

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบการเข้าออกสำนักงานโดยใช้หลักการตรวจสอบจากลายนิ้วมือ

OFFICE ACCESS CHECKING SYSTEM BY FINGERPRINT



T117518



เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

117518

5 ต.ค. 2554

b.....
i.....

ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบการเข้าออกสำนักงานโดยใช้หลักการตรวจสอบจากลายนิ้วมือ
OFFICE ACCESS CHECKING SYSTEM BY FINGERPRINT

โดย

นายอรรณวิท ศรีคำไทย

49011155

นายนรภัทร ปัตตาเทสัง

49011160

อาจารย์ที่ปรึกษา

ผศ.ดร.สุทธิชัย นพนาศิพงษ์

ผศ. สุรพล บุญจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

ผ่านการตรวจงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2552

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบการเข้าออกสำนักงานโดยใช้หลักการตรวจสอบจากลายนิ้วมือ

OFFICE ACCESS CHECKING SYSTEM BY FINGERPRINT

ผู้จัดทำ

1. นายอรรณวิทย์ ศรีคำไทย 49011155
2. นายนรภัทร ปัตตาเทสัง 49011160



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จได้ด้วยความกรุณา ของ ผศ.ดร.สุทธิชัย นพนาถิพงษ์ และ ผศ.สุรพล บุญจันทร์ อาจารย์ที่ปรึกษาปริญญาบัตร ซึ่งได้ให้คำปรึกษา ข้อชี้แนะและความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ที่คอยช่วยเหลือเป็นกำลังใจให้ข้าพเจ้า

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดีพร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

นายอรรณวิท ศรีคำไทย
นายนรภัทร ปัตตาเทสัง
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบการเข้าออกสำนักงานโดยใช้หลักการ
ตรวจสอบจากลายนิ้วมือ
OFFICE ACCESS CHECKING SYSTEM BY
FINGERPRINT

โดย นายอรรถวิท ศรีคำไทย 49011155
นายนรภัทร ปัดดาเทสัง 49011160

อาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาถิพงษ์
ผศ. สุรพล บุญจันทร์

บทคัดย่อ

โครงการนี้เป็นการออกแบบและสร้างระบบตรวจสอบการเข้า - ออกสำนักงาน โดยใช้ลายนิ้วมือและรหัสพนักงานในการตรวจสอบ ประกอบด้วย 4 ส่วน ได้แก่ ส่วนรับข้อมูล ส่วนประมวลผล ส่วนฐานข้อมูล และส่วนแสดงผล ส่วนรับข้อมูลทำหน้าที่รับข้อมูลลายนิ้วมือและรหัสพนักงาน ส่วนประมวลผลทำหน้าที่ตรวจสอบและควบคุมการทำงานของส่วนต่างๆ ส่วนฐานข้อมูลทำหน้าที่จัดเก็บข้อมูล และส่วนแสดงผลทำหน้าที่แสดงสถานะการทำงานให้ผู้ใช้ทราบ การใช้งานระบบจะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วนผู้ใช้งาน และส่วนของผู้ดูแลระบบ

ABSTRACT

This project presents and designs the office access checking by fingerprint by using fingerprint and password of employees for checking. Consists of four parts following information part , processing part , database part and display part. The information part has a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

duty to get fingerprint and password data of employees. the processing part has a duty to check and control system. The database part uses for record datas and the display part shows working status of user processing. The applications will be divided into two parts that first is a user and second is a administrator.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า	
กิตติกรรมประกาศ	I	
บทคัดย่อ	II	
สารบัญ	IV	
สารบัญรูป	VII	
สารบัญตาราง	X	
บทที่ 1	บทนำ	1
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	2
	1.3 ขอบเขตของโครงการ	2
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง	3
	2.1 เครื่องสแกนลายนิ้วมือ U.ARE.U รุ่น 4000B	3
	2.1.1 คุณสมบัติ U.ARE.U 4000B	4
	2.1.2 รายละเอียดการใช้งาน	4
	2.2 ลักษณะทั่วไปของลายนิ้วมือ	5
	2.2.1 แบบแผนพื้นฐานของลายนิ้วมือ	5
	2.2.2 จุดสังเกตลายนิ้วมือ	9
	2.3 ไมโครคอนโทรลเลอร์ PIC18F458	10
	2.3.1 โครงสร้างของไมโครคอนโทรลเลอร์ PIC18F458	11
	2.3.2 คุณสมบัติทางเทคนิคของ PIC18F458	13
	2.4 โมดูล LCD (LIQUID CRYSTAL DISPLAY)	14
	2.4.1 การติดต่อแบบ 8 บิต	15
	2.4.2 การติดต่อแบบ 4 บิต	17
	2.5 การเชื่อมต่อผ่านพอร์ทอนุกรม RS-232	18

สารบัญ (ต่อ)

	หน้า
2.5.1 การจัดหาของ CONNECTOR อนุกรมแบบ DB9 และหน้าที่ใช้งานต่างๆ	19
2.5.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9	20
2.5.3 ระดับสัญญาณของ RS232	21
2.5.4 อัตราการส่งข้อมูล (BAUD RATE)	22
2.5.5 รูปแบบการสื่อสารแบบอนุกรม	22
2.5.6 ลักษณะการรับ-ส่งข้อมูลแบบอนุกรม	23
บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์	24
3.1 การออกแบบ	24
3.1.1 ส่วนรับข้อมูล	24
3.1.2 ส่วนประมวลผล	26
3.1.3 ส่วนฐานข้อมูล	27
3.1.4 ส่วนแสดงผล	30
3.2 การออกแบบโปรแกรมในส่วนผู้ใช้งาน	30
3.3 การออกแบบโปรแกรมในส่วนของผู้ดูแลระบบ	32
3.4 การออกแบบ โปรแกรมของส่วนแสดงผล	34
3.4.1 การทำงานสถานะปกติ	34
3.4.2 การทำงานสถานะตรวจสอบ	34
3.4.3 ส่วนแสดงผลได้รับคำสั่ง A	36
3.4.4 ส่วนแสดงผลได้รับคำสั่ง B	37
3.4.5 ส่วนแสดงผลได้รับคำสั่ง C	37
3.4.6 ส่วนแสดงผลได้รับคำสั่ง D	37
3.4.7 ส่วนแสดงผลได้รับคำสั่ง E	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.4.8 ส่วนแสดงผลได้รับคำสั่งปรับเวลา	37
บทที่ 4 ผลการทดลอง	38
4.1 การเชื่อมต่อส่วนแสดงผลเข้ากับส่วนประมวลผล	40
4.2 การทำงานในส่วนของผู้ใช้งาน	41
4.2.1 สถานะปกติ	41
4.2.2 สถานะรหัสประจำตัวไม่ถูกต้อง	41
4.2.3 สถานะรอกการสแกนลายนิ้วมือ	42
4.2.4 สถานะลายนิ้วมือไม่ถูกต้อง	43
4.2.5 สถานะตรวจสอบลายนิ้วมือถูกต้อง	43
4.3 การทำงานในส่วนของผู้ดูแลระบบ	44
4.3.1 การเพิ่มผู้ใช้งานเข้าระบบ	44
4.3.2 การลบผู้ใช้งานออกจากระบบ	47
บทที่ 5 บทสรุปและวิจารณ์	49
5.1 สรุปผลการทำงานของระบบ	49
5.2 บทวิจารณ์	49
บรรณานุกรม	50
ภาคผนวก ก	
ภาคผนวก ข	
ภาคผนวก ค	
ภาคผนวก ง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	เครื่องสแกนลายนิ้วมือ U.ARE.U รุ่น 4000B	3
2.2	รายละเอียดของตัวเครื่องเชิงกล	5
2.3	ลายนิ้วมือแบบก้นหอย	6
2.4	ลายนิ้วมือแบบมัดหวาย	6
2.5	ลายนิ้วมือแบบโค้ง	6
2.6	ลายนิ้วมือแบบก้นหอยปิดขวา	7
2.7	ลายนิ้วมือแบบก้นหอยธรรมดา	7
2.8	ลายนิ้วมือแบบก้นหอยปิดซ้าย	7
2.9	ลายนิ้วมือแบบมัดหวายวนขวา	8
2.10	ลายนิ้วมือแบบมัดหวายวนซ้าย	8
2.11	ลายนิ้วมือแบบมัดหวายคู่	8
2.12	ลายนิ้วมือแบบโค้งราบ	8
2.13	ลายนิ้วมือแบบโค้งกระโจม	8
2.14	แบบแผนลายเส้นพื้นฐานและพื้นที่ทั้งหมดของแบบแผนลายเส้น	9
2.15	จุดศูนย์กลางและจุดมโนทัยของลายนิ้วมือ	9
2.16	การทำงานพื้นฐานของไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมแบบฮาร์วาร์ด	10
2.17	ไดอะแกรมของกระบวนการ PIPELINE ที่ใช้ในไมโครคอนโทรลเลอร์ PIC	11
2.18	โครงสร้างของไมโครคอนโทรลเลอร์ PIC18F458	12
2.19	ขาใช้งานของ PIC18F458	14
2.20	จอแสดงผลแอลซีดี	14
2.21	การต่อใช้งาน LCD แบบ 8 BIT DATA	16
2.22	การต่อใช้งาน LCD แบบ 4 BIT DATA	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.23	การเชื่อมต่อระหว่างคอมพิวเตอร์กับบอร์ดของไมโครคอนโทรลเลอร์	19
2.24	พอร์ตอนุกรมของ PC DB9 ตัวผู้และพอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย	19
2.25	การเรียงขาของ DB9 เมื่อมองจากด้านหลัง	19
2.26	การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ NULL MODEM	20
2.27	การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น	20
2.28	ระดับสัญญาณของ RS232 และระดับสัญญาณของ TTL	21
2.29	สัญญาณนาฬิกาและสัญญาณข้อมูล	22
2.30	สัญญาณแบบอะซิงโครนัส (ASYNCHRONOUS)	23
3.1	BLOCK DIAGRAM รวมของโครงการ	24
3.2	KEYPAD	25
3.3	เครื่องสแกนลายนิ้วมือ	26
3.4	รายละเอียดในการเชื่อมต่อพอร์ตอนุกรม	27
3.5	ตาราง ENROLL ในมุมมองแผ่นข้อมูล	28
3.6	ตาราง ENROLL ในมุมมองออกแบบ	28
3.7	ตาราง LISTOFTIMESCAN ในมุมมองแผ่นข้อมูล	29
3.8	ตาราง LISTOFTIMESCAN ในมุมมองออกแบบ	29
3.9	ตาราง USER_INFO ในมุมมองแผ่นข้อมูล	30
3.10	ตาราง USER_INFO ในมุมมองออกแบบ	30
3.11	โฟลวชาร์ตการทำงานของระบบในส่วนผู้ใช้งาน	32
3.12	โฟลวชาร์ตการทำงานในส่วนของผู้ดูแลระบบ	33
3.13	โฟลวชาร์ตการทำงานของส่วนแสดงผล	36
4.1	ตรวจสอบหมายเลขพอร์ต RS-232	38
4.2	กล่องข้อความเตือนให้ใส่หมายเลขคอมพอร์ทก่อนใช้งาน	39

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.3	ใส่หมายเลขคอมพิวเตอร์ไม่ถูกต้อง	39
4.4	การเชื่อมต่อถูกต้อง ระบบพร้อมที่จะใช้งาน	40
4.5	จอแสดงผล LCD ในสถานะปกติ	40
4.6	สถานะปกติ	41
4.7	สถานะรหัสประจำตัวไม่ถูกต้อง	42
4.8	สถานะรอกการสแกนลายนิ้วมือ	42
4.9	สถานะลายนิ้วมือไม่ถูกต้อง	43
4.10	สถานการณ์ลายนิ้วมือถูกต้อง	44
4.11	กรอกข้อมูลครบถ้วนแล้วทำการถ่ายรูป	44
4.12	กล่องข้อความเตือนให้กรอกข้อมูล	45
4.13	รอกการสแกนลายนิ้วมือ	45
4.14	ทำการบันทึกข้อมูลแล้ว	46
4.15	ตาราง ENROLL ก่อนและหลังทำการบันทึก	46
4.16	ตาราง USER_INFO ก่อนและหลังทำการบันทึก	47
4.17	ค้นหาพนักงาน โดยใช้รหัสพนักงาน	48
4.18	กล่องข้อความยืนยันการลบ	48

สารบัญตาราง

ตารางที่		หน้า
2.1	รายละเอียดของเครื่องสแกนลายนิ้วมือ U.ARE.U 4000B	4
2.2	โครงสร้างของไมโครคอนโทรลเลอร์	12
2.3	ชื่อและหน้าที่ของขาสัญญาณต่างๆของ LCD	15
2.4	หน้าที่ใช้งานของขาต่างๆของ DB9	20
3.1	เงื่อนไขการทำงานของส่วนแสดงผล	34



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาในการจัดทำปริญญาณิพนธ์

ในอดีตจนถึงปัจจุบันวิธีการระบุตัวบุคคลที่ได้รับความนิยม คือ การตรวจสอบหมายเลขบัตรประจำตัวประชาชน, ตำแหน่งรอยแผลเป็นตามร่างกาย เป็นต้น ซึ่งเป็นวิธีที่ง่ายและสะดวก ทำให้สามารถทำงานได้อย่างรวดเร็ว แต่ในทางตรงกันข้ามก็มีความผิดพลาดค่อนข้างสูง เพราะวิธีการดังกล่าวสามารถลอกเลียนได้ง่ายและมีความเป็นเอกลักษณ์ค่อนข้างต่ำ เพราะฉะนั้นนักวิจัยจึงได้มีการทำการวิจัยเพื่อค้นหาวิธีการพิสูจน์และระบุตัวบุคคลที่มีประสิทธิภาพและแม่นยำมากขึ้นเพื่อเลี่ยงข้อผิดพลาดต่างๆ จากการศึกษาพบว่า การพิสูจน์และระบุตัวตนของบุคคล โดยการอาศัยข้อมูลพื้นฐานทางด้านร่างกายของบุคคลหรือเรียกว่า ชีวมาตร (Biometrics) ซึ่งสามารถทำได้หลายวิธี เช่น การตรวจสอบโครงสร้างใบหน้า การตรวจสอบม่านตา การตรวจสอบดีเอ็นเอ และการตรวจสอบลายนิ้วมือ เป็นต้น ซึ่งวิธีชีวมาตรเป็นวิธีการระบุตัวบุคคลที่มีประสิทธิภาพมากที่สุดในปัจจุบัน เพราะเป็นวิธีที่ยากต่อการเลียนแบบ มีความคงสภาพสูง และมีเอกลักษณ์ของแต่ละบุคคล จึงเป็นวิธีที่แพร่หลายในปัจจุบันและมีแนวโน้มที่จะเพิ่มมากขึ้นในอนาคต

การระบุตัวบุคคลด้วยการใช้ลายนิ้วมือนั้นได้รับความนิยมอย่างแพร่หลายในปัจจุบัน เพราะผิวหนังนิ้วมือของคนเราที่ประกอบด้วยส่วนที่เป็นร่อง (Furrow) และส่วนที่เป็นเส้นหรือสัน (Ridge) ที่เราเรียกรวมกันว่าลายนิ้วมือ ซึ่งมีความเป็นเอกลักษณ์เฉพาะบุคคลและมีความคงสภาพไม่เปลี่ยนแปลง นอกจากนี้ลายนิ้วมือยังสามารถนำมาตรวจสอบและประมวลผลได้ง่ายและรวดเร็ว อีกทั้งโครงสร้างลายนิ้วมือมีความซับซ้อนยากต่อการซ้ำซ้อนกับบุคคลอื่น ซึ่งทำให้การระบุตัวบุคคลโดยการใช้ลายนิ้วมือเป็นวิธีการตรวจสอบทางชีวมาตรที่ง่ายกว่าวิธีชีวมาตรวิธีอื่นๆ ในปัจจุบัน ดังนั้น จากเหตุผลดังกล่าว ผู้ศึกษาจึงสนใจทำปริญญาณิพนธ์ในเรื่องการตรวจสอบการเข้า-ออกสำนักงาน โดยใช้ลายนิ้วมือเพื่อใช้ตรวจสอบควบคุมความปลอดภัยในด้านการระบุตัวบุคคลซึ่งเป็นประโยชน์ต่อสังคม องค์กร สำนักงานต่างๆ ต่อไป

1.2 วัตถุประสงค์ของปฏิญานิพนธ์

เพื่อศึกษาและออกแบบการสร้างเครื่องตรวจสอบการเข้า – ออกสำนักงาน โดยใช้ลายนิ้วมือและรหัสพนักงานพร้อมทั้งบันทึกเวลาเข้า - ออกและแสดงสถานะผ่านจอ LCD เพื่อประโยชน์ต่อสังคม องค์กร สำนักงานในด้านความปลอดภัย ประหยัด อย่างมีประสิทธิภาพ

1.3 ขอบเขตของปฏิญานิพนธ์

1.3.1 ศึกษาการทำงานของโปรแกรม Microsoft Access ในการใช้เก็บข้อมูล โดยแบ่งเป็น 3 ตาราง ได้แก่ ตารางที่ใช้เก็บรายละเอียดของพนักงาน ตารางที่ใช้เก็บข้อมูลลายนิ้วมือและตารางที่ใช้เก็บเวลาที่มีผู้เข้า-ออก

1.3.2 ศึกษาการใช้งานโปรแกรม Microsoft Visual C# ให้สามารถทำงานร่วมกับโปรแกรม Microsoft Access

1.3.3 ศึกษาการใช้งานโปรแกรม Microsoft Visual C# ในการรับข้อมูลจากเครื่องอ่านลายนิ้วมือมาทำการประมวลผลกับข้อมูลลายนิ้วมือในฐานข้อมูล

1.3.4 ศึกษาการใช้งานไมโครคอนโทรลเลอร์ PIC18F458 ในการรับข้อมูลผ่านทางพอร์ตอนุกรมและแสดงผลออกทางจอ LCD

1.3.5 การตรวจสอบ ตรวจสอบเฉพาะรหัสพนักงานและลายนิ้วมือ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เครื่องสแกนลายนิ้วมือ U.are.U รุ่น 4000B

เป็นเครื่องสแกนลายนิ้วมือที่ใช้เทคโนโลยีเซ็นเซอร์แสง สำหรับเทคโนโลยีสแกนลายพิมพ์นิ้วมือที่นิยมใช้ในปัจจุบัน โดยทั่วไปแล้วอาจแบ่งได้ 3 ลักษณะ ได้แก่ เซ็นเซอร์ประจุไฟฟ้า (capacitive sensor) เซ็นเซอร์อุณหภูมิ (thermal sensor) และเซ็นเซอร์แสง (optical sensor) เทคโนโลยีสองแบบแรก ผู้ใช้ต้องแตะไปที่เซ็นเซอร์โดยตรง ทำให้อายุใช้งานสั้น และมีปัญหาด้านการบำรุงรักษา โดยเฉพาะในต่างประเทศที่มีอากาศหนาวจะเกิดไฟฟ้าสถิตทำลายผิวเซ็นเซอร์ได้ ขณะที่แบบแสงจะมีความทนทานกว่าและมีคุณสมบัติรวมของเทคโนโลยีอื่นมาไว้ด้วย หลักการทำงานของเซ็นเซอร์แสง คือการสะท้อนกลับหมดของแสง ด้วยการอาศัยแสงส่องลายนิ้วมือที่วางอยู่บนเลนส์สะท้อนกับตัวหัวเซ็นเซอร์ เช่นเดียวกับการใช้นิ้วจับแก้วที่มีน้ำอยู่ภายใน ซึ่งการสะท้อนของแสง ทำให้เรามองเห็นลายนิ้วมือได้อย่างชัดเจน



รูปที่ 2.1 เครื่องสแกนลายนิ้วมือ U.are.U รุ่น 4000B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติ U.are.U 4000B

- 2.1.1.1 ความสามารถของหัวอ่านในการแยกจุด 512 dpi
- 2.1.1.2 พื้นที่ทำการสแกนกว้าง 14.6 มม.(ที่กึ่งกลาง) ยาว 18.1 มม.
- 2.1.1.3 ให้ภาพ 8 บิต (256 ระดับสี) ขาว/ดำ
- 2.1.1.4 ขนาดเครื่องโดยประมาณยาว 79 มม. กว้าง 49 มม. สูง 19 มม.
- 2.1.1.5 รองรับการทำงานต่อมาตรฐาน USB 1.0, 1.1 และ 2.0

2.1.2 รายละเอียดการใช้งาน

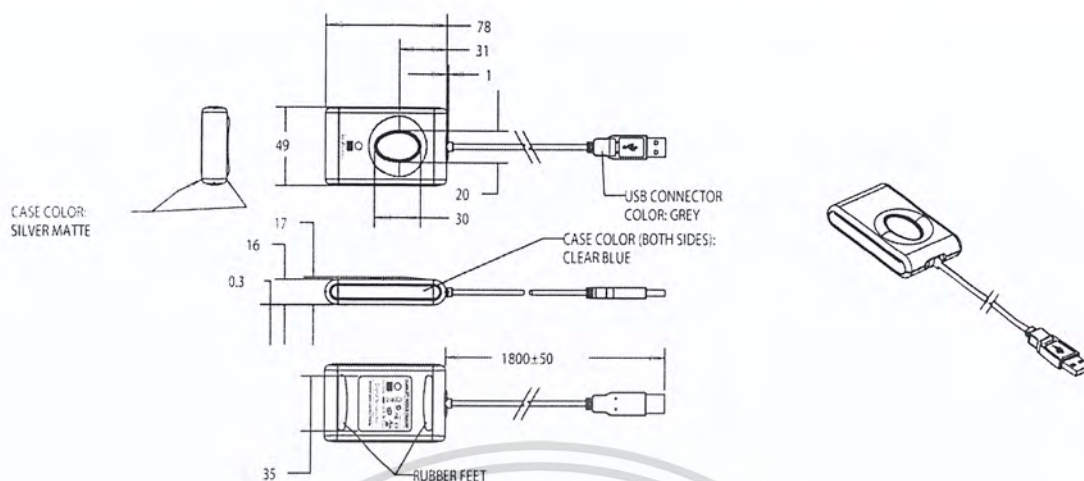
ตัวเครื่องมีรายละเอียดต่างๆ ดังต่อไปนี้

ตารางที่ 2.1 รายละเอียดของเครื่องสแกนลายนิ้วมือ U.are.U 4000B

Description	Specification
Connection type	USB Port
Supply Voltage	5.0V \pm 5V supplied by USB
Supply Current – scanning	190 mA (typ.)
Supply Current – idle mode	140 mA (typ.)
Supply Current – suspend mode	1.5 mA (max.)
ESD Susceptibility	>15 KV, mounted in case
Temperature, Operating	0 - 40 C
Humidity, Operating	20% - 80% non-condensing
Humidity, Storage	20% - 90% non-condensing
Scan data	8-bit grayscale
Standards Compliance	FCC Class B, CE, VCCI
Software Support	win98, ME, XP

รายละเอียดของตัวเครื่องเชิงกลสามารถแสดงได้ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 รายละเอียดของตัวเครื่องเชิงกล

2.2 ลักษณะทั่วไปของลายนิ้วมือ

ลายเส้นผิวหนัง มาจากคำภาษาอังกฤษว่า dermal ridge หรือ dermatoglyphics หมายถึง ลายเส้นบนฝ่ามือ (palmprint) ลายนิ้วมือ (fingerprint) ลายฝ่าเท้า (footprint) มีลักษณะเป็น เส้นนูนปรากฏบนผิวหนังนิ้วมือ และนิ้วเท้าของทุกคน เป็นลักษณะเฉพาะของแต่ละบุคคล แม้แต่ฝาแฝดที่เกิดจากไข่ใบเดียวกัน (identical twins) ก็มีลักษณะลายเส้นผิวหนังแตกต่างกัน การสร้างลายเส้นบนนิ้วมือถูกควบคุมด้วยยีนบนโครโมโซมร่างกายมากถึง 7 ตำแหน่ง และเป็นการถ่ายทอดทางพันธุกรรมที่สิ่งแวดล้อมมีอิทธิพลร่วมด้วย ยีนส์หลายคู่มีปฏิกริยาร่วมกับสิ่งแวดล้อม ในระยะตัวอ่อนในครรภ์ (prenatal stress) มีผลให้แต่ละคนมีเส้นลายนิ้วมือที่แตกต่างกันไป ดังนั้นจึงมีการนำลายเส้นผิวหนังโดยเฉพาะลายนิ้วมือไปใช้ประโยชน์ในด้านนิติวิทยาศาสตร์ คือ การพิสูจน์บุคคล และด้านการแพทย์ในการช่วยวินิจฉัยโรคพันธุกรรมได้อีกด้วย ซึ่งประกอบด้วยลักษณะดังนี้

2.2.1 แบบแผนพื้นฐานของลายนิ้วมือ

การจำแนกแบบแผนลายเส้นพื้นฐาน อาจแบ่งได้หลากหลาย แต่ที่นิยมใช้กันมากที่สุด แบ่งได้เป็น 3 แบบหลักๆ ได้แก่ ก้นหอย (Whorl) มัดหวาย (Loop) และโค้ง (Arch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายนิ้วมือแบบลายก้นหอย (Whorl) มีประมาณ 30% ลายนิ้วมือแบบลายมัดหวาย (Loop) มีประมาณ 65% และลายนิ้วมือแบบลายโค้ง (Arch) มีประมาณ 5% การแบ่งลายนิ้วมือออกเป็นหลายประเภทนี้เพื่อวัตถุประสงค์ในการเพิ่มความรวดเร็วในการตรวจสอบลายนิ้วมือขั้นพื้นฐาน แต่ไม่ได้เป็นสิ่งที่ใช้ในการบอกความเหมือนหรือความแตกต่างระหว่างลายนิ้วมือ แต่จะใช้ลักษณะของสัน (Ridge) ของลายนิ้วมือ เช่น การสิ้นสุดของสัน (Ridge Ending) สันแบบลายจุด (Dots) สันที่แตกแขนง (Bifurcations) หรือรูปแบบต่างๆของสันที่เกิดขึ้น เป็นสิ่งที่ใช้ในการเปรียบเทียบลายนิ้วมือ แบบแผนลายนิ้วมือที่นิยมใช้กันมากที่สุดสามารถแสดงได้ดังรูปที่ 2.3 – 2.5



Whorl

รูปที่ 2.3 ลายนิ้วมือแบบก้นหอย



Loop

รูปที่ 2.4 ลายนิ้วมือแบบมัดหวาย



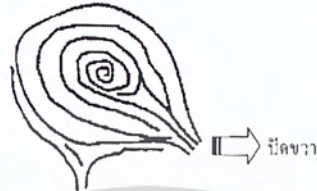
Arch

รูปที่ 2.5 ลายนิ้วมือแบบโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 รูปแบบของลายนิ้วมือแบบก้นหอย

ลายนิ้วมือแบบก้นหอย แบ่งออกเป็นก้นหอยปิดขวา ก้นหอยธรรมดาและก้นหอยปิดซ้ายสามารถแสดงได้ดังรูปที่ 2.6 – 2.8



รูปที่ 2.6 ลายนิ้วมือแบบก้นหอยปิดขวา

รูปที่ 2.7 ลายนิ้วมือแบบก้นหอยธรรมดา

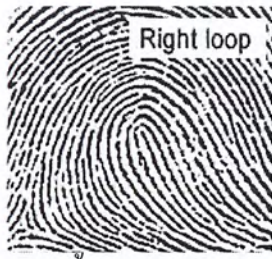


รูปที่ 2.8 ลายนิ้วมือแบบก้นหอยปิดซ้าย

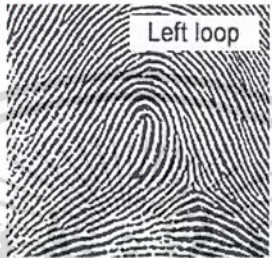
2.2.1.2 รูปแบบของลายนิ้วมือแบบมัดหวาย

ลายนิ้วมือแบบมัดหวาย แบ่งออกเป็นมัดหวายวนขวา มัดหวายวนซ้ายและมัดหวายคู่สามารถแสดงได้ดังรูปที่ 2.9 – 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ลายนิ้วมือแบบมัดหวายวนขวา



รูปที่ 2.10 ลายนิ้วมือแบบมัดหวายวนซ้าย



รูปที่ 2.11 ลายนิ้วมือแบบมัดหวายคู่

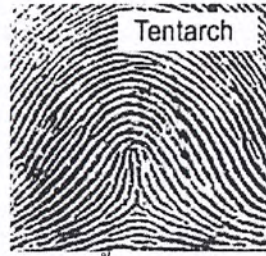
2.2.1.3 รูปแบบของลายนิ้วมือแบบโค้ง

ลายนิ้วมือแบบ โค้ง แบ่งออกเป็น โค้งราบและ โค้งกระโจม



รูปที่ 2.12 ลายนิ้วมือแบบ โค้งราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



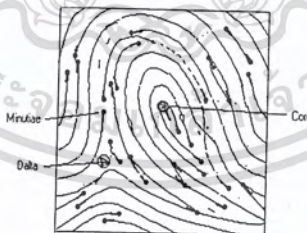
รูปที่ 2.13 ลายนิ้วมือแบบ โค้งกระโจม

2.2.2 จุดสังเกตลายนิ้วมือ

ลักษณะลายนิ้วมือที่มองเห็นได้ด้วยตาเปล่า ประกอบด้วยแบบแผนลายเส้นพื้นฐาน (basic ridge pattern) พื้นที่ทั้งหมดของแบบแผนลายเส้น (pattern area) จุดใจกลาง (core area) สามเหลี่ยมเคลด้าหรือสันคอง (delta, triradius) ชนิดของเส้น (typelines) และจำนวนเส้นลายนิ้วมือ (ridge count) จุดสังเกตลายนิ้วมือสามารถแสดงได้ดังรูปที่ 2.14 – 2.15



รูปที่ 2.14 แบบแผนลายเส้นพื้นฐานและพื้นที่ทั้งหมดของแบบแผนลายเส้น



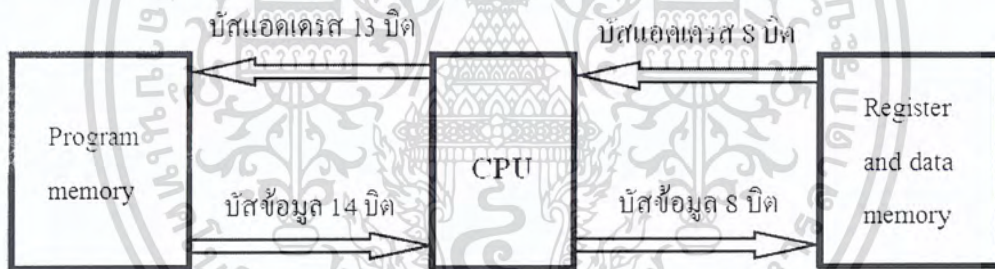
รูปที่ 2.15 จุดศูนย์กลางและจุดมีนุเทียของลายนิ้วมือ

ลายนิ้วมือบริเวณพื้นที่ของแบบแผนลายเส้นจะถูกนำมาทำการหาจุดศูนย์กลางและจุดมีนุเทียเพื่อนำไปใช้ในการตรวจสอบกับลายนิ้วมือต้นฉบับว่าเป็นลายนิ้วมือเดียวกันหรือไม่

ในการบันทึกลายนิ้วมือ ก็จะมีการบันทึกตำแหน่งของจุดสังเกต ซึ่งปกติแล้วจะบันทึกไว้ประมาณ 30-40 จุดต่อ 1 ลายนิ้วมือ และนำข้อมูลมาเปรียบเทียบกับลายนิ้วมือที่สแกนเข้ามาว่า ตรงกันหรือไม่ สำหรับการตรวจดูลายนิ้วมือ 2 รายการว่าเป็นลายนิ้วมือเดียวกันหรือไม่ นั้น มีหลากหลายวิธีด้วยกัน แต่วิธีที่เป็นที่นิยม กันมากที่สุด ได้แก่การเปรียบเทียบจุดสังเกตเล็ก ๆ (minutiae) จุดที่ว่านี้คือจุดที่เส้นลายนิ้วมือ (ridge) มาบรรจบกัน หรือ แยกออกจากกัน หรือ เป็นจุดจบของเส้น และจุดที่ว่านี้ของแต่ละคนจะไม่เหมือนกัน (FBI ได้สรุปแล้วว่า คนแต่ละคนจะไม่มีทางมีจุดสังเกตที่ว่านี้เหมือนกัน เกินกว่า 8 จุด)

2.3 ไมโครคอนโทรลเลอร์ PIC18F458

ไมโครคอนโทรลเลอร์ตระกูล PIC เป็นไมโครคอนโทรลเลอร์ที่มีสถาปัตยกรรมแบบฮาร์วาร์ด (Harward Achitecture) คือ มีการแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกันมีบัสสำหรับติดต่อแยกกัน ดังรูป 2.16

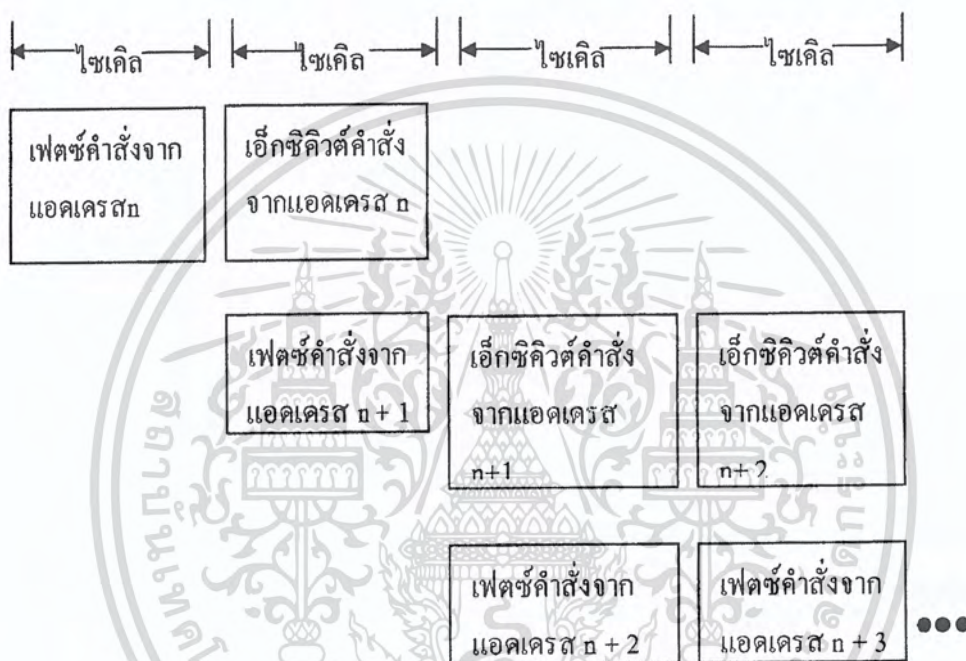


รูปที่ 2.16 การทำงานพื้นฐานของไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมแบบฮาร์วาร์ด

จะเห็นว่าซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยจำความโปรแกรมด้วยบัสแอดเดรส 13 บิต และบัสข้อมูลหน่วยความจำโปรแกรม 14 บิต ในขณะที่บัสสำหรับติดต่อกับหน่วยความจำข้อมูลและรีจิสเตอร์แบบภายในเป็นแบบ 8 บิต ทั้งบัสแอดเดรสและบัสข้อมูล

นอกจากสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งบนไมโครคอนโทรลเลอร์ PIC ยังใช้กระบวนการที่เรียกว่า ไปป์ไลน์ (pipeline) ทำให้สามารถอ่าน (fetch) คำสั่งถัดไปในขณะที่

คำสั่งกระทำ (execute) คำสั่งในปัจจุบัน ส่งผลให้ความเร็วในการทำงานของ ไมโครคอนโทรลเลอร์เพิ่มมากขึ้น นั่นจึงเป็นที่มาของความสามารถ ในการกระทำคำสั่ง 1 คำสั่ง ภายในสัญญาณนาฬิกา 1 ลูก (เฟตซ์: fetch เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำ โปรแกรมแล้วแปลงเป็นเลขฐาน 16 เพื่อให้ CPU เข้าใจ ส่วนกระบวนการเอ็กซีคิวท์ (Execute) เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ ตามคำสั่งนั้นๆกำหนด)



รูปที่ 2.17 ไตอะแกรมของกระบวนการ Pipeline ที่ใช้ในไมโครคอนโทรลเลอร์ PIC

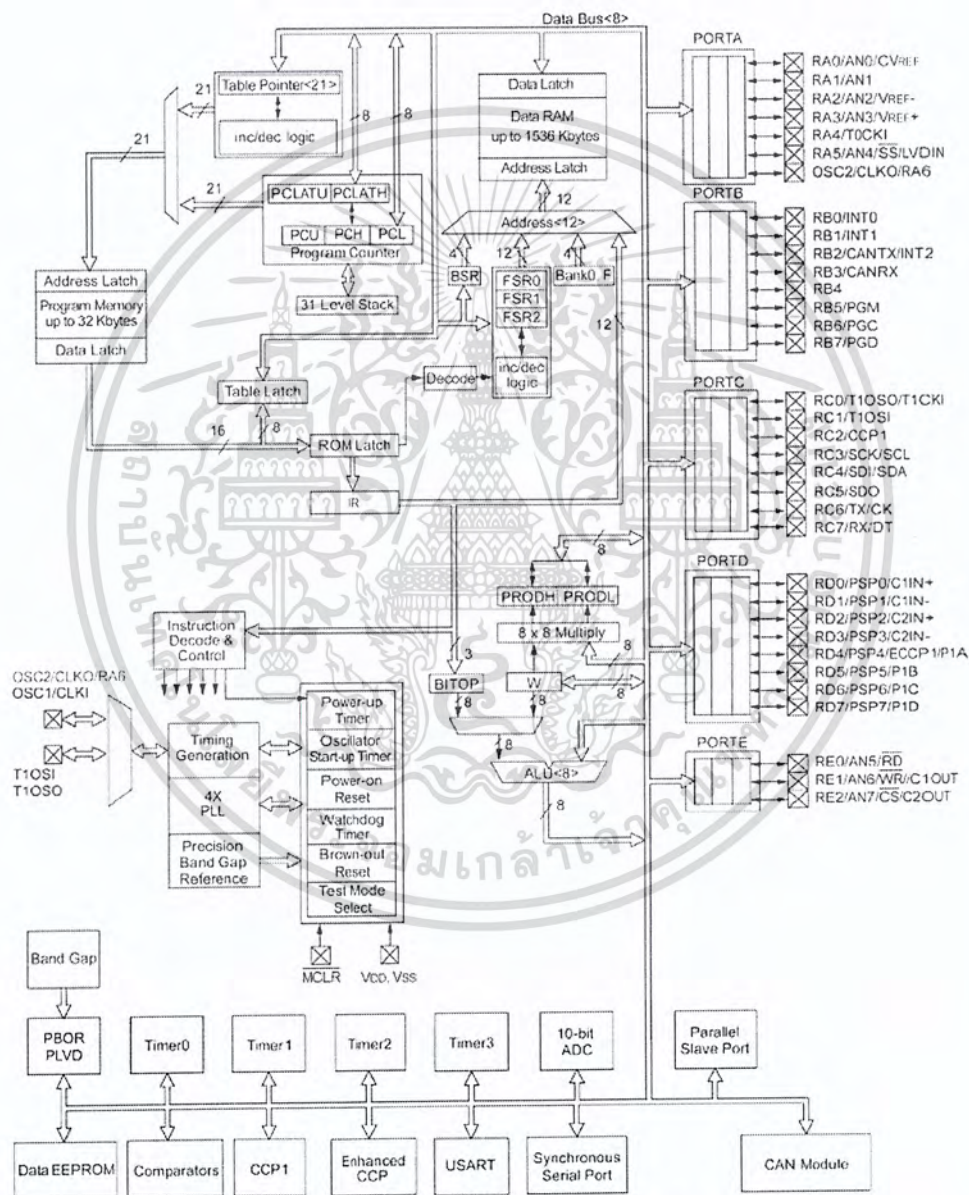
2.3.1 โครงสร้างของไมโครคอนโทรลเลอร์ PIC18F458

ส่วนประกอบหลักก็เหมือนไมโครคอนโทรลเลอร์ PIC16F877 ตามตาราง ที่ 2.2 แต่จะมีส่วนที่เพิ่มเติมเข้ามามากพอสมควร ได้แก่ หน่วยความจำของตัว ไมโครคอนโทรลเลอร์ที่มากขึ้น โดยมีขนาดหน่วยความจำมากถึง 32 Kbyte (Program Memory) และแรมที่เพิ่มมากขึ้น ดังแสดงในตารางที่ 2.2 และรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 โครงสร้างของไมโครคอนโทรลเลอร์

ชิพ	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล		CAN Module	จำนวน I/O (บิต)	OSC (MHz)	Timer	PLL
		RAM	EEPROM					
16F877	8K Word	368	256	ไม่มี	33	4-20	3	ไม่มี
18F458	32KB	1536	256	มี	34	40	4	มี



รูปที่ 2.18 โครงสร้างของไมโครคอนโทรลเลอร์ PIC18F458

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 คุณสมบัติทางเทคนิคของ PIC18F458

2.3.2.1 ซีพียูเป็นแบบ RISC (Reduced Instruction-Set Computer) มีคำสั่งใช้งาน เพียง 35 คำสั่ง

2.3.2.2 สามารถกระทำคำสั่ง โดยใช้สัญญาณเพียงหนึ่งลูก ยกเว้นคำสั่งกระโดด

2.3.2.3 ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 40 MHz(PLL)

2.3.2.4 หน่วยความจำโปรแกรม 32 KByte

2.3.2.5 หน่วยความจำข้อมูลแรมหรือรีจิสเตอร์ 1536 Byte

2.3.2.6 ขนาดหน่วยความจำข้อมูลอีอีพรอม 256 Byte

2.3.2.7 มีสแต็ก 31 ระดับ

2.3.2.8 มีวงจรเพาเวอร์ออนรีเซต (POR)

2.3.2.9 มีเพาเวอร์อัปไทมเมอร์ (PWTR) และออสซิลเลเตอร์สตาร์ทอัปไทมเมอร์ (OST)

2.3.2.10 มีวงจร วอตช์ด็อกไทมเมอร์ (WDT) มีวงจรออสซิลเลเตอร์ในตัว ทำให้มีความน่าเชื่อถือในการทำงานสูง

2.3.2.11 เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลสามารถเลือกระดับป้องกันได้

2.3.2.12 มีโหมดประหยัดพลังงาน

2.3.2.13 สามารถโปรแกรมโดยใช้แรงดัน +5 V (Low level) ได้

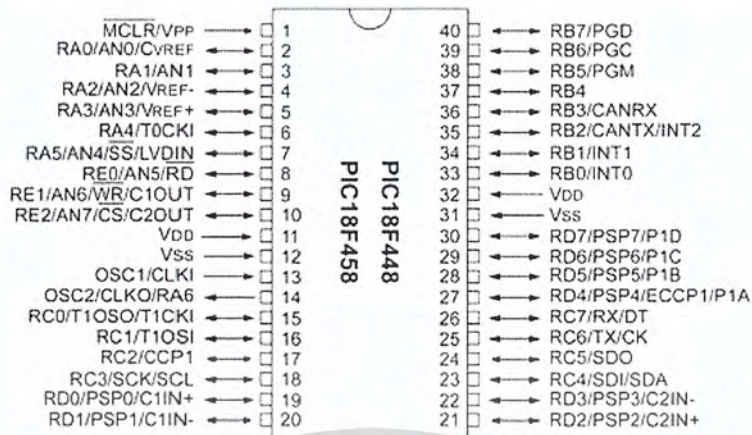
2.3.2.14 ตรวจสอบคำสั่งข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ IDC (in-circuit Debugger) ผ่านบอร์ดเพียง 2 ขา

2.3.2.15 ซีพียูสามารถอ่านและเขียนหน่วยความจำโปรแกรมได้

2.3.2.16 ไฟเลี้ยงที่ใช้งานตั้งแต่ +2 V ถึง +5.5 V

2.3.2.17 กระแสซิงก์และซอร์สของแต่ละพอร์ตที่ 25 mA

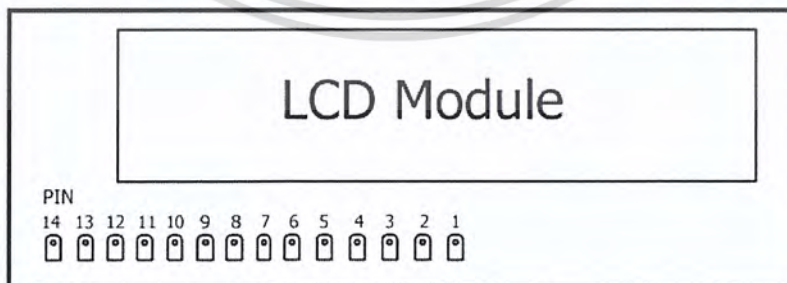
ขาต่อใช้งานของ PIC18F458 สามารถแสดงได้ดังรูปที่ 2.19



รูปที่ 2.19 ขาใช้งานของ PIC18F458

2.4 โมดูล LCD (Liquid Crystal Display)

เป็นอุปกรณ์กำเนิดแสงชนิดหนึ่งที่นิยมนำไปใช้เป็นตัวแสดงผลข้อความหรือตัวเลขมากกว่านำไปใช้เพื่อเป็นอุปกรณ์กำเนิดแสงสว่าง เนื่องจากมีความเข้มของแสงต่ำมาก และกินกำลังไฟฟ้าน้อยกว่าหลอดไส้ ดังนั้นจึงนิยมนำ LCD มาใช้ในเครื่องคิดเลขและนาฬิกาดิจิตอล LCD เป็นจอแสดงผลที่ได้รับความนิยมอย่างสูง ในปัจจุบัน LCD ถูกนำมาใช้งานแทนที่ 7-SEGMENT เป็นจำนวนมากเนื่องจาก LCD สามารถแสดงตัวอักษรและรายละเอียดได้มากกว่า 7-SEGMENT ทำให้มีการนำไปใช้งานในเครื่องมือต่างๆ มากขึ้น เพราะการสื่อสารระหว่างผู้ใช้กับเครื่องมือต่างๆ โดยผ่าน LCD นั้นมีความสะดวก ซึ่งในขั้นนี้จะขอกล่าวถึงเฉพาะ LCD ที่นำมาใช้ในโครงการซึ่งเป็นแบบ 16 ตัวอักษร 2 บรรทัด เท่านั้น จอแสดงผล LCD มีลักษณะดังรูปที่ 2.20



รูปที่ 2.20 จอแสดงผลแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

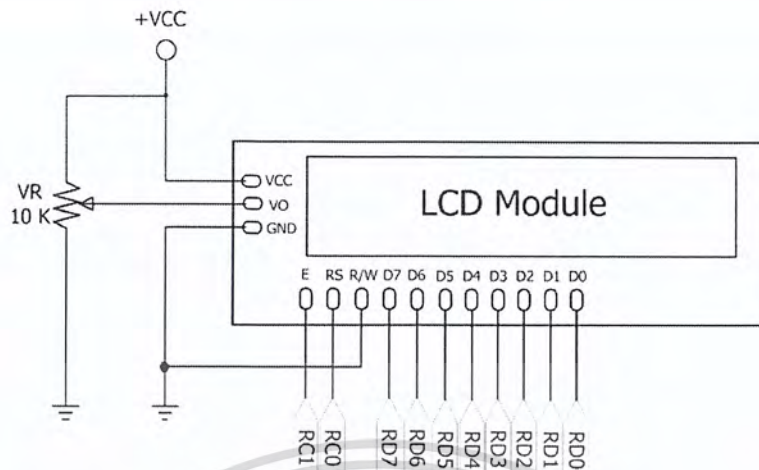
ตารางที่ 2.3 ชื่อและหน้าที่ของขาสัญญาณต่างๆของ LCD

ขาที่	ชื่อขา	หน้าที่ใช้งาน
1	GND	ต่อกับกราวด์ของวงจร
2	+V _{DD}	ต่อกับไฟเลี้ยง +5V
3	V _O	เป็นขาสำหรับป้อนแรงดันเพื่อปรับความสว่างของจอแสดงผลแอลซีดี
4	RS	เป็นขาสื่อเลือกการติดต่อกับรีจิสเตอร์คำสั่งหรือรีจิสเตอร์ข้อมูล 0 : จะติดต่อกับรีจิสเตอร์คำสั่ง 1 : จะติดต่อกับรีจิสเตอร์ข้อมูลเพื่อนำข้อมูลไปแสดงผล
5	R/W	เป็นขาสื่อเลือกการอ่านหรือเขียนข้อมูลกับ โมดูล LCD
6	E	เป็นขาสำหรับป้อนสัญญาณพัลส์ Enable ให้โมดูล LCD ทำงาน
7-14	D0-D7	เป็นขาข้อมูล 8 บิต โดยใช้ขา 7 คือ D0 ไปจนถึงขา 14 คือ D7
15-16	BKL1-BKL2	ขาต่อไฟเลี้ยง Blacklight ต่อเข้ากับไฟ +4.2V หรือ 120~180 mA DC

การติดต่อกับ โมดูล LCD 16 ตัวอักษร 2 บรรทัด มีอยู่ด้วยกัน 2 แบบคือ แบบ 8 บิต และแบบ 4 บิต โดยปกติจะมีการใช้แบบ 8 บิต มากกว่า แต่หากมีข้อจำกัดเรื่องจำนวนของพอร์ต ควรเลือกใช้การติดต่อแบบ 4 บิต ซึ่งจะมีขั้นตอนที่เพิ่มขึ้นเพียงเล็กน้อย แต่จะใช้สายสัญญาณเพียง 6 เส้น ในขณะที่แบบ 8 บิต จะใช้สายสัญญาณ 10 เส้น

2.4.1 การติดต่อแบบ 8 บิต

เป็นการเชื่อมต่อโมดูล LCD กับ CPU แบบ 8 บิต ขา D0-D7 ของโมดูล LCD เชื่อมต่อกับขา RD0-RD7 ขา RS ต่อกับ RC0 และ E ต่อกับ RC1 ส่วนขา R/W ให้ต่อลงกราวด์ เพื่อให้แอลซีดีทำงานในลักษณะเขียนข้อมูลอย่างเดียว ในขณะที่ขา V_O ต่อกับตัวต้านทานปรับค่าได้ 10 k Ω เพื่อปรับความสว่างของจอแสดงผล การต่อใช้งานแบบ 8 บิต Data สามารถแสดงได้ดังรูปที่ 2.21



รูปที่ 2.21 การต่อใช้งาน LCD แบบ 8 Bit Data

การเขียนโปรแกรมควบคุมให้จอแอลซีดีแสดงผลนั้น ในขั้นแรกจะต้องทำการกำหนดฟังก์ชันการทำงานต่างๆ ของแอลซีดีเสียก่อน หรือ เรียกว่าการ อินิเชียลแอลซีดี (Initial LCD) ซึ่งก็คือการเขียนข้อมูลคำสั่งไปยังรีจิสเตอร์คำสั่งภายใน โมดูล LCD เพื่อเตรียมความพร้อมให้แก่โมดูล LCD ซึ่งในการอินิเชียล LCD ก็คือ การกำหนดให้แอลซีดีมีการทำงานในรูปแบบต่างๆ เช่น กำหนดตำแหน่งของเคอร์เซอร์มาอยู่ที่จุดเริ่มต้นที่ตำแหน่งซ้ายมือสุด เปิดจอแสดงผล เปิด-ปิดเคอร์เซอร์ กำหนดรูปแบบการแสดงผลของตัวอักษร กำหนดจำนวนบรรทัด และการกำหนดโหมดในการติดต่อ โดยในการส่งข้อมูลไปยัง LCD จะแบ่งข้อมูลออกเป็น 2 ชนิด คือ ข้อมูลคำสั่ง (Command) และข้อมูลในการแสดงผล (Data) ซึ่งการส่งข้อมูลทั้งสองชนิดจะมีลำดับขั้นตอนเหมือนกันแต่จะต่างกันตรง การกำหนดข้อมูลที่บิต RS เพื่อแยกชนิดของข้อมูลซึ่งมีลำดับขั้นตอน ดังนี้

2.4.1.1 การเขียนข้อมูลคำสั่งไปยังจอแอลซีดีสำหรับการติดต่อแบบ 8 บิต

- 1) ทำให้ขา RS เป็น “0” เพื่อแจ้งให้โมดูล LCD ทราบว่า ข้อมูลที่ขา Data เป็นข้อมูลคำสั่ง
- 2) ส่งข้อมูลคำสั่งที่ต้องการไปยังขา Data ทั้ง 8 เส้น
- 3) ส่งพัลส์ Enable ไปยังขา E

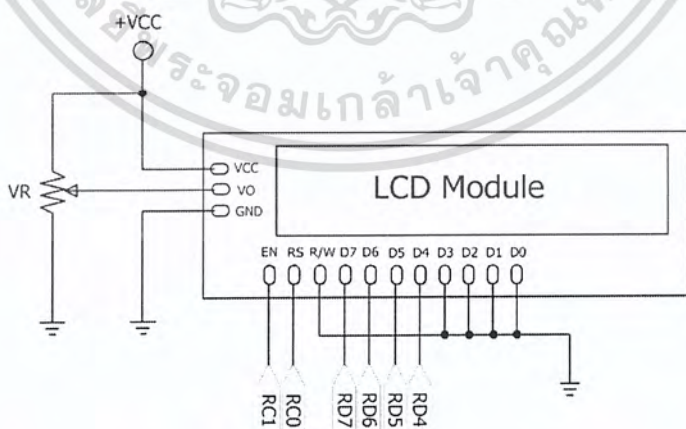
2.4.1.2 การเขียนข้อมูล (DATA) เพื่อแสดงผลข้อมูลสำหรับการติดต่อ

แบบ 8 บิต

- 1) ทำให้ขา RS เป็น “1” เพื่อแจ้งให้โมดูล LCD ทราบว่า ข้อมูลที่ขา DATA เป็นข้อมูลที่จะแสดงผล
- 2) เขียนข้อมูลที่ต้องการไปยังขา Data ทั้ง 8 เส้น
- 3) ส่งพัลส์ Enable ไปยังขา E

2.4.2 การติดต่อแบบ 4 บิต

เป็นการเชื่อมต่อ โมดูล LCD กับ CPU แบบ 8 บิต ขาของข้อมูลที่ใช้มีเพียง 4 เส้น คือ D4-D7 ซึ่งต่อเข้ากับ RD4-RD7 ของ CPU สำหรับขา D0-D3 ของโมดูล LCD ให้ต่อลงกราวด์ ส่วนขา RS ต่อเข้ากับ RC0 และ E ต่อเข้ากับ RC1 จุดที่แตกต่างจากการติดต่อแบบ 8 บิต ในการเขียนโปรแกรมคือ ต้องทำการส่งข้อมูล 2 ครั้ง คือส่ง 4 บิตบนของข้อมูลก่อน จากนั้นจึงส่งข้อมูล 4 บิตล่างตามไป สำหรับการอินิเชียนั้นมีสิ่งที่จะต้องทำก่อนเสมอ คือ ต้องส่งข้อมูล 03h (0011) ออกไปให้ LCD ที่ขา D7-D4 แล้วทำการส่งสัญญาณ Enable จำนวน 2 ครั้ง เพื่อจัดสถานะการทำงานของแอสซีดี จากนั้นส่งข้อมูลคำสั่ง 02h ออกไปที่ขา D7-D4 แล้ว Enable อีกเช่นกันเพื่อกำหนดให้แอสซีดีทำงานในโหมด 4 บิต เท่านั้นแอสซีดีก็จะพร้อมที่จะทำงานในโหมด 4 บิตแล้ว ส่วนการกำหนดค่าอื่นๆ สามารถทำต่อจากนี้ได้เลยแต่ต้องส่งข้อมูลในแบบ 4 บิต การต่อใช้งานแบบ 4 บิต Data สามารถแสดงได้ดังรูปที่ 2.22



รูปที่ 2.22 การต่อใช้งาน LCD แบบ 4 Bit Data

2.4.2.1 การเขียนข้อมูลคำสั่งไปยังจอแอลซีดีสำหรับการติดต่อแบบ 4 บิต

1. ทำให้ขา RS เป็น “0” เพื่อแจ้งให้โมดูล LCD ทราบว่า ข้อมูลที่ขา DATA เป็นข้อมูลคำสั่ง

Data ทั้ง 4 เส้น

2. ส่งข้อมูลคำสั่ง 4 บิตบน (บิต 7 -> บิต 4) ที่ต้องการไปยังขา

3. ส่งพัลส์ Enable ไปยังขา E

4. ส่งข้อมูลคำสั่ง 4 บิตล่าง (บิต 3 -> บิต 0) ที่ต้องการไปยังขา

Data ทั้ง 4 เส้น

5. ส่งพัลส์ Enable ไปยังขา E

2.4.2.2 การเขียนข้อมูล (DATA) เพื่อแสดงผลข้อมูลสำหรับการติดต่อแบบ 4 บิต จะต้องทำดังนี้

1. ทำให้ขา RS เป็น “1” เพื่อแจ้งให้โมดูล LCD ทราบว่า ข้อมูลที่ขา DATA เป็นข้อมูลในการแสดงผล

2. ส่งข้อมูล (Data) 4 บิตบน (บิต 7 -> บิต 4) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น

3. ส่งพัลส์ Enable ไปยังขา E

4. ส่งข้อมูล (Data) 4 บิตล่าง (บิต 3 -> บิต 0) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น

5. ส่งพัลส์ Enable ไปยังขา E

2.5 การเชื่อมต่อผ่านพอร์ตอนุกรม RS-232

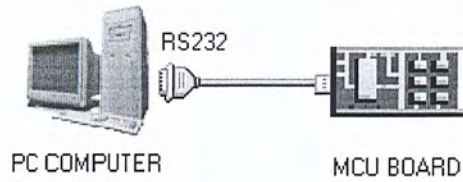
การสื่อสารแบบอนุกรม นับว่ามีความสำคัญต่อการใช้งาน ไมโครคอนโทรลเลอร์มาก เพราะการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์นั้นใช้สายสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้น คือ

1) สายส่งสัญญาณ TX

2) สายรับสัญญาณ RX

3) และสาย GND

การเชื่อมต่อระหว่างคอมพิวเตอร์กับบอร์ดของไมโครคอนโทรลเลอร์สามารถแสดง
ได้ดังรูปที่ 2.23



รูปที่ 2.23 การเชื่อมต่อระหว่างคอมพิวเตอร์กับบอร์ดของไมโครคอนโทรลเลอร์

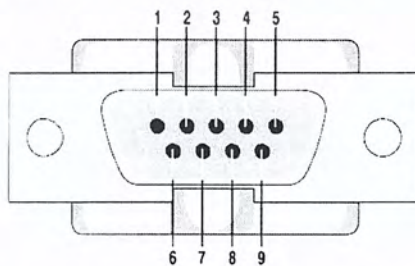
โดยปกติพอร์ตอนุกรม RS-232 จะสามารถต่อสายได้ยาว 50 ฟุตหรือ 15 เมตร
โดยประมาณขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

รูปที่ 2.24 พอร์ตอนุกรมของ PC DB9 ตัวผู้และพอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย

พอร์ตอนุกรมของ PC จะเป็นคอนเน็คเตอร์แบบ DB9 ตัวผู้ (Male) และ พอร์ต
อนุกรม ของอุปกรณ์ภายนอก จะเป็นคอนเน็คเตอร์แบบ DB9 ตัวเมีย (Female)

2.5.1 การจัดขาของ Connector อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ

ขาของ DB 9 มีทั้งหมด 9 ขา โดยเมื่อมองจากด้านหลังซึ่งเป็นด้านที่ใช้
เชื่อมต่อกับสายสัญญาณจะมีการเรียงลำดับของขา ดังรูปที่ 2.25



รูปที่ 2.25 การเรียงขาของ DB9 เมื่อมองจากด้านหลัง

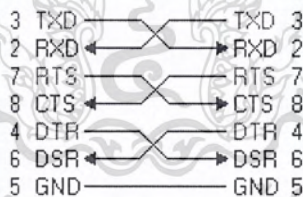
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 หน้าที่ใช้งานของขาต่างๆ ของ DB9

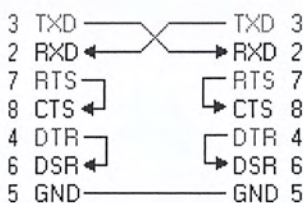
Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

2.5.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9

การเชื่อมต่อผ่าน DB9 สามารถทำการเชื่อมต่อกับสายสัญญาณได้ 2 รูปแบบดังแสดงในรูปที่ 2.26 - 2.27



รูปที่ 2.26 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem



รูปที่ 2.27 การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น

การทำงานของขาสัญญาณ DB9

- **TXD** เป็นขาที่ใช้ส่งข้อมูล
- **RXD** เป็นขาที่ใช้รับข้อมูล
- **DTR** แสดงสถานะพอร์ตว่าเปิดใช้งาน
- **DSR** ตรวจสอบว่าพอร์ตที่ติดต่อด้วยเปิดอยู่หรือไม่

เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าการติดต่อด้วย ในขณะที่เดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

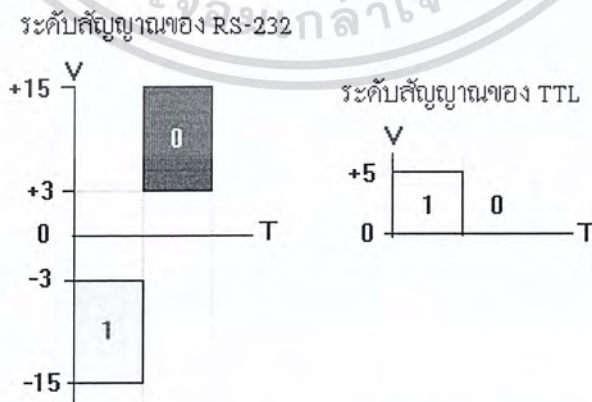
- **RTS** แสดงสถานะพอร์ตว่าต้องการส่งข้อมูล
- **CTS** ตรวจสอบว่าพอร์ตที่ติดต่อด้วย ต้องการส่งข้อมูลหรือไม่

เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

- **GND** ขา Ground

2.5.3 ระดับสัญญาณของ RS232

สัญญาณรบกวนที่เกิดขึ้นในสายนำสัญญาณมักจะมีแรงดันเป็นบวกเมื่อเทียบกับกราวด์ เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดัน ของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3 V ถึง -15 V ส่วนแรงดัน ของโลจิก "0" อยู่ในช่วง +3 V ถึง +15 V และเหตุที่ระดับสัญญาณ ของ RS232 อยู่ในช่วง +15 V ถึง -15 V ก็เพื่อให้ต่อสายสัญญาณ ไปได้ไกลขึ้น ดังนั้นจึงจำเป็นจะต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL ดังแสดงในรูปที่ 2.28



รูปที่ 2.28 ระดับสัญญาณของ RS232 และระดับสัญญาณของ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 อัตราการส่งข้อมูล (Baud rate)

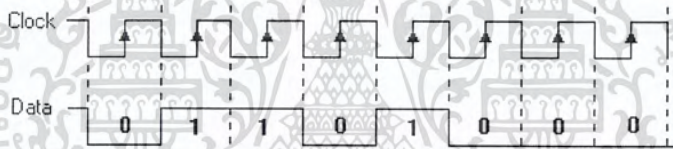
คือความเร็วของการรับ-ส่งข้อมูลเป็นจำนวนบิตต่อวินาทีเช่น 300 1,200 2,400 4,800 9,600 14,400 19,200 38,400 56,000 เป็นต้น การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับ ชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

2.5.5 รูปแบบการสื่อสารแบบอนุกรม

รูปแบบการสื่อสารมีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous)

2.5.5.1 การสื่อสารแบบซิงโครนัส (Synchronous)

การรับส่งข้อมูลจะมีสัญญาณนาฬิกาซึ่งเป็นตัวกำหนดจังหวะเวลาการส่งข้อมูลรวมอยู่ด้วยอีกเส้นหนึ่งใช้คู่กับสัญญาณข้อมูล ดังแสดงในรูปที่ 2.29



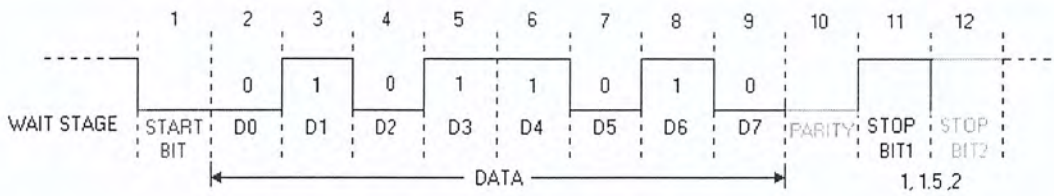
รูปที่ 2.29 สัญญาณนาฬิกาและสัญญาณข้อมูล

2.5.5.2 การสื่อสารแบบอะซิงโครนัส (Asynchronous)

การรับส่งข้อมูลโดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วย แต่จะใช้ให้ตัวส่งและตัวรับมีอัตราส่งข้อมูลที่เท่ากัน รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

- 1) บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
- 2) บิตข้อมูล (Data bit) มีขนาด 5 6 7 หรือ 8 บิต
- 3) บิตตรวจสอบพาริตี (Parity bit) มีขนาด 1 บิตหรือไม่มี
- 4) บิตหยุด (Stop bit) มีขนาด 1 1.5 2 บิต

การสื่อสารแบบอะซิงโครนัสมีลักษณะสัญญาณดังรูปที่ 2.30



รูปที่ 2.30 สัญญาณแบบอะซิงโครนัส (Asynchronous)

เมื่อไม่มีการส่งข้อมูล data จะมีสถานะเป็น โลจิก "1" หรือสถานะหยุดรอ (Waiting stage) เมื่อเริ่มต้นส่งข้อมูลจะให้ค่า data เป็น โลจิก "0" เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit) จากนั้นก็จะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำไปก่อน (LSB) แล้วตามด้วยพาริตีบิต (จะมีหรือไม่ก็ได้ ขึ้นอยู่กับการติดตั้งค่าของทั้งสองฝ่าย) สุดท้ายตามด้วยโลจิก "1" อย่างน้อย 1 บิต (มีขนาด 1, 1.5, หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

การเริ่มใช้พอร์ตอนุกรมต้องตั้งค่าต่างๆ สำหรับการสื่อสาร ดังต่อไปนี้

- 1) ความเร็วของการส่ง
- 2) ความยาวของรหัส 1 อักขระ
- 3) บิตตรวจสอบ
- 4) จำนวน stop บิต

2.5.6 ลักษณะการรับ-ส่งข้อมูลแบบอนุกรม

ลักษณะการรับ-ส่งข้อมูลแบบอนุกรมแบ่งออกเป็น 3 แบบ ได้แก่

2.5.7.1 แบบซิมเพล็กซ์ (Simplex) เป็นการส่งหรือรับข้อมูลแบบทิศทางเดียว เท่านั้น

2.5.7.2 แบบฮาล์ฟดูเพล็กซ์ (Half Duplex) เป็นการส่งและรับข้อมูลแบบสลับกัน คือ เมื่อด้านหนึ่งส่งอีกด้านหนึ่งเป็นฝ่ายรับสลับกันไม่สามารถรับ-ส่งในเวลาเดียวกันได้

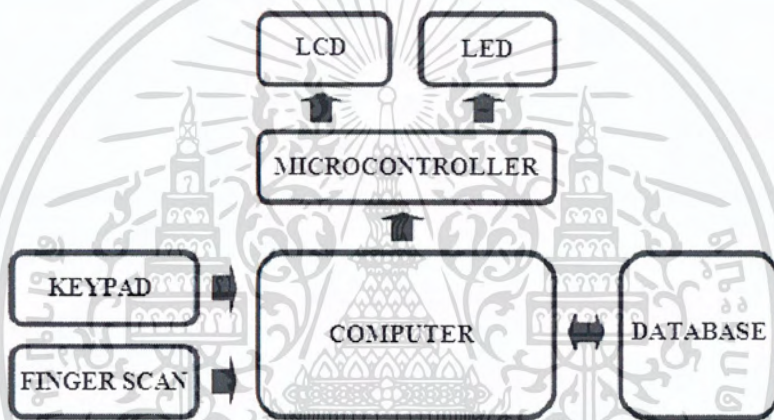
2.5.7.3 แบบฟูลดูเพล็กซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

บทที่ 3

การออกแบบและการจัดทำปฏิญานិพนธ์

3.1 การออกแบบ

บล็อกไดอะแกรมรวมของโครงการสามารถแสดงได้ดังรูปที่ 3.1



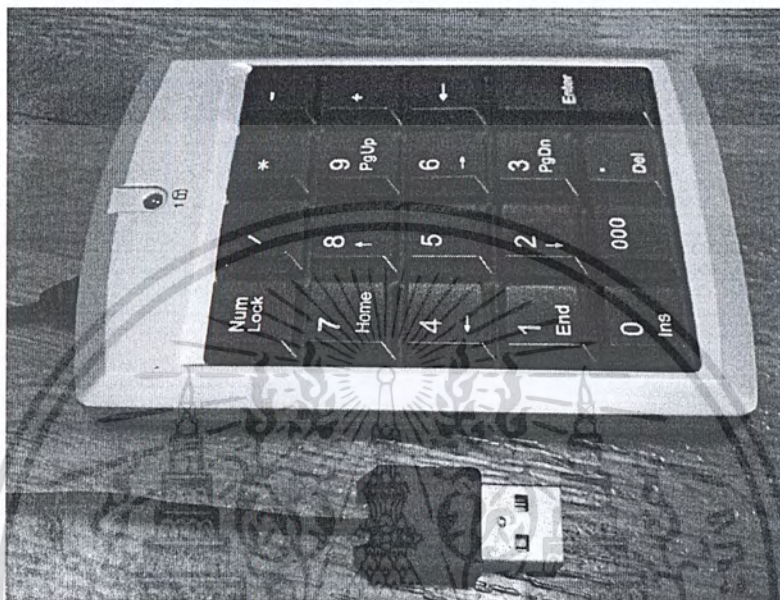
รูปที่ 3.1 Block diagram รวมของโครงการ

จากบล็อกไดอะแกรมจะเห็นได้ว่าระบบแบ่งออกเป็น 4 ส่วน ได้แก่ ส่วนรับข้อมูล ส่วนประมวลผล ส่วนฐานข้อมูลและส่วนแสดงผล

3.1.1 ส่วนรับข้อมูล

ส่วนรับข้อมูลประกอบด้วย 2 อุปกรณ์ ได้แก่ คีย์แพดและเครื่องสแกนลายนิ้วมือดังรูปที่ 3.2 – 3.3

3.1.1.1 Keypad เชื่อมต่อกับคอมพิวเตอร์ผ่านทาง USB พอร์ต ทำหน้าที่รับข้อมูลรหัสพนักงานส่งต่อให้ส่วนประมวลผลตรวจสอบว่ามีผู้ใช้งานนี้หรือไม่ โดยส่วนประมวลผลรับข้อมูลเข้ามาเป็น ASCII ดังรูปที่ 3.2



รูปที่ 3.2 Keypad

3.1.1.2 เครื่องสแกนลายนิ้วมือ (U.are.U 4000B) เชื่อมต่อกับคอมพิวเตอร์ผ่านทาง USB พอร์ต ทำหน้าที่ถ่ายภาพลายนิ้วมือของผู้ใช้งานแล้วส่งต่อให้ส่วนประมวลผลเป็นข้อมูล Long binary รูปตัวเครื่องสแกนลายนิ้วมือแสดงดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 เครื่องสแกนลายนิ้วมือ

3.1.2 ส่วนประมวลผล

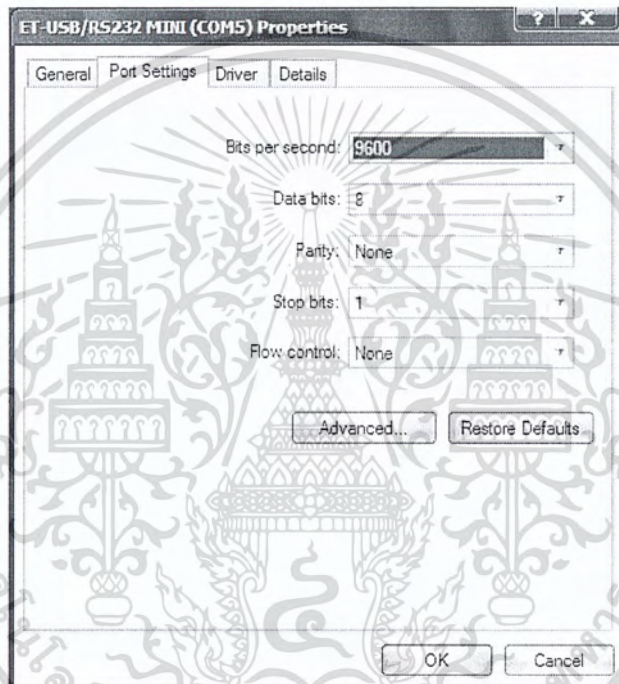
ทำหน้าที่ควบคุมและตรวจสอบ ออกแบบโดยใช้โปรแกรม Microsoft visual C# เขียนและทำงานบนคอมพิวเตอร์ ทำงานร่วมกับทั้ง 3 ส่วนและยังแบ่งการทำงานออกเป็น 2 ประเภท คือ

3.1.2.1 การใช้งานโดยผู้ดูแลระบบ ผู้ดูแลระบบสามารถเพิ่ม ลบหรือแก้ไขข้อมูลของผู้ใช้งานได้ โดยรับข้อมูลพนักงานผ่านทางคีย์บอร์ดคอมพิวเตอร์และเครื่องสแกนลายนิ้วมือ ข้อมูลพนักงานที่รับเข้ามาเมื่อทำการบันทึกจะถูกนำไปเก็บยังฐานข้อมูลซึ่งใช้โปรแกรม Microsoft office access แต่มีข้อมูลอยู่ 2 อย่างที่จัดเก็บในไฟล์เตอร์โดยใช้คลาส Path ในการจัดการรูปถ่ายและรูปลายนิ้วมือของผู้ใช้งานเป็นภาพสกุล .JPEG ส่วนข้อมูลลายนิ้วมือ (template) จัดเก็บโดยใช้ชนิดข้อมูลเป็น OLE Object

3.1.2.2 การใช้งานโดยผู้ใช้งาน ผู้ใช้งานจะทำการใช้งานผ่านทางตัวเครื่องซึ่งทำได้เฉพาะการตรวจสอบ ไม่สามารถแก้ไขหรือเพิ่มเติมข้อมูลของตัวเองได้ ส่วนประมวลผลจะทำหน้าที่รอรับรหัสพนักงานและข้อมูลลายนิ้วมือมาทำการตรวจสอบแล้วส่งคำสั่งให้ส่วนแสดงผลแสดงผลการใช้งานให้ผู้ใช้งานทราบดังตารางที่ 3.1

ข้อมูลที่ใช้ในการตรวจสอบเป็นข้อมูลที่อยู่ในฐานข้อมูล โดยที่ การตรวจสอบรหัสพนักงานจะใช้เมธอด checkIDInDB ของคลาส DBFingerClass และการ ตรวจสอบลายนิ้วมือจะใช้เมธอด Verify ของคลาส AxGrFingerXLib

การส่งคำสั่งให้ส่วนแสดงผล คำสั่งที่ใช้ในการส่งเป็นข้อมูล ชนิด String เช่น A B C D และกำหนดรายละเอียดในการรับ – ส่ง คือ ความเร็ว 9,600 บิตต่อวินาที บิตข้อมูล 8 บิต บิตตรวจสอบ 0 บิต สตอปบิต 1 บิต ดังรูปที่ 3.4



รูปที่ 3.4 รายละเอียดในการเชื่อมต่อพอร์ทอนุกรม

3.1.3 ส่วนฐานข้อมูล

การออกแบบฐานข้อมูลสำหรับโครงการนี้ใช้โปรแกรม Microsoft office access โดยแบ่งข้อมูลออกเป็น 3 ตารางโดยที่มุมมองแผ่นข้อมูลใช้เฉพาะตรวจสอบข้อมูลและ มุมมองออกแบบใช้ในการกำหนดค่าต่างๆของตาราง เช่น ชื่อของฟิลด์ ชนิดของข้อมูล

3.1.3.1 ฐานข้อมูลเก็บค่าของฟังก์เกอร์โค้ด

ชื่อตาราง enroll เป็นตารางที่ประกอบด้วย 2 ฟิวด์ คือ ID เก็บข้อมูลชนิด Text และ template เก็บข้อมูลชนิด OLE Object ในมุมมองแผ่นข้อมูลสามารถแสดงได้ดังรูปที่ 3.5 และในมุมมองออกแบบสามารถแสดงได้ดังรูปที่ 3.6

ID	template
100101	ข้อมูล Long binary
100102	ข้อมูล Long binary
100103	ข้อมูล Long binary
100104	ข้อมูล Long binary

รูปที่ 3.5 ตาราง enroll ในมุมมองแผ่นข้อมูล

รูปที่ 3.6 ตาราง enroll ในมุมมองออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.2 ฐานข้อมูลเก็บค่าของเวลาใช้งาน

ชื่อตาราง listOfTimeScan เป็นตารางที่ประกอบด้วย 2 ฟิลด์ คือ ID เก็บข้อมูลชนิด Text และ dateTimeScan เก็บข้อมูลชนิด Text ในมุมมองแผ่นข้อมูลสามารถแสดงได้ดังรูปที่ 3.7 และในมุมมองออกแบบสามารถแสดงได้ดังรูปที่ 3.8

ID	dateTimeScan
100101	16/1/2554 21:34:02
100101	16/1/2554 21:34:28
100101	16/1/2554 21:34:33
100101	16/1/2554 21:34:37

รูปที่ 3.7 ตาราง listOfTimeScan ในมุมมองแผ่นข้อมูล

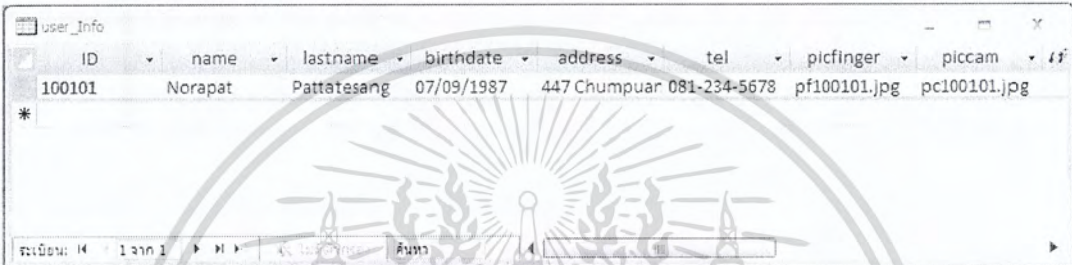
ID	dateTimeScan

รูปที่ 3.8 ตาราง listOfTimeScan ในมุมมองออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

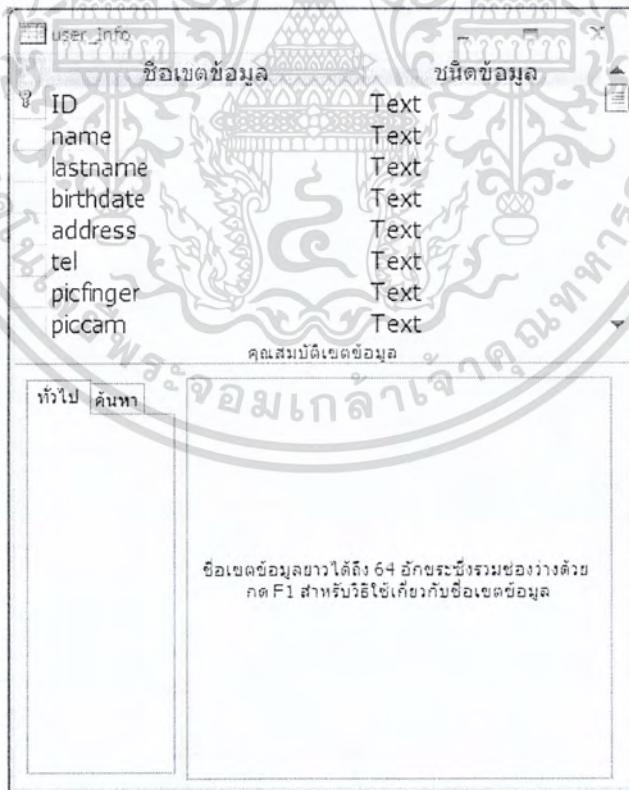
3.1.3.4 ฐานข้อมูลเก็บรายละเอียดข้อมูลพนักงาน

ชื่อตาราง user_info เป็นตารางที่ประกอบด้วย 8 필ด์ คือ 필ด์ ID เก็บข้อมูลชนิด Text 필ด์ name เก็บข้อมูลชนิด Text 필ด์ lastname เก็บข้อมูลชนิด Text 필ด์ Birthdate เก็บข้อมูลชนิด Text 필ด์ address เก็บข้อมูลชนิด Text 필ด์ tel เก็บข้อมูลชนิด Text 필ด์ picfinger เก็บข้อมูลชนิด Text และฟิลด์ piccam เก็บข้อมูลชนิด Text ในมุมมองแผ่นข้อมูลสามารถแสดงได้ดังรูปที่ 3.9 และในมุมมองออกแบบสามารถแสดงได้ดังรูปที่ 3.10



ID	name	lastname	birthdate	address	tel	picfinger	piccam
100101	Norapat	Pattatesang	07/09/1987	447 Chumpuar	081-234-5678	pf100101.jpg	pc100101.jpg

รูปที่ 3.9 ตาราง user_info ในมุมมองแผ่นข้อมูล



ชื่อเขตข้อมูล	ชนิดข้อมูล
ID	Text
name	Text
lastname	Text
birthdate	Text
address	Text
tel	Text
picfinger	Text
piccam	Text

คุณสมบัติเขตข้อมูล

ทั่วไป ค้นหา

ชื่อเขตข้อมูลยาวได้ถึง 64 อักขระซึ่งรวมช่องว่างด้วย
กด F1 สำหรับวิธีใช้เกี่ยวกับชื่อเขตข้อมูล

รูปที่ 3.10 ตาราง user_info ในมุมมองออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

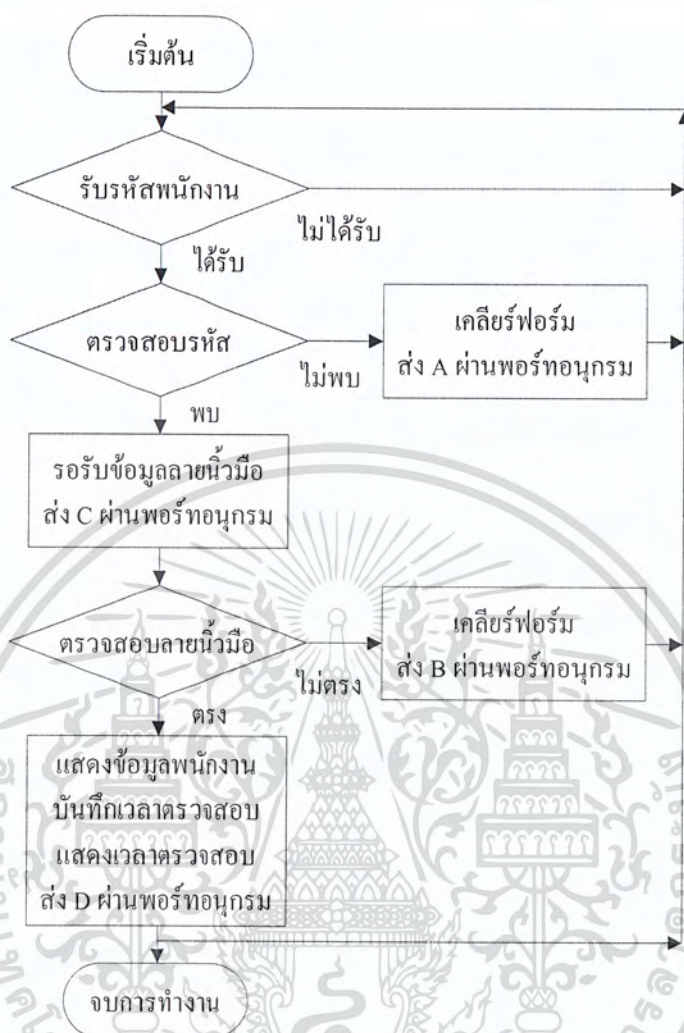
3.1.4 ส่วนแสดงผล

ส่วนแสดงผลประกอบด้วยไมโครคอนโทรลเลอร์ จอแสดงผลแอลซีดี และหลอดแอลอีดี ทำหน้าที่แจ้งสถานการณ์ทำงานให้ผู้ใช้งานทราบ โดยรับคำสั่งจากส่วนประมวลผลแล้วแสดงสถานะตามเงื่อนไขของโปรแกรมตามหัวข้อ 3.4

3.2 การออกแบบโปรแกรมในส่วนของผู้ใช้งาน

ระบบจะเริ่มกระบวนการทำงานจากขั้นตอนของการรอรับรหัสประจำตัวจากคีย์แพด ต่อจากนั้นโปรแกรมจะทำการตรวจสอบรหัส โดยหากไม่พบรหัสประจำตัวที่รับเข้ามาในฐานข้อมูลระบบจะส่งคำสั่งให้ส่วนแสดงผลทำการแสดงข้อความ ID Invalid! แจ้งผู้ใช้งานทราบ แต่หากพบว่ารหัสประจำตัวที่รับเข้ามาอยู่ในฐานข้อมูลระบบจะทำการดึงข้อมูลลายนิ้วมือของผู้ใช้งานผู้นั้นขึ้นมาซึ่งเป็นข้อมูล Long Binary ดังรูปที่ 3.4 เพื่อการเปรียบเทียบกับลายนิ้วมือของผู้ใช้งานที่รับเข้ามาจากเครื่องสแกนลายนิ้วมือ (Long Binary) และทำการส่งคำสั่งให้ส่วนแสดงผลทำการแสดงข้อความ Place Finger! แจ้งผู้ใช้งานเพื่อให้ผู้ใช้งานทำการวางนิ้วมือลงบนตำแหน่งสแกนลายนิ้วมือ เมื่อระบบได้รับข้อมูลลายนิ้วมือจากเครื่องสแกน ระบบจะทำการตรวจสอบลายนิ้วมือที่รับเข้ามาว่าเป็นลายนิ้วมือเดียวกันกับลายนิ้วมือที่ระบบได้ดึงขึ้นมาจากฐานข้อมูลเพื่อรอการตรวจสอบก่อนหน้าหรือไม่ หากพบว่าเป็นคนละลายนิ้วมือกันระบบจะส่งคำสั่งให้ส่วนแสดงผลทำการแสดงข้อความ Finger Invalid! ให้ผู้ใช้งานทราบแล้วกลับไปทำงานที่จุดเริ่มต้น คือ การรอรับรหัสพนักงาน แต่หากตรวจสอบแล้วพบว่าเป็นลายนิ้วมือเดียวกัน ระบบจะส่งคำสั่งให้ส่วนแสดงผลทำการแสดงข้อความ Finish! ให้ผู้ใช้งานทราบและทำการบันทึกเวลา ณ ขณะนั้นเก็บลงในฐานข้อมูลแล้วกลับไปทำงานที่จุดเริ่มต้นอีกครั้ง

ในส่วนนี้ใช้ภาษา C# ในการเขียนโปรแกรมทำงานอยู่บนคอมพิวเตอร์ อยู่ในคลาส Form Main โฟลวชาร์ทการทำงานของระบบในส่วนของผู้ใช้งานสามารถแสดงได้ดังรูปที่ 3.11

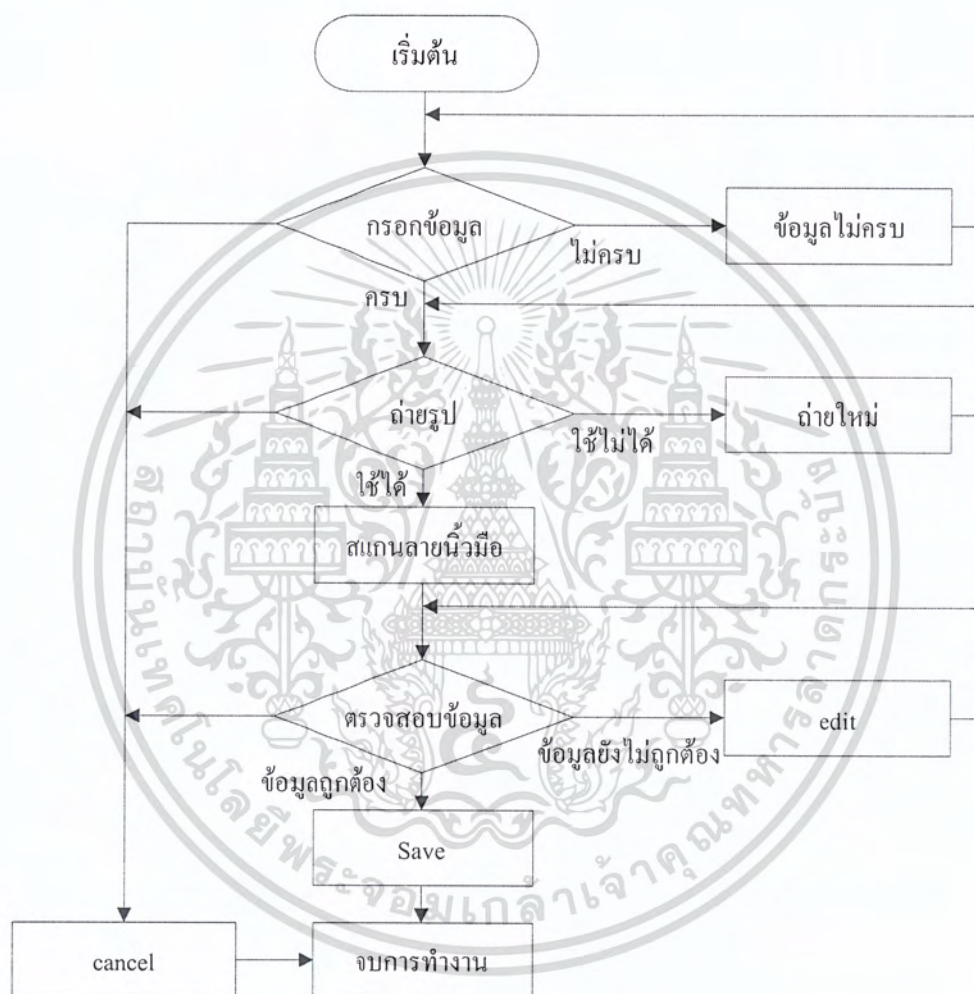


รูปที่ 3.11 โฟลวชาร์ตการทำงานของระบบในส่วนผู้ใช้งาน

3.3 การออกแบบโปรแกรมส่วนของผู้ดูแลระบบ

ระบบจะเริ่มกระบวนการทำงานจากขั้นตอนการกรอกข้อมูล เมื่อกรอกข้อมูลครบทุกช่องแล้วจึงทำการถ่ายภาพผู้ใช้งาน โดยที่ถ้ากรอกข้อมูลไม่ครบถ้วนระบบจะแสดงกล่องข้อความเตือนให้กรอกข้อมูลให้ครบ เมื่อทำการถ่ายภาพแล้วให้ทำการคลิกที่ปุ่ม scan ระบบจะทำการรอรหัสลายนิ้วมือจากเครื่องสแกนลายนิ้วมือ เมื่อระบบได้รับลายนิ้วมือแล้วจะนำข้อมูลที่ได้ทั้งหมดมาแสดงในหน้าต่างของฟอร์มหลักของโปรแกรม ในขั้นตอนนี้ให้ผู้ดูแลระบบตรวจสอบ

ความถูกต้องของข้อมูลผู้ใช้งาน ถ้าข้อมูลผู้ใช้งานถูกต้องแล้วจึงทำการบันทึกลงฐานข้อมูล โดยคลิกที่ปุ่ม save แต่หากข้อมูลผู้ใช้งานยังไม่ถูกต้อง เช่น ชื่อผู้ใช้งานผิด ที่อยู่ผิดหรือวันเดือนปีเกิดผิด สามารถคลิกที่ edit เพื่อแก้ไขได้ โฟลวชาร์ทการทำงานในส่วนของผู้ดูแลระบบสามารถแสดงได้ดังรูปที่ 3.12



รูปที่ 3.12 โฟลวชาร์ทการทำงานในส่วนของผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบโปรแกรมของส่วนแสดงผล

การทำงานของส่วนแสดงผลจะแบ่งออกเป็น 2 สถานะ คือ สถานะปกติและสถานะตรวจสอบ


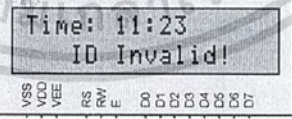
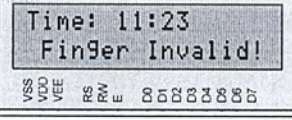
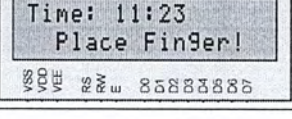
3.4.1 การทำงานสถานะปกติ

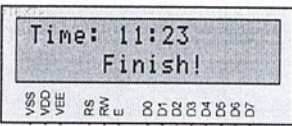
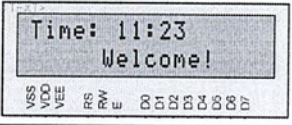

เป็นสถานะเมื่อไม่มีผู้มาใช้งาน หลอด LED จะกระพริบต่อเนื่องและจอแสดงผล LCD จะแสดงเวลาปัจจุบันและข้อความ Welcome!

3.4.2 การทำงานสถานะตรวจสอบ

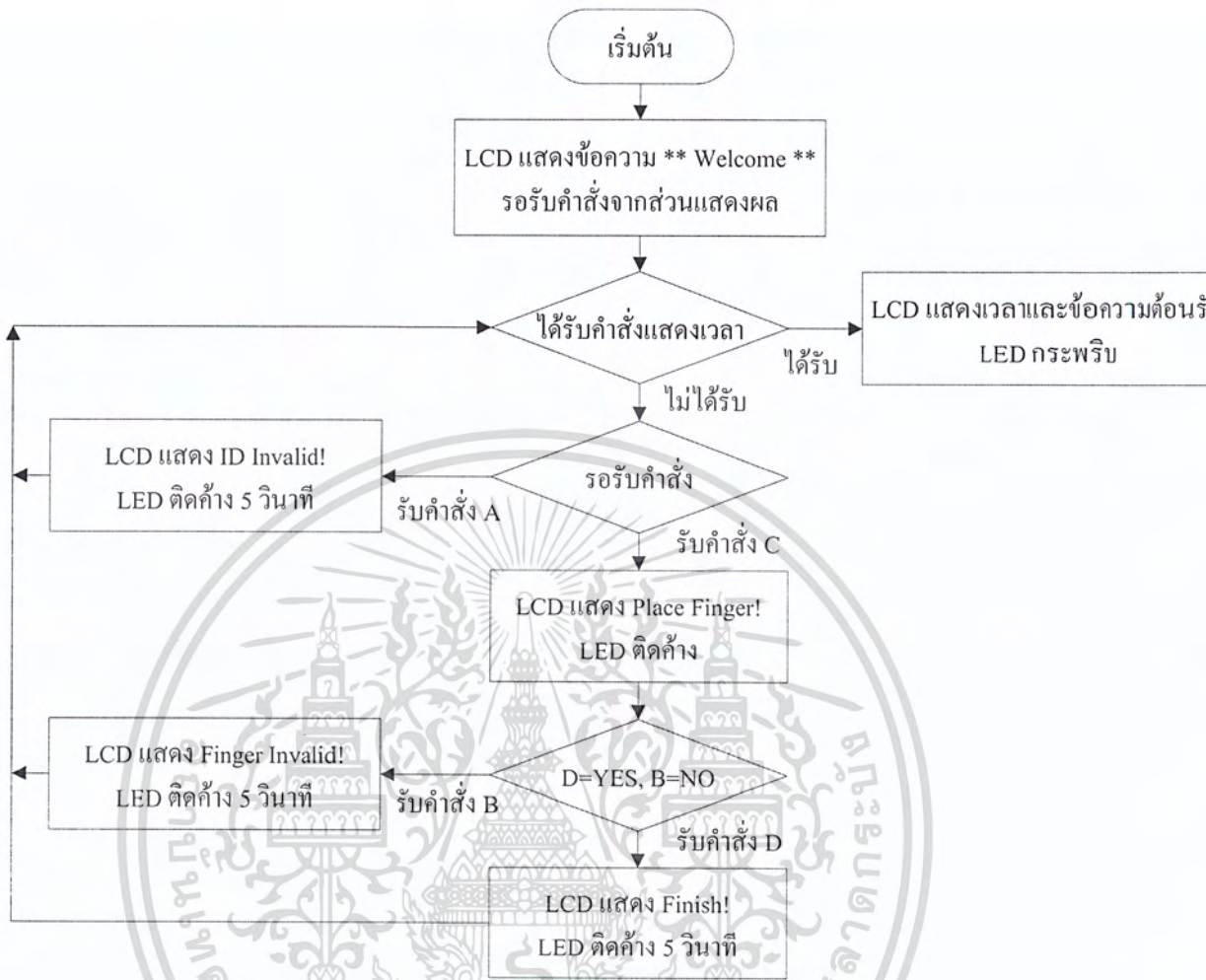
เป็นสถานะเมื่อมีผู้มาใช้งาน หลอด LED จะค้างและจอแสดงผล LCD จะแสดงเวลาปัจจุบันและข้อความแจ้งให้ผู้ใช้ทราบตามคำสั่งที่ได้รับจากส่วนประมวลผล

ตารางที่ 3.1 เงื่อนไขการทำงานของส่วนแสดงผล

ผลการตรวจสอบ	ข้อมูลที่ได้รับ	จอ LCD	หลอด LED
สถานะเริ่มต้น	-		กระพริบ
รหัสประจำตัวผิด	A		ติดค้าง 5 วินาที
ลายนิ้วมือผิด	B		ค้าง 5 วินาที
รหัสประจำตัวถูก	C		ค้าง

ลายนิ้วมือถูก	D		ค้าง 5 วินาที
สถานะปกติ	E		กระพริบ
รับค่าแสดงเวลา	T11:40/		กระพริบ

จากตารางที่ 3.1 การทำงานของส่วนแสดงผลจะเริ่มทำงานจากการรอรับคำสั่งจากส่วนประมวลผลที่ส่งผ่านมาทางพอร์ตอนุกรม ในกรณีที่เปิดเครื่องใหม่ส่วนแสดงผลจะแสดงข้อความ ** Welcome ** แล้วเมื่อมีการเชื่อมต่อกับส่วนประมวลผลแล้วจะแสดงผลเป็นสถานะปกติหลังจากนั้นส่วนแสดงผลจะรอรับคำสั่งจากส่วนประมวลผลอีกครั้งแล้วจะทำงานตามเงื่อนไขของคำสั่งที่ได้รับ โดยที่คำสั่งที่ได้รับเป็น ASCII ตามตารางที่ 3.1 โฟลวชาร์ตการทำงานของส่วนแสดงผลสามารถแสดงได้ดังรูปที่ 3.13



รูปที่ 3.13 โฟลวชาร์ทการทำงานของส่วนแสดงผล

3.4.3 ส่วนแสดงผลได้รับคำสั่ง A

เมื่อส่วนแสดงผลได้รับคำสั่ง A จากส่วนประมวลผลจะทำการแสดงข้อความ ID Invalid! ที่จอแสดงผล LCD และหลอด LED จะติดค้างเป็นเวลา 5 วินาทีแล้วจะได้รับคำสั่ง E จากส่วนประมวลผลจึงจะกลับเข้าสู่สถานะปกติ

3.4.4 ส่วนแสดงผลได้รับคำสั่ง B

เมื่อส่วนแสดงผลได้รับคำสั่ง B จากส่วนประมวลผลจะทำการแสดงข้อความ Finger Invalid! ที่จอแสดงผล LCD และหลอด LED จะติดค้างเป็นเวลา 5 วินาทีแล้วจะ
ได้รับคำสั่ง E จากส่วนประมวลผลจึงจะกลับเข้าสู่สถานะปกติ

3.4.5 ส่วนแสดงผลได้รับคำสั่ง C

เมื่อส่วนแสดงผลได้รับคำสั่ง C จากส่วนประมวลผลจะทำการแสดงข้อความ Place Finger! ที่จอแสดงผล LCD และหลอด LED จะติดค้าง

3.4.6 ส่วนแสดงผลได้รับคำสั่ง D

เมื่อส่วนแสดงผลได้รับคำสั่ง D จากส่วนประมวลผลจะทำการแสดงข้อความ Finish! ที่จอแสดงผล LCD และหลอด LED จะติดค้างเป็นเวลา 5 วินาทีแล้วจะ
ได้รับคำสั่ง E จากส่วนประมวลผลจึงจะกลับเข้าสู่สถานะปกติ

3.4.7 ส่วนแสดงผลได้รับคำสั่ง E

เมื่อส่วนแสดงผลได้รับคำสั่ง E จากส่วนประมวลผลจะทำการแสดงข้อความ Welcome! ที่จอแสดงผล LCD และหลอด LED จะกระพริบ

3.4.8 ส่วนแสดงผลได้รับคำสั่งปรับเวลา

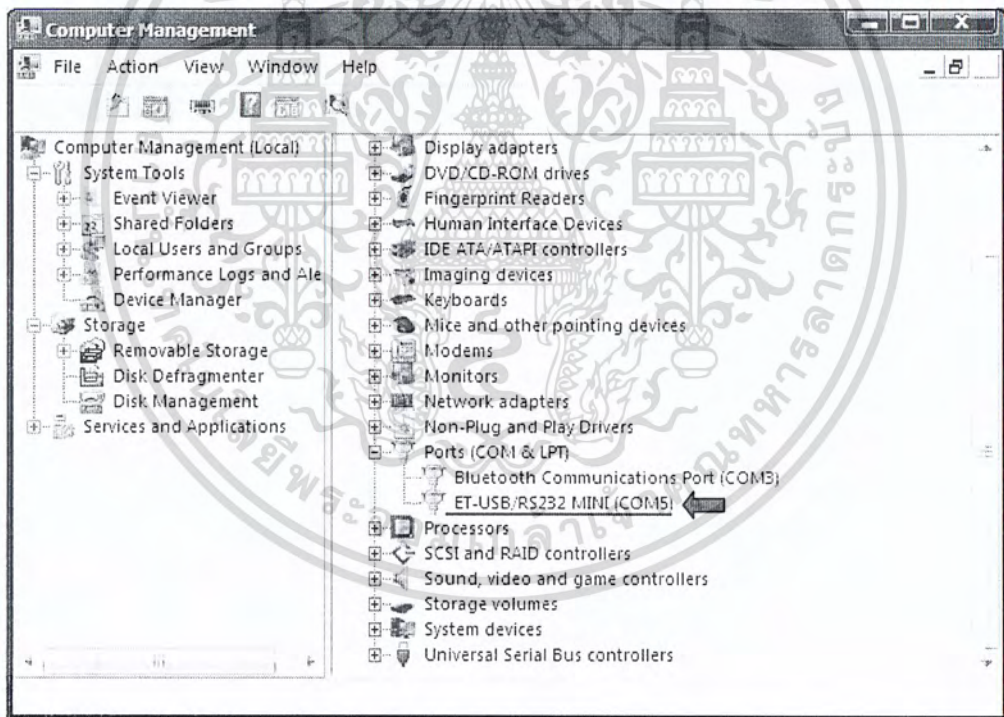
เมื่อส่วนแสดงผลได้รับคำสั่ง T11:11/ ซึ่งเป็นคำสั่งให้ทำการแสดงเวลาตามคำสั่งที่ได้รับ ส่วนแสดงผลจะทำตามเงื่อนไข โปรแกรม จากคำสั่งตัวอย่างข้างต้นส่วนแสดงผลจะแสดงเวลา 11:11 น.

บทที่ 4

ผลการทดลอง

4.1 การเชื่อมต่อของส่วนแสดงผลเข้ากับส่วนประมวลผล

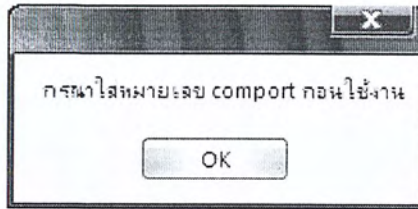
การเชื่อมต่อระหว่างทั้งสองส่วนนี้ทำการเชื่อมต่อผ่านทางพอร์ตอนุกรม โดยการใช้งานระบบจะต้องทำการกำหนดพอร์ตกอมพิวเตอร์ให้ถูกต้องจึงจะสามารถใช้งานระบบได้ พอร์ตการเชื่อมต่อสามารถทำการตรวจสอบได้ดังรูปที่ 4.1



รูปที่ 4.1 ตรวจสอบหมายเลขพอร์ท RS-232

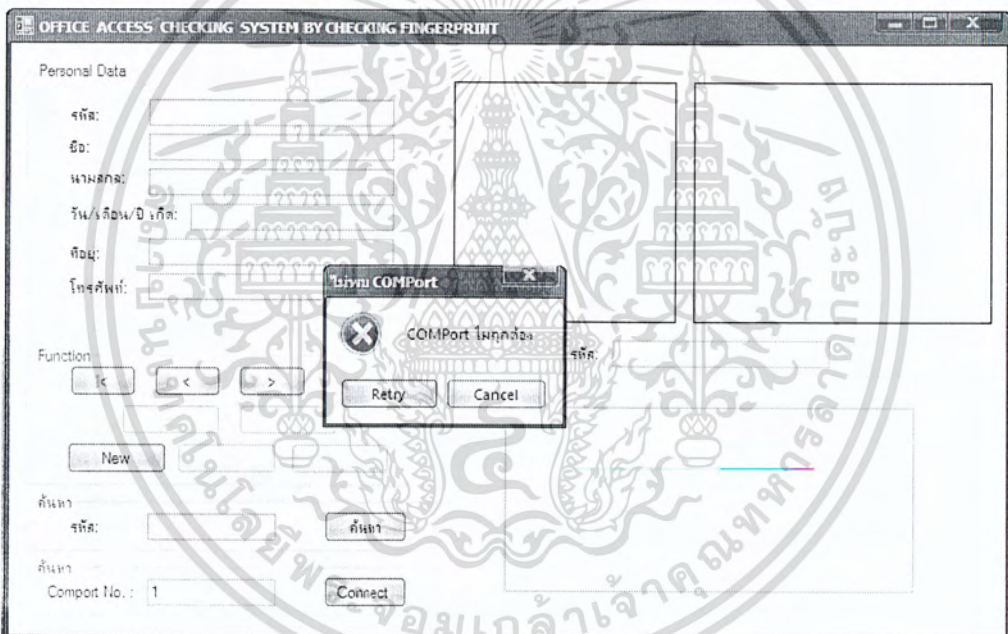
เมื่อทำการเริ่ม โปรแกรมจะมีกล่องข้อความเตือนให้กำหนดพอร์ตกอมพิวเตอร์
ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 กดลองข้อความเตือนให้ใส่หมายเลขคอมพอร์ทก่อนใช้งาน

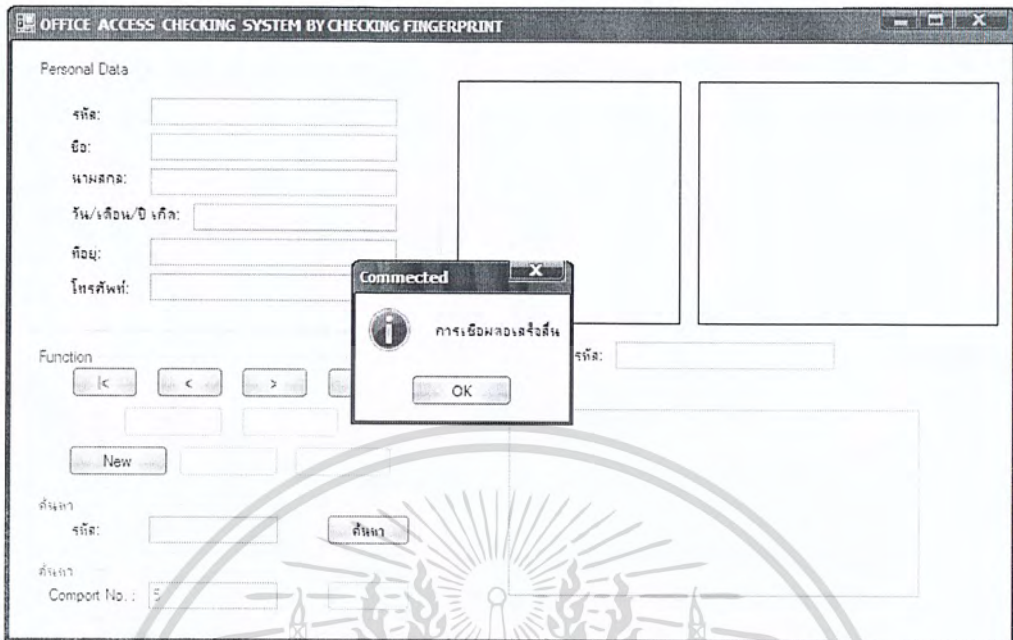
ในกรณีที่ระบุหมายเลขคอมพอร์ทไม่ถูกต้องจะมีกล่องข้อความแจ้งว่า COMPort ไม่ถูกต้องและให้เลือกที่จะใส่เลขคอมพอร์ทอีกครั้งหรือปิดการทำงานของโปรแกรม ดังรูปที่ 4.3



รูปที่ 4.3 ใส่หมายเลขคอมพอร์ทไม่ถูกต้อง

ในกรณีที่ระบุหมายเลขคอมพอร์ทถูกต้องจะมีกล่องข้อความแจ้งว่า การเชื่อมต่อเสร็จสิ้น ดังรูปที่ 4.4 หลังจากนั้นระบบพร้อมที่จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การเชื่อมต่อถูกต้อง ระบบพร้อมที่จะใช้งาน

หลังจากมีการเชื่อมต่อแล้วส่วนแสดงผลจะแสดงเวลาที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Welcome! ที่จอแสดงผล LCD บรรทัดล่างดังรูปที่ 4.5



รูปที่ 4.5 จอแสดงผล LCD ในสถานะปกติ

เมื่อการเชื่อมต่อกับส่วนแสดงผลเรียบร้อยแล้วระบบจะทำการรอรับรหัสประจำตัวของผู้ใช้งานและผู้ดูแลระบบสามารถเพิ่มเติมหรือแก้ไขข้อมูลของผู้ใช้งานได้

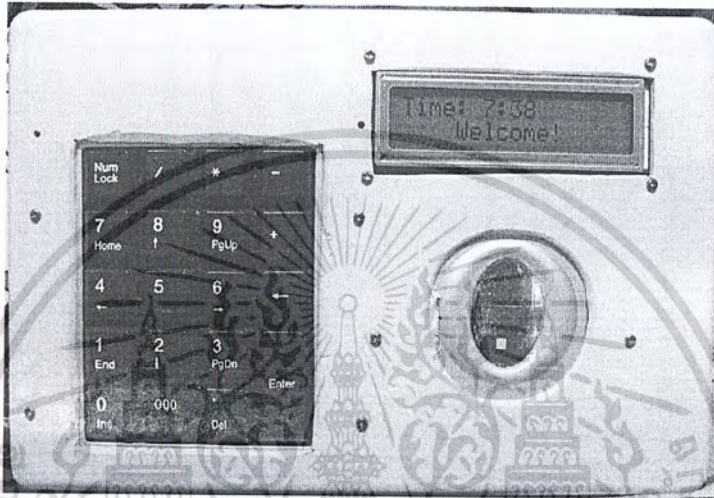
4.2 การทำงานในส่วนของผู้ใช้งาน

การทำงานในส่วนของผู้ใช้แบ่งเป็นสถานะต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 สถานะปกติ

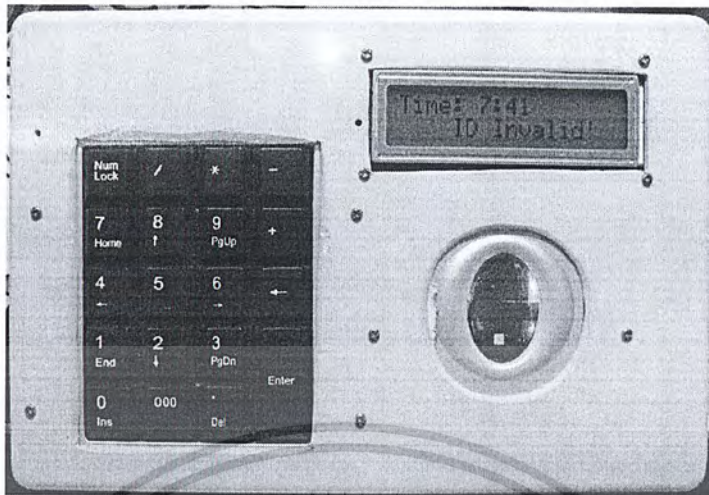
สถานะปกติเป็นสถานะของเครื่องเมื่อไม่มีผู้เข้ามาใช้งานตัวเครื่องจะแสดงเวลาปัจจุบันที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Welcome! ที่บรรทัดล่างของจอแสดงผล LCD ส่วนหลอด LED ตติกระพริบดังรูปที่ 4.6



รูปที่ 4.6 สถานะปกติ

4.2.2 สถานะรหัสประจำตัวไม่ถูกต้อง

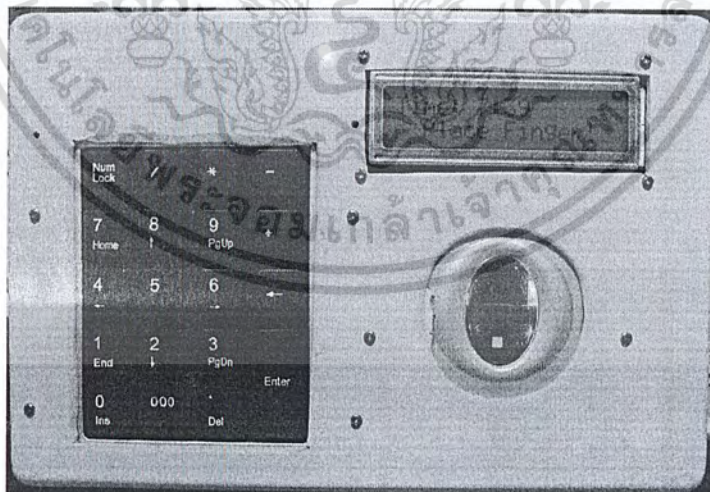
เป็นสถานะที่ผู้ใช้งานกรอกรหัสประจำตัวไม่ถูกต้องส่วนแสดงผลจะแสดงเวลาปัจจุบันที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Welcome! ที่บรรทัดล่างของจอแสดงผล LCD ส่วนหลอด LED จะติดค้าง 5 วินาทีหลังจากนั้นจะติคกระพริบแล้วกลับไปสถานะปกติดังรูปที่ 4.7



รูปที่ 4.7 สถานะรหัสประจำตัวไม่ถูกต้อง

4.2.3 สถานะรอการสแกนลายนิ้วมือ

เป็นสถานะรอการสแกนลายนิ้วมือของผู้ใช้งานเพื่อที่จะนำมาตรวจสอบกับลายนิ้วมือที่ถูกตั้งขึ้นมารอตรวจสอบหลังจากรหัสประจำตัวผ่านการตรวจสอบแล้วว่าถูกต้อง ส่วนแสดงผลจะแสดงเวลาปัจจุบันที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Place Finger! ที่บรรทัดล่างของจอแสดงผล LCD ส่วนหลอด LED จะติดค้างดังรูปที่ 4.8

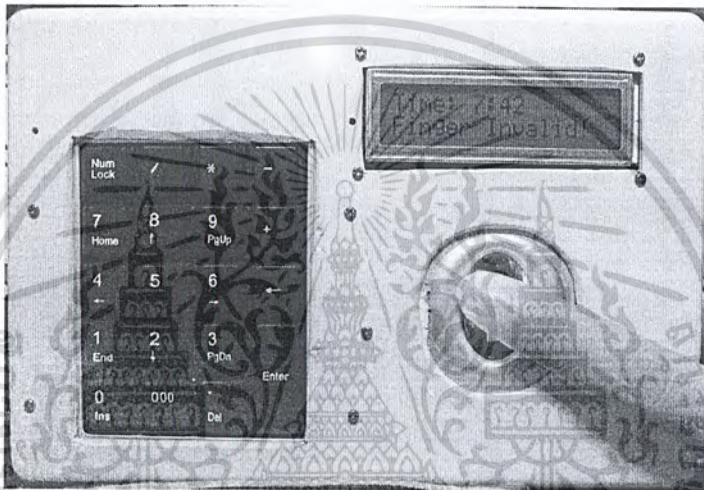


รูปที่ 4.8 สถานะรอการสแกนลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 สถานะลายนิ้วมือไม่ถูกต้อง

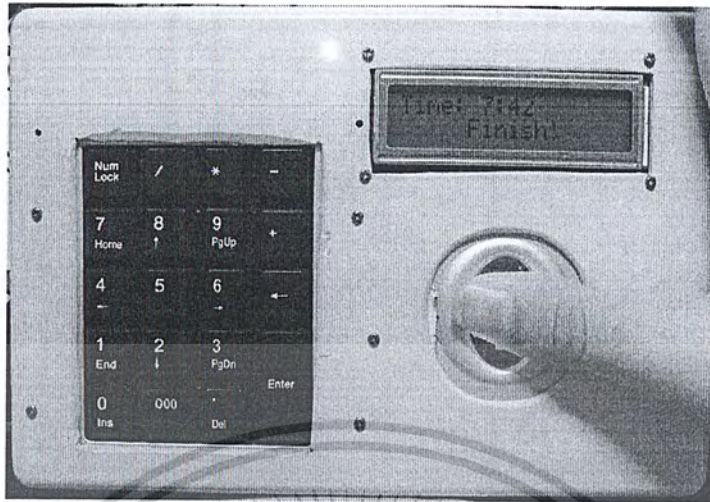
เป็นสถานะที่ผู้ใช้งานสแกนลายนิ้วมือเข้าสู่ส่วนประมวลผลเพื่อจะนำมาตรวจสอบแล้วปรากฏว่าลายนิ้วมือไม่ถูกต้อง ไม่ตรงกับลายนิ้วมือที่ใช้อ้างอิงของระบบ ส่วนแสดงผลจะแสดงเวลาปัจจุบันที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Finger Invalid! ที่บรรทัดล่างของจอแสดงผล LCD ส่วนหลอด LED จะติดค้าง 5 วินาทีหลังจากนั้นจะติดกระพริบแล้วกลับไปสถานะปกติดังรูปที่ 4.9



รูปที่ 4.9 สถานะลายนิ้วมือไม่ถูกต้อง

4.2.5 สถานะตรวจสอบลายนิ้วมือถูกต้อง

เป็นสถานะที่ผู้ใช้งานสแกนลายนิ้วมือเข้าสู่ส่วนประมวลผลเพื่อจะนำมาตรวจสอบแล้วปรากฏว่าลายนิ้วมือถูกต้อง ตรงกับลายนิ้วมือที่ใช้อ้างอิงของระบบ ส่วนแสดงผลจะแสดงเวลาปัจจุบันที่บรรทัดบนของจอแสดงผล LCD และแสดงข้อความ Finish! ที่บรรทัดล่างของจอแสดงผล LCD ส่วนหลอด LED จะติดค้าง 5 วินาทีหลังจากนั้นจะติดกระพริบแล้วกลับไปสถานะปกติดังรูปที่ 4.10



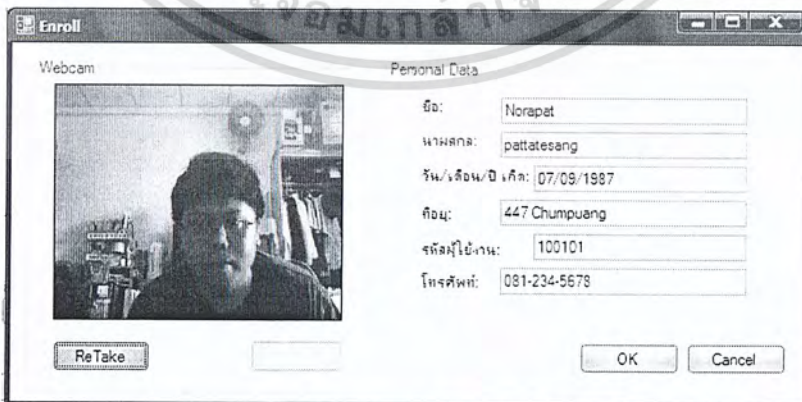
รูปที่ 4.10 สถานะถายนิ้วมือถูกต้อง

4.3 การทำงานในส่วนของผู้ดูแลระบบ

การทำงานในส่วนของผู้ดูแลระบบหลังจากขั้นตอนที่ 4.1 แล้วผู้ดูแลระบบจะมีหน้าที่ในการเพิ่มหรือแก้ไขข้อมูลของผู้ใช้งานในระบบ

4.3.1 การเพิ่มผู้ใช้งานเข้าระบบ

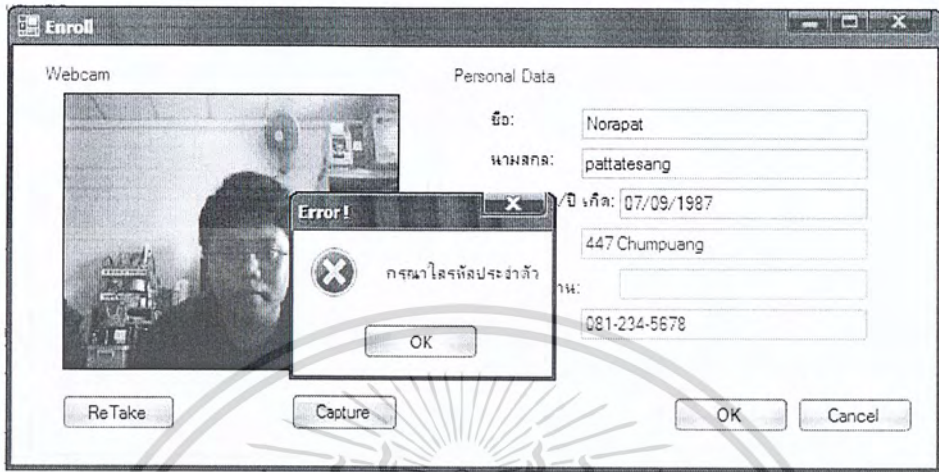
การเพิ่มข้อมูลเข้าระบบเป็นการนำข้อมูลต่างๆ ของผู้ใช้งานมาทำการบันทึกลงฐานข้อมูลเพื่อใช้เป็นข้อมูลในการตรวจสอบดังรูปที่ 4.11



รูปที่ 4.11 กรอกข้อมูลครบถ้วนแล้วทำการถ่ายรูป

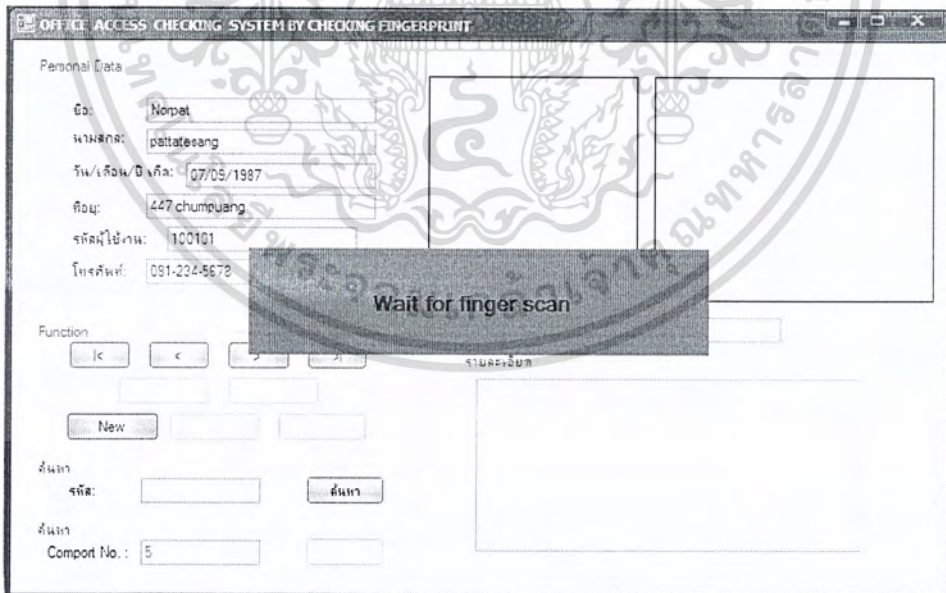
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่กรอกข้อมูลไม่ครบจะมีกล่องข้อความเตือนดังรูปที่ 4.12



รูปที่ 4.12 กล่องข้อความเตือนให้กรอกข้อมูล

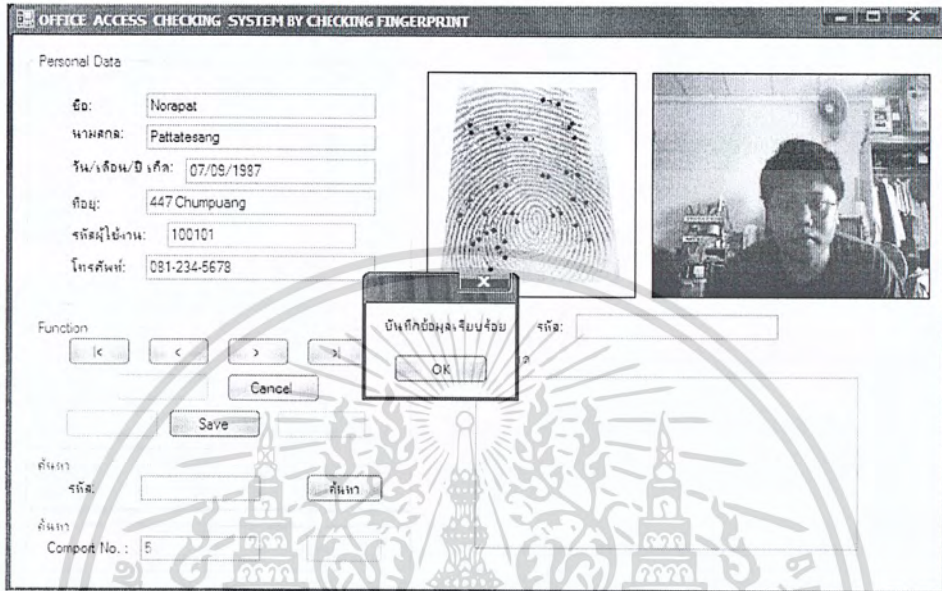
เมื่อกรอกข้อมูลเรียบร้อยแล้วดังรูปที่ 4.11 แล้วจะทำการเก็บข้อมูลลายนิ้วมือ ดังรูปที่ 4.13



รูปที่ 4.13 รอการสแกนลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการสแกนและตรวจสอบข้อมูลเรียบร้อยแล้วทำการบันทึกข้อมูลลงฐานข้อมูลดังรูปที่ 4.14 – 4.16



รูปที่ 4.14 ทำการบันทึกข้อมูลแล้ว



รูปที่ 4.15 ตาราง enroll ก่อนและหลังทำการบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนบันทึกข้อมูล

ID	name	lastname	birthdate	address	tel	picfinger	piccam

หลังบันทึกข้อมูล

ID	name	lastname	birthdate	address	tel	picfinger	piccam
100101	Norapat	Pattatesang	07/09/1987	447 Chumpuar	081-234-5678	pf100101.jpg	pc100101.jpg

รูปที่ 4.16 ตาราง user_info ก่อนและหลังทำการบันทึก

จากรูปที่ 4.15 – 4.16 จะเห็นว่าในตารางฐานข้อมูลยังไม่มีข้อมูลผู้ใช้งาน เมื่อทำการบันทึกข้อมูลจึงมีข้อมูลผู้ใช้งานอยู่ในฐานข้อมูล

4.3.2 การลบผู้ใช้งานออกจากระบบ

การลบข้อมูลของผู้ใช้งานออกจากระบบเป็นการลบข้อมูลทุกอย่างของบุคคลนั้นออกจากฐานข้อมูล โดยสามารถค้นหาผู้ใช้งานที่จะทำการลบ โดยใช้ฟังก์ชันค้นหาจากรหัสพนักงานดังรูปที่ 4.17 – 4.18

OFFICE ACCESS CHECKING SYSTEM BY CHECKING FINGERPRINT

Personal Data

ชื่อ: Norapat
 นามสกุล: Pattatesang
 วัน/เดือน/ปีเกิด: 07/09/1987
 ที่อยู่: 447 Chumpuang
 รหัสผู้ใช้งาน: 100101
 โทรศัพท์: 081-234-5678

Function

Navigation: < > < >
 Edit
 New Delete

ค้นหา
 รหัส: 100101 ค้นหา

ค้นหา
 Comport No.: 5

รหัส:

จำนวนรอบ

ลำดับที่	วันที่	เวลา
1	7/3/2554	15:47:55
2	7/3/2554	15:48:00
3	7/3/2554	15:48:04
4	7/3/2554	15:48:09
5	7/3/2554	15:48:13
6	7/3/2554	15:48:17

รูปที่ 4.17 ค้นหาพนักงานโดยใช้รหัสพนักงาน

OFFICE ACCESS CHECKING SYSTEM BY CHECKING FINGERPRINT

Personal Data

ชื่อ: Norapat
 นามสกุล: Pattatesang
 วัน/เดือน/ปีเกิด: 07/09/1987
 ที่อยู่: 447 Chumpuang
 รหัสผู้ใช้งาน: 100101
 โทรศัพท์: 081-234-5678

Function

Navigation: < > < >
 Edit
 New Delete

ค้นหา
 รหัส: 100101 ค้นหา

ค้นหา
 Comport No.: 5

Warning!
 ต้องการลบ ?
 OK Cancel

จำนวนรอบ

ลำดับที่	วันที่	เวลา
1	7/3/2554	15:47:55
2	7/3/2554	15:48:00
3	7/3/2554	15:48:04
4	7/3/2554	15:48:09
5	7/3/2554	15:48:13
6	7/3/2554	15:48:17

รูปที่ 4.18 กล่องข้อความยืนยันการลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการทำงานของระบบ

การทำงานของระบบแบ่งออกเป็น 2 ส่วน ได้แก่

5.1.1 การทำงานในส่วนของผู้ดูแลระบบ

ผู้ดูแลระบบสามารถทำการเพิ่ม - ลบข้อมูลต่างๆของพนักงานและตรวจสอบข้อมูลการเข้าออกหรือข้อมูลส่วนตัวต่างๆของพนักงานได้โดยผ่านทางเครื่องคอมพิวเตอร์ การเพิ่มผู้ใช้งานเข้าระบบจะต้องระบุข้อมูลชื่อ นามสกุล วันเกิด ที่อยู่ รหัสพนักงาน หมายเลขโทรศัพท์ ภาพถ่ายและข้อมูลลายนิ้วมือ (ได้มาจากการสแกนลายนิ้วมือ เป็นข้อมูล Long binary) ผู้ใช้งานจึงจะสามารถใช้งานระบบได้

5.1.2 การทำงานในส่วนของผู้ใช้งาน

ผู้ใช้งานจะใช้งานระบบผ่านทางตัวเครื่องโดยจะต้องระบุรหัสพนักงานก่อนที่จะไปถึงขั้นตรวจสอบลายนิ้วมือ เพื่อเพิ่มความปลอดภัยของระบบ

5.2 บทวิจารณ์

เนื่องจากการเชื่อมต่อระหว่างตัวเครื่องเข้ากับคอมพิวเตอร์ต้องเชื่อมต่อผ่านทาง USB พอร์ต ซึ่งมีความจำกัดในด้านระยะทางทำให้ไม่สามารถติดตั้งตัวเครื่องห่างจากคอมพิวเตอร์ได้ไม่เกิน 5 เมตรและต้องเปิดเครื่องคอมพิวเตอร์ไว้ตลอดเวลาที่ใช้งาน แนวทางการพัฒนาต่อพัฒนาโดยใช้การเชื่อมต่อไร้สายเพื่อแก้ปัญหาด้านความจำกัดของระยะทางของตัวเครื่องกับคอมพิวเตอร์และพัฒนาโดยการสร้างเซิร์ฟเวอร์ฐานข้อมูลเพื่อให้ผู้ดูแลระบบสามารถทำการตรวจสอบข้อมูลการเข้า - ออกและข้อมูลของผู้ใช้งานผ่านทางอินเทอร์เน็ตได้

บรรณานุกรม

- [1] บัญชา ปะสีละเตสัง. พัฒนาแอปพลิเคชันด้วย Visual C# 2008. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2552.
- [2] จีระสิทธิ์ อึ้งรัตนวงศ์. คู่มือการใช้โปรแกรม Microsoft Access 2007. กรุงเทพฯ : สำนักพิมพ์ สวีสวี ไอที, 2551.
- [3] ทีมงานสมาร์ทเลิร์นนิ่ง. PIC Microcontroller Learning-By-Doing ด้วยภาษา C. พิมพ์ครั้งที่ 1. กรุงเทพฯ : สมาร์ทเลิร์นนิ่ง, 2550.
- [4] สัจจะ จรัสรุ่งรวีวร. คู่มือ Visual C# 2005 ฉบับสมบูรณ์. นนทบุรี : ไอดีซี, 2550.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Code FormMain //
```

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using System.Data.OleDb;  
using System.Runtime.InteropServices;  
using GrFingerXLib;  
using System.IO.Ports;  
using System.Threading;  
using System.IO;
```

```
namespace WindowsFormsApplication1  
{
```

```
    // กำหนดตัวแปรต่างๆ //
```

```
    public partial class frmMain : Form
```

```
    {  
        private SerialPort comport;  
        int score,busyComport=0;  
        string state = "" ;  
        int stateFinger = 0;  
        string strTime;  
        string id, name, lastname, birthdate, address, tel,picFingerName,picCamName;  
        string pathPicFinger = Application.StartupPath + "/picFinger/";  
        string pathPicCam = Application.StartupPath + "/picCam/";  
        [DllImport("user32.dll", EntryPoint = "GetDC")]  
        public static extern IntPtr GetDC(IntPtr ptr);  
        [DllImport("user32.dll", EntryPoint = "ReleaseDC")]  
        public static extern IntPtr ReleaseDC(IntPtr hWnd, IntPtr hDc);  
        TTemplate template;  
        DBFingerClass DBClass = new DBFingerClass();  
        alertFrm popupFrm = new alertFrm();
```

```
    public frmMain()
```

```
    {  
        InitializeComponent();  
    }
```

```
        // เริ่มโปรแกรม //
```

```
    private void frmMain_Load(object sender, EventArgs e)
```

```
    {  
        txtID.Select();  
        int err = InitializeGrFinger(axGrFingerXCtrl1);  
        DBClass.openDB(Application.StartupPath + "/student.mdb");  
        btnCancel.Enabled = false;  
        btnDelete.Enabled = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    btnEdit.Enabled = false;
    btnSave.Enabled = false;
    btnAdd.Enabled = false;
    MessageBox.Show("กรุณาใส่หมายเลข comport ก่อนใช้งาน");
    txtComportNo.Select();
}

void port_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
}

public int InitializeGrFinger(AxGrFingerXLib.AxGrFingerXCtrl grfingerx)
{
    GRConstants result;
    AxGrFingerXLib.AxGrFingerXCtrl _grfingerx;
    _grfingerx = grfingerx;
    if (template == null)
        template = new TTemplate();
    result = (GRConstants)_grfingerx.Initialize();
    if (result < 0) return (int)result;
    return (int)_grfingerx.CapInitialize();
}

// ตั้ง //
private void axGrFingerXCtrl1_SensorPlug(object sender,
AxGrFingerXLib._IGrFingerXCtrlEvents_SensorPlugEvent e)
{
    axGrFingerXCtrl1.CapStartCapture(e.idSensor);
}

private void axGrFingerXCtrl1_SensorUnplug(object sender,
AxGrFingerXLib._IGrFingerXCtrlEvents_SensorUnplugEvent e)
{
    axGrFingerXCtrl1.CapStopCapture(e.idSensor);
}

private void axGrFingerXCtrl1_ImageAcquired(object sender,
AxGrFingerXLib._IGrFingerXCtrlEvents_ImageAcquiredEvent e)
{
    int ret;
    if (stateFinger == 1)
    {
        IntPtr hdc = GetDC(System.IntPtr.Zero);
        System.Drawing.Image handle = null;
        axGrFingerXCtrl1.CapRawImageToHandle(ref e.rawImage, (int)e.width, (int)e.height,
hdc.ToInt32(), ref handle);
        handle = null;
        ReleaseDC(System.IntPtr.Zero, hdc);
        ret = ExtractTemplate(e);
        axGrFingerXCtrl1.BiometricDisplay(ref template.tptData, ref e.rawImage, e.width, e.height,
e.res, hdc.ToInt32(), ref handle, -1);
        if (handle != null)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        img.Image = handle;
        img.Update();
    }
    popupFrm.Hide();
}
}

public int ExtractTemplate(AxGrFingerXLib._IGrFingerXCtrlEvents_ImageAcquiredEvent e)
{
    int result;
    template.size = (int)GRConstants.GR_MAX_SIZE_TEMPLATE;
    result = (int)axGrFingerXCtrl1.Extract(ref e.rawImage, e.width, e.height, e.res, ref
template.tptData, ref template.size, (int)GRConstants.GR_DEFAULT_CONTEXT);

    if (result < 0)
    {
        template.size = 0;
    }
    return result;
}

// เคลียร์ข้อมูลในช่องชื่อ , นามสกุล , ที่อยู่ , วันเกิด , รหัสและหมายเลขโทรศัพท์ //
void clearForm()
{
    txtFirstName.Text = "";
    txtLastName.Text = "";
    txtAddress.Text = "";
    txtBirthDate.Text = "";
    txtIDStudent.Text = "";
    txtTel.Text = "";
}

// ทำให้ข้อมูลในช่องชื่อ , นามสกุล , ที่อยู่ , วันเกิด , รหัสและหมายเลขโทรศัพท์สามารถแก้ไขได้ //
void txtReadOnlyFalse()
{
    txtFirstName.ReadOnly = false;
    txtLastName.ReadOnly = false;
    txtAddress.ReadOnly = false;
    txtBirthDate.ReadOnly = false;
    txtTel.ReadOnly = false;
}

// ทำให้ข้อมูลในช่องชื่อ , นามสกุล , ที่อยู่ , วันเกิด , รหัสและหมายเลขโทรศัพท์สามารถแก้ไขไม่ได้ //
void txtReadOnlyTrue()
{
    txtFirstName.ReadOnly = true;
    txtLastName.ReadOnly = true;
    txtAddress.ReadOnly = true;
    txtBirthDate.ReadOnly = true;
    txtTel.ReadOnly = true;
}

// แสดงเวลาการใช้งานของพนักงานในช่อง listbox1 //

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void listTimeScan(string id)
{
    string[] list;
    string[] dtScan;
    list = DBClass.queryTimeScan(id);
    int num = list.Length;
    listBox1.Items.Clear();

listBox1.Items.Add("=====");
    listBox1.Items.Add(" ลำดับที่ | วันที่ | เวลา ");

listBox1.Items.Add("=====");

    for (int i = 0; i < num; i++)
    {
        dtScan = list[i].Split(' ');
        string x = string.Format("{0,10}{1,25}{2,32}", i+1, dtScan[0].ToString(), dtScan[1].ToString());
        listBox1.Items.Add(x);
    }

listBox1.Items.Add("=====");
}
// กำหนดตัวแปร x ให้เป็นเวลาปัจจุบัน //
void addTimeScan(string id)
{
    DateTime x = new DateTime();
    x = DateTime.Now;
    DBClass.insertTimeScan(id, x);
}
// แสดงข้อมูลผู้ใช้งานจากฐานข้อมูลลงในช่องต่างๆ //
void showUserData(string id)
{
    try
    {
        DBClass.queryData(ref id, ref name, ref lastname, ref birthdate, ref address, ref tel, ref
picFingerName, ref picCamName);
        txtFirstName.Text = name;
        txtLastName.Text = lastname;
        txtAddress.Text = address;
        txtBirthDate.Text = birthdate;
        txtIDStudent.Text = id;
        txtTel.Text = tel;
        btnEdit.Enabled = true;
        btnDelete.Enabled = true;
        img.Image = Image.FromFile(pathPicFinger + picFingerName);
        picWebcam.Image = Image.FromFile(pathPicCam + picCamName);
        img.Update();
        picWebcam.Update();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch
{
}
}
// ส่งเวลาผ่านทางพอร์ทอนุกรมให้ส่วนแสดงผล //
void sendSerial(string data)
{
strTime = DateTime.Now.Hour.ToString() + ":" + DateTime.Now.Minute.ToString();
try
{
if (!comport.IsOpen && busyComport == 0)
comport.Open();

busyComport = 1;
comport.Write("T" + strTime + "/" + data);
busyComport = 0;
}
catch
{
MessageBox.Show("กรุณาติดต่อ comport");
this.Close();
this.Dispose();
}
}
// เพิ่มข้อมูลพนักงานเข้าระบบ //
private void btnAdd_Click(object sender, EventArgs e)
{
state = "ADD";
enrollFrm enrollDlg = new enrollFrm();
if(enrollDlg.ShowDialog() == DialogResult.OK)
{
string path;
txtFirstName.Text = enrollDlg.getTxtFirstName();
txtLastName.Text = enrollDlg.getTxtLastName();
txtAddress.Text = enrollDlg.getTxtAddress();
txtBirthDate.Text = enrollDlg.getTxtBirthDate();
txtIDStudent.Text = enrollDlg.getTxtIDStudent();
txtTel.Text = enrollDlg.getTxtTel();
path = enrollDlg.getPathPicCam();
stateFinger = 1;
popupFrm.ShowDialog();
FileStream fs = new FileStream(path, FileMode.Open, FileAccess.ReadWrite);
picWebcam.Image = System.Drawing.Image.FromStream(fs);
fs.Close();
picWebcam.Tag = path;
btnAdd.Enabled = false;
btnDelete.Enabled = false;
btnEdit.Enabled = false;
btnSave.Enabled = true;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    btnCancel.Enabled = true;
    listBox1.Items.Clear();
    enrollDlg = null;
}
}
// บันทึกข้อมูลพนักงานลงฐานข้อมูล //
private void btnSave_Click(object sender, EventArgs e)
{
    if (state == "ADD")
    {
        if (DBClass.checkIDInDB(txtIDStudent.Text.ToString()) == 0)
        {
            img.Image.Save(pathPicFinger + "pf" + txtIDStudent.Text + ".jpg");
            DBClass.saveTemplate(template, txtIDStudent.Text.ToString());
            if (DBClass.insertData(txtIDStudent.Text.ToString(), txtFirstName.Text.ToString(),
txtLastName.Text.ToString(), txtBirthDate.Text.ToString(), txtAddress.Text.ToString(),
txtTel.Text.ToString(), "pf" + txtIDStudent.Text.ToString() + ".jpg", "pc" + txtIDStudent.Text.ToString()
+ ".jpg"))
            {
                MessageBox.Show("บันทึกข้อมูลเรียบร้อยแล้ว");
            }
            else
            {
                MessageBox.Show("ขัดข้อง! ไม่สามารถบันทึกข้อมูลได้");
            }
        }
        else
        {
            MessageBox.Show("ข้อมูลรหัสนักศึกษาซ้ำ");
            clearForm();
            picWebcam.Image = null;
            picWebcam.Update();
            img.Image = null;
            img.Update();
        }
    }
}
else if (state == "EDIT")
{
    if (DBClass.updateData(txtIDStudent.Text.ToString(), txtFirstName.Text.ToString(),
txtLastName.Text.ToString(), txtBirthDate.Text.ToString(), txtAddress.Text.ToString(),
txtTel.Text.ToString())
    {
        MessageBox.Show("บันทึกข้อมูลเรียบร้อยแล้ว");
    }
    else
    {
        MessageBox.Show("ขัดข้อง! ไม่สามารถบันทึกข้อมูลได้");
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
state = "SAVE";
txtReadOnlyTrue();
btnSave.Enabled = false;
btnCancel.Enabled = false;
btnAdd.Enabled = true;
stateFinger = 0;
txtID.Select();
}
// ยกเลิกการเพิ่มข้อมูลพนักงานเข้าระบบ //
private void btnCancel_Click(object sender, EventArgs e)
{
    state = "CANCEL";
    clearForm();

    img.Image = null;
    img.Update();

    picWebcam.Image = null;
    picWebcam.Update();
    btnCancel.Enabled = false;
    btnAdd.Enabled = true;
    btnSave.Enabled = false;
    txtID.Select();
}
// แก้ไขข้อมูลพนักงาน //
private void btnEdit_Click(object sender, EventArgs e)
{
    state = "EDIT";
    txtReadOnlyFalse();
    btnSave.Enabled = true;
    btnEdit.Enabled = false;
    btnCancel.Enabled = true;
}

private void btnDelete_Click(object sender, EventArgs e)
{
    DialogResult del = MessageBox.Show("ต้องการลบ ?", "Warning !",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
    if (del == DialogResult.OK)
    {
        btnDelete.Enabled = false;
        DBClass.deleteData(txtIDStudent.Text.ToString());
        DBClass.deleteTemplate(txtIDStudent.Text.ToString());
        DBClass.deleteListOfTimeScan(txtIDStudent.Text.ToString());
        img.Image = null;
        img.Update();
        picWebcam.Image = null;
        picWebcam.Update();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MessageBox.Show("ลบข้อมูลเรียบร้อยแล้ว");
    listBox1.Items.Clear();
    txtID.Select();
    clearForm();
}
else
{
}
}

void txtID_KeyDown(object sender, KeyEventArgs e)
{
    int ansVerify = 0;
    TTemplate refTptFromDB = new TTemplate();

    if (e.KeyValue == 13)
    {
        if (txtID.Text != "")
        {
            timer1.Stop();
            if (DBClass.checkIDInDB(txtID.Text.ToString()) == 1)
            {
                stateFinger = 1;
                sendSerial("C");
                popupFrm.ShowDialog();
                refTptFromDB = DBClass.getTemplateFromID(txtID.Text.ToString());
                ansVerify = axGrFingerXCtrl1.Verify(ref refTptFromDB.tptData, ref template.tptData,
                ref score, (int)GrFingerXLib.GRConstants.GR_DEFAULT_CONTEXT);
                if (ansVerify == 1)
                {
                    id = txtID.Text.ToString();
                    showUserData(id);
                    addTimeScan(id);
                    listTimeScan(id);
                    txtID.Text = "";
                    stateFinger = 0;
                    sendSerial("D");
                    timer3.Start();
                    timer2.Start();
                    timer1.Start();
                }
            }
            else
            {
                sendSerial("B");
                clearForm();
                picWebcam.Image = null;
                picWebcam.Update();
                img.Image = null;
                img.Update();
                btnEdit.Enabled = false;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
}
}

private void btnConnect_Click(object sender, EventArgs e)
{
    comport = new SerialPort("COM"+txtComportNo.Text.ToString(), 9600, Parity.None, 8,
StopBits.One);
    comport.Handshake = Handshake.None;
    comport.ReadTimeout = 500;
    comport.WriteTimeout = 500;
    comport.DataReceived += new SerialDataReceivedEventHandler(this.port_DataReceived);
    try
    {
        comport.Open();
        btnConnect.Enabled = false;
        txtComportNo.Enabled = false;
        timer1.Start();
        txtID.Select();
        btnAdd.Enabled = true;

        strTime = DateTime.Now.Hour.ToString() + ":" + DateTime.Now.Minute.ToString();
        if (!comport.IsOpen && busyComport == 0)
            comport.Open();
        busyComport = 1;
        comport.Write("T" + strTime + "/E");
        busyComport = 0;
        MessageBox.Show("การเชื่อมต่อเสร็จสิ้น", "Connected", MessageBoxButtons.OK,
MessageBoxIcon.Information);

    }
    catch
    {
        DialogResult dlg = MessageBox.Show("COMPort ไม่ถูกต้อง", "ไม่พบ
COMPort", MessageBoxButtons.RetryCancel, MessageBoxIcon.Stop);
        if (dlg == DialogResult.Cancel)
        {
            this.Close();
            this.Dispose();
        }
        else
        {
            txtComportNo.Text = "";
            txtComportNo.Select();
        }
    }
}

private void timer1_Tick(object sender, EventArgs e)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strTime = DateTime.Now.Hour.ToString() + ":" + DateTime.Now.Minute.ToString();
if (!comport.IsOpen && busyComport == 0)
    comport.Open();
    busyComport = 1;
    comport.Write("T" + strTime + "/E");
    busyComport = 0;
}

private void timer2_Tick(object sender, EventArgs e)
{
    timer2.Stop();
    sendSerial("E");
}

private void btnFirst_Click(object sender, EventArgs e)
{
    DBClass.cmdFirstRow(ref id, ref name, ref lastname, ref birthdate, ref address, ref tel, ref
picFingerName, ref picCamName);
    showUserData(id);
    listTimeScan(id);
    txtID.Select();
}

private void btnPrevious_Click(object sender, EventArgs e)
{
    DBClass.cmdPrevRow(ref id, ref name, ref lastname, ref birthdate, ref address, ref tel, ref
picFingerName, ref picCamName);
    showUserData(id);
    listTimeScan(id);
    txtID.Select();
}

private void btnNext_Click(object sender, EventArgs e)
{
    DBClass.cmdNextRow(ref id, ref name, ref lastname, ref birthdate, ref address, ref tel, ref
picFingerName, ref picCamName);
    showUserData(id);
    listTimeScan(id);
    txtID.Select();
}

private void btnLast_Click(object sender, EventArgs e)
{
    DBClass.cmdLastRow(ref id, ref name, ref lastname, ref birthdate, ref address, ref tel, ref
picFingerName, ref picCamName);
    showUserData(id);
    listTimeScan(id);
    txtID.Select();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void timer3_Tick(object sender, EventArgs e)
{
    clearForm();
    listBox1.Items.Clear();
    img.Image = null;
    img.Update();
    picWebcam.Image = null;
    picWebcam.Update();
    timer3.Stop();
}

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

}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Code enroll //

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace WindowsFormsApplication1
{
    public partial class enrollFrm : Form
    {
        string pathPicCam = Application.StartupPath + "/picCam/";
        string pathPicFinger = Application.StartupPath + "/picFinger/";

        public enrollFrm()
        {
            InitializeComponent();
        }
        private void enrollFrm_Load(object sender, EventArgs e)
        {
            picWebcam.Image = null;
            btnOK.Enabled = false;
            this.webCamCapture.CaptureHeight = 480;
            this.webCamCapture.CaptureWidth = 640;
            webCamCapture.TimeToCapture_milliseconds = 10;
            webCamCapture.Start(0);
        }
        private void enrollFrm_close(object sender, EventArgs e)
        {
            webCamCapture.Stop();
        }
        private void WebCamCapture_ImageCaptured(object source, WebCam_Capture.WebcamEventArgs
e)
        {
            this.picWebcam.Image = e.WebCamImage;
        }
        private void capture_Click(object sender, EventArgs e)
        {
            if (txtFirstName.Text == "")
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MessageBox.Show("กรุณาใส่ชื่อ", "Error !", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}
else if (txtLastName.Text == "")
{
    MessageBox.Show("กรุณาใส่นามสกุล", "Error !", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}
else if (txtBirthDate.Text == "")
{
    MessageBox.Show("กรุณาใส่วันเกิด", "Error !", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}
else if (txtAddress.Text == "")
{
    MessageBox.Show("กรุณาใส่ที่อยู่", "Error !", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}
else if (txtIDStudent.Text == "")
{
    MessageBox.Show("กรุณาใส่รหัสประจำตัว", "Error !", MessageBoxButtons.OK, MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}
else if (txtTel.Text == "")
{
    MessageBox.Show("กรุณาใส่หมายเลขโทรศัพท์", "Error !", MessageBoxButtons.OK,
MessageBoxIcon.Stop);
    webCamCapture.Stop();
    webCamCapture.Start(0);
}

else
{
    webCamCapture.Stop();
    try
    {
        picWebcam.Image.Save(pathPicCam + "pc" + txtIDStudent.Text + ".jpg");
    }
    catch
    {
    }
    capture.Enabled = false;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FileStream fs = new FileStream(pathPicCam + "pc" + txtIDStudent.Text + ".jpg", FileMode.Open,
FileAccess.ReadWrite);
        this.picWebcam.Image = System.Drawing.Image.FromStream(fs);
        fs.Close();
        this.picWebcam.Image.Tag = pathPicCam + "pc" + txtIDStudent.Text + ".jpg";
        btnOK.Enabled = true;
    }

}

public string getTxtFirstName()
{
    return txtFirstName.Text;
}
public string getTxtLastName()
{
    return txtLastName.Text;
}
public string getTxtBirthDate()
{
    return txtBirthDate.Text;
}
public string getTxtAddress()
{
    return txtAddress.Text;
}
public string getTxtIDStudent()
{
    return txtIDStudent.Text;
}
public string getTxtTel()
{
    return txtTel.Text;
}
public string getPathPicCam()
{
    return picWebcam.Image.Tag.ToString();
}
private void btnCancel_Click(object sender, EventArgs e)
{
    webCamCapture.Stop();
}
private void btnOK_Click(object sender, EventArgs e)
{
}
private void ReTake_Click(object sender, EventArgs e)
{
    webCamCapture.Start(0);
    capture.Enabled = true;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Code Database //

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    class DBFingerClass
    {
        OleDbConnection connection = new OleDbConnection();

        public bool openDB(string pathDB)
        {
            connection.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + pathDB + "";
            try
            {
                connection.Open();
            }
            catch
            {
                return false;
            }
            return true;
        }
        public bool closeDB()
        {
            if (connection.State != ConnectionState.Closed)
                connection.Close();
            return true;
        }
        public bool clearDB()
        {
            OleDbCommand cmdClear = null;
            cmdClear = new OleDbCommand("DELETE FROM enroll", connection);
            if (connection.State == ConnectionState.Open)
                cmdClear.ExecuteNonQuery();
            return true;
        }
        public bool saveTemplate(TTemplate template , string id)
        {
            OleDbCommand cmdInsert = null;
            OleDbParameter dbParamInsert = null;
            try

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    cmdInsert = new OleDbCommand("INSERT INTO enroll(ID,template) values(@ID,@template)",
connection);
    dbParamInsert = new OleDbParameter("@ID", OleDbType.VarChar);
    dbParamInsert.Value = id;
    cmdInsert.Parameters.Add(dbParamInsert);
    dbParamInsert = new OleDbParameter("@template", OleDbType.VarBinary, template.size);
    dbParamInsert.Value = template.tptData;
    cmdInsert.Parameters.Add(dbParamInsert);
    if (connection.State == ConnectionState.Open)
        cmdInsert.ExecuteNonQuery();
    return true;
}
catch
{
    return false;
}
}
public bool deleteTemplate(string id)
{
    OleDbCommand cmdDel = null;
    try
    {
        cmdDel = new OleDbCommand("DELETE from enroll where id = '" + id + "'", connection);
        if (connection.State == ConnectionState.Open)
            cmdDel.ExecuteNonQuery();
        return true;
    }
    catch
    {
        return false;
    }
}
public TTemplate getTemplateFromID(string id)
{
    OleDbCommand cmdSelect = null;
    OleDbDataReader dr = null;
    TTemplate templateFromDB = new TTemplate();
    long readedBytes;
    templateFromDB.size = 0;
    try
    {
        cmdSelect = new OleDbCommand("SELECT * FROM enroll WHERE ID = '" + id + "'", connection);
        dr = cmdSelect.ExecuteReader();
        dr.Read();
        System.Byte[] temp = new
System.Byte[(int)GrFingerXLib.GRConstants.GR_MAX_SIZE_TEMPLATE];
        readedBytes = dr.GetBytes(1, 0, temp, 0, temp.Length);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

System.Array.Copy(temp, 0, templateFromDB.tptData, 0, (int)readedBytes);
templateFromDB.size = (int)readedBytes;

}
catch
{
    dr.Close();
}
return templateFromDB;
}
public int checkIDInDB(string id)
{
    OleDbCommand cmdSelect = null;
    OleDbDataReader dr = null;
    cmdSelect = new OleDbCommand("SELECT * FROM enroll WHERE ID = '"+ id + "'", connection);
    dr = cmdSelect.ExecuteReader();
    dr.Read();
    if (dr.HasRows) //เช็คว่ามีข้อมูลไหม
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
public bool insertData(string id ,string name , string lastname,string birthdate, string address , string
tel,string pathPicfinger , string pathPiccam )
{
    OleDbCommand cmdInsert = null;
    try
    {
        cmdInsert = new OleDbCommand("INSERT INTO
user_Info(ID,name,lastname,birthdate,address,tel,picfinger,piccam) values('"+ id + "', '"+ name + "', '"+
lastname + "', '"+ birthdate + "', '"+ address + "', '"+ tel + "', '"+ pathPicfinger + "', '"+ pathPiccam + "')",
connection);
        if (connection.State == ConnectionState.Open)
            cmdInsert.ExecuteNonQuery();
        return true;
    }
    catch
    {
        return false;
    }
}
public bool deleteData(string id)
{
    OleDbCommand cmdDel = null;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try
{
    cmdDel = new OleDbCommand("DELETE from user_Info where id = " + id + "'", connection);
    if (connection.State == ConnectionState.Open)
        cmdDel.ExecuteNonQuery();

    return true;
}
catch
{
    return false;
}
}
public bool deleteListOfTimeScan(string id)
{
    OleDbCommand cmdDel = null;
    try
    {
        cmdDel = new OleDbCommand("DELETE from listOfTimeScan where id = " + id + "'",
connection);
        if (connection.State == ConnectionState.Open)
            cmdDel.ExecuteNonQuery();

        return true;
    }
    catch
    {
        return false;
    }
}
public bool updateData(string id ,string name , string lastname,string birthdate, string address ,
string tel)
{
    OleDbCommand cmdUpdate = null;
    try
    {
        cmdUpdate = new OleDbCommand("UPDATE user_Info SET [name] = " + name + ",[lastname]
= " + lastname + ",[birthdate]= " + birthdate + ",[address] = " + address + ", [tel]=" + tel + " WHERE
[ID] = " + id + "'", connection);
        if (connection.State == ConnectionState.Open)
            cmdUpdate.ExecuteNonQuery();
        return true;
    }
    catch
    {
        return false;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public bool queryData(ref string id, ref string name, ref string lastname, ref string birthdate, ref
string address, ref string tel,ref string pathFinger, ref string pathCam)
{
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;
    try
    {
        cmdQuery = new OleDbCommand("SELECT * FROM user_Info where id = '" + id + "'",
connection);
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())
        {
            name = dr.GetValue(1).ToString();
            lastname = dr.GetValue(2).ToString();
            birthdate = dr.GetValue(3).ToString();
            address = dr.GetValue(4).ToString();
            tel = dr.GetValue(5).ToString();
            pathFinger = dr.GetValue(6).ToString();
            pathCam = dr.GetValue(7).ToString();
        }
        dr.Close();
        return true;
    }
    catch
    {
        return false;
    }
}

public string[] queryTimeScan(string id)
{
    string[] listDate = new string[0];
    int i = 0;
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;
    try
    {
        cmdQuery = new OleDbCommand("SELECT * FROM listOfTimeScan where id = '" + id + "'",
connection);
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())
        {
            Array.Resize(ref listDate, i+1);
            listDate[i] = dr.GetValue(1).ToString();
            i++;
        }
    }
    catch { };
    return listDate;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
public bool insertTimeScan(string id, DateTime dt)
{
    OleDbCommand cmdInsert = null;
    try
    {
        cmdInsert = new OleDbCommand("INSERT INTO listOfTimeScan(ID,dateTimeScan) values('" + id
+ "', '" + dt + "')", connection);
        if (connection.State == ConnectionState.Open)
            cmdInsert.ExecuteNonQuery();
        return true;
    }
    catch
    {
        return false;
    }
}
public bool cmdFirstRow(ref string nowID, ref string name, ref string lastname, ref string birthdate,
ref string address, ref string tel, ref string pathFinger, ref string pathCam)
{
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;
    try
    {
        cmdQuery = new OleDbCommand("SELECT top 1 * FROM (select * from enroll order by id)",
connection);
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())
        {
            nowID = dr.GetValue(0).ToString();
            name = dr.GetValue(1).ToString();
            lastname = dr.GetValue(2).ToString();
            birthdate = dr.GetValue(3).ToString();
            address = dr.GetValue(4).ToString();
            tel = dr.GetValue(5).ToString();
            pathFinger = dr.GetValue(6).ToString();
            pathCam = dr.GetValue(7).ToString();
        }
        dr.Close();
        return true;
    }
    catch
    {
        return false;
    }
}
public bool cmdNextRow(ref string nowID, ref string name, ref string lastname, ref string birthdate,
ref string address, ref string tel, ref string pathFinger, ref string pathCam)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;
    try
    {
        cmdQuery = new OleDbCommand("SELECT TOP 1 * FROM enroll WHERE (ID > " + nowID + ")
ORDER BY ID ", connection);
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())
        {
            nowID = dr.GetValue(0).ToString();
            name = dr.GetValue(1).ToString();
            lastname = dr.GetValue(2).ToString();
            birthdate = dr.GetValue(3).ToString();
            address = dr.GetValue(4).ToString();
            tel = dr.GetValue(5).ToString();
            pathFinger = dr.GetValue(6).ToString();
            pathCam = dr.GetValue(7).ToString();
        }
        dr.Close();
        return true;
    }
    catch
    {
        return false;
    }
}

public bool cmdPrevRow(ref string nowID, ref string name, ref string lastname, ref string birthdate,
ref string address, ref string tel, ref string pathFinger, ref string pathCam)
{
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;

    try
    {
        if (nowID != null)
        {
            cmdQuery = new OleDbCommand("SELECT TOP 1 * FROM enroll WHERE (ID < " + nowID + ")
ORDER BY ID DESC ", connection);
        }
        else
        {
            cmdQuery = new OleDbCommand("SELECT top 1 * FROM (select * from enroll order by id)",
connection);
        }
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    nowID = dr.GetValue(0).ToString();
    name = dr.GetValue(1).ToString();
    lastname = dr.GetValue(2).ToString();
    birthdate = dr.GetValue(3).ToString();
    address = dr.GetValue(4).ToString();
    tel = dr.GetValue(5).ToString();
    pathFinger = dr.GetValue(6).ToString();
    pathCam = dr.GetValue(7).ToString();
}
dr.Close();
return true;
}
catch
{
    return false;
}
}
public bool cmdLastRow(ref string nowID, ref string name, ref string lastname, ref string birthdate,
ref string address, ref string tel, ref string pathFinger, ref string pathCam)
{
    OleDbCommand cmdQuery = null;
    OleDbDataReader dr = null;
    try
    {
        if (nowID != null)
        {
            cmdQuery = new OleDbCommand("SELECT top 1 * FROM (select * from enroll order by id
desc) ", connection);
        }
        else
        {
            cmdQuery = new OleDbCommand("SELECT top 1 * FROM (select * from enroll order by id)",
connection);
        }
        dr = cmdQuery.ExecuteReader();
        while (dr.Read())
        {
            nowID = dr.GetValue(0).ToString();
            name = dr.GetValue(1).ToString();
            lastname = dr.GetValue(2).ToString();
            birthdate = dr.GetValue(3).ToString();
            address = dr.GetValue(4).ToString();
            tel = dr.GetValue(5).ToString();
            pathFinger = dr.GetValue(6).ToString();
            pathCam = dr.GetValue(7).ToString();
        }
        dr.Close();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return true;
}
catch
{
    return false;
}
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Code Microcontroller //

// use Port D3 = enable
//   D1 = rs
//   D2 = rw
// and LCD type 2
#define LCD_ENABLE_PIN PIN_D3
#define LCD_RS_PIN PIN_D1
#define LCD_RW_PIN PIN_D2
#define LCD_TYPE 2
#include <lcd.c>
#include <string.h>
#include <delay.h>

void LED(unsigned int led);
void main();

////////// LED is switching//////////

void LED(unsigned int led)
{
  if(led == 1)
  {
    output_high(PIN_B7); // Led on
  }

  else
  {
    output_low(PIN_B7); // Led off
  }
}

////////// Main command //////////

void main()
{
  int busy = 0;
  char index;
  int pauseLed = 0;
  long int count = 0;
  int i = 0;
  char y[20];
  char t[50] = "Time: ";

  // set status in microcontroller

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_psp(PSP_DISABLED);
setup_spi(FALSE);
setup_wdt(WDT_OFF);
setup_timer_0(RTCC_INTERNAL);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
setup_oscillator(False);

```

```

lcd_init(); // initial value in LCD from library (lcd.c)
set_tris_b(0B01111111); // set register B -> bit7 is output

```

```

// LCD show ** Welcome ** in first line when program starts.
lcd_putc("\f ** Welcome **\n");

```

```

while (TRUE) // loop process
{
if(pauseLed == 0) // set LED is switching
{
count++;

if(count < 40000)
{
LED(0); // go to function LED and send value 0
}

else if(count > 40000)
{
LED(1); // go to function LED and send value 1
}

else if(count > 100000)
{
count = 0;
}
}
}

```

```

else if(pauseLed == 1) // set LED on continuous

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  LED(1); // go to function LED and send value 1
}

```

```

////////// start check input signal from serial port //////////

```

```

if(kbhit())
{
  index = getc(); // get a string in index

  if(index == 84) // 84 is the string of T
  {
    busy = 1;
    i = 0;
  }
}

```

```

////////// get clock from COM //////////

```

```

if(busy == 1)
{
  if(index != 47 && index != 84) // 47 and 84 are not the string of T and slash(/)
  {
    y[i] = index; // get a word in y(i) >> i=1,2,3,...
    i++;
    y[i] = '\n'; // put \n in the end of string for finish process
  }

  else if(index == 47) // 47 is the string of slash(/)
  {
    y[i] = '\n'; // put \n in the end of string
    strcpy(t,"Time: "); // copy string(t) = "Time: "
    strcat(t,y); // string(t) = string(t) + string(y)
    busy = 0;
  }
}
}

```

```

////////// Get result from COM //////////

```

```

if (busy == 0)
{

```

```

  // ID invalid!

```

```

  if (index == 65) // 65 is the string of A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    //busy = 1;    // protect interrupt another string
    pauseLed = 1;
    printf(lcd_putc, "\f%s\n", t);    // LCD shows "Time: ..." in first line.
    delay_ms(1);
    lcd_putc("\n ID Invalid!");    // LCD shows "ID Invalid!" in second line
    delay_ms(1);
    // busy = 0;    // finish the protect interrupt another string
}

else if (index == 66) // 66 is the string of B
{
    //busy = 1;    // protect interrupt another string
    pauseLed = 1;
    printf(lcd_putc, "\f%s\n", t);    // LCD shows "Time: ..." in first line.
    delay_ms(1);
    lcd_putc("\nFinger Invalid!");    // LCD shows "Finger Invalid!" in second line.
    delay_ms(1);
    //busy = 0;    // finish the protect interrupt another string
}

else if (index == 67) // 67 is the string of C
{
    //busy = 1;    // protect interrupt another string
    pauseLed = 1;
    printf(lcd_putc, "\f%s\n", t);    // LCD shows "Time: ..." in first line.
    delay_ms(1);
    lcd_putc("\n Place Finger!");    // LCD shows "Place Finger!" in second line.
    delay_ms(1);
    //busy = 0;    // finish the protect interrupt another string
}

else if (index == 68) // 68 is the string of D
{
    //busy = 1;    // protect interrupt another string
    pauseLed = 1;
    printf(lcd_putc, "\f%s\n", t);    // LCD shows "Time: ..." in first line.
    delay_ms(1);
    lcd_putc("\n Finish!");    // LCD shows "Finish!" in second line.
    delay_ms(1);
    //busy = 0;    // finish the protect interrupt another string
}

else if (index == 69) // 69 is the string of E
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//busy = 1; // protect interrupt another string
pauseLed = 0;
printf(lcd_putc, "\f%s\n", t); // LCD shows "Time: ..." in first line.
delay_ms(1);
lcd_putc("\n Welcome!"); // LCD shows "Welcome!" in second line.
delay_ms(1);
//busy = 0; // finish the protect interrupt another string
}
}
}
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้