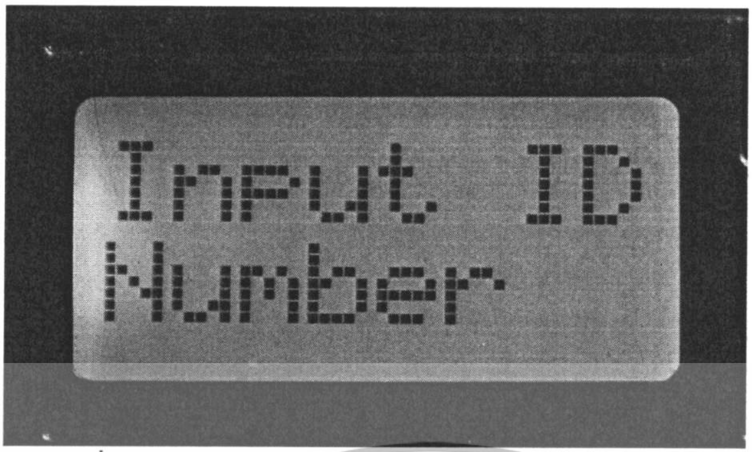
รูปที่ 4.17 หน้าจอแอลซีดีแสดงข้อความ “No Finger”

7. ถ้ากด “3” หน้าจอแอลซีดีจะแสดงข้อความ



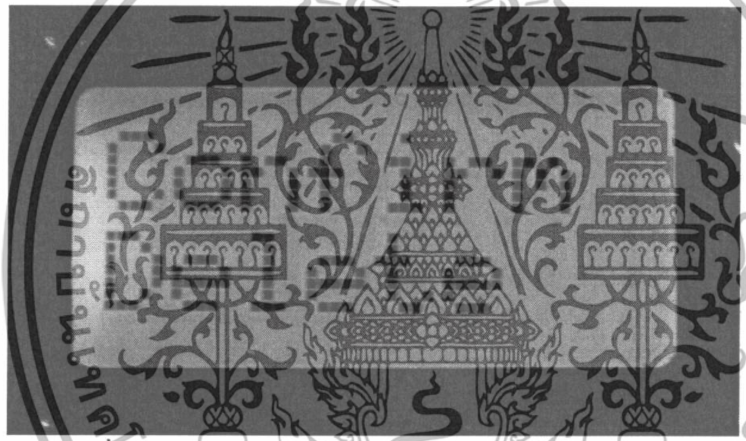
รูปที่ 4.18 หน้าจอแอลซีดีแสดงข้อความ “Delete Member”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



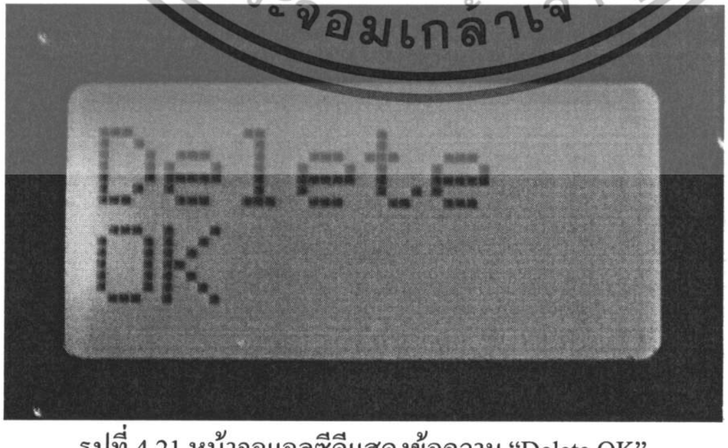
รูปที่ 4.19 หน้าจอแอลซีดีแสดงข้อความ “Input ID Number”

8.จากนั้นระบบจะทำการตรวจสอบ ID Number ถ้าไม่ผิดพลาดหน้าจอแอลซีดีจะแสดงข้อความ



รูปที่ 4.20 หน้าจอแอลซีดีแสดงข้อความ “Confirm Delete”

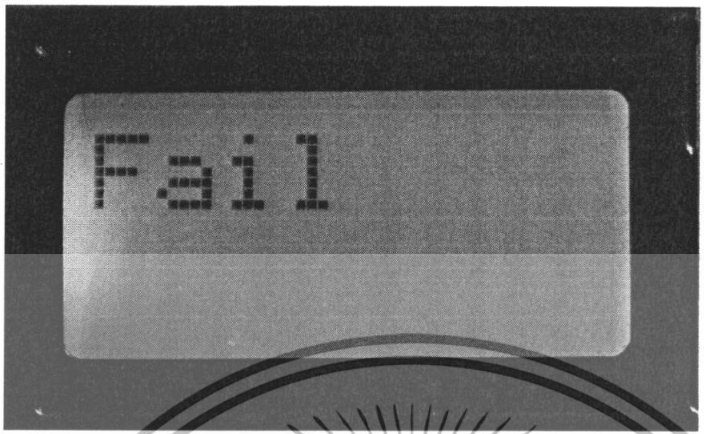
จากนั้นถ้ากด “7” หน้าจอแอลซีดีจะแสดงข้อความ



รูปที่ 4.21 หน้าจอแอลซีดีแสดงข้อความ “Delete OK”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าไม่ถูกต้องหน้าจอแอลซีดีจะแสดงข้อความ



รูปที่ 4.22 หน้าจอแอลซีดีแสดงข้อความ “Fail”

9. ถ้ากด “4” หน้าจอแอลซีดีจะแสดงข้อความ

รูปที่ 4.23 หน้าจอแอลซีดีแสดงข้อความ“Delete All”

จากนั้นถ้ากด “7” หน้าจอแอลซีดีจะแสดงข้อความ

รูปที่ 4.24 หน้าจอแอลซีดีแสดงข้อความ “Delete All OK”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

จากการทำโครงงานนี้ระบบนั้นจะมีวงจรอยู่ 5 ส่วนด้วยกัน ส่วนที่หนึ่งคือส่วนของเครื่องแสกนลายนิ้วมือซึ่งประกอบด้วยฟิงเกอร์พรินต์เซนเซอร์และบอร์ดควบคุมทำหน้าที่เก็บข้อมูลลายนิ้วมือของแต่ละบุคคล ส่วนที่สองเป็นไมโครคอนโทรลเลอร์ PIC16F877 ควบคุมการทำงานของจอแอลซีดี บัซเซอร์ รีเลย์ ตลอดจนรับคำสั่งที่ผู้ใช้ป้อนเข้าทางคีย์แพด เป็นจุดศูนย์กลางที่จะควบคุมการทำงานทั้งหมดของวงจร ส่วนที่สามคือจอแอลซีดีทำหน้าที่แสดงข้อมูลที่เป็นข้อความและตัวเลข ส่วนที่สี่คือคีย์แพดทำหน้าที่เป็นตัวป้อนข้อมูลให้กับระบบและเลือกการทำงานต่างๆของระบบ ส่วนที่ห้าคือรีเลย์ทำหน้าที่ควบคุมอุปกรณ์ต่อพ่วงเพื่อป้องกันหรือส่งสัญญาณเตือนเมื่อมีการโจรกรรมได้

5.1 ปัญหาที่พบในการทำโครงงานนี้

1.) ขาดความรู้ความชำนาญในการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ PIC16F877

2.) การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ PIC16F877 ให้เข้าถึงข้อมูลในบอร์ดควบคุมฟิงเกอร์พรินต์เซนเซอร์ทำได้ยากมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1.ขวัญชัย เรืองบุญคุ้มสุข, สุเมธ สำเร็จศิลป์, แมนศักดิ์ บวรจิโรภาส, "ระบบตรวจสอบลายนิ้วมือระบุบุคคล", ปริญญานิพนธ์ ปีการศึกษา 2546

2. C.V. Kameshwara Roo, B.Prasad and K.R. Shama, "Automatic fingerprint a syntactic approach", Proc 3 rd Tnt. Conf. on pattern Recognition, San Diego, November 1976, PP (778-782)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของคีย์แพด

```
#define use_portb_kbd TRUE

#if defined(__PCH__)
#if defined use_portb_kbd
   #byte kbd = 0xF81          // This puts the entire structure
#else
   #byte kbd = 0xF83          // This puts the entire structure
#endif
#else
#if defined use_portb_kbd
   #byte kbd = 6           // on to port B (at address 6)
#else
   #byte kbd = 8           // on to port D (at address 8)
#endif
#endif

#if defined use_portb_kbd
   #define set_tris_kbd(x) set_tris_b(x)
#else
   #define set_tris_kbd(x) set_tris_d(x)
#endif

//Keypad connection:   (for example column 0 is B2)
//            Bx:

#define COL0 (1 << 6)
#define COL1 (1 << 5)
#define COL2 (1 << 4)

#define ROW0 (1 << 3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define ROW1 (1 << 2)
#define ROW2 (1 << 1)
#define ROW3 (1 << 0)


#define ALL_ROWS (ROW0|ROW1|ROW2|ROW3)
#define ALL_PINS (ALL_ROWS|COL0|COL1|COL2)

// Keypad layout:
char const KEYS[4][3] = {{'1','2','3'},
                         {'4','5','6'},
                         {'7','8','9'},
                         {'*','0','#'}};

#define KBD_DEBOUNCE_FACTOR 33    // Set this number to apx n/333 where
                                  // n is the number of times you expect
                                  // to call kbd_getc each second


void kbd_init() {
}


char kbd_getc( ) {
   static byte kbd_call_count;
   static short int kbd_down;
   static char last_key;
   static byte col;

   byte kchar;
   byte row;

   kchar='\0';
   if(++kbd_call_count>KBD_DEBOUNCE_FACTOR) {
      switch (col) {
        case 0   : set_tris_kbd(ALL_PINS&~COL0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        kbd=~COL0&ALL_PINS;
        break;
  case 1  : set_tris_kbd(ALL_PINS&~COL1);
        kbd=~COL1&ALL_PINS;
        break;
  case 2  : set_tris_kbd(ALL_PINS&~COL2);
        kbd=~COL2&ALL_PINS;
        break;
}

if(kbd_down) {
  if((kbd & (ALL_ROWS))==(ALL_ROWS)) {
   kbd_down=false;
   kchar=last_key;
   last_key='\0';
  }
} else {
  if((kbd & (ALL_ROWS))!=(ALL_ROWS)) {
    if((kbd & ROW0)==0)
      row=0;
    else if((kbd & ROW1)==0)
      row=1;
    else if((kbd & ROW2)==0)
      row=2;
    else if((kbd & ROW3)==0)
      row=3;
    last_key =KEYS[row][col];
    kbd_down = true;
  } else {
    ++col;
    if(col==3)
      col=0;
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    kbd_call_count=0;
  }
 set_tris_kbd(ALL_PINS);
 return(kchar);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของแอลซีดี

```
struct lcd_pin_map {              // This structure is overlayed
        boolean enable;         // on to an I/O port to gain
        boolean rs;             // access to the LCD pins.
        boolean rw;             // The bits are allocated from
        boolean unused;         // low order up.  ENABLE will
        int     data : 4;       // be pin B0.
     } lcd;


#if defined(__PCH__)
#if defined use_portb_lcd
  #byte lcd = 0xF81                // This puts the entire structure
#else
  #byte lcd = 0xF83                // This puts the entire structure
#endif
#else
#if defined use_portb_lcd
  #byte lcd = 6              // on to port B (at address 6)
#else
  #byte lcd = 8              // on to port D (at address 8)
#endif
#endif


#if defined use_portb_lcd
  #define set_tris_lcd(x) set_tris_b(x)
#else
  #define set_tris_lcd(x) set_tris_d(x)
#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define lcd_type 2          // 0=5x7, 1=5x10, 2=2 lines
#define lcd_line_two 0x40   // LCD RAM address for the second line



byte CONST LCD_INIT_STRING[4] = {0x20 | (lcd_type << 2), 0xc, 1, 6};
                    // These bytes need to be sent to the LCD
                    // to start it up.



                    // The following are used for setting
                    // the I/O port direction register.

STRUCT lcd_pin_map const LCD_WRITE = {0,0,0,0,0}; // For write mode all pins are out
STRUCT lcd_pin_map const LCD_READ = {0,0,0,0,15}; // For read mode data pins are in




byte lcd_read_byte() {
    byte low,high;
    set_tris_lcd(LCD_READ);
    lcd.rw = 1;
    delay_cycles(1);
    lcd.enable = 1;
    delay_cycles(1);
    high = lcd.data;
    lcd.enable = 0;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(1);
    low = lcd.data;
    lcd.enable = 0;
    set_tris_lcd(LCD_WRITE);
    return( (high<<4) | low);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void lcd_send_nibble( byte n ) {
    lcd.data = n;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(2);
    lcd.enable = 0;
}

void lcd_send_byte( byte address, byte n ) {

    lcd.rs = 0;
    while ( bit_test(lcd_read_byte(),7) ) ;
    lcd.rs = address;
    delay_cycles(1);
    lcd.rw = 0;
    delay_cycles(1);
    lcd.enable = 0;
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init() {
   byte i;
   set_tris_lcd(LCD_WRITE);
   lcd.rs = 0;
   lcd.rw = 0;
   lcd.enable = 0;
   delay_ms(15);
   for(i=1;i<=3;++i) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      lcd_send_nibble(3);
      delay_ms(5);
    }
    lcd_send_nibble(2);
    for(i=0;i<=3;++i)
      lcd_send_byte(0,LCD_INIT_STRING[i]);
}


void lcd_gotoxy( byte x, byte y) {
   byte address;

   if(y!=1)
     address=lcd_line_two;
   else
     address=0;
   address+=x-1;
   lcd_send_byte(0,0x80|address);
}

void lcd_putc( char c) {
   switch (c) {
     case '\f'   : lcd_send_byte(0,1);
                   delay_ms(2);
                                  break;
     case '\n'   : lcd_gotoxy(1,2);        break;
     case '\b'   : lcd_send_byte(0,0x10);  break;
     default     : lcd_send_byte(1,c);     break;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของ EEPROM

```
#define EEPROM_SDA  PIN_C4
#define EEPROM_SCL  PIN_C3
#use i2c(master,sda=EEPROM_SDA, scl=EEPROM_SCL,restart_wdt)

void eeprom_init()
{
  output_float(EEPROM_SCL);
  output_float(EEPROM_SDA);
}

void write_eeprom(unsigned int16 add, byte datas)
{
  i2c_start();
  i2c_write(0xa0);
  i2c_write(add>>8);
  i2c_write(add);
  i2c_write(datas);
  i2c_stop();
  delay_ms(11);
}

int read_eeprom(unsigned int16 add) {
  byte data;

  i2c_start();
  i2c_write(0xa0);
  i2c_write(add>>8);
  i2c_write(add);
  i2c_start();
  i2c_write(0xa1);
  data=i2c_read(0);
  i2c_stop();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(data);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของวงจรและเครื่องแสกนลายนิ้วมือ

```
#include <16F877.h>
#device *=16
#use delay(clock=20000000,RESTART_WDT)
#fuses HS,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT
#use rs232(baud=115200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

#byte porta = 0x05
#byte porte = 0x09
#byte TXREG = 0x19
#byte RCREG = 0x1A
#byte PIR1 = 0x0C
#byte TXSTA = 0x98
#bit RCIF = PIR1.5
#bit TRMT = TXSTA.1
#bit TXEN = TXSTA.5
#bit LED = porta.2
#bit DOOR = porta.3

#include <LCD.C>
#include <KBD_choak.C>
#include <rom.c>

#define CMD_OPEN 0x64
#define CMD_ISFINGER 0x6F
#define CMD_ENROLLSTART 0x7A
#define CMD_ENROLLPRON 0x7B
#define CMD_ENROLLEND 0x7C
#define CMD_CAPTUREIMAGE 0x70
#define CMD_IDENTIFYIMAGE 0x89
#define CMD_DELETEALL 0xA4
#define CMD_DELETEID 0xA2
#define CMD_CHECKID 0x99
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define CMD_GETENROLLCOUNT 0x9B

int8 data[14];
int8 outparam[4];
int1 manager_flag=0;
int8 user_id=0;
int input_id=0;
unsigned int16 count1sec = 0;
unsigned char sec=0,min=0,hr=0;

#int_TIMER0      //1.6 ms
TIMER0_isr() {
  restart_wdt();
  if(++count1sec == 625)
  {
    count1sec = 0;
    if(++sec==60)
    {
      sec = 0;
      if(++min==60)
      {
        min = 0;
        if(++hr==24)
        {
          hr = 0;
        }
      }
    }
  }
}

void display_log(unsigned int address)
{
  byte log_hr=0,log_min=0,log_user=0,log_count=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 log_count = read_eeprom(0);
 log_user=read_eeprom(address);
 log_hr=read_eeprom(address+1);
 log_min=read_eeprom(address+2);
 lcd_putc("\f(");
 lcd_putc((address/3+1)+0x30);
 lcd_putc("/");
 lcd_putc(log_count+0x30);
 lcd_putc(")");
 lcd_putc((log_user/100)+0x30);
 lcd_putc(((log_user/10)%10)+0x30);
 lcd_putc((log_user%10)+0x30);
 lcd_putc("\n");
 lcd_putc((log_hr/10)+0x30);
 lcd_putc((log_hr%10)+0x30);
 lcd_putc(":");
 lcd_putc((log_min/10)+0x30);
 lcd_putc((log_min%10)+0x30);
}

void set_time()
{
  char temp;

  lcd_putc("\fSET TIME");
  delay_ms(1000);
  lcd_putc("\f");
  Do
  {
    temp = kbd_getc();
  }while((temp=='\0') || (temp=='*') || (temp=='#') || (temp=='3') || (temp=='4') || (temp=='5') ||
(temp=='6') || (temp=='7') || (temp=='8') || (temp=='9'));
  hr = (temp-0x30)*10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 lcd_putc(temp);
 Do
 {
   temp = kbd_getc();
 }while((temp=='\0') || (temp=='*') || (temp=='#'));
 hr += (temp-0x30);
 lcd_putc(temp);
 lcd_putc(":");
 Do
 {
   temp = kbd_getc();
 }while((temp=='\0') || (temp=='*') || (temp=='#') || (temp=='6') || (temp=='7') || (temp=='8') ||
(temp=='9'));
 min = (temp-0x30)*10;
 lcd_putc(temp);
 Do
 {
   temp = kbd_getc();
 }while((temp=='\0') || (temp=='*') || (temp=='#'));
 min += (temp-0x30);
 lcd_putc(temp);
 delay_ms(1000);
 lcd_putc("\fOK");
 delay_ms(1000);
}

void Senddata()
{
  int i;

  for(i=0;i<14;i++)
  {
    TXREG = data[i];
    TXEN = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      while(!TRMT);
      TXEN = 0;
    }
  }

int1 Acc()
{
  char data_acc[6];
  int i;

  while(!RCIF);
  for(i=0;i<6;i++)
  {
    while(!RCIF);
    data_acc[i] = RCREG;
  }

  if(data_acc[3] == 0x03)
    return 1;
  else
    return 0;
}

void result()
{
  char data_result[10];
  int i;
  int32 out;

  while(!RCIF);
  for(i=0;i<10;i++)
  {
    while(!RCIF);
    data_result[i] = RCREG;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  for(i=0;i<4;i++)
  {
    outparam[i] = data_result[i+4];
  }
}

int1 send_package(char command)
{
  int1 ac;
  char i;
  unsigned int16 temp=0;

  data[0] = 0x55;
  data[1] = 0xAA;
  data[2] = 0x01;
  data[3] = command;
  data[8] = 0x00;
  data[9] = 0x00;
  data[10] = 0x00;
  data[11] = 0x00;
  for(i=0;i<12;i++)
  {
    temp += data[i];
  }
  data[12] = temp & 0x00FF;
  data[13] = (temp & 0xFF00)>>8;

  Senddata();

  if(Acc()==1)
    result();
  else
    return 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

int1 finger_open()
{
  data[4] = 0x00;
  data[5] = 0x00;
  data[6] = 0x01;
  data[7] = 0x00;
  if(send_package(CMD_OPEN))
  {
    if(outparam[0] == 1)
      return 1;
    else
      return 0;
  }
  else
    return 0;
}

int1 finger_pressed()
{
  data[4] = 0x00;
  data[5] = 0x00;
  data[6] = 0x00;
  data[7] = 0x00;
  if(send_package(CMD_ISFINGER))
  {
    if(outparam[0] == 1)
      return 1;
    else
      return 0;
  }
  else
    return 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

int1 finger_capture()
{
  data[4] = 0x00;
  data[5] = 0x00;
  data[6] = 0x00;
  data[7] = 0x00;
  if(send_package(CMD_CAPTUREIMAGE))
  {
    if(outparam[0] == 1)
      return 1;
    else
      return 0;
  }
  else
    return 0;
}

int1 finger_identify()
{
  data[4] = 0;
  data[5] = 0;
  data[6] = 0;
  data[7] = 0;
  if(send_package(CMD_IDENTIFYIMAGE))
  {
    if((outparam[0]==0) || (outparam[0]==-1) || (outparam[0]==-100))
      return 0;
    else
    {
      user_id = outparam[0];
      manager_flag = outparam[3] & 0x01;
      return 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
  }
}

int1 finger_deleteall()
{
  data[4] = 0;
  data[5] = 0;
  data[6] = 0;
  data[7] = 0;
  if(send_package(CMD_DELETEALL))
  {
    if(outparam[0]==1)
      return 1;
    else
      return 0;
  }
  else
    return 0;
}

int1 finger_checkid()
{
  data[4] = input_id;
  data[5] = 0;
  data[6] = 0;
  data[7] = 0;
  if(send_package(CMD_CHECKID))
  {
    if(outparam[0]==1)
      return 1;
    else
      return 0;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else
    return 0;
}

int1 finger_enrollstart(int1 flag)
{
  data[4] = input_id;
  data[5] = 0;
  data[6] = 0x01;
  if(flag)
    data[7] = 0x11;
  else
    data[7] = 0x10;
  if(send_package(CMD_ENROLLSTART))
  {
    if(outparam[0]==1)
      return 1;
    else
      return 0;
  }
  else
    return 0;
}

int1 finger_enrollimage(int8 i)
{
  data[4] = i+1;
  data[5] = 0;
  data[6] = 0;
  data[7] = 0;
  if(send_package(CMD_ENROLLPRON))
  {
    if(outparam[0]==1)
      return 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      else
        return 0;
    }
    else
      return 0;
  }

 int1 finger_enrollend()
 {
   data[4] = 0;
   data[5] = 0;
   data[6] = 0;
   data[7] = 0;
   if(send_package(CMD_ENROLLEND))
   {
     if((outparam[0]==1) && (outparam[2]==0))
       return 1;
     else
       return 0;
   }
   else
     return 0;
 }

 int1 finger_deleteid()
 {
   data[4] = input_id;
   data[5] = 0;
   data[6] = 0x01;
   data[7] = 0;
   if(send_package(CMD_DELETEID))
   {
     if(outparam[0]==1)
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 1;
      }
      else
        return 0;
   }
   else
     return 0;
}

int1 finger_count()
{
   data[4] = 0;
   data[5] = 0;
   data[6] = 0;
   data[7] = 0;
   if(send_package(CMD_GETENROLLCOUNT))
   {
     return 1;
   }
   else
     return 0;
}

void main() {

   char temp,temp2,log_count=0,i;
   byte address=0;

   port_b_pullups(TRUE);
   setup_adc_ports(NO_ANALOGS);
   setup_adc(ADC_CLOCK_DIV_2);
   setup_psp(PSP_DISABLED);
   setup_spi(FALSE);
   setup_counters(RTCC_INTERNAL,RTCC_DIV_32);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
enable_interrupts(INT_TIMER0);
enable_interrupts(global);
set_tris_a(0x00);
set_tris_e(0x00);
eeprom_init();
LED = 1;
DOOR = 1;
lcd_init(); // 8x8 LCD
kbd_init();

delay_ms(3000);
if(finger_open())
  lcd_putc("\fSYSTEM\nSTARTED");
else
  lcd_putc("\fERROR!!");
delay_ms(2000);

set_time();

while(TRUE)
{
  finger_count();
  if(outparam[0]!=0)
  {
  lcd_putc("\f");
  lcd_putc((hr/10)+0x30);
  lcd_putc((hr%10)+0x30);
  if(sec%2==0)
    lcd_putc(":");
  else
    lcd_putc(" ");
  lcd_putc((min/10)+0x30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcd_putc((min%10)+0x30);
if(finger_pressed())
{
  if(finger_capture())
  {
    if(finger_identify())
    {
      if(manager_flag == 0)
      {
        lcd_putc("\fOK!\nID:");
        lcd_putc((user_id/100)+0x30);
        lcd_putc(((user_id/10)%10)+0x30);
        lcd_putc((user_id%10)+0x30);
        log_count = read_eeprom(0);
        write_eeprom(0,log_count+1);
        address = log_count*3+1;
        write_eeprom(address,user_id);
        write_eeprom(address+1,hr);
        write_eeprom(address+2,min);
        LED = 0;
        DOOR = 0;
        delay_ms(5000);
        LED = 1;
        DOOR = 1;
      }
      else
      {
        //MANAGER MODE
        lcd_putc("\f0:Edit\n#:Pass");
        Do
        {
          temp = kbd_getc();
        }while((temp=='\0') || ((temp!='#') && (temp!='0')));
        if(temp=='#')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  //Open Door
  lcd_putc("\fOK!\nID:");
  lcd_putc((user_id/100)+0x30);
  lcd_putc(((user_id/10)%10)+0x30);
  lcd_putc((user_id%10)+0x30);
  DOOR = 0;
  LED = 0;
  delay_ms(5000);
  LED = 1;
  DOOR = 1;
}
else if(temp=='0')
{
  //Admin Mode
  lcd_putc("\fSelect\nMenu");
  Do
  {
    temp = kbd_getc();
  switch(temp)
  {
    case '1':
      MODE:
      //*********************************MASTER REGISTER
      input_id=0;
      lcd_putc("\fMaster\nRegister");
      delay_ms(2000);
      lcd_putc("\fID:");
      for(i=0;i<3;i++)
      {
        Do
        {
          temp2=kbd_getc();
        }while((temp2=='\0') || (temp2=='*') || (temp2=='#'));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    switch(i)
    {
      case 0:
        input_id += (temp2-0x30)*100;
        break;
      case 1:
        input_id += (temp2-0x30)*10;
        break;
      case 2:
        input_id += (temp2-0x30);
        break;
    }
    lcd_putc(temp2);
  }
  if(finger_checkid())
  {
    lcd_putc("\fERROR");
  }
  else
  {
    if(finger_enrollstart(1))
    {
      for(i=0;i<3;i++)
      {
        lcd_putc("\fPress\nFinger");
        temp2 = (i+1)+0x30;
        lcd_putc(temp2);
        while(!finger_pressed());
        if(finger_capture())
        {
          if(finger_enrollimage(i))
          {
            lcd_putc("\fTake\nOff");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            else
            {
              lcd_putc("\fERROR");
            }
          }
          else
          {
            lcd_putc("\fERROR");
          }
          while(finger_pressed());
        }
        if(finger_enrollend())
        {
          lcd_putc("\fRegister\nOK!");
          delay_ms(2000);
        }
        else
        {
          lcd_putc("\fERROR");
        }
      }
      else
      {
        lcd_putc("\fERROR");
      }
    }
    lcd_putc("\fSelect\nMenu");
    break;
  case '2':
    //*********************************USER REGISTER
    input_id=0;
    lcd_putc("\fUser\nRegister");
    delay_ms(2000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcd_putc("\fID:");
for(i=0;i<3;i++)
{
  Do
  {
    temp2=kbd_getc();
  }while((temp2=='\0') || (temp2=='*') || (temp2=='#'));

  switch(i)
  {
    case 0:
      input_id += (temp2-0x30)*100;
      break;
    case 1:
      input_id += (temp2-0x30)*10;
      break;
    case 2:
      input_id += (temp2-0x30);
      break;
  }
  lcd_putc(temp2);
}
if(finger_checkid())
{
  lcd_putc("\fERROR");
}
else
{
  if(finger_enrollstart(0))
  {
    for(i=0;i<3;i++)
    {
      lcd_putc("\fPress\nFinger");
      temp2 = (i+1)+0x30;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcd_putc(temp2);
    while(!finger_pressed());
    if(finger_capture())
    {
      if(finger_enrollimage(i))
      {
        lcd_putc("\fTake\nOff");
      }
      else
      {
        lcd_putc("\fERROR");
      }
    }
    else
    {
      lcd_putc("\fERROR");
    }
    while(finger_pressed());
  }
  if(finger_enrollend())
  {
    lcd_putc("\fRegister\nOK!");
    delay_ms(2000);
  }
  else
  {
    lcd_putc("\fERROR");
  }
}
else
{
  lcd_putc("\fERROR");
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  lcd_putc("\fSelect\nMenu");
  break;
case '3':
  //**********************************DELETE MEMBER
  input_id=0;
  lcd_putc("\fDelete\nMember");
  delay_ms(2000);
  lcd_putc("\fID:");
  for(i=0;i<3;i++)
  {
    Do
    {
      temp2=kbd_getc();
    }while((temp2=='\0') || (temp2=='*') || (temp2=='#'));

    switch(i)
    {
      case 0:
        input_id += (temp2-0x30)*100;
        break;
      case 1:
        input_id += (temp2-0x30)*10;
        break;
      case 2:
        input_id += (temp2-0x30);
        break;
    }
    lcd_putc(temp2);
  }
 lcd_putc("\fDelete?");
 Do
 {
   temp2 = kbd_getc();
 }while((temp2=='\0') || ((temp2!='7') && (temp2!='9')));
```
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(temp2=='7')
{
  if((finger_deleteid()) && (input_id>0) && (input_id<=255))
  {
    lcd_putc("\fDELETED");
    delay_ms(2000);
  }
  else
  {
    lcd_putc("\fERROR");
  }
}
else if(temp2=='9')
{
  lcd_putc("\fCANCEL");
  delay_ms(2000);
}
lcd_putc("\fSelect\nMenu");
break;
case '4':
//**********************************DELETE ALL
lcd_putc("\fDelete\nAll?");
delay_ms(2000);
Do
{
  temp2 = kbd_getc();
}while((temp2=='\0') || ((temp2!='7') && (temp2!='9')));

if(temp2=='7')
{
  if(finger_deleteall())
  {
    lcd_putc("\fDELETED");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      delay_ms(2000);
      goto MODE;
    }
    else
    {
      lcd_putc("\fERROR");
    }
  }
  else if(temp2=='9')
  {
    lcd_putc("\fCANCEL");
    delay_ms(2000);
  }
  lcd_putc("\fSelect\nMenu");
  break;
case '5':
  set_time();
  lcd_putc("\fSelect\nMenu");
  break;
case '6':
  lcd_putc("\fErase\nLog?");
  Do
  {
    temp2 = kbd_getc();
  }while((temp2=='\0') || ((temp2!='7') && (temp2!='9')));
  if(temp2=='7')
  {
    write_eeprom(0,0);
    lcd_putc("\fDELETED");
    delay_ms(2000);
  }
  else if(temp2=='9')
  {
    lcd_putc("\fCANCEL");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay_ms(2000);
      }
      lcd_putc("\fSelect\nMenu");
      break;
    case '7':
      log_count = read_eeprom(0);
      address = 1;
      if(log_count!=0)
      {
        display_log(1);
        do
        {
          if(temp2=='*')
          {
            if(address>1)
            {
              address -= 3;
              display_log(address);
            }
          }
          else if(temp2=='#')
          {
            if(address<((log_count-1)*3+1))
            {
              address += 3;
              display_log(address);
            }
          }
          do
          {
            temp2 = kbd_getc();
          }while((temp2=='\0') || ((temp2!='#') && (temp2!='*') && (temp2!='0')));
        }while(temp2!='0');
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else
            {
              lcd_putc("\fNO DATA");
              delay_ms(2000);
            }
            lcd_putc("\fSelect\nMenu");
            break;
          }
        }while((temp=='\0')||(temp!='0'));
      }
      else
      {
        lcd_putc("\fERROR");
      }
    }
  }
  else
  {
    lcd_putc("\fNO DATA");
  }
 }
 else
 {
   lcd_putc("\fERROR");
 }
 while(finger_pressed());
}
}
else
{
  Goto MODE;
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BioPrima OEM2000P-44B0

Fingerprint Identification Board

Datasheet

(Ver.1.0)

APRO TECHNOLOGY (BANGKOK) CO., LTD.

February 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Notices

Information in this document is subject to change without notice.
NO WARRANTY OF ANY KIND IS MADE WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
No liability is assumed for errors contained herein or for incidental damages in connection with the furnishing, performance, use of this material.
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of APRO Technology (Bangkok) Co., Ltd.
Other products and corporate names may be trademarks or registered trademarks of other companies and are only for explanation without intent to infringe.

Copyright © All rights reserved.
Version 1.0

We hope that all the contents and information contained herein will be of assistance to you. Welcome to comments, opinions & suggestions.

Contact Us:

Telephone : +66-2234-2523-4 , +66-26345871

Fax : +66-2634-0072

E-mail : apro-bangkok@aprotechnology.com

URL : http://www.aprotechnology.com

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE OF CONTENTS

1. CONFIGURATION 4

2. PRODUCT SPECIFICATION 5

2.1. FEATURES 5
2.2. TECHNICAL SPECIFICATIONS 5
2.3. ELEMENT LAYOUT 6
2.4. DIMENSION OF OPTICAL SENSOR 6
2.5. RS-232C CONNECTOR 7
2.6. DIPSW 7

3. COMMUNICATION PROTOCOL 8

3.1. COMMAND MODE 8
3.2. RECEIVE MODE 9
3.3. SEND MODE 10
3.4. PACKET STRUCTURE 11
3.4.1. Command Packet (Command) 11
3.4.2. Command Response Packet (Acknowledge) 12
3.4.3. Packet Result (Result) 12
3.4.4. Block Data Packet (Block Data) 13
3.4.5. Transmitted Data Description 13

4. COMMAND DESCRIPTION 14

4.1. COMMAND TABLE 14
4.2. COMMAND DETAIL 15

5. ERROR CODE 29

6. HOW TO USE COMMANDS 30

6.1. BASIC PERFORMANCE OF OEM2000P-44B0 30
6.2. EXAMPLES OF APPLYING FINGERPRINT PROCESSING ALGORITHM 31
6.2.1. Initialize OEM2000P-44B0 31
6.2.2. Enrollment 32
6.2.3. 1:1 Verify 33
6.2.4. 1:N Identify 33
6.2.5. How to use command for Fingerprint 34
6.2.6. How to Use Command for Fingerprint Database 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Configuration

OEM2000P-44B0 consists of fingerprint processor and optical input sensor.
It is the High-Speed Fingerprint identification OEM module, and consists of CPU, Flash memory, SDRAM, etc.

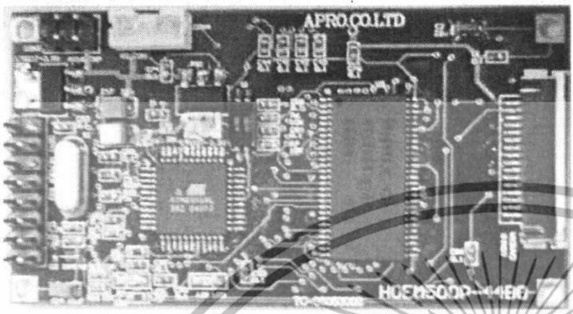


Fig. 1 OEM2000P-44b0 module

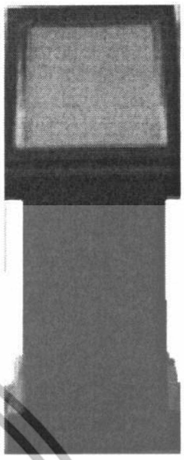


Fig. 2 Optical Sensor

Item	
Fingerprint Processing Board	1
Optical Fingerprint Sensor	1
Datasheet	1
Sample Program	
Sample Source Code	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Product specification

2.1. Features

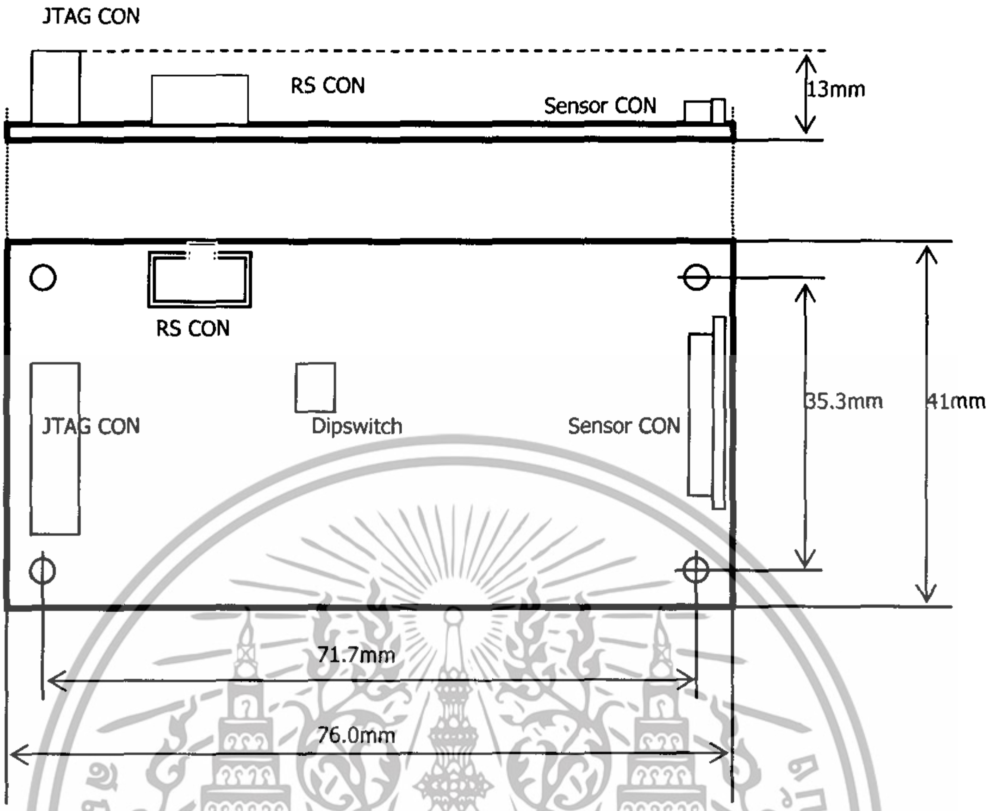
- High speed fingerprint match technology
- Optical sensor
- 1:N, 1:1 match
- Read /write fingerprint minutiae data
- Reasonable control of pass rate
- Command reception and process through parallel communication of standard RS-232C

2.2. Technical Specifications

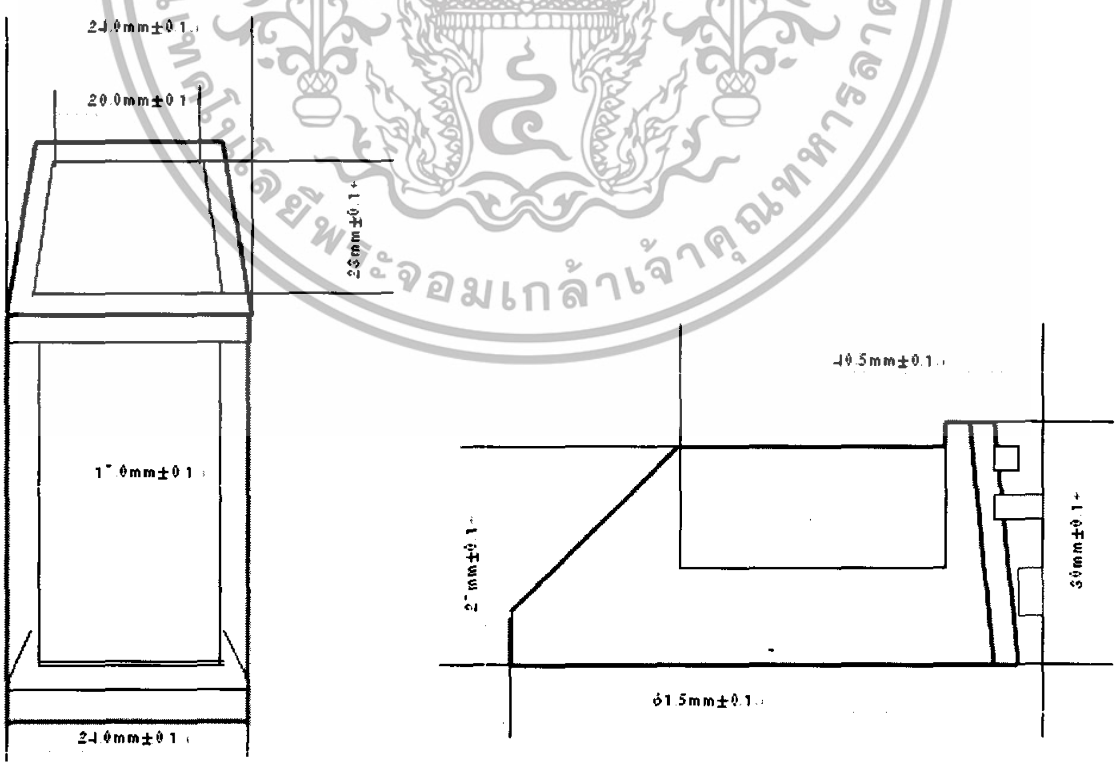
Item		Feature
Mode of combination(I/F)		RS232C (3 line) Baudrate=9600/19200/38400bps/115200bps Parity=NONE StopBit=1 FlowControl=NONE
Size of fingerprint enroll data		1404 Byte
Max. fingerprint capacity		2,000 fingerprints
Matching mode		1 : N / 1 : 1
Enrolling mode		Thrice
Pass Rate value		1 ~ 5
FAR		< 0.0001% (in case pass rate is 3)
FRR		< 0.01% (in case pass rate is 3)
Enrollment speed		< 3 sec.
Verification speed (500 prints)		< 1 sec.
Operating voltage		DC 5V
Operating current		200 mA
Operating Environment	Temperature	0℃ ~ +60℃
	Humidity	20% ~ 80%
Storage Environment	Temperature	0℃ ~ +70℃
	Humidity	10% ~ 80%
Size of board		76 x 41 x 12 mm (W x H x D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3. Element Layout



2.4. Dimension of Optical Sensor

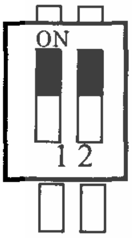


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5. RS-232C Connector

Item	Pin No	Contents
+5V	1	+5V supply
GND	2	GND
TXD0	3	Data transmission of RS232C level for serial port 0
RXD0	4	Data reception of RS232C level for serial port 0

2.6. DIPSW



- SW1 = ON : Maintenance of Current port setting.
  = OFF : Default setting.
    - Baud = 115200
    - DataBit = 8
    - Parity = None
    - Stop = 1

- SW2 = ON : Maintenance of current flash data.
  = OFF : Initialization of flash data.

- DIPSW Setting in use :
  SW1 = ON
  SW2 = ON

- How to initialize :
  SW1 = OFF
  SW2 = OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

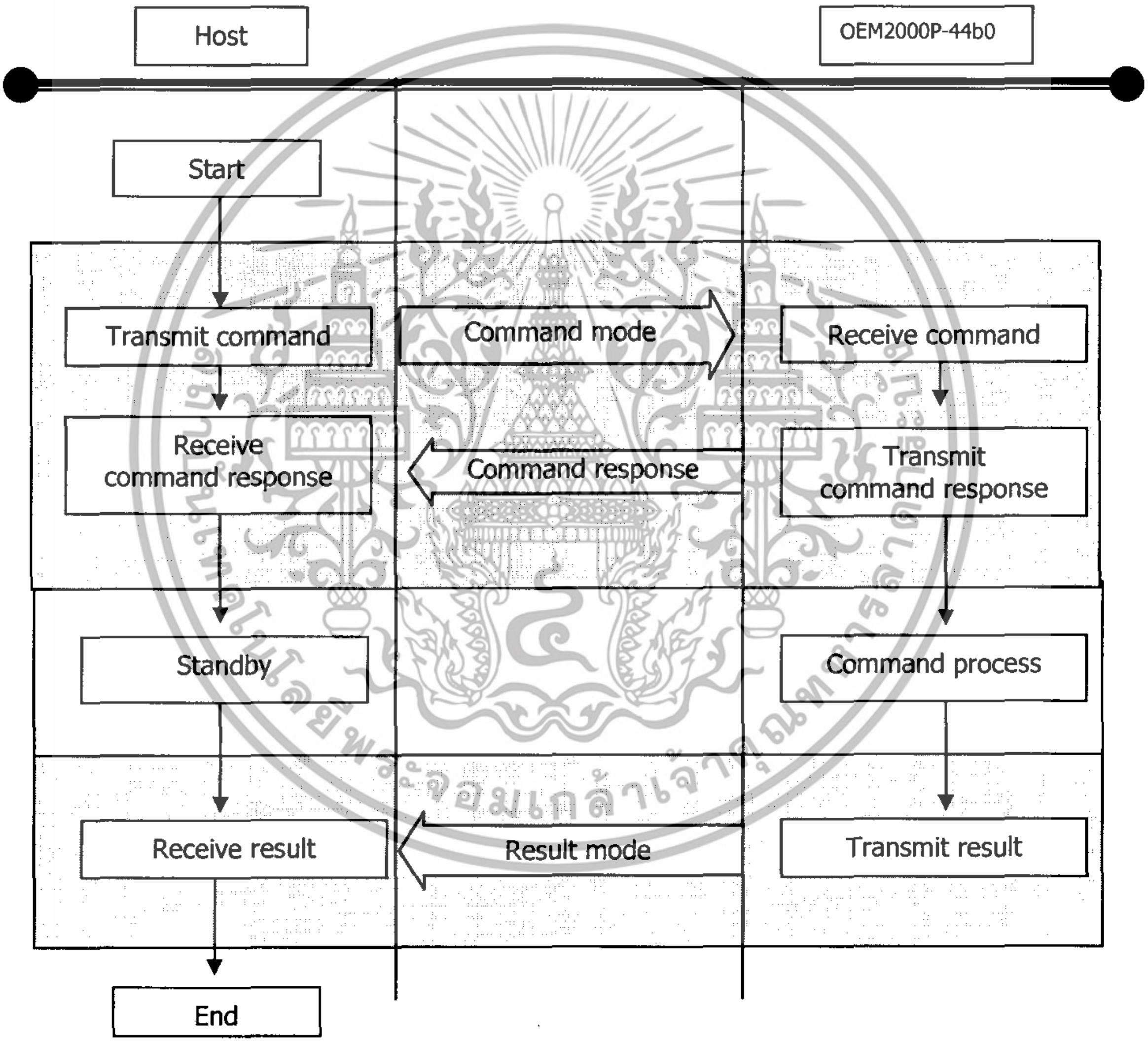
3. Communication Protocol

There are three kinds of RS232C communication protocols.

1) Command Mode
2) Receive Mode
3) Send Mode

3.1. Command Mode

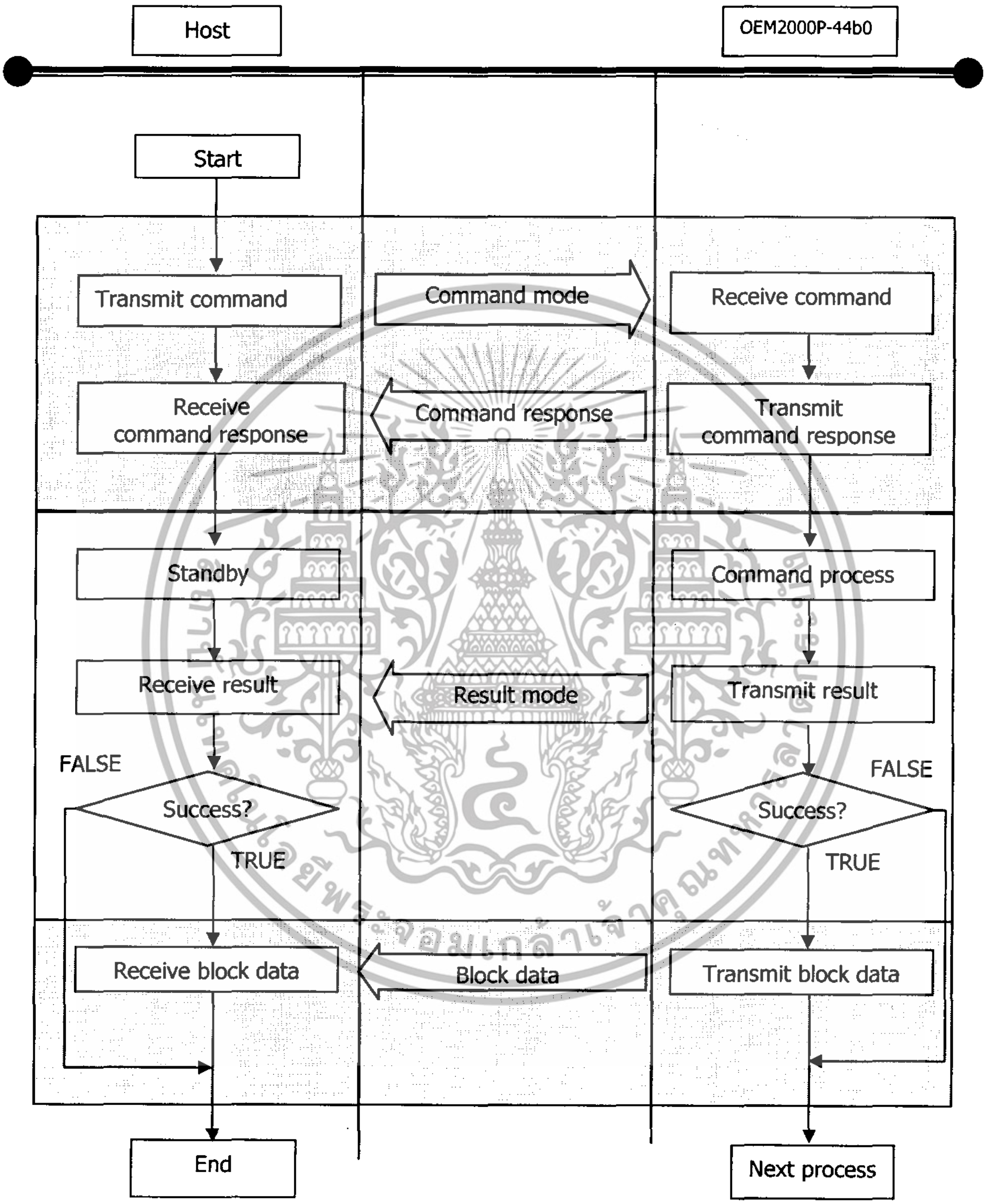
Only commands are transmitted and received.
There are no transmission and reception of block data.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2. Receive Mode

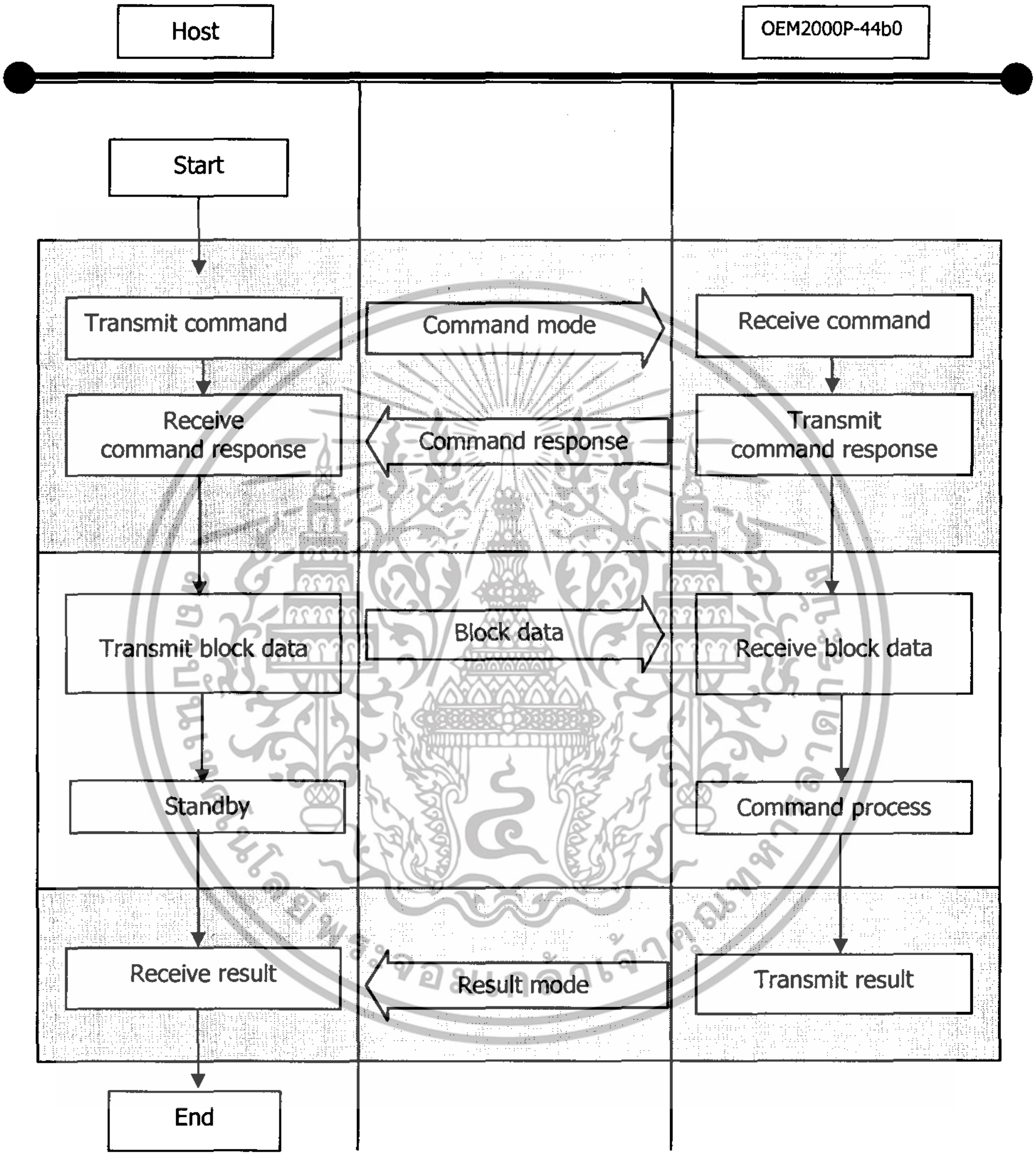
Block data are transmitted from OEM2000P-44B0 to Host as a result of command execution.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3. Send Mode

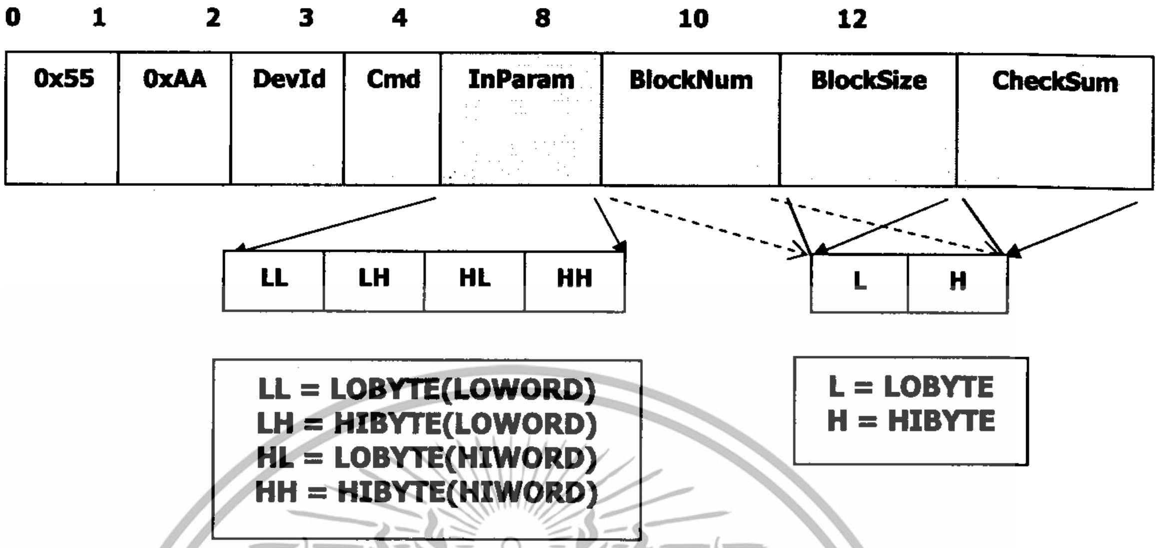
OEM2000P-44B0 receives block data transmitted from Host and executes a command.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4. Packet Structure

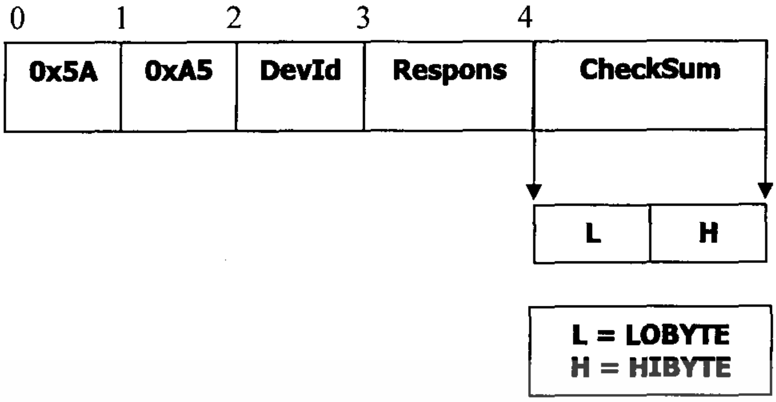
3.4.1. Command Packet (Command)



OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Command start code 1
1	0xAA	BYTE	Command start code 2
2	DevId	BYTE	Device number (0x01-0xff) Usually, set as 0x01.
3	Cmd	BYTE	Command code
4	InParam	DWORD	Input parameter
8	BlockNum	WORD	Block number used at the time of transmission of block data. ReceiveMode and SendMode use this command.
10	BlockSize	WORD	Block size used at the time of transmission of block data. ReceiveMode and SendMode use this command.
12	CheckSum	WORD	Check Sum OFFSET[0] + ... + OFFSET[11] =CheckSum

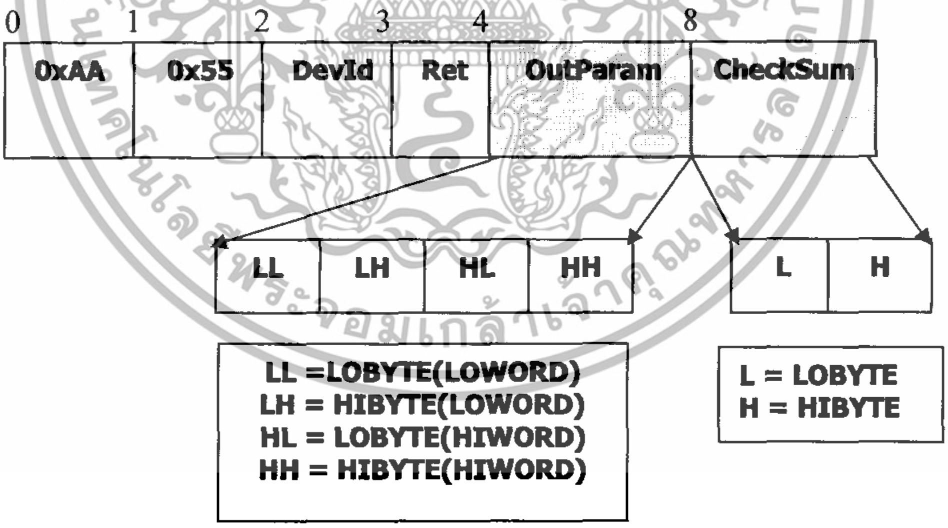
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2. Command Response Packet (Acknowledge)



OFFSET	ITEM	TYPE	DESCRIPTION
0	0x5A	BYTE	Response start code 1
1	0xA5	BYTE	Response start code 2
2	DevId	BYTE	Device number (0x01-0xff) Usually, set as 0x01.
3	Respons	BYTE	Response=0x02:Unusual response (NAK) Response=0x03: Normal response (ACK)
4	CheckSum	WORD	Check Sum OFFSET[0] + ... + OFFSET[3] =CheckSum

3.4.3. Packet Result (Result)



OFFSET	ITEM	TYPE	DESCRIPTION
0	0xAA	BYTE	Result start code 1
1	0x55	BYTE	Result start code 2
2	DevId	BYTE	Device number (0x01-0xff) Usually, set as 0x01.
3	Ret	BYTE	Result code 0x01: Command process success 0x00: Command process failure
4	OutParam	DWORD	Output parameter
8	CheckSum	WORD	Check Sum OFFSET[0] + ... + OFFSET[7]=CheckSum

เอกสารนี้เป็นเอกสารที่สงวนไว้... ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้าง...

3.4.4. Block Data Packet (Block Data)

0	1	2	3	- - -	n	n+1	n+2
0x55	0xAA	Byte(0)	Byte(1)	………	Byte(n-2)	Byte(n-1)	CheckSum

CheckSum:

L	H

n : shows block size.

L = LOBYTE
H = HIBYTE

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Block data start code 1
1	0xAA	BYTE	Block data start code 2
2	BYTE[0]	BYTE	The 1st data (Byte)
3	BYTE[1]	BYTE	The 2nd data (Byte)
: :	: :	: :	: :
n+1	BYTE[n-1]	BYTE	n(BlockSize)th byte data
n+2	CheckSum	WORD	Check Sum OFFSET[0] + ... + OFFSET[n+1] =CheckSum

3.4.5. Transmitted Data Description

1) Block size.
   Block size is set as 1024.

2) Enrollment Information(User Info)

(1) Manager flag (0 : User / 1 : Manager)
   It is a flag set to identify whether he is a user or a manager and to use the restricted right at the time of enrollment / verification of fingerprint.
   Usually set as user = 0

(2) Enroll number (1 ~ 32767)
   Enroll number is a decimal number of 4 figures with the length of 2 bytes at most.

(3) Finger number (1 ~ 10)
   Finger number is a decimal number of 2 figures.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Command Description

4.1. Command Table

No.	Alias	Command Code		Description
		DEC	HEX	
1	Open	100	0x64	Initialize device
2	Close	101	0x65	
3	SetThreshold	102	0x66	Set up threshold of fingerprint identification engine
4	DeviceSetup	103	0x67	Set up device ID, RS-232 port
5	IsPressFinger	111	0x6F	Check whether a finger is now placed on the fingerprint input sensor or not.
6	CaptureImage	112	0x70	Capture the fingerprint image from fingerprint input sensor.
7	GetImage	113	0x71	Transmit 256*256 fingerprint image to Host.
8	PutImage	115	0x73	Transmit 256*256 fingerprint image through Device.
9	EnrollStartN	122	0x7A	Start 3 fingerprints enrollment.
10	EnrollaProNImage	123	0x7B	Preprocess nth fingerprint image.
11	EnrollEndN	124	0x7C	Enroll 3 fingerprint images.
12	VerifyImage	131	0x83	Verify fingerprint image with specified ID number and fingerprint number.
13	VerifyIDImage	132	0x84	Verify fingerprint image with specified ID number.
14	IdentifyImage	137	0x89	1:N Identification.
15	InputFpData	140	0x8C	Transmit fingerprint feature data through device.
16	EnrollFpData	141	0x8D	Enroll fingerprint feature data transmitted.
17	VerifyFpData	142	0x8E	Verify fingerprint feature data transmitted with specified ID number and fingerprint number.
18	VerifyIDFpData	143	0x8F	Verify fingerprint feature data transmitted with specified ID number.
19	IdentifyFpData	144	0x90	Identify (1:N Identify) fingerprint feature data transmitted.
20	SearchUnusedID	150	0x96	Search unused ID.
21	SearchUnusedFinger	151	0x97	Search unused Finger No. of specified ID.
22	CheckIDFinger	152	0x98	Check whether fingerprint data are enrolled in specified ID, FingerNo.
23	CheckID	153	0x99	Check whether fingerprint data are enrolled in specified ID.
24	CheckManager	154	0x9A	Check whether specified ID, FingerNo is manager or not.
25	GetEnrollCount	155	0x9B	Get the number of currently enrolled fingerprint data.
26	ReadFpData	160	0xA0	Read fingerprint data from enrollment database.
27	WriteFpDB	161	0xA1	Write fingerprint data to enrollment database.
28	DeleteIDFinger	162	0xA2	Delete fingerprint data with specified ID, FingerNo from enrollment database.
29	DeleteID	163	0xA3	Delete fingerprint data with specified ID from enrollment database.
30	DeleteAll	164	0xA4	Delete all fingerprint data from enrollment database.
31	WriteProductID	185	0xB9	Write ID of product.
32	ReadProductID	186	0xBA	Read ID of product.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2. Command Detail

1) Open

Command	**100**	
Alias	**Open**	
Description	Open fingerprint identification module. Initialize fingerprint data record pointer. Initialize fingerprint input sensor. Initialize fingerprint identification engine. Perform inspection of device and software of SlaveCPU. Notes: When executing this command, do not place a finger on the sensor. It may cause an error.	
HandShake	CommandMode	
InParam	Refer to following	Unuse
Result	1	Unuse
OutParam	1	Success
	SENSOR_ERR	Fingerprint input sensor error
	DEV_ERR	Device error

InParam			
HH	**HL**	**LH**	**LL**
Sensor Type Must be 1.		Enroll Type Must be 0.	

2) Close

Command	**101**	
Alias	**Close**	
Description	Close fingerprint identification module.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	0	Success

3) SetThreshold

Command	**102**	
Alias	**SetThreshold**	
Description	Set up threshold of fingerprint identification engine. Integer above 900 can be selected. Default is 1050.	
HandShake	CommandMode	
InParam	n	Threshold to select. The integer above 900 can be selected.
Result	1	Unuse
OutParam	1	Success
	SECURITY_ERR	Input parameter error It is the case where it is set up except the integer above 900.
	DB_ERR	Writing error in enrollment database (FLASH Memory).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) DeviceSetup

Command	103	
Alias	DeviceSetup	
Description	Set up device ID and communication port settings of fingerprint identification module. Data bit is 8 bit.	
HandShake	CommandMode	
InParam	Refer to following.	
Result	1	Unuse
OutParam	1	Success.
	DEVSETUP_ERR	Input parameter error.

InParam			
HH	**HL**	**LH**	**LL**
Device ID (1 ~ 255)	Baud rate (0 ~ 4) 0: 9600 bps 1: 19200 bps 2: 38400 bps 3: 57600 bps 4: 115200 bps	Parity bit (0 ~ 2) 0: None 1: Odd 2: Even	Stop bit (0, 1) 0: 1bit 1: 2bit

5) IsPressFinger

Command	**111**	
Alias	**IsPressFinger**	
Description	Check whether a finger is now placed on the fingerprint input sensor or not.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	1	A finger is placed on the fingerprint input sensor.
	IMAGE_ERR	A finger is not placed on the fingerprint input sensor.
	SENSOR_ERR	An image is not sent from fingerprint input sensor.

6) CaptureImage

Command	**112**	
Alias	**CaptureImage**	
Description	After inputting a fingerprint image from fingerprint input sensor, checks whether the finger is placed on the sensor or not. Fingerprint image data are saved. The image data are used for enrollment and identification.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	1	A finger is placed on the fingerprint input sensor.
	IMAGE_ERR	A finger is not placed on the fingerprint input sensor.
	SENSOR_ERR	An image is not sent from fingerprint input sensor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) GetImage

Command	**113**	
Alias	**GetImage**	
Description	Transmit 256*256 fingerprint image obtained by **CaptureImage** command to Host.	
HandShake	ReceiveMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	0	Unuse

8) PutImage

Command	**115**	
Alias	**PutImage**	
Description	Transmit 256*256fingerprint image through device. This image is used in **IdentifyImage**, **EnrollProNImage** and **VerifyImage** commands. Refer to the description of these commands.	
HandShake	SendMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	0	Unuse

9) EnrollStartN

Command	**122**	
Alias	**EnrollStartN**	
Description	Start enrollment.	
HandShake	CommandMode	
InParam	Refer to following	Input data
Result	1	Unuse
OutParam	1	Successful EnrollStart.
	ID_ERR	ID is out of the range of enrollment.
	FINGERNO_ERR	FingerNo is out of the range of enrollment.
	OVER_ERR	Enrollment database is full.
	DEV_ERR	Device error

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
check bit	manager	Reserved	Finger number		
0 ~ 1	0:user 1:manager	0	1 ~ 10		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) EnrollProNImage

Command	**123**	
Alias	**EnrollProNImage**	
Description	Get nth fingerprint data when three fingerprints are enrolled. Get nth fingerprint data from fingerprint image data obtained by **PutImage** or **CaptureImage** command.	
HandShake	CommandMode	
InParam	n	nth (from 1 to 3)
Result	1	Unuse
OutParam	1	Success.
	TURN_ERR	EnrollStartN has not been executed.
	NTH_ERR	Incorrect number.
	IMAGE_ERR	The fingerprint image cannot be enrolled.
	DEV_ERR	Device error

11) EnrollEndN

Command	**124**	
Alias	**EnrollEndN**	
Description	Enroll the fingerprint data by compounding three fingerprint data obtained by EnrollProNImage when three fingerprints are enrolled. At this moment it can be checked whether current fingerprint has already enrolled or not. If it is not check state then check is not done but it is checked whether current fingerprint has already enrolled or not. If yes then already enrolled ID, FingerNo is returned.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	1	Successful Enrollment.
	> 1	ID, FingerNo of already enrolled fingerprint when InParam is 1. Refer to following.
	TURN_ERR	EnrollStartN has not been executed.
	MERGE_ERR	Compounding error.
	DB_ERR	Writing error in enrollment database(FLASH Memory).
	DEV_ERR	Device error

OutParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
1	0:user 1:manager	0	1 ~ 10		

- **Note!**
  Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12) VerifyImage

Command	**131**	
Alias	**VerifyImage**	
Description	Verify (1:1 Verify) fingerprint image obtained by **PutImage** or **CaptureImage** command.	
HandShake	CommandMode	
InParam	Refer to following	Fingerprint data ID, FingerNo to verify.
Result	1	Unuse
OutParam	Refer to following	Successful verification. The verified fingerprint data ID,FingerNo are return. The same value as InParam.
	0	Verification failure
	NOTENROLL_ERR	It is not enrolled.
	IMAGE_ERR	The fingerprint image cannot be verify.
	ID_ERR	ID is out of the range of enrollment. There is no fingerprint data with ID.
	FINGERNO_ERR	FingerNo is out of the range of enrollment. There is no fingerprint data with FingerNo.
	DEV_ERR	Device error

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

OutParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0:user 1:manager	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13) VerifyIDImage

Command	**132**	
Alias	**VerifyIDImage**	
Description	Verify (1:1 Verify) fingerprint image obtained by **PutImage** or **CaptureImage** command with specified ID.	
HandShake	CommandMode	
InParam	nnn	Fingerprint data ID to verify.
Result	1	Unuse
OutParam	Refer to following	Successful verification. The verified fingerprint data ID is returned. The same value as InParam.
	0	Verification failure.
	NOTENROLL_ERR	It is not enrolled.
	IMAGE_ERR	The fingerprint image cannot be verified.
	ID_ERR	ID is out of the range of enrollment. There is no fingerprint data with specified ID.
	DEV_ERR	Device error

OutParam					
HH		HL		LH	LL
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
1	0:user 1:manager	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

14) IdentifyImage

Command	**137**	
Alias	**IdentifyImage**	
Description	Identify (1:N Identify) fingerprint image obtained by **PutImage** or **CaptureImage** command.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	Refer to following	Successful identification. The identified fingerprint data ID is returned.
	0	Identification failure. There is no identified fingerprint data.
	IMAGE_ERR	The fingerprint image cannot be identify.
	DEV_ERR	Device error

OutParam					
HH		HL		LH	LL
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0:user 1:manager	0	1 ~ 10		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note!
Enrollment ID can be input from 1 to 32767.

15) InputFpdata.

Command	**140**	
Alias	**InputFpdata**	
Description	Transmit one of fingerprint data in external database to a fingerprint identification module through RS-232C. Fingerprint image data are used for enrollment and identification.	
HandShake	SendMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	0	Data are transmitted correctly.
	COMM_ERR	CheckSum of transmitted data is different.

16) EnrollFpdata

Command	**141**	
Alias	**EnrollFpdata**	
Description	Enroll fingerprint data transmitted by **InputFpdata** command.	
HandShake	CommandMode	
InParam	Refer to following	Input data
Result	1	Unuse
OutParam	0	Successful enrollment.
	FPDATA_ERR	The fingerprint data cannot be enrolled.
	ID_ERR	ID is out of the range of enrollment.
	OVER_ERR	Enrollment database is full.
	DB_ERR	Writing error in enrollment database(FLASH Memory).

<table>
<tr><th colspan="6">InParam</th></tr>
<tr><th colspan="2">HH</th><th colspan="2">HL</th><th>LH</th><th>LL</th></tr>
<tr><td>7654</td><td>3210</td><td>7654</td><td>3210</td><td colspan="2" rowspan="3">Enrollment ID</td></tr>
<tr><td>Reserved</td><td>manager</td><td>Reserved</td><td>Finger number</td></tr>
<tr><td>0</td><td>0:user
1:manager</td><td>0</td><td>1 ~ 10</td></tr>
</table>

Note!

Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17) VerifyFpdata

Command	**142**	
Alias	**VerifyFpdata**	
Description	Verify (1:1 Verify) fingerprint data obtained by **InputFpdata** command.	
HandShake	CommandMode	
InParam	Refer to following	Fingerprint data ID to verify.
Result	1	Unuse
OutParam	Refer to following	Successful verification. The verified fingerprint data ID is returned. The same value as InParam.
	0	Verification failure.
	FPDATA_ERR	The fingerprint data cannot be verified.
	ID_ERR	ID is out of the range of enrollment. There is no fingerprint data with specified ID.
	DEV_ERR	Device error

InParam					
HH		HL		LH	LL
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

OutParam					
HH		HL		LH	LL
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0:user 1:manager	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18) VerifyIDFpdata

Command	**143**	
Alias	**VerifyIDFpdata**	
Description	Verify (1:1 Verify) fingerprint data obtained by **InputFpdata** command with specified ID.	
HandShake	CommandMode	
InParam	nnn	Fingerprint data ID to verify.
Result	1	Unuse
OutParam	Refer to following	Successful verification. The verified fingerprint data ID is returned. The same value as InParam.
	0	Verification failure.
	FPDATA_ERR	The fingerprint data cannot be verified.
	ID_ERR	ID is out of the range of enrollment. There is no fingerprint data with specified ID.
	DEV_ERR	Device error

OutParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0:user 1:manager	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

19) IdentifyFpdata

Command	**144**	
Alias	**IdentifyFpdata**	
Description	Identify (1: N Identify) fingerprint data obtained by **InputFpdata** command.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	Refer to following	Successful identification. The identified fingerprint data ID is returned.
	0	Identification failure. There is no identified fingerprint data.
	FPDATA_ERR	The fingerprint data cannot be identified.
	DEV_ERR	Device error

OutParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0:user 1:manager	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20) SearchUnsedID

Command	**150**	
Alias	**SearchUnsedID**	
Description	Search uneused ID.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	> 0	Successful searching. The searched ID is returned.
	0	Searching failure. There is no searched ID.
	DEV_ERR	Device error.

21) SearchUnusedFinger

Command	**151**	
Alias	**SearchUnusedFinger**	
Description	Search unused FingerNo of specified ID.	
HandShake	CommandMode	
InParam	n	ID number
Result	1	Unuse
OutParam	> 0	Successful searching. The searched FingerNo is returned.
	0	Searching failure. There is no searched FingerNo.
	DEV_ERR	Device error.

22) CheckIDFinger

Command	**152**	
Alias	**CheckIDFinger**	
Description	Check whether fingerprint of specified ID is enrolled or not.	
HandShake	CommandMode	
InParam	Refer to following	ID, FingerNo to check
Result	1	Unuse
OutParam	1	There is specified ID, FingerNo.
	0	The fingerprint of specified ID, FingerNo is not enrolled.
	ID_ERR	ID is out of the range.
	FINGERNO_ERR	FingerNo is out of the range.

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23) CheckID

Command	**153**	
Alias	**CheckID**	
Description	Check whether specified ID is enrolled or not.	
HandShake	CommandMode	
InParam	nnn	ID to check
Result	1	Unuse
OutParam	1	There is specified ID.
	0	The fingerprint of specified ID is not enrolled.
	ID_ERR	ID is out of the range.

Note!
Enrollment ID can be input from 1 to 32767.

24) CheckManager

Command	**154**	
Alias	**CheckManager**	
Description	Check whether specified ID is a manager or not.	
HandShake	CommandMode	
InParam	nnn	ID to check
Result	1	Unuse
OutParam	1	Specified ID is a manager.
	0	The fingerprint of specified ID is not enrolled.
	ID_ERR	ID is out of the range.

25) GetEnrollCount

Command	**155**	
Alias	**GetEnrollCount**	
Description	Get the number of currently enrolled fingerprint data.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	nnn	Number of currently enrolled fingerprint data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

26) ReadFpdata

Command	**160**	
Alias	**ReadFpdata**	
Description	Read fingerprint data from enrollment database.	
HandShake	ReceiveMode	
InParam	Refer to following	
Result	1	Unuse
OutParam	0	Unuse

InParam = 0 : Read all fingerprint data of enrollment database.
HIWORD(InParam) = 0 : Read fingerprint data of specified ID of enrollment database.
HIWORD(InParam) != 0 : Read fingerprint data of specified ID, FingerNo of enrollment database.

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

27) WriteFpdata

Command	**161**	
Alias	**WriteFpdata**	
Description	Write fingerprint data to enrollment database.	
HandShake	ReceiveMode	
InParam	Refer to following	ID of fingerprint data to write.
Result	1	Unuse
OutParam	0	Unuse

InParam = 0 : Write all fingerprint data of enrollment database.
HIWORD(InParam) = 0 : Write fingerprint data of specified ID of enrollment database.
HIWORD(InParam) != 0 : Write fingerprint data of specified ID, FingerNo of enrollment database.

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28) DeleteIDFinger

Command	**162**	
Alias	**DeleteIDFinger**	
Description	Delete fingerprint data with specified ID, FingerNo from enrollment database.	
HandShake	CommandMode	
InParam	Refer to following	ID, FingerNo of fingerprint data to delete.
Result	1	Unuse
OutParam	1	Success
	ID_ERR	Specified ID is out of range.
	FINGERNO_ERR	Specified FingerNo is out of range.
	DB_ERR	Writing error in enrollment database(FLASH Memory).

InParam					
HH		**HL**		**LH**	**LL**
7654	3210	7654	3210	Enrollment ID	
Reserved	manager	Reserved	Finger number		
0	0	0	1 ~ 10		

Note!
Enrollment ID can be input from 1 to 32767.

29) DeleteID

Command	**163**	
Alias	**DeleteID**	
Description	Delete fingerprint data with specified ID from enrollment database.	
HandShake	CommandMode	
InParam	nnn	ID of fingerprint data to delete.
Result	1	Unuse
OutParam	1	Success
	ID_ERR	Specified ID is out of range.
	FINGERNO_ERR	Specified FingerNo is out of range.
	DB_ERR	Writing error in enrollment database(FLASH Memory).

30) DeleteAll

Command	**164**	
Alias	**DeleteAll**	
Description	Delete all fingerprint data from enrollment database. Immediately after this command is executed, all fingerprint data in database are deleted.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	1	Success
	DB_ERR	Writing error in enrollment database(FLASH Memory).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

31) WriteProductID

Command	**185**	
Alias	**WriteProductID**	
Description	Change Product ID of product. Manufacturer can write his ProductID to Flash memory and identify products using this. Product ID is not deleted when power off. Product ID is unsigned long type of integer. Default of Product ID is 0xFFFFFFFF.	
HandShake	CommandMode	
InParam	nnn	Product ID to write.
Result	1	Unuse
OutParam	1	Success
	DEV_ERR	Device error.

32) ReadProductID

Command	**186**	
Alias	**ReadProductID**	
Description	Read Product ID of product. Manufacturer can write his ProductID to Flash memory and identify products using this. Product ID is not deleted when power off. Product ID is unsigned long type of integer. Default of Product ID is 0xFFFFFFFF.	
HandShake	CommandMode	
InParam	0	Unuse
Result	1	Unuse
OutParam	nnn	Product ID

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

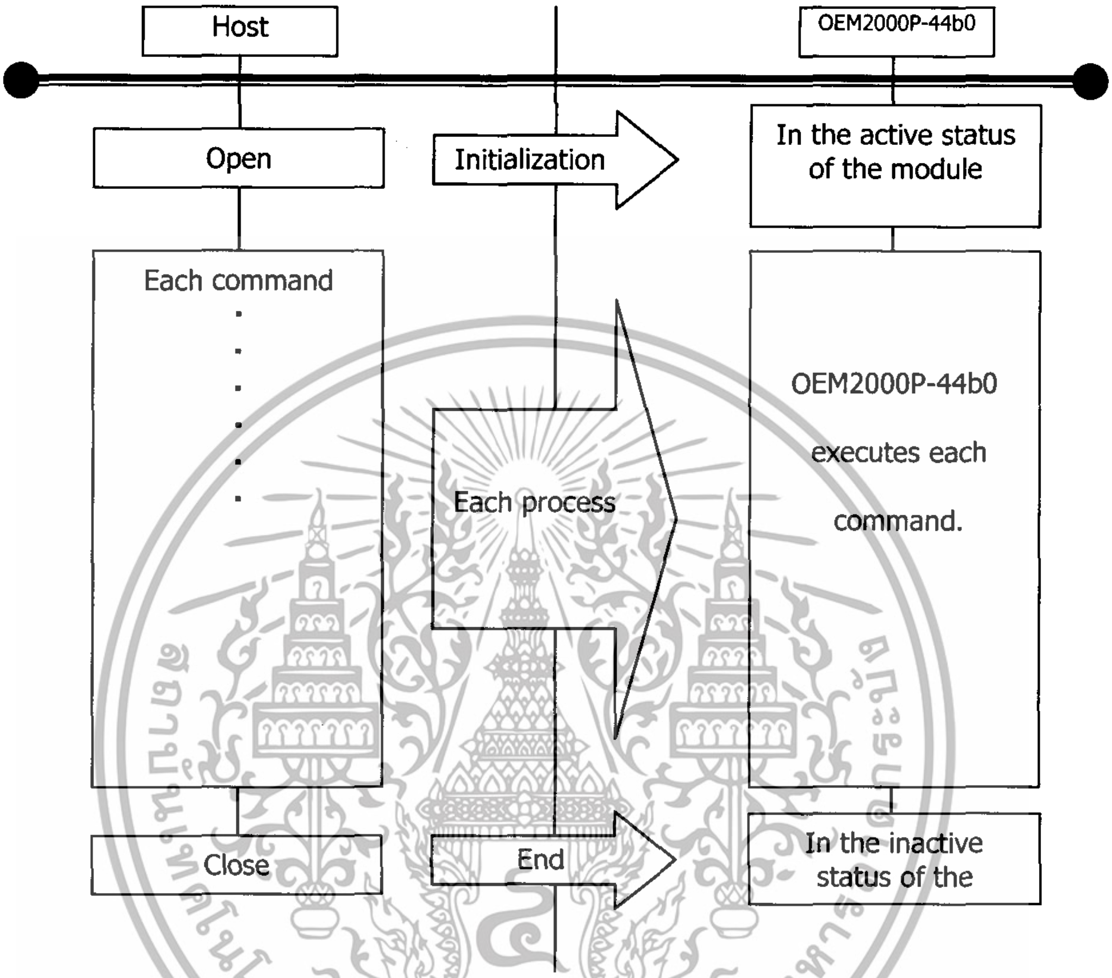
5. Error Code

Error	Value
IMAGE_ERR	-1
FPDATA_ERR	-2
ID_ERR	-3
OVER_ERR	-4
SECURITY_ERR	-5
SENSOR_ERR	-6
DEVSETUP_ERR	-7
FINGERNO_ERR	-8
DB_ERR	-9
COMM_ERR	-10
OTHER_ERR	-11
PARAM_ERR	-12
MERGE_ERR	-13
NTH_ERR	-14
TURN_ERR	-15
NOTENROLL_ERR	-16
INPARAM_ERR	-32
BLOCK_SIZE_ERR	-33
BLOCK_NUM_ERR	-34
DEV_ERR	-100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. How to Use Commands

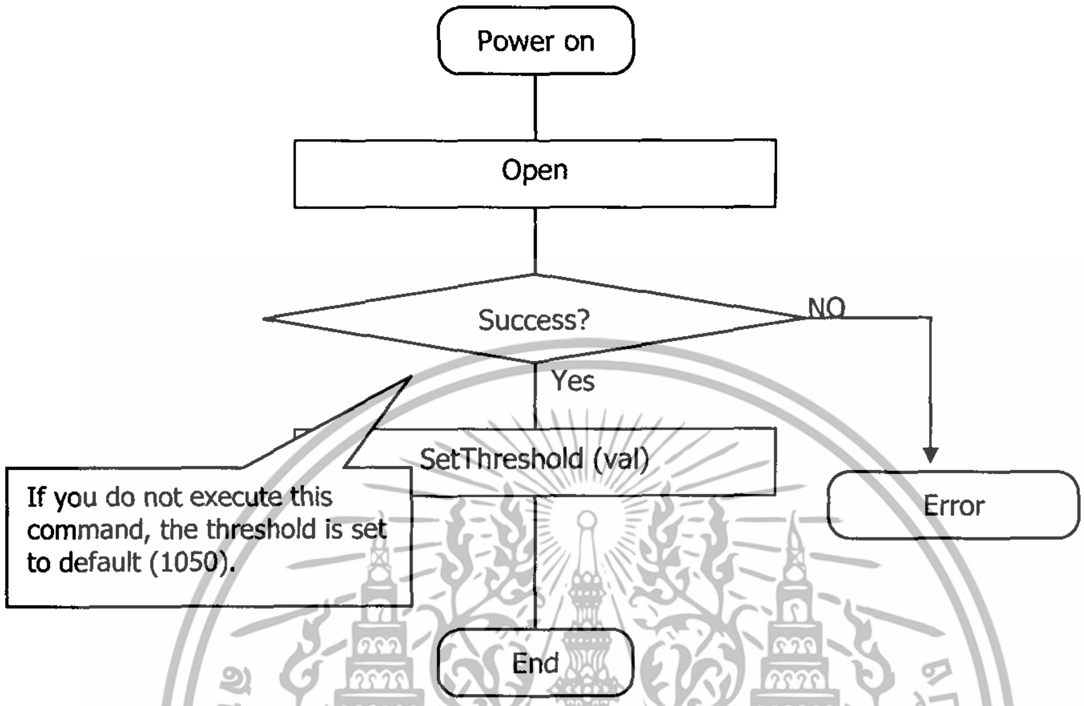
6.1. Basic Performance of OEM2000P-44B0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

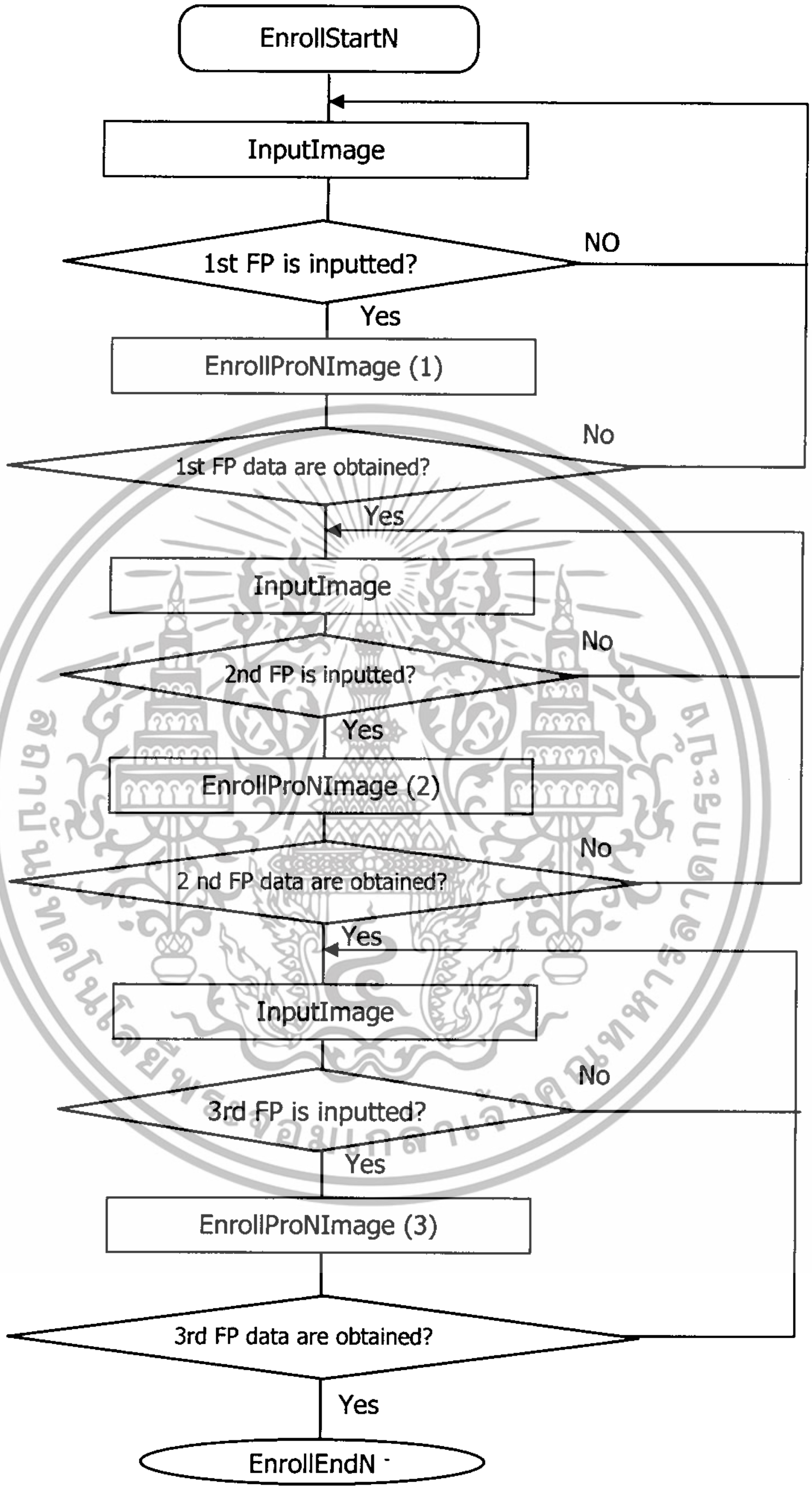
6.2. Examples of Applying Fingerprint Processing Algorithm

6.2.1. Initialize OEM2000P-44B0



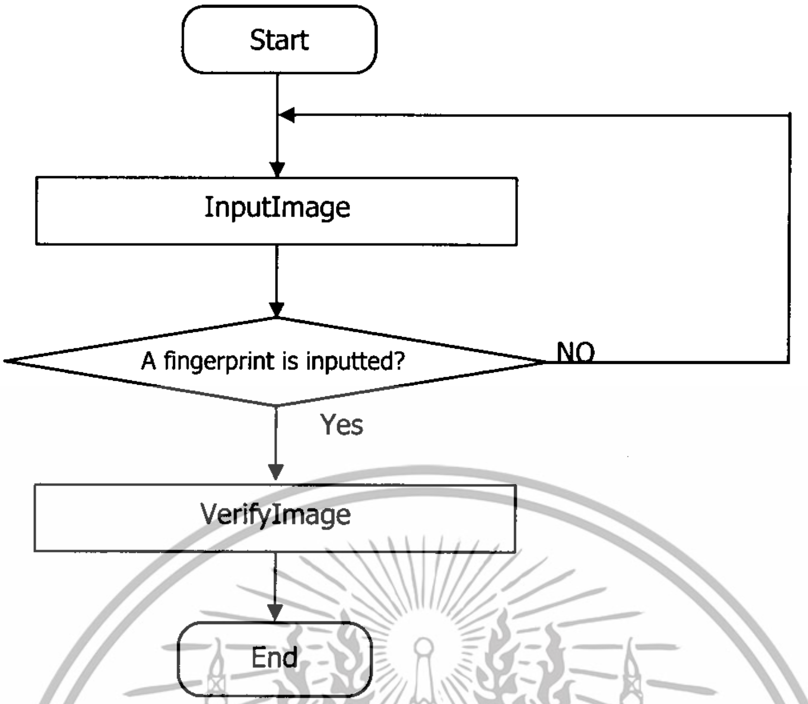
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2. Enrollment

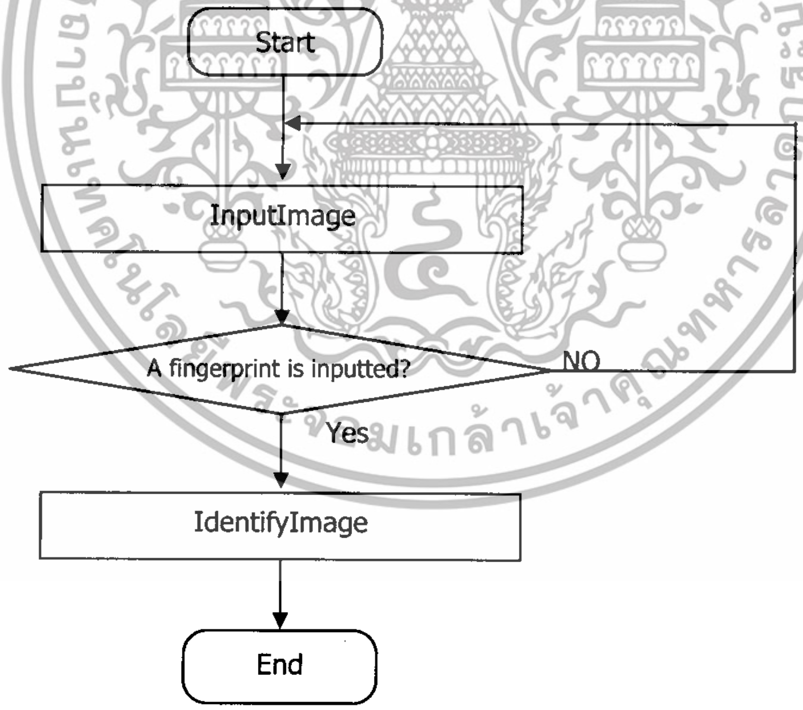


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3. 1:1 Verify



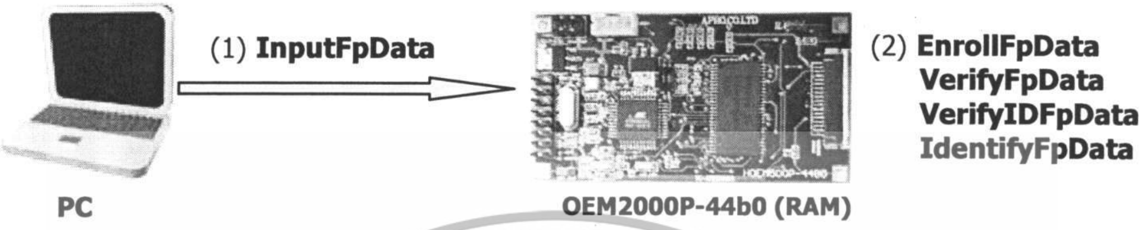
6.2.4. 1:N Identify



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.5. How to use command for Fingerprint

- InputFpData
- EnrollFpData
- VerifyFpData
- VerifyIDFpData
- IdentifyFpData



First InputFpData is called and feature data of PC are transmitted to RAM of OEM board.
Feature data of PC is one obtained from ReadFpDB.
In enrollment time EnrollFpData is called and transmitted feature data are enrolled in flash database of ARM7OEM board.
In 1:1 Verification time VerifyFpData is called.
In 1:ID Verification time VerifyIDFpData is called.
VerifyFpData, VerifyIDFpData and IdentifyFpData are called in 1:1 Verification, 1:ID Verification and 1:N Identification time respectively.

6.2.6. How to Use Command for Fingerprint Database

- WriteFpDB
- ReadFpDB

These functions are used to manipulate the database of OEM board
To read or write one fingerprint data through RS-232C from or to flash database, appoint ID and FingerNo in InParam(refer to 4-2) and call WriteFpDB, ReadFpDB.
To read or write all fingerprint data enrolled in a certain ID through RS-232C from or to flash database, appoint ID in InParam(refer to 4-2) and call WriteFpDB, ReadFpDB.
To read or write all fingerprint data enrolled in flash database through RS-232C from flash database, set as InParam = 0 (refer to 4-2) and call WriteFpDB, ReadFpDB.

This document may be modified without any notice.
This document is protected by copyright law and international treaties.
Unauthorized reproduction or distribution of all or a part of this document may result in severe civil and criminal penalties.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARALLAX

Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

24LC128 (#602-00013)

General Description

The 24LC128 is a 128 Kb (16K X 8) I²C CMOS Serial EEPROM that can provide additional non-volatile data storage for the BASIC Stamp® 2 Modules. The 8-pin DIP package is easy to interface to the BS2, BS2e, BS2sx, BS2p24/40, BS2pe and BS2px24.

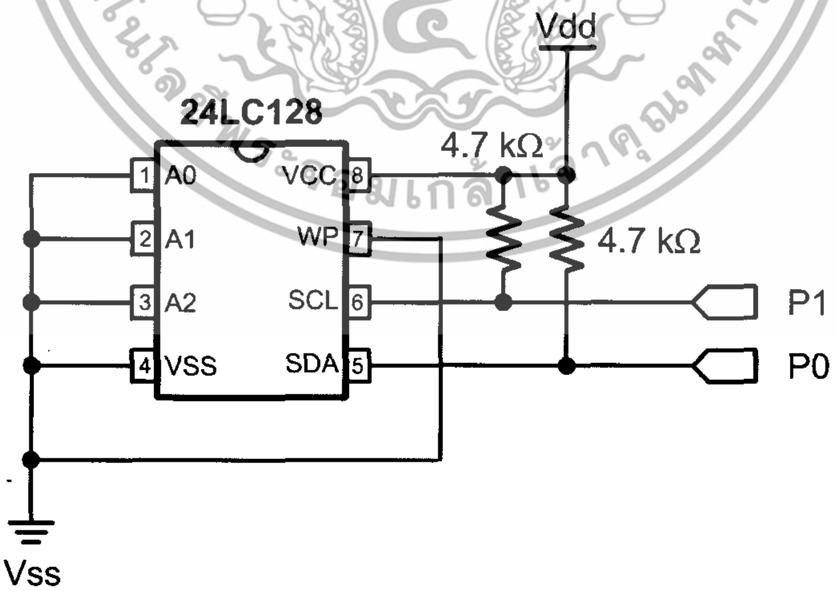
Features

- Low-power CMOS technology
- Maximum write current 3 mA at 5.5 V
- Maximum read current 400 µA at 5.5 V
- Standby current 100 nA typical at 5.5 V
- 5 mS max write cycle time
- 1,000,000 erase/write cycles
- 2-wire serial interface, I²C™ compatible
- 8-pin .300" DIP Package
- Compatible with all BASIC Stamp 2 and SX Modules

Application Ideas

- Data storage
- Data tables

Quick Start Circuit



เอกสารนี้เป็นเอกสาร... เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connecting and Testing

Connect the 24LC128 as shown in the Quick Start Circuit, then use the appropriate demo program for your BASIC Stamp module to verify correct operation. If you are using an SX Microcontroller you will need to write or edit some existing I²C code to access the EEPROM.

Resources and Downloads

Check out the 24LC128 product page for example programs, articles and more:

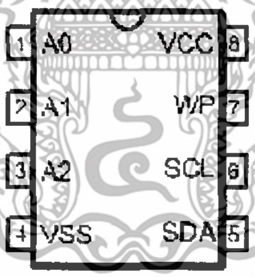
http://www.parallax.com/detail.asp?product_id=602-00013

Device Information

Pin Definitions and Ratings

Pin	Name	Function
1	A0	Address Pin 0
2	A1	Address Pin 1
3	A2	Address Pin 2
4	Vss	Ground -> 0 V
5	SDA	Serial Data Input/Output
6	SCL	Synchronous Clock Input
7	WP	Write-Protect Pin
8	Vcc	+5 V Power Input

24LC128



Symbol	Quantity	Minimum	Typical	Maximum	Units
Vcc	Supply Voltage	2.5	5.0	6.5	V
GND	Ground		0		V
Icc (sb)	Standyby Supply Current		100		nA
Ir	Read Current		400		uA
Iw	Write Current		3		mA

-* Ratings from manufacturer datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้