สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรอเรียกตามบัตรคิว

WAITING QUEUE SYSTEM

โดย

นายปิติ กิตติธรรมวงศ์

นางสาวลลิดา ชาญฤทธิเสน

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิพล สิทธิชีวภาค

ผศ.เกรียงไกร วงศ์โรจนภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50216
วัน,เดือน,ปี 27 เม.ย. 2547

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรอเรียกตามบัตรคิว

WAITING QUEUE SYSTEM

โดย

นายปิติ กิตติธรรมวงศ์ 42010200

นางสาวลลิดา ชาญฤทธิเสน 42010302

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิพล สิทธิชีวภาค

ผศ.เกรียงไกร วงศ์โรจนภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบรอเรียกตามบัตรคิว**

WAITING QUEUE SYSTEM

ผู้จัดทำ

1. นายปิติ กิตติธรรมวงศ์ 42010200

2. นางสาวลลิดา ชาญฤทธิเสน 42010302

..... อาจารย์ที่ปรึกษา

(รศ.ดร.สุวิพล สิทธิชีวภาค)

..... อาจารย์ที่ปรึกษา

(ผศ.เกรียงไกร วงศ์โรจนภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรอเรียกตามบัตรคิว
WAITING QUEUE SYSTEM

โดย นายปิติ กิตติธรรมวงศ์ 42010200
นางสาวลลิดา ชาญฤทธิเสน 42010302

อาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิทธิชีวภาค
ผศ.เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

ระบบรอเรียกตามบัตรคิว เป็นการออกแบบระบบโดยประยุกต์การใช้งานไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์โดยใช้มาตรฐาน RS-232 ในส่วนไมโครคอนโทรลเลอร์จะเชื่อมต่อกับปุ่มกดของพนักงานให้บริการและจอแสดงผล เซเว่น-เซกเมนต์ (seven-segment) เพื่อแสดงหมายเลขให้ตรงกับเสียงประกาศ และในส่วนคอมพิวเตอร์เชื่อมจะเชื่อมต่อกับเครื่องพิมพ์ เพื่อพิมพ์หมายเลขบัตรคิว เมื่อมีผู้กดปุ่มเพื่อขอรับบริการที่หน้าจอแสดงผล และทำหน้าที่เล่นเสียงตามหมายเลขที่ได้รับการบริการในลำดับต่อไป

ABSTRACT

The designed automatic queuing system is composed of two major sections, which are micro-controller application, and computer, connected together via RS-232 standard. For microcontroller application, it is linked to two main devices. The first device is pressing button. This is employed to signal to the system. The other device is seven-segment display screen, which is used to show results of queuing prioritization. For computer section, it is connected with a printer for the purpose of issuing queuing ticket.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อหา		หน้าที่
บทที่ 1	บทนำ	1
บทที่ 2	หลักการและทฤษฎี	2
	2.1 คอมพิวเตอร์	2
	2.2 ไมโครคอนโทรลเลอร์ PIC16F877	4
บทที่ 3	การสร้างและการออกแบบ	24
	3.1 ในส่วนของไมโครคอนโทรลเลอร์	24
	3.2 ในส่วนของคอมพิวเตอร์	26
บทที่ 4	การทดลองและผลการทดลอง	32
	4.1 การควบคุมเครื่องพิมพ์บัตรคิว	32
	4.2 การรับข้อมูลจากเคาน์เตอร์รับบริการ	32
	4.3 ระบบเสียงพูด	34
	4.4 การควบคุมการแสดงผลตัวเลข	36
บทที่ 5	สรุปและวิจารณ์	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เนื้อหา	หน้าที่
รูปที่ 2.1 แสดงการทำงานของระบบรอเรียกตามบัตรคิว	2
รูปที่ 2.2 แสดงขาของ PIC16F87X	5
รูปที่ 2.3 แสดงบล็อคไดอาแกรมของขา RA0/RA3/RA5	7
รูปที่ 2.4 แสดงบล็อคไดอาแกรมของขา RA4/T0CKI	7
รูปที่ 2.5 แสดงแสดงบล็อคไดอาแกรมของขา RB0/RB3	8
รูปที่ 2.6 แสดงแสดงบล็อคไดอาแกรมของขา RB4/RB7	9
รูปที่ 2.7 แสดงแสดงบล็อคไดอาแกรมของขา RC0/RC2/RC5/RC7	10
รูปที่ 2.8 แสดงบล็อคไดอาแกรมของขา RC3/RC4	10
รูปที่ 2.9 แสดงบล็อคไดอาแกรมของพอร์ท D	11
รูปที่2.10 แสดงบล็อคไดอาแกรมพอร์ท E	12
รูปที่ 2.11 แสดงรูปแบบของสัญญาณอนุกรม	15
รูปที่ 2.12 แสดงโครงสร้างของรีจิสเตอร์ PIR1	15
รูปที่ 2.13 แสดงโครงสร้างของรีจิสเตอร์ RCSTA	16
รูปที่ 2.14 แสดงโครงสร้างของรีจิสเตอร์ TXSTA	17
รูปที่ 2.15 แสดงโครงสร้างของ PIE1	17
รูปที่ 2.16 แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว	19
รูปที่ 2.17 แสดงการสื่อสารแบบสองทิศทาง	20
รูปที่ 2.18 แสดงลักษณะหัวต่อแบบ D-Type	20
รูปที่ 2.19 แสดงการเชื่อมต่อแบบ Null Modem ที่นิยมใช้	21
รูปที่ 2.20 แสดงการต่อแบบ Loop Back	22
รูปที่ 2.21 RS-232 logic wave form	22
รูปที่ 3.1 แสดงผังการทำงานของไมโครคอนโทรลเลอร์	24
รูปที่ 3.2 แสดงผังการทำงานของไมโครคอนโทรลเลอร์ (ต่อ)	25
รูปที่ 3.3 แสดงการต่อไมโครคอนโทรลเลอร์เข้ากับจอแสดงผล	25
รูปที่ 3.4 แสดงการต่อไมโครคอนโทรลเลอร์เข้ากับจอแสดงผล	26
รูปที่ 3.5 แสดงหน้าจอที่ใช้เพื่อกดพิมพ์บัตรคิว	26
รูปที่ 3.6 แสดงหน้าจอของหมายเลขลำดับคิวและจำนวนคิวที่รอ	26
รูปที่ 3.7 แสดงหน้าจอของวันที่และเวลา	27
รูปที่ 3.8 แสดงหน้าจอการออกแบบที่สมบูรณ์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เนื้อหา	หน้าที่
รูปที่ 3.9 แสดงหน้าจอการแก้ไขระบบในส่วนของเคาน์เตอร์ที่ให้บริการ	27
รูปที่ 3.10 แสดงหน้าจอการแก้ไขระบบในส่วนของหมายเลขคิวก่อนหน้า	28
รูปที่ 3.11 แสดงหน้าจอการแก้ไขระบบในส่วนของที่เก็บไฟล์เสียง	28
รูปที่ 3.12 แสดงหน้าจอการแก้ไขระบบในส่วนของหมายเลขคิว	29
รูปที่ 3.13 แสดงหน้าจอการแก้ไขระบบที่เสร็จสมบูรณ์	29
รูปที่ 3.14 แสดงผังการทำงานของโปรแกรม	30
รูปที่ 3.15 แสดงผังการทำงานของโปรแกรม (ต่อ)	31
รูปที่ 4.1 แสดงหน้าจอเมื่อทำการรันโปรแกรม	32
รูปที่ 4.2 แสดงตัวอย่างของบัตรคิวที่พิมพ์ออกมา	33
รูปที่ 4.3 แสดงหน้าจอของโปรแกรมเมื่อมีการกดเข้ามาของเคาน์เตอร์รับบริการ	33
รูปที่ 4.4 แสดงกราฟเสียงพูด “หนึ่ง.”	34
รูปที่ 4.5 แสดงกราฟเสียงพูด “สอง”	34
รูปที่ 4.6 แสดงกราฟเสียงพูด “ขอเชิญคิวลำดับที่”	35
รูปที่ 4.7 แสดงกราฟเสียงพูด “ที่ช่องบริการหมายเลข”	35
รูปที่ 4.8 แสดงกราฟเสียงพูด “ครับ”	36
รูปที่ 4.9 แสดงกราฟเสียงพูด “ขอเชิญคิวลำดับที่หนึ่งที่ช่องบริการหมายเลขสองครับ”	36
รูปที่ 4.10 แสดงหน้าจอแสดงผล	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เนื้อหา	หน้าที่
ตารางที่ 2.1 แสดงรายละเอียดของแต่ละขา	12
ตารางที่ 2.2 D-Type 9 Pin and D-Type 25 Pin Connectors	20
ตารางที่ 2.3 แสดงการทำงานของแต่ละขา	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันนี้ หน่วยงานหรือองค์กรต่างๆ ที่มีหน้าที่เกี่ยวข้องกับการให้บริการแก่บุคคลจำนวนมาก เช่น การรอคิวจ่ายเงินค่าไฟฟ้า ค่าโทรศัพท์ ค่าน้ำประปา ทำธุรกรรมของธนาคาร หรือที่ทำการไปรษณีย์ เป็นต้น จะมีเครื่องอำนวยความสะดวกในการจัดลำดับการให้บริการอย่างเป็นระเบียบ โดยส่วนมากจะใช้การประมวลผลโดยคอมพิวเตอร์ จะมีส่วนแสดงผล เสียงประกาศ และเครื่องพิมพ์เพื่อพิมพ์หมายเลขในการรอรับบริการ

ระบบจัดการลำดับนั้น ขั้นตอนพื้นฐานในการใช้อันดับแรกคือ ผู้ขอรับบริการแจ้งความจำนงในการขอรับบริการด้วยการกดปุ่ม และเครื่องพิมพ์จะทำการพิมพ์หมายเลขในการรอรับบริการ และเมื่อถึงหมายเลขนั้น ก็จะมีเสียงประกาศเรียกหมายเลขนั้น

จากเดิมที่จะต้องยืนรอรับบริการทำให้เกิดแถวยาวเกะกะสถานที่ และไม่สะดวกต่อผู้รอรับบริการต่อมาได้ใช้ระบบเรียกตามบัตรคิวนี้ จะเห็นได้ว่าสร้างความสะดวกสบายในการทำธุรกรรมต่างๆอย่างเห็นได้ชัด

ในชิ้นงานของระบบรอเรียกตามบัตรคิวที่จัดทำขึ้นนี้แบ่งการควบคุมเป็น 2 ส่วนใหญ่ๆ คือ

1.1 คอมพิวเตอร์

โดยการเขียนโปรแกรม Visual Basic เพื่อทำการประมวลผลเพื่อแสดงภาพทางหน้าจอ ประกาศเสียง และติดต่อกับไมโครคอนโทรลเลอร์

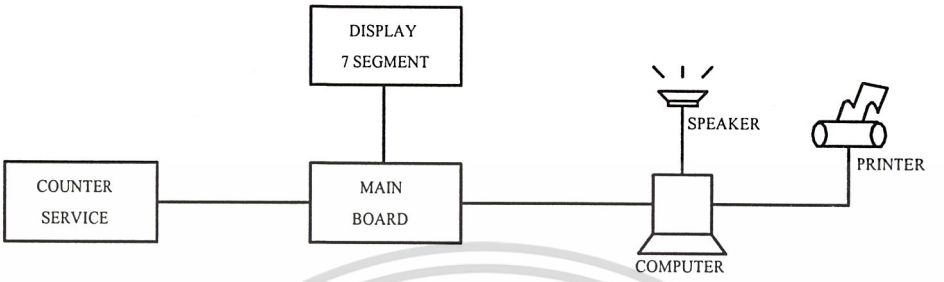
1.2 ไมโครคอนโทรลเลอร์

โดยการเขียนโปรแกรม Picbasicpro เข้าไปในไมโครคอนโทรลเลอร์ เพื่อติดต่อกับคอมพิวเตอร์ แสดงผลทางเซเว่น-เซกเมนต์ และรับค่าจากปุ่มกดของพนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎี

หลักการทำงานของเครื่องจัดลำดับอัตโนมัติ เป็นดังต่อไปนี้



รูปที่ 2.1 แสดงการทำงานของระบบรอเรียกตามบัตรคิว

2.1.COMPUTER

ในส่วนนี้จะทำการเขียนโปรแกรม Visual Basic ติดต่อระหว่างส่วนประมวลผลหลักที่มีไมโครคอนโทรลเลอร์ และเครื่องพิมพ์ จะทำหน้าที่รับข้อมูลจากส่วนประมวลผลหลัก และสั่งงานให้เครื่องพิมพ์พิมพ์ตามที่ต้องการ และอีกหน้าที่หนึ่งคือเป็นหน้าจอรับอินพุท(input)จากลูกค้า

การเขียนโปรแกรม Visual Basic ติดต่อกับพอร์ทอนุกรมสามารถทำได้ 2 วิธี คือ

1. การติดต่อแบบอินเตอร์รัพท์

ขบวนการอินเตอร์รัพท์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพท์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2. การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่างเทอร์มินอลกับซีพียูกรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบคีย์บอร์ดว่ามีข้อมูลส่งมาหรือไม่ โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทางคีย์บอร์ด การที่ซีพียูส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า “เว็ทโพล”(Wet Poll) ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้นเราเลี่ยงไปใช้เทคนิค การโพลแบบ “ราวน์โรบิน”(Round Robin) แทน แต่ในVBเราจะใช้การตรวจสอบข้อมูลที่มาจากพอร์ทอนุกรมตลอดโดยจะใช้คอนโทรลไทม์เมอร์เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มีเหตุการณ์ (Event)ที่ใช้เพียงเหตุการณ์เดียวเท่านั้น คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพท์ การเขียนโปรแกรมติดต่อพอร์ทอนุกรมแบบธรรมดาจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ComEvent, ComEvReceive, ComEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ

องค์ประกอบในการใช้ MSComm

การตั้งค่าติดต่อกับพอร์ท

- **ComPort** คือ กำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)รายละเอียดดูในเมนูด้านซ้าย Serial Port Detail
- **Setting** คือ กำหนดอัตรา Baud, Parity, Data(จำนวนบิต), Stop ตัวอย่าง 1200, n, 8, 1 เป็นต้น
- **HandShaking** คือ กำหนดได้ 4 แบบ 1.comNone 2.comXonXoff 3. comRTS 4.comTRSXonXoff

การใช้ Buffer ในการรับส่งข้อมูล

- **InBuffersize** คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา
- **OutBuffersize** คือ การกำหนด Buffer ในการส่งข้อมูลออกไป
- **Rthreshold** คือ การที่กำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- **Sthreshold** คือ การที่กำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- **Inputlen** คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- **EOFEnable** คือ การที่บอกว่าสิ้นสุดของไฟล์(EOF) End of File

ด้านฮาร์ดแวร์

- **ParityReplace** คือ ค่าของคาแลกเตอร์ที่จะแทนเมื่อเกิด Parity Error
- **NullDiscard** คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- **RTSEnable** คือ ทำให้มีสัญญาณ RTS (Request To Send)
- **DTSEnable** คือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

การกำหนดคุณสมบัติของ **MSComm Control** ให้สามารถติดต่อกับพอร์ตได้

1. **Property** ชื่อ **CommPort** คือ การเลือกคอมพอร์ตที่เราจะต่อใช้งาน

   ตัวอย่าง MSComm1.CommPort=1 ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมฯ

2. **Property** ชื่อ **Settings** คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด (Baud Rate) ของอุปกรณ์ที่จะทำการติดต่อด้วยว่าเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings = "Baud (อัตราการรับส่งข้อมูล), Parity (ถ้าไม่ใช้ใส่ N, จำนวนบิตข้อมูล, บิตสต๊อบ" ตัวอย่าง MSComm1.Settings = "1200, N, 8, 1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.Property ชื่อ InputLen คือ การกำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัพเฟอร์

ตัวอย่าง MSComm1.InputLen = 1

4. Property ชื่อ PortOpen คือ การจะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด = True ถ้าปิด = False

ตัวอย่าง MSComm1.PortOpen = True

5. Property ชื่อ Rthreshold คือ การทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล(Comport)จะทำให้เกิดCommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold = 1

วิธีของการรับส่งข้อมูลจาก Serial Port

จากวิธีเขียนโค๊ดด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

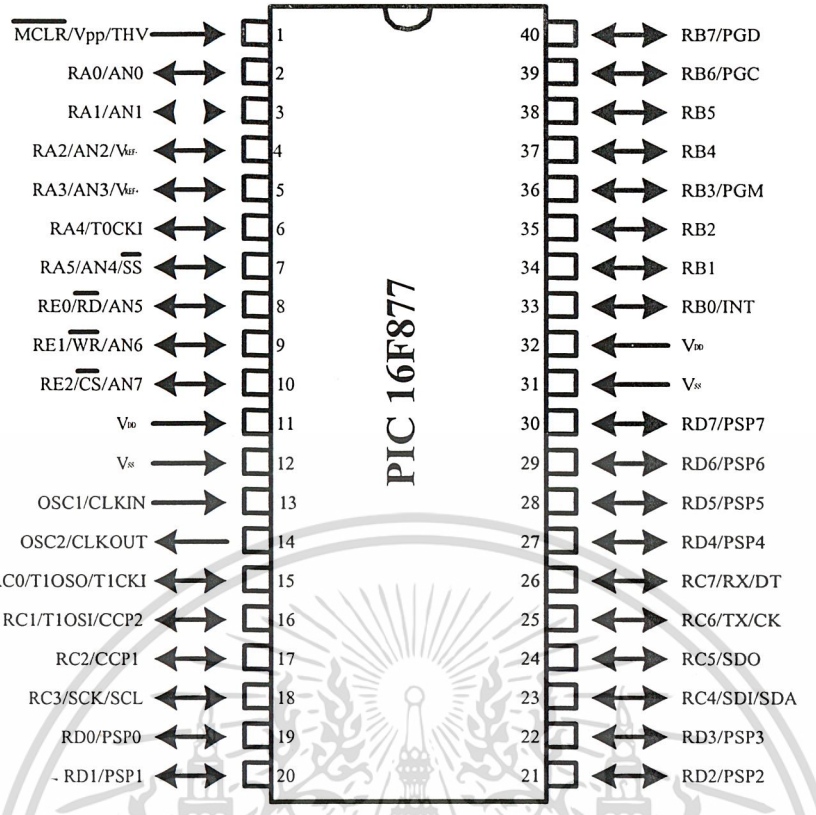
Output = ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input = เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS-232

2.2 ไมโครคอนโทรลเลอร์ PIC16F877

PIC เป็น Microcontroller ชนิดหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่งลักษณะเด่นของตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัวของอุปกรณ์เองไม่ว่าจะเป็น PROGRAM MEMROY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้มันเหมือนกับเป็นซีพียูตัวหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงขาของ PIC16F87X

PIC16F87X

คุณสมบัติทางเทคนิคของ PIC16F87X

- ซีพียูเป็นแบบ RISC (Reduce Instruction-Set Computer) มีคำสั่งใช้งานเพียง 35 คำสั่ง
- ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20 MHz (สูงสุด)
- ขนาดหน่วยความจำโปรแกรม (FLASH Program Memory) 112 กิโลเวิร์ด
- หน่วยความจำแรมข้อมูล (Data Memory or RAM) 2,944 ไบต์
- หน่วยความจำข้อมูลอีอีพรอม (EEPROM data memory) 2,048 ไบต์
- ตอบสนองแหล่งกำเนิดอินเตอร์รัปท์ได้ 14 แหล่ง
- มีสแต็ก 8 ระดับ
- มีวงจรเพาเวอร์ออนรีเซต (Power-on Reset or POR), เพาเวอร์อัปไทม์เมอร์ (Power-up Tymer : PWRT) และ ออสซิลเลเตอร์สตาร์ทอัปไทม์เมอร์ (Oscillator Start-up Timer : OST)
- มีวอทช์ด็อกไทม์เมอร์ (Watchdog Timer : WDT) ที่มีวงจรออสซิลเลเตอร์ในตัว ทำให้มีความเชื่อถือในการทำงานสูง
- เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล และเลือกระดับการป้องกันได้
- สามารถเลือกใช้วงจรกำเนิดสัญญาณนาฬิกาได้
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

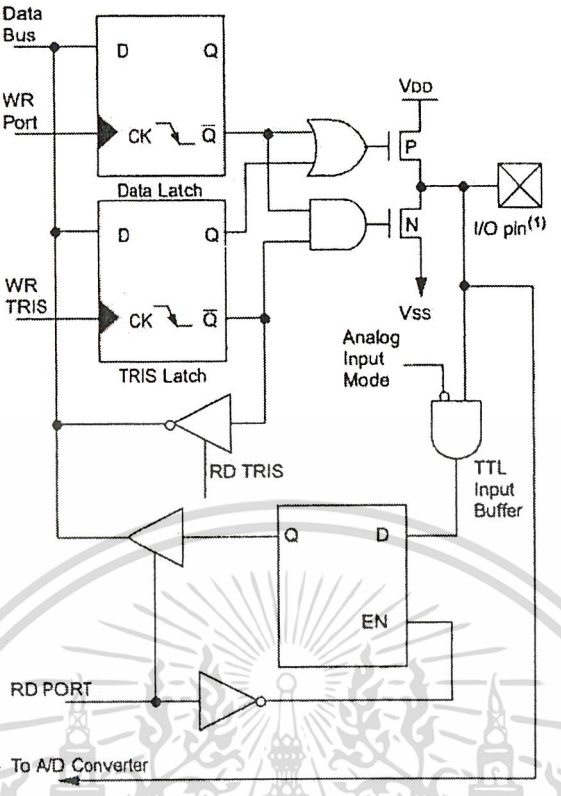
- ไฟเลี้ยง +2 ถึง +5.5V
- กระแสซิงก์และซอร์สของพอร์ท 25mA
- ไทม์เมอร์ 3 ตัว (ไทม์เมอร์ 0, ไทม์เมอร์ 1, ไทม์เมอร์ 2)
- มีโมดูล CCP (Capture/Compare/PWM) 1 ชุด
- มีโมดูลเปรียบเทียบแรงดันอนาลอก 2 ชุด
- มีโมดูลสร้างแรงดันอ้างอิง
- มีโมดูลสื่อสารข้อมูลอนุกรม USART
- มีวงจรตรวจจับระดับแรงดันไฟเลี้ยงหรือบราวน์เอาท์ดีเท็กชั่น (Brown-out Detection) เพื่อสร้างสัญญาณรีเซตซีพียูหรือเรียกว่า บราวน์เอาท์รีเซต (Brown-out Reset : BOR)
- การใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด
  - น้อยกว่า 2 mA ที่ +5V และสัญญาณนาฬิกา 4MHz
  - 20 μA ที่ +3V และสัญญาณนาฬิกา 32 kHz
  - น้อยกว่า 1 μA ในโหมดประหยัดพลังงานหรือสแตนด์บายที่ไฟเลี้ยง +3V

รายละเอียดของพอร์ทแต่ละพอร์ท

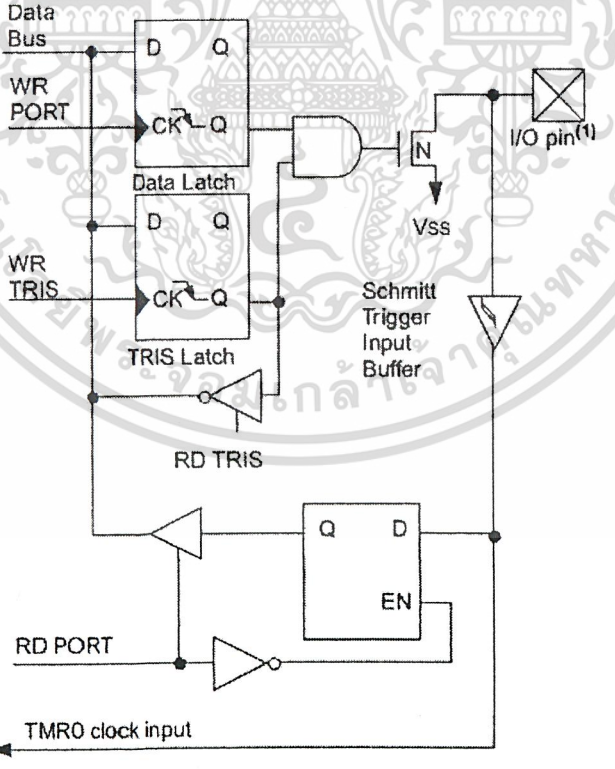
PORTA มีขนาด 6 บิต ซึ่งเป็นพอร์มที่เป็นได้ทั้งอินพุทและเอาท์พุท โดยต้องเลือกแบบใดแบบหนึ่ง สามารถเลือกได้จากรีจิสเตอร์ที่มีชื่อว่า TRISA ซึ่งถ้า TRISA bit ถูกเซตเป็น '1' PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็ทำงานเป็นอินพุท(ทำให้พอร์ทนั้นอยู่ในสถานะที่มีความต้านทางสูง) ส่วนถ้า TRISA bit ถูกเซตเป็น '0' PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็จะทำงานเป็นเอาท์พุท(พอร์ทจะอยู่ในสถานะ output latch) การอ่านค่า PORTA register คือการอ่านค่าสถานะของขา PORTA ในขณะนั้น ส่วนการเขียนค่าไปยัง PORTA คือการเขียนไปยังแลทช์ (Latch) ของพอร์ทลักษณะการเขียนจะเป็นแบบ read-modify-write operations ซึ่งหมายความว่า ในการเขียนไปยังพอร์ทจะเริ่มด้วยการอ่านค่าพอร์ทนั้นมาก่อนแล้วทำการเปลี่ยนแปลงค่า จากนั้นก็ทำการเขียนกลับไปยังพอร์ทแลทช์ อีกครั้งหนึ่ง ขา RA4 จะ multiplexed กับ Timer0 module clock input ซึ่งจะเรียกรวมๆ ว่า RA4/T0CKI โดยที่ ขา RA4/T0CKI จะเป็นลักษณะอินพุทแบบ ชมิตต์ทริกเกอร์ (Schmitt Trigger) และเอาท์พุทแบบเดรนเปิด (Open Drain)

Port RA ทั้งหมด จะมี TTL input level และ มีเอาท์พุต แบบ full CMOS drivers ส่วน PORTA ขาอื่นๆ จะมัลติเพล็กซ์กับอนาลอกอินพุต และ Vref ของ A/D input ซึ่งการกำหนดการทำงานของแต่ละขาสามารถเลือกได้โดยเคลียร์หรือ เซตคอนโทรลบิตใน ADCON1 register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงบล็อคไดอาแกรมของขา RA0/RA3/RA5



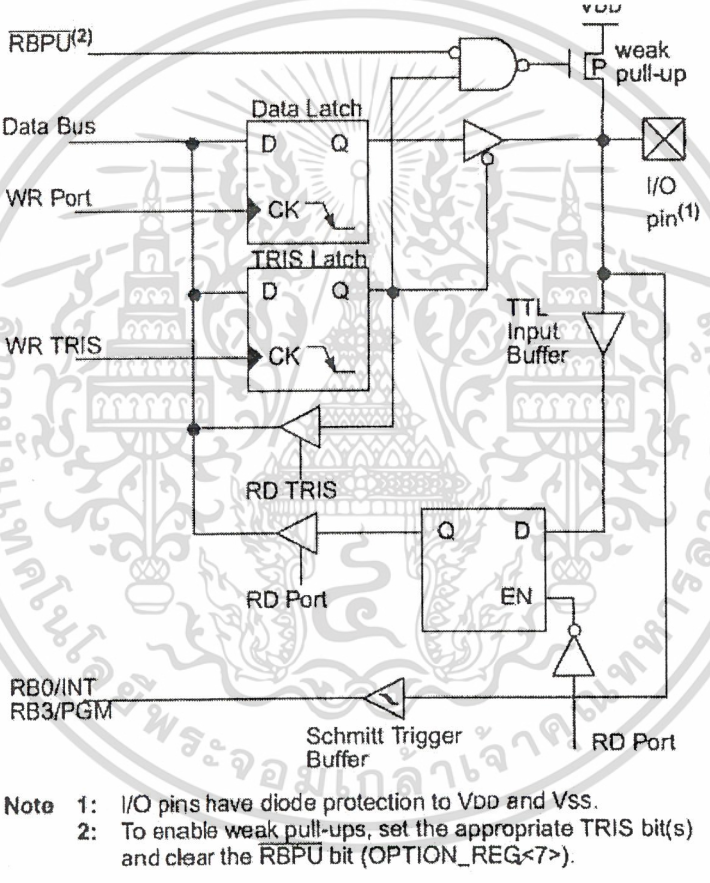
Note 1: I/O pin has protection diodes to Vss only.

รูปที่ 2.4 แสดงบล็อคไดอาแกรมของขา RA4/T0CKI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

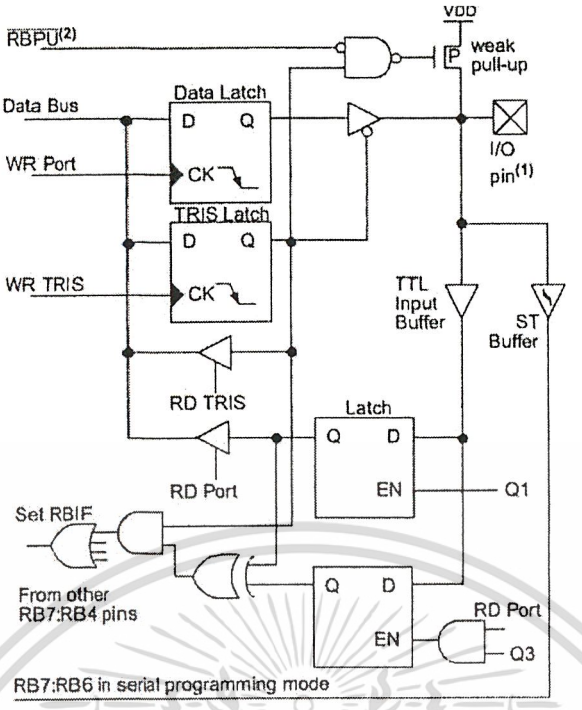
PORTB เป็นลักษณะแบบพอร์ทแบบสองทิศทาง ซึ่งรีจิสเตอร์จะเป็นตัวกำหนดว่าพอร์ทใดจะเป็นแบบอินพุทหรือเอาท์พุทซึ่งจะถูกกำหนดโดย TRISB register ถ้าเซตTRISB bit ใด (=1) PORTB ที่บิตนั้นก็จะเป็นอินพุท ถ้าเคลียร์ TRISB bit ใด (=0) PORTB ที่บิตนั้นก็จะเป็นเอาท์พุท ขาสามขาของPORTB จะ multiplexed กับ Low Voltage Programming function ซึ่งได้แก่ RB3/PGM, RB6/PGC และ RB7/PGD

สำหรับ PORTB นั้น ขา RB4-RB7 จะมีลักษณะเพิ่มเติมก็คือ การกำหนดให้เกิดอินเตอร์รัปเมื่อเกิดการเปลี่ยนแปลงของสถานะของสัญญาณไฟฟ้าที่ขา RB4-RB7 (โดยถ้าขาใดขาหนึ่งเกิดเปลี่ยนสถานะก็จะทำให้เกิด RB Port Change Interrupt ขึ้น ซึ่งจะทำให้ RBIF (INTCON.0) flag ถูกเซตโดยที่อินเตอร์รัปประเภทนี้สามารถทำการ “wake” microcontroller จากสถานะ sleep mode ได้



รูปที่ 2.5 แสดงแสดงบล็อคไดอาแกรมของขา RB0/RB3

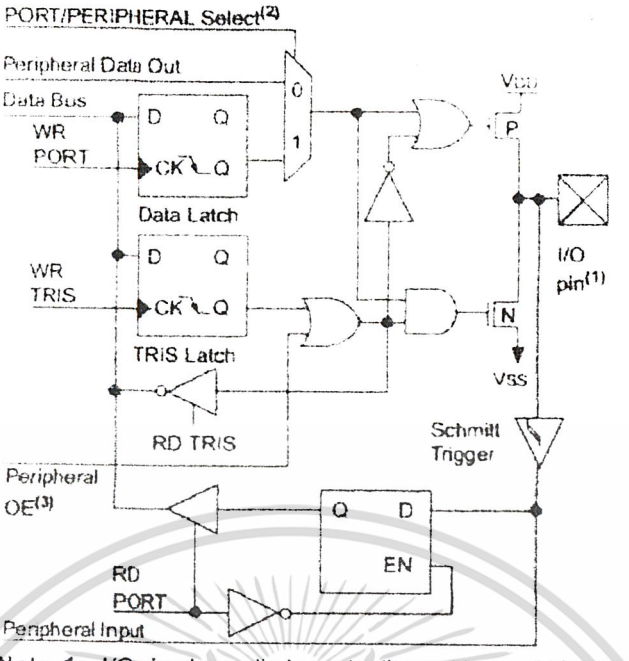
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงแสดงบล็อคไดอาแกรมของขา RB4/RB7

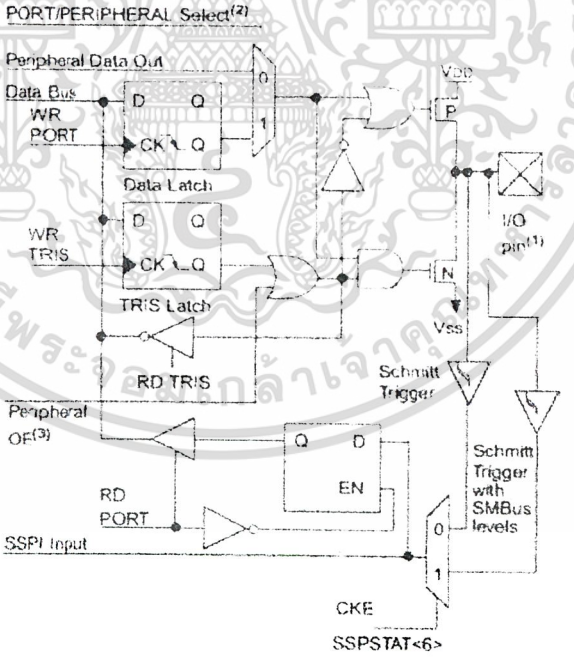
PORTC เป็นลักษณะแบบพอร์ทสองทิศทาง ซึ่งรีจิสเตอร์ที่จะเป็นตัวกำหนดว่าพอร์ทใดจะเป็นแบบอินพุทหรือเอาท์พุทนั้นจะถูกกำหนดโดย TRISC register ถ้าเซตTRISC bit ใด (=1) PORTB ที่บิตนั้นก็จะเป็นอินพุท ถ้าเคลียร์ TRISC bit ใด (=0) PORTC ที่บิตนั้นก็จะเป็นเอาท์พุท ที่ PORTC จะมีคุณสมบัติเพิ่มเติม เช่น IIC, UART, SPI, PWM, CAPTURE ขึ้นอยู่กับการเลือกใช้งาน โดยเมื่อเราทำการเอนเนเบิล (Enable) คุณสมบัติเพิ่มเติมต่างๆ ที่ PORTC เราต้องระวังในเรื่องของการตั้งค่า TRISC ของแต่ละขาของ PORTC เพราะในการเอนเนเบิลคุณสมบัติบางตัวที่อยู่ที่ PORTC (เช่น UART) ตัวมันเองก็จะทำการเปลี่ยนบิต TRISC โดยอัตโนมัติ ดังนั้นไม่ควรที่จะตั้งค่า TRISC โดยตรงกับขาใดของ PORTC ที่ทำการเอนเนเบิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

รูปที่ 2.7 แสดงแสดงบล็อคไดอาแกรมของขา RC0/RC2/RC5/RC7

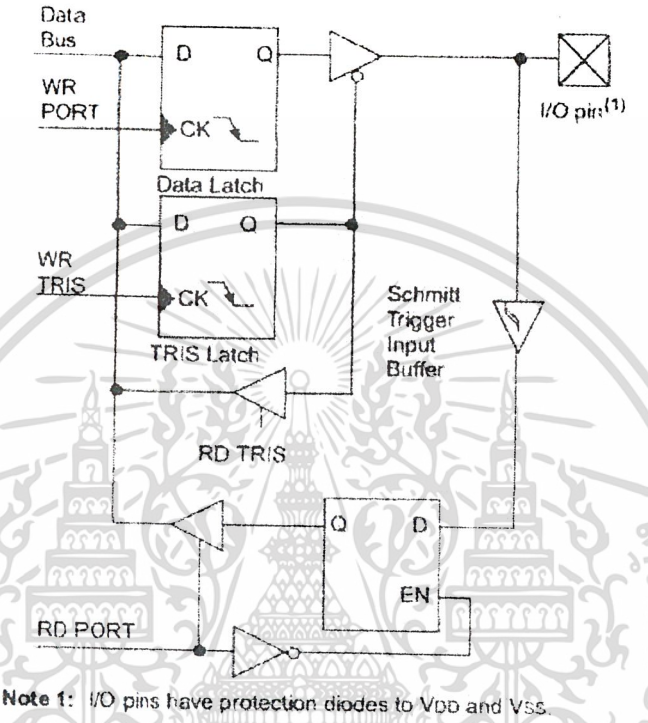


Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

รูปที่ 2.8 แสดงบล็อคไดอาแกรมของขา RC3/RC4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORTD จะเป็นพอร์ทขนาด 8 bits ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของพอร์ทให้เป็นอินพุทหรือเอาท์พุทได้โดยอิสระจากกัน PORTD สามารถที่จะทำตัวเป็นพอร์ทแบบขนาน (parallel slave port) ได้อีกด้วย โดยทำได้โดยการเซตPSPMODE bit (TRISE <4>) ซึ่งในโหมดนี้ บัฟเฟอร์ภายในจะกลายเป็นแบบ TTL

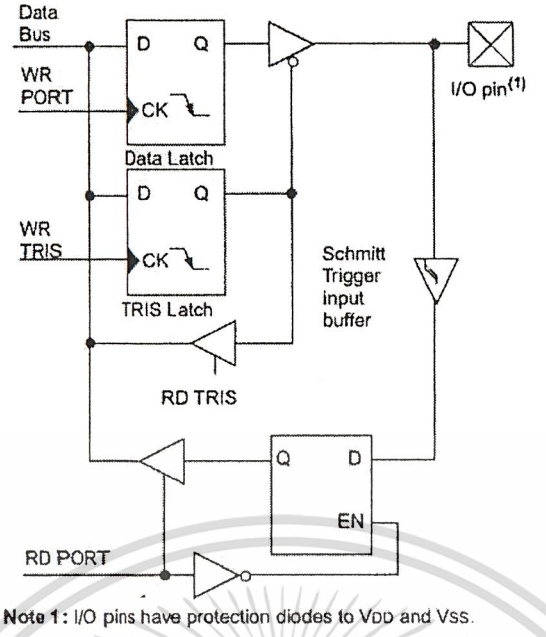


รูปที่ 2.9 แสดงบล็อคไดอาแกรมของพอร์ท D

PORTE จะมีทั้งหมด 3 ขา คือ RE0/(RD\)/AN5, RE1/(WR\)/AN6 และ RE2/(CS\)/AN7 ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของพอร์ทให้เป็นอินพุทหรือเอาท์พุต สามารถกลายเป็นคอนโทรลอินพุทสำหรับพอร์ทของไมโครโปรเซสเซอร์ได้ เมื่อ ทำการเซต PSPMODE(TRISE<4>) bit ข้อควรระวังเมื่ออยู่ในโหมดนี้ก็คือ ต้องตรวจดูให้ดีว่า TRISE ตั้งแต่บิต 0-2 ถูกเซต (อยู่ในสถานะอินพุท) และต้องแน่ใจว่า ADCON1 ถูกเซตให้อยู่ในโหมด digital I/O ซึ่งในโหมดนี้อินพุทบัฟเฟอร์จะเป็น TTL

PORT E จะมีลักษณะคือ จะมัลติเพล็กซ์กับอนาลอกอินพุท โดยเมื่อ PORTE ถูกเซตเป็นอนาลอกอินพุทแล้ว ขาเหล่านี้เมื่อทำการอ่านค่าจะมีค่าเป็น 0 ส่วน TRISE ซึ่งเป็นคอนโทรลรีจิสเตอร์นั้นจะต้องเซตให้เป็นอินพุทเมื่อเซตให้อยู่ในโหมดอนาลอกอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.10 แสดงบล็อคไดอาแกรมพอร์ท E

ตารางที่ 2.1 แสดงรายละเอียดของแต่ละขา

Pin Name	Description
OSC1/CLKIN	ขาต่อคริสตอล, อินพุตสัญญาณนาฬิกาหลัก
OSC2/CLKOUT	ขาต่อคริสตอล, เอาท์พุตสัญญาณนาฬิกาหลัก เมื่อทำงานในโหมด RC มีความที่เท่ากับ $1/4$ เท่าของความถี่ที่ขา OSC1
$\overline{\text{MCLR}}$ /Vpp	เป็นขารีเซตหลักของอุปกรณ์ จะทำการรีเซตอุปกรณ์เมื่อมีสถานะเป็นแอคทีฟโลว์ (Active low)
	PORT A เป็น I/O พอร์ท 2 ทิศทาง
RA0/AN0	- ขาพอร์ท RA0 - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 0
RA1/AN1	- ขาพอร์ท RA1 - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 1
RA2/AN2/ V_{REF-}	- ขาพอร์ท RA2 - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 2 - เอาท์พุตแรงดันอ้างอิงลบ
RA3/AN3/ V_{REF+}	- ขาพอร์ท RA3 - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	- เอาท์พุตแรงดันอ้างอิงบวก
RA4/T0CKI	- ขาพอร์ท RA4 กรณีใช้งานเป็นพอร์ทเอาท์พุตโครงสร้างเป็นแบบเดรนเปิด
	- อินพุตสัญญาณนาฬิกาของไทเมอร์ 0
RA5/$\overline{SS}$/AN4	- ขาพอร์ท RA5
	- อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 4
	- ขาเลือกเสลฟ (Slave select)สำหรับการส่งแบบซิงโครนัสในพอร์ทอนุกรม
RB0/INT	- ขาพอร์ท RB0
	- อินพุตรับสัญญาณอินเตอร์รัปจากภายนอก
RB1	- ขาพอร์ท RB1
RB2	- ขาพอร์ท RB2
RB3/PGM	- ขาพอร์ท RB3
	- สามารถกำเนิดสัญญาณอินเตอร์รัปอันเนื่องมาจากมีการเปลี่ยนแปลงลอจิดที่ขานี้ ในกรณีเอ็นนาเบิลไว้
	- อินพุตรับแรงดัน +5V ในกรณีที่ต้องการโปรแกรมไมโครคอนโทรลเลอร์ด้วยแรงดันต่ำ
RB4	- ขาพอร์ท RB4
	- สามารถกำเนิดสัญญาณอินเตอร์รัปอันเนื่องมาจากมีการเปลี่ยนแปลงลอจิดที่ขานี้ ในกรณีเอ็นนาเบิลไว้
RB5	- ขาพอร์ท RB5
	- สามารถกำเนิดสัญญาณอินเตอร์รัปอันเนื่องมาจากมีการเปลี่ยนแปลงลอจิดที่ขานี้ ในกรณีเอ็นนาเบิลไว้
RB6/PGC	- ขาพอร์ท RB6
	- สามารถกำเนิดสัญญาณอินเตอร์รัปอันเนื่องมาจากมีการเปลี่ยนแปลงลอจิดที่ขานี้ ในกรณีเอ็นนาเบิลไว้
	- ขาสัญญาณนาฬิกาของการโปรแกรมข้อมูลในโหมดอนุกรม
RB7/PGD	- ขาพอร์ท RB7
	- สามารถกำเนิดสัญญาณอินเตอร์รัปอันเนื่องมาจากมีการเปลี่ยนแปลงลอจิดที่ขานี้ ในกรณีเอ็นนาเบิลไว้
	- ขาสัญญาณข้อมูลของการโปรแกรมข้อมูลในโหมดอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	PORT C เป็น I/O พอร์ท 2 ทิศทาง
RC0/T1OSO/T1CKI	- ขาพอร์ท RC0
	- ขาเอาท์พุตสัญญาณออสซิเลเตอร์ของไทม์เมอร์ 1
	- อินพุตรับสัญญาณนาฬิกาสำหรับไทม์เมอร์ 1
RC1/T1OSI/CCP2	- ขาพอร์ท RC1
	- ขาอินพุตสัญญาณออสซิลเลเตอร์ของไทม์เมอร์ 1
	- ขาอินพุต/เอาท์พุตของโมดูล CCP
RC2/CCP1	- ขาพอร์ท RC2
	- ขาอินพุต/เอาท์พุตของโมดูล CCP
RC3/SCK/SCL	-RC3 สามารถเป็นได้ทั้งอินพุตและเอาพุตของซิงโครนัส
RC4/SDI/SDA	สำหรับ SPI
RC5/SDO	-RC4สามารถเป็นข้อมูล SPI
RC6/TX/CK	-RC5 เป็น SPI DATA OUT
	- ขา RC6
	- ขาส่งข้อมูลของวงจรสื่อสารข้อมูลอนุกรม USART
	- ขาติดต่อสัญญาณนาฬิกาซิงโครนัส
RC7/RX/DT	- ขา RC7
	- ขารับข้อมูลของวงจรสื่อสารข้อมูลอนุกรม USART
	- ขาติดต่อสัญญาณข้อมูลซิงโครนัส
	PORT D เป็น I/O พอร์ท 2 ทิศทางหรือ พอร์ตสเลฟแบบขนาน (parallel slave port) เมื่อทำการเชื่อมต่อกับบัสของไมโครโปรเซสเซอร์
RD0/PSP0	- ขาพอร์ท RD0
RD1/PSP1	- ขาพอร์ท RD1
RD2/PSP2	- ขาพอร์ท RD2
RD3/PSP3	- ขาพอร์ท RD3
RD4/PSP4	- ขาพอร์ท RD4
RD5/PSP5	- ขาพอร์ท RD5
RD6/PSP6	- ขาพอร์ท RD6
RD7/PSP7	- ขาพอร์ท RD7
	PORT E เป็น I/O พอร์ท 2 ทิศทาง
RE0/ $\overline{RD}$ /AN5	- ขาพอร์ท RE0
	- ขาควบคุมการเขียนข้อมูลสำหรับสเลฟแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

RE1/ $\overline{WR}$ /AN6	(parallel slave) - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 5 - ขาพอร์ท RE1 - ขาควบคุมการเขียนข้อมูลสำหรับสเลฟแบบขนาน
RE2/ $\overline{CS}$ /AN7	- อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 6 - ขาพอร์ท RE2 - ขาควบคุมการเขียนข้อมูลสำหรับสเลฟแบบขนาน - อินพุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 7
V_{SS}	- ขาต่อกราวด์
V_{DD}	- ขาต่อไฟเลี้ยงบวก

รูปแบบสัญญาณอนุกรม

รูปที่ 2.11 แสดงรูปแบบของสัญญาณอนุกรม

ข้อมูลแบบอนุกรมจะมีลักษณะการส่งด้วยการเริ่มต้นด้วยบิตเริ่มต้น (Start Bit) ซึ่งจะมีค่าเป็น '0' แล้วตามด้วยบิต 0 จนถึงบิต 7 (ในที่นี้บิต 0 คือบิตซ้ายสุดของไบท์) จากนั้นก็จะปิดท้ายด้วยบิตสิ้นสุด (Stop Bit) ซึ่งจะมีค่าเป็น '1'

- ในบางครั้งอาจจะมีพาริตี้บิต (Parity Bit) ซึ่งจะเป็นบิตที่เป็นตัวตรวจสอบความผิดพลาดของข้อมูลขนาด 1 ไบท์ ที่เราทำการส่งซึ่งจะอยู่ระหว่างบิต 7 และบิตสิ้นสุดแต่ในโครงงานนี้เราจะไม่ใช้พาริตี้บิต
- บิตสิ้นสุดสามารถมีขนาด 1, 1+1/2 หรือ 2 บิต ก็ได้แล้วแต่กำหนด ซึ่งในโครงงานนี้เราให้มีขนาด 1 bit

ความเร็วในการส่ง

เราใช้ความเร็วในการส่งขนาด 19200 bps (bps ย่อมาจาก bit per second) พูดกันง่ายๆ ก็คือ ส่งข้อมูลขนาด 19200 บิต ใน 1 วินาที ซึ่งก็หมายความได้อีกอย่างหนึ่งว่าสัญญาณขนาด 1 บิต มีความกว้าง (1/19200) = 0.052 msec

รีจิสเตอร์ของ PIC ที่เกี่ยวข้องกับการติดต่อสื่อสารแบบอนุกรม

เนื่องจากเราจะทำการส่งแบบอะซิงโครนัสดังนั้นเราจะพิจารณารีจิสเตอร์ต่างๆ ในกรณีแบบอะซิงโครนัส

1. PIR1 register บิตที่เกี่ยวข้องก็คือ RCIF และ TXIF

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
BIT 7							BIT 0

รูปที่ 2.12 แสดงโครงสร้างของรีจิสเตอร์ PIR1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ RCIF = 1 หมายถึงบัฟเฟอร์ที่ทำหน้าที่รับข้อมูลอนุกรมนั้นเต็ม

เมื่อ RCIF = 0 หมายถึงบัฟเฟอร์ที่ทำหน้าที่รับข้อมูลอนุกรมนั้นว่าง

เมื่อ TXIF = 1 หมายถึงบัฟเฟอร์ที่ทำหน้าที่ส่งข้อมูลอนุกรมนั้นเต็ม

เมื่อ TXIF = 0 หมายถึงบัฟเฟอร์ที่ทำหน้าที่ส่งข้อมูลอนุกรมนั้นว่าง

2. RCSTA register เป็น register ที่เกี่ยวข้องกับการรับข้อมูล

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
BIT 7							BIT 0

รูปที่ 2.13 แสดงโครงสร้างของรีจิสเตอร์ RCSTA

Bit7 SPEN

= 1 หมายถึง อนุญาตให้ใช้งานพอร์ทอนุกรมได้

= 0 หมายถึง ไม่อนุญาตให้ใช้งานพอร์ทอนุกรม

Bit6 RX9

= 1 หมายถึง ขนาดข้อมูล 1 ไบท์ ที่รับมีขนาด 9 บิต

= 0 หมายถึง ขนาดข้อมูล 1 ไบท์ ที่รับมีขนาด 8 บิต

Bit5 SREN

= เราไม่สนใจบิตนี้ในกรณีที่ใช้แบบอะซิงโครนัส

Bit4 CREN

= 1 หมายถึง กำหนดให้มีการรับข้อมูลที่เข้ามาอย่างต่อเนื่อง

= 0 หมายถึง ไม่ให้มีการรับข้อมูลที่เข้ามาอย่างต่อเนื่อง

Bit3 ADDEN

= เนื่องจากเราส่งข้อมูลแบบ 8 บิต ดังนั้นเราจะไม่สนใจ บิตนี้

Bit2 FERR

= 1 หมายถึง มีการผิดพลาดของเฟรมเกิดขึ้นกับข้อมูลที่ได้รับ

= 0 หมายถึง ไม่มีการผิดพลาดของเฟรมเกิดขึ้นกับข้อมูลที่ได้รับ

Bit1 OERR

= 1 หมายถึง มี Overrun error เกิดขึ้นกับข้อมูลที่ได้รับ

= 0 หมายถึง ไม่มี Overrun error เกิดขึ้นกับข้อมูลที่ได้รับ

Bit0 RX9D

= เป็นบิตที่ 9 ของข้อมูลในกรณีที่ข้อมูลเป็นแบบ 9 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3. RCREG register คือข้อมูลขนาด 1 ไบท์ ที่รับได้จากภายนอกในขณะนั้น

4. TXSTA register เป็น register ที่เกี่ยวข้องกับการส่งข้อมูล

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
BIT 7							BIT 0

รูปที่ 2.14 แสดงโครงสร้างของรีจิสเตอร์ TXSTA

Bit7 CSRC

= เราไม่สนใจบิตนี้ในกรณีที่ใช้แบบอะซิงโครนัส

Bit6 TX9

= 1 หมายถึง กำหนดให้เป็นการส่งแบบ 9 บิต

= 0 หมายถึง กำหนดให้เป็นการส่งแบบ 8 บิต

Bit5 TXEN

= 1 หมายถึง อนุญาตให้มีการส่งข้อมูล

= 0 หมายถึง ไม่อนุญาตให้มีการส่งข้อมูล

Bit4 SYNC

= 1 หมายถึง เป็นการส่งแบบซิงโครนัส

= 0 หมายถึง เป็นการส่งแบบอะซิงโครนัส

Bit3 ไม่มีการใช้งาน

Bit2 BRGH ในโหมดอะซิงโครนัสหมายถึง

= 1 หมายถึง เป็นแบบความเร็วสูง

= 0 หมายถึง เป็นแบบความเร็วต่ำ

Bit1 TRMT หมายถึง บิตที่แสดงสถานะของ Transmitt Shift register

= 1 หมายถึง เป็น TSR ว่าง

= 0หมายถึง เป็น TSR เต็ม

Bit0 TX9D เป็นข้อมูล bit ที่ 9 ถ้าจะทำการส่งแบบ 9 บิต

5. PIE1 บิตที่เกี่ยวข้องคือ RCIE,TXIE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IF	TMR1IE
BIT 7							BIT 0

รูปที่ 2.15 แสดงโครงสร้างของ PIE1

Bit5 RCIE

= 1 หมายถึงกำหนดให้มีการอินเตอร์รัปเกิดขึ้นได้เมื่อทำการรับข้อมูลทางอนุกรมเสร็จ 1 ไบท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50216

= 0 หมายถึงไม่ให้มีการอินเตอร์รัปเกิดขึ้นได้เมื่อทำการรับข้อมูลทางอนุกรมเสร็จ 1 ไบท์

Bit4 TXIE

= 1 หมายถึงกำหนดให้มีการอินเตอร์รัปเกิดขึ้นได้เมื่อทำการส่งข้อมูลทางอนุกรมเสร็จ 1 ไบท์

= 0 หมายถึงไม่ให้มีการอินเตอร์รัปเกิดขึ้นได้เมื่อทำการส่งข้อมูลทางอนุกรมเสร็จ 1 ไบท์

Serial Interfacing

เป็นวิธีการ Interface ของ PC ชนิดหนึ่งเพื่อทำการเชื่อมต่อกับอุปกรณ์อื่นๆ ซึ่งวิธีนี้จะเป็นการติดต่อแบบอนุกรมผ่านทาง Serial Port ซึ่งการเชื่อมต่อนั้น การต่อแบบ Serial Port นั้นจะมีความเร็วในการเชื่อมต่อช้ากว่าแบบ Parallel Port แต่อุปกรณ์ที่ใช้ใน Parallel Port จะมีราคาแพงกว่าและจะต้องใช้สายสัญญาณมากกว่า การส่งข้อมูลทาง Serial Port นั้นในเครื่อง PC จะใช้มาตรฐาน RS-232C ซึ่งเป็นมาตรฐานการส่งข้อมูลไม่เกิน 15 m. ส่วน Chip ที่ใช้ควบคุมการทำงานนั้นนิยมใช้ 8250 UART หรือตัวที่พัฒนามาใหม่อย่างเช่น 16550 UART ซึ่งมีการทำงานที่ดีขึ้น เราจึงต้องทำความเข้าใจมาตรฐานที่ใช้อยู่ เพื่อนำไปประยุกต์และใช้งานตามความต้องการได้

มาตรฐาน RS-232C

เพื่อที่จะให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมาตรฐานที่ใช้กันกว้างขวางที่สุดคือ RS-232C ถูกประกาศในปี 1969 โดย Electronic Industries Association ซึ่งเป็นการกำหนดมาตรฐานต่างๆ เช่น ลักษณะการเชื่อมต่อ สัญญาณทางไฟฟ้าที่ใช้

เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลบนสายเส้นเดียวกัน อุปกรณ์จึงถูกแบ่งเป็น 2 ประเภท คือ

1. DTE (Data Terminal Equipment) มักเป็นอุปกรณ์สำหรับส่งข้อมูล อย่างเช่น Computer
2. DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับทำการติดต่อ อย่างเช่น MODEM, HUB

ตามมาตรฐาน RS-232C อุปกรณ์ DTE ควรใช้หัวต่อตัวผู้ และอุปกรณ์ DCE ควรใช้หัวต่อตัวเมีย ซึ่งหัวต่อที่นิยมใช้กันอยู่จะเป็นชนิด D-Type ชนิด 9 และ25 ขา (บางครั้งเรียก DB-9 และ DB-25) อย่างไรก็ตามผู้ผลิตไม่ได้ปฏิบัติตามกฎนี้เสมอ ดังนั้นจึงไม่อาจแยกแยะอุปกรณ์ DTE และ DCE โดยการมองผ่านๆ ได้เสมอไป

เมื่อเราทราบว่าอุปกรณ์หนึ่งเป็น DTE และอีกตัวเป็น DCE ในทางทฤษฎีแล้วสามารถเชื่อมต่อได้อย่างง่ายดาย โดยการเชื่อมต่อสายสัญญาณที่หมายเลข(PIN) ตรงกันเรียกว่าการเชื่อมต่อแบบตรงไปตรงมา

การสื่อสารทางเดียว

สัญญาณหลักที่ใช้ในการสื่อสารมีอยู่ 3 สัญญาณ ได้แก่

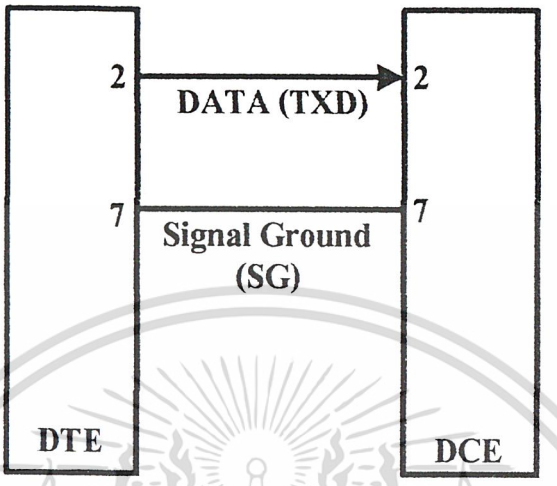
สายสัญญาณที่ 2 สำหรับข้อมูลจาก DTE →DCE

สายสัญญาณที่ 3 สำหรับข้อมูลจาก DCE→DTE

สายสัญญาณที่ 7 สำหรับSingle Ground (SG) ซึ่งเป็นจุดอ้างอิงร่วมสำหรับขั้วและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันไฟฟ้าของสายอื่น ตัวอย่างของการสื่อสารอย่างง่ายที่สุดซึ่งมีเพียงอุปกรณ์หนึ่งส่งและอีกตัวรับจะใช้สายเพียง 2 เส้นก็เพียงพอในการเชื่อมต่อ คือ สายเส้นที่ 2 และสายเส้นที่ 7



รูปที่ 2.16 แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว

Hardware Handshaking

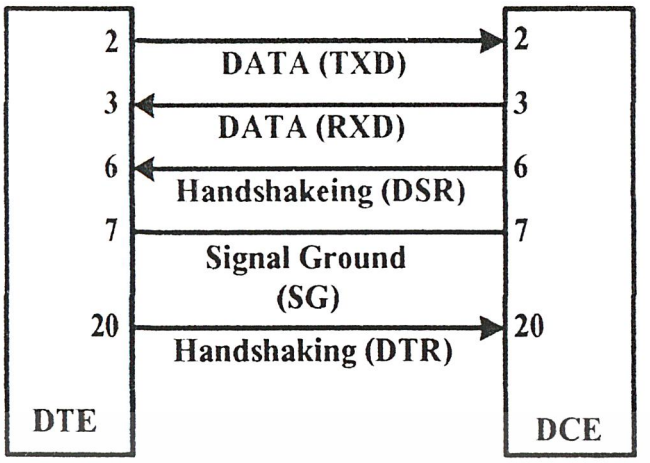
ในหลายๆ กรณีอุปกรณ์ฝ่ายส่งต้องรู้ว่าอุปกรณ์ฝ่ายรับพร้อมที่จะรับข้อมูลหรือไม่ ตัวอย่างเช่น การส่งข้อมูลไปยังเครื่องพิมพ์ ความเร็วของการสื่อสารอาจเร็วกว่าความเร็วของเครื่องพิมพ์ เครื่องพิมพ์จะต้องระงับการส่งข้อมูลของคอมพิวเตอร์จนกว่ามันจะพร้อมที่จะรับข้อมูล กรณีเช่นนี้จะต้องมีสัญญาณส่งจากฝ่ายรับบอกไปยังอุปกรณ์ฝ่ายส่ง เพื่อแจ้งสถานะการทำงานต่างๆ ว่าพร้อมหรือไม่ สัญญาณนี้เรียกว่า Flow Control หรือ Hand Shaking โดยใช้สัญญาณทางไฟฟ้าที่เปลี่ยนไปเป็น Flow Control และ Software Handshake จะต้องใช้อักขระพิเศษ หรือข้อมูลที่แตกต่างจากการสื่อสารปกติเป็น Flow Control

การใช้ Hardware Handshake จะต้องมีสายสัญญาณต่อเพิ่มขึ้นเพื่อใช้เป็นสายในการส่งสัญญาณ Flow Control จึงทำให้สายสัญญาณแบ่งได้เป็น 3 กลุ่ม คือ Data, Handshake และ Signal Ground

การสื่อสารสองทาง

ในหลายกรณีที่ข้อมูลถูกส่งผ่านในสองทิศทาง โดยเฉพาะเมื่อคอมพิวเตอร์สองตัวติดต่อสื่อสารกัน จำนวนสายที่น้อยที่สุดในการติดต่อสื่อสารแบบสองทิศทางคือ 3 เส้น ได้แก่ สายข้อมูลในแต่ละทิศทาง และสาย Signal Ground การเพิ่มเติมสาย Handshake ในแต่ละทิศทางจะทำให้จำนวนสายเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงการสื่อสารแบบสองทิศทาง

ลักษณะหัวต่อแบบ D-Type

แบบ 9 pin แบบ 25 pin

รูปที่ 2.18 แสดงลักษณะหัวต่อแบบ D-Type

หัวต่อแบบ D-Type มักใช้เป็นพอร์ทอนุกรมซึ่งมีอยู่ 2 ลักษณะคือ 9 pin และ 25 pin โดยจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่จัดเรียงไว้ไม่เหมือนกัน

D-Type 25Pin NO.	D-Type 9 Pin NO.	Abbreviation	Full Name
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Transmit Ready
Pin 22	Pin 9	RI	Ring Indicator

ตารางที่ 2.2 D-Type 9 Pin and D-Type 25 Pin Connectors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

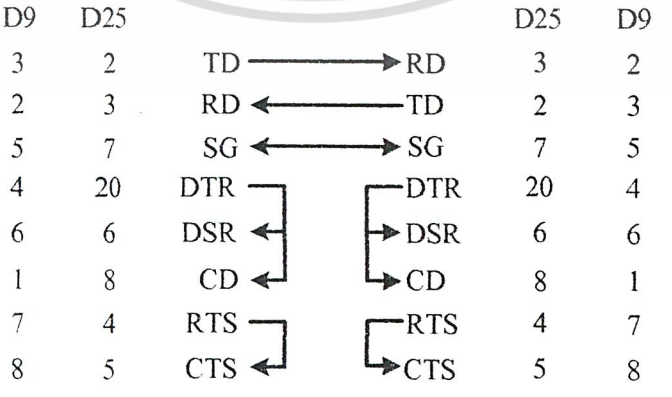
Pin Function

Abbreviation	Full Name	Function
TD	Transmit Data	เอาท์พุตของข้อมูลแบบอนุกรม (TXD)
RD	Receive Data	อินพุตของข้อมูลแบบอนุกรม (RXD)
CTS	Clear To Send	เป็นตัวบ่งชี้ว่าโมเด็มพร้อมที่จะทำการแลกเปลี่ยนข้อมูลแล้ว
DCD	Data Carrier Detect	เมื่อโมเด็มสามารถตรวจจับ carrier จากสายโทรศัพท์อื่นๆ ได้แล้ว จะมีสถานะเป็นแอ็คทีฟ
DSR	Data Set Ready	เป็นตัวบอก UART ว่าโมเด็มพร้อมที่จะทำการสร้างทางเชื่อมต่อแล้ว
DTR	Data Terminal Ready	เป็นตัวบอก UART ว่าโมเด็มพร้อมที่จะทำการเชื่อมต่อแล้ว
RTS	Request To Send	สายนี้จะเป็นตัวบอกโมเด็มว่า UART พร้อมที่จะทำการแลกเปลี่ยนข้อมูลแล้ว
RI	Ring Indicator	จะเปลี่ยนสถานะเป็นแอ็คทีฟเมื่อโมเด็มตรวจจับสัญญาณกระดิ่งจากโทรศัพท์บ้าน

ตารางที่ 2.3 แสดงการทำงานของแต่ละขา

การต่อแบบ Null Modem

เป็นการเชื่อมต่อกันระหว่าง DTE กับ DTE เข้าด้วยกัน เช่น การเชื่อมต่อกันระหว่าง PC กับ PC เพื่อทำการติดต่อรับ-ส่งข้อมูลระหว่างกัน ในลักษณะเช่นนี้ต้องทำการเชื่อมต่อสาย 2 สายบนอุปกรณ์ตัวแรกกับสาย 3 บนอุปกรณ์ตัวที่สอง และสาย 3 บนอุปกรณ์ตัวแรกกับสาย 2 บนอุปกรณ์ตัวที่สอง และสาย Handshake ก็ต้องถูกไขว้ในทำนองเดียวกัน



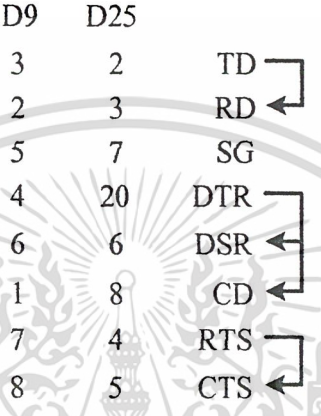
รูปที่ 2.19 แสดงการเชื่อมต่อแบบ Null Modem ที่นิยมใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเราสามารถทำการ Loop Back สัญญาณ Handshake ในตัวอุปกรณ์เองเพื่อไม่ต้องไขว้สายสัญญาณภายนอกได้ ทำให้ใช้สายสัญญาณในการติดต่อเพียง 3 เส้นเท่านั้น (TD, RD, SG)

จากรูปได้ทำ Loop Back สัญญาณ DTR เข้ากับ DSR และ CD เมื่อสัญญาณ DTR active (เครื่องพร้อมที่จะรับข้อมูล) สัญญาณนี้จะ Loop Back ทำให้ DSR และ CD Active เสมือนว่ามีอุปกรณ์ DCE ต่ออยู่จริง ในทางกลับกันการต่อแบบนี้ก็สามารถใช้ได้กับการต่อระหว่าง DCE กับ DCE ได้อีกด้วย

Loop Back Plug



รูปที่ 2.20 แสดงการต่อแบบ Loop Back

จากรูปจะเป็นการต่อเพื่อทำการรับและส่งในพอร์ทเดียวกัน เหมาะสำหรับทำการตรวจสอบพอร์ท และตรวจสอบการในการทำงานโปรแกรม เช่น การทดสอบโปรแกรม talk สิ่งที่เราพิมพ์ออกไปก็จะได้รับเข้ามาด้วย

สัญญาณทางไฟฟ้า

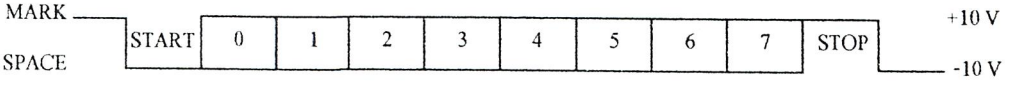
มาตรฐาน RS-232 ได้กำหนดลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่อแบบอนุกรม นี้มี 2 ลักษณะ คือ SPACE หมายถึง logic “0” และ MARK หมายถึง logic”1” โดย

SPACE จะเป็นแรงดันไฟฟ้าบวก Output ในช่วง +5 ถึง +15 V.

Input ในช่วง +3 ถึง +15 V.

MARK จะเป็นแรงดันไฟฟ้าลบ Output ในช่วง -5 ถึง -15 V.

Input ในช่วง -3 ถึง -15 V.



รูปที่ 2.21 RS-232 logic wave form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแตกต่างของอินพุทและเอาท์พุตมีไว้เผื่อกรณีแรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสายสัญญาณ และจะพบว่าเมื่อให้สายสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงเกินขอบเขตที่ยอมรับได้ นอกจากนี้ความจุไฟฟ้าที่เกิดขึ้นจะมีผลกับคุณภาพของสัญญาณทำให้สถานะจากแรงดันไฟฟ้าบวกและลบไม่ชัดเจนทำให้การติดต่อไม่ได้ไกลนัก แต่ถ้าจำเป็นต้องใช้จำเป็นจะต้องมีอุปกรณ์อื่นเพิ่มเติม เช่น Line driver

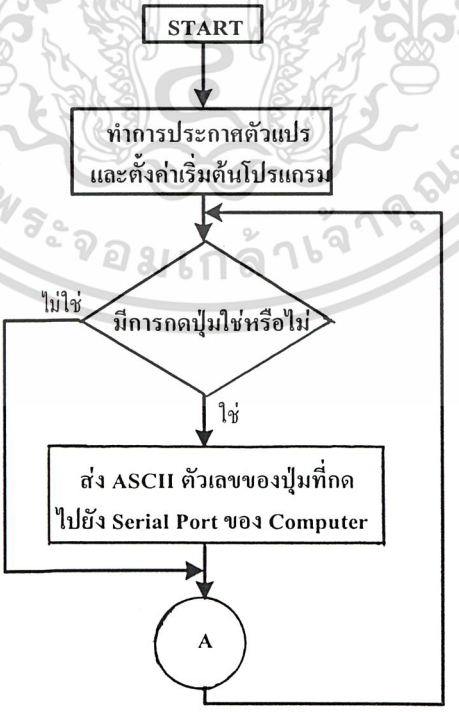
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการออกแบบ

การสร้างและการออกแบบระบบนี้ แบ่งเป็น 2 ส่วนคือ การสร้างและการออกแบบในส่วนของไมโครคอนโทรลเลอร์และการสร้างและการออกแบบในส่วนของคอมพิวเตอร์

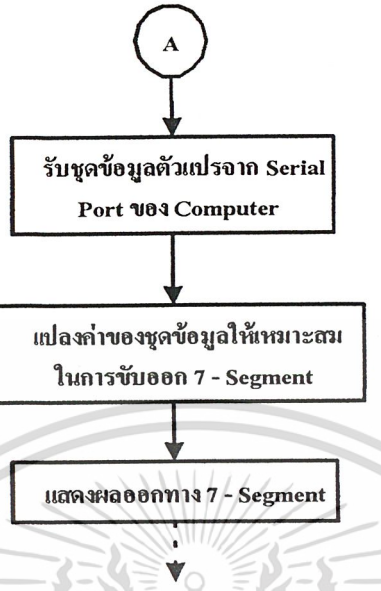
3.1 ในส่วนของไมโครคอนโทรลเลอร์

จากบล็อกไดอาแกรมรูปที่ 2.1 ในบทที่ 2 แสดงถึงการทำงานของส่วนต่างๆ สามารถที่จะนำมาเขียนโปรแกรมควบคุมได้ โดยเมื่อเริ่มทำงานคอมพิวเตอร์จะส่งค่ามาให้กับไมโครคอรโทรลเลอร์เพื่อทำการเคลียร์ค่าในไมโครคอรโทรลเลอร์ จากนั้นเมื่อมีการกดสวิตซ์เข้ามาจากเคาน์เตอร์บริการไมโครคอนโทรล เลอร์จะทำการส่งค่าที่ได้กลับไปยังคอมพิวเตอร์เพื่อให้คอมพิวเตอร์ประมวลผลว่าเคาน์เตอร์ได้เป็นผู้กดเข้ามา หลังจากนั้นคอมพิวเตอร์ก็จะส่งหมายเลขคิวที่ยังไม่ได้รับบริการพร้อมทั้งช่องที่สามารถให้บริการกลับไปยังไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์นั้นนำค่าที่ได้มาประมวลผลและแสดงบนจอแสดงผล ในกรณีที่ไม่มีคิวรอแต่มีการกดจากเคาน์เตอร์บริการ ไมโครคอนโทรลเลอร์ก็จะทำการส่งค่าที่ได้จากการกดสวิตซ์ไปยังคอมพิวเตอร์ แต่คอมพิวเตอร์จะไม่ส่งค่ากลับมาจนกว่าจะมีผู้มากดปุ่มที่หน้าจอเพื่อรับบัตรคิว ในโครงงานนี้จะทำการเขียนโปรแกรม PicBasicPro โปรแกรมใน PIC16F877



รูปที่ 3.1 แสดงผังการทำงานของไมโครคอนโทรลเลอร์

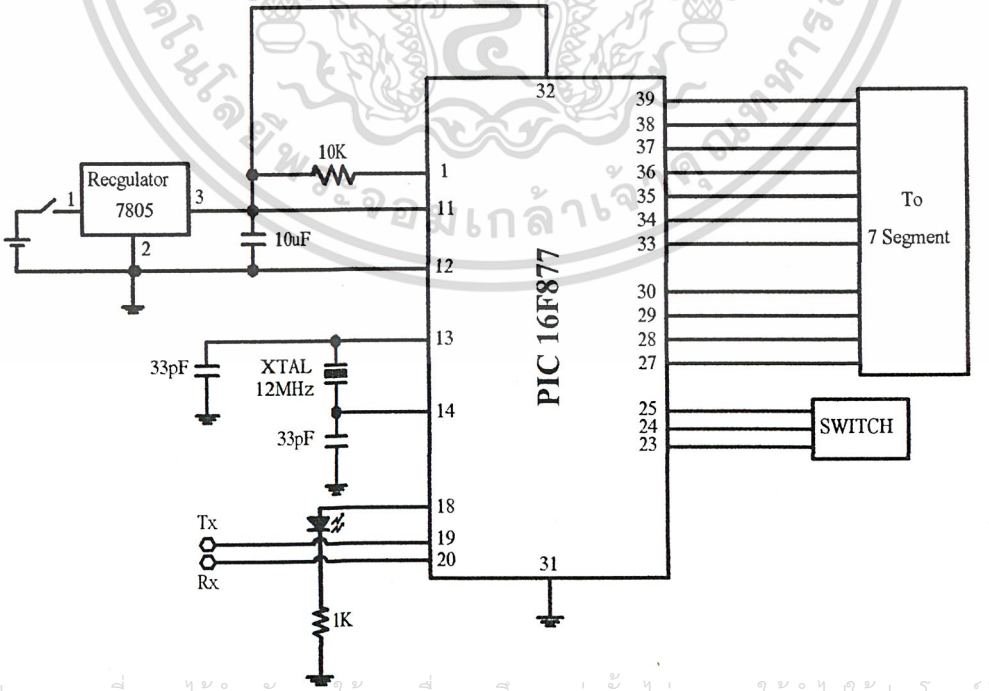
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงผังการทำงานของไมโครคอนโทรลเลอร์ (ต่อ)

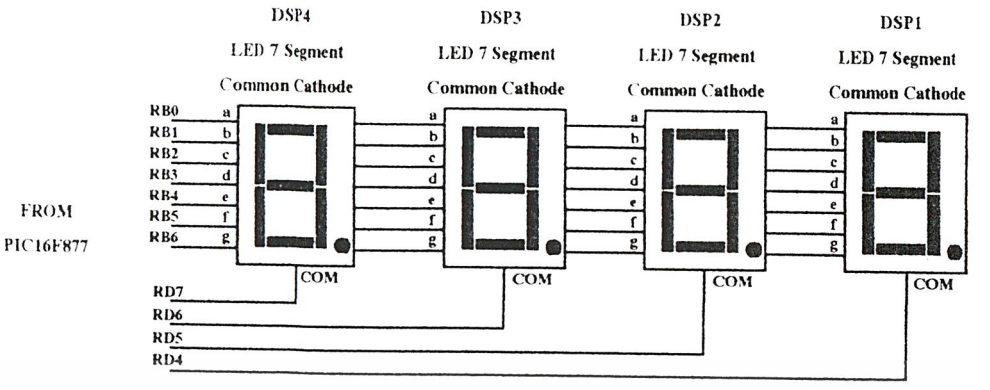
การประมวลผลเพื่อแสดงผลบนจอแสดงผล

การทำงานในส่วนนี้ ไมโครคอรโทรลเลอร์จะคอยรับข้อมูลที่เป็นค่าหมายเลขคิวและหมายเลขเคาน์เตอร์ที่ส่งมาจากคอมพิวเตอร์ และนำมาประมวลผล จากนั้นจึงนำค่าที่ได้มาแสดงผลบนจอแสดงผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงการต่อไมโครคอนโทรลเลอร์เข้ากับจอแสดงผล



รูปที่ 3.4 แสดงการต่อไมโครคอนโทรลเลอร์เข้ากับจอแสดงผล

3.2 ในส่วนของคอมพิวเตอร์

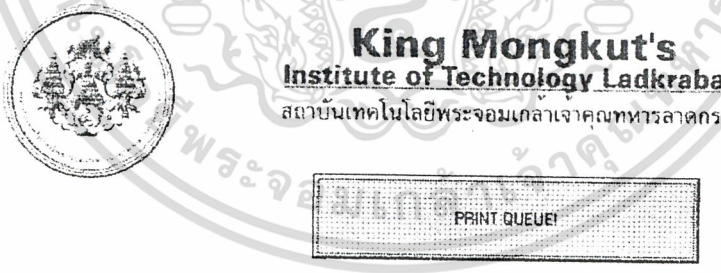
ในส่วนนี้จะใช้โปรแกรม Visual Basic ออกแบบระบบ ซึ่งจะแบ่งการออกแบบออกเป็น 2 ส่วน คือ 1. ส่วนของหน้าจอที่ใช้ในการพิมพ์บัตรคิว

2. ส่วนของหน้าจอที่ใช้ในการทดสอบและแก้ไขระบบ

1. ส่วนของหน้าจอที่ใช้ในการพิมพ์บัตรคิว

ในส่วนนี้เราจะทำการออกแบบหน้าจอที่เราต้องการให้ระบบพิมพ์บัตรคิวออกมา โดยจะแบ่งหน้าจอออกเป็น 3 ส่วนหลักๆ คือ

1. ส่วนที่ใช้กดเพื่อพิมพ์บัตรคิว



รูปที่ 3.5 แสดงหน้าจอที่ใช้เพื่อกดพิมพ์บัตรคิว

2. ส่วนแสดงหมายเลขปัจจุบันและจำนวนคิวที่รอ

รูปที่ 3.6 แสดงหน้าจอของหมายเลขลำดับคิวและจำนวนคิวที่รอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนแสดงวันที่และเวลา

3/24/03
1:41:15 AM

รูปที่ 3.7 แสดงหน้าจอของวันที่และเวลา

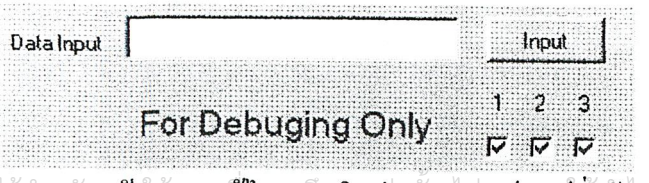
ซึ่งเมื่อทำการออกแบบตามที่กำหนดไว้แล้วจะได้หน้าจอที่มีลักษณะดังนี้

รูปที่ 3.8 แสดงหน้าจอการออกแบบที่สมบูรณ์

2. ส่วนของหน้าจอที่ใช้ในการทดสอบและแก้ไขระบบ

ในส่วนนี้เราจะทำการออกแบบหน้าจอที่ใช้ในการทดสอบและแก้ไขระบบ ซึ่งจะแบ่งออกเป็น 4 ส่วนคือ

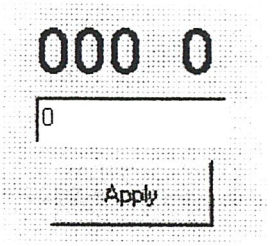
1. ส่วนที่ใช้ในการป้อนช่องที่สามารถให้บริการได้



รูปที่ 3.9 แสดงหน้าจอการแก้ไขระบบในส่วนของเคาน์เตอร์ที่ให้บริการ

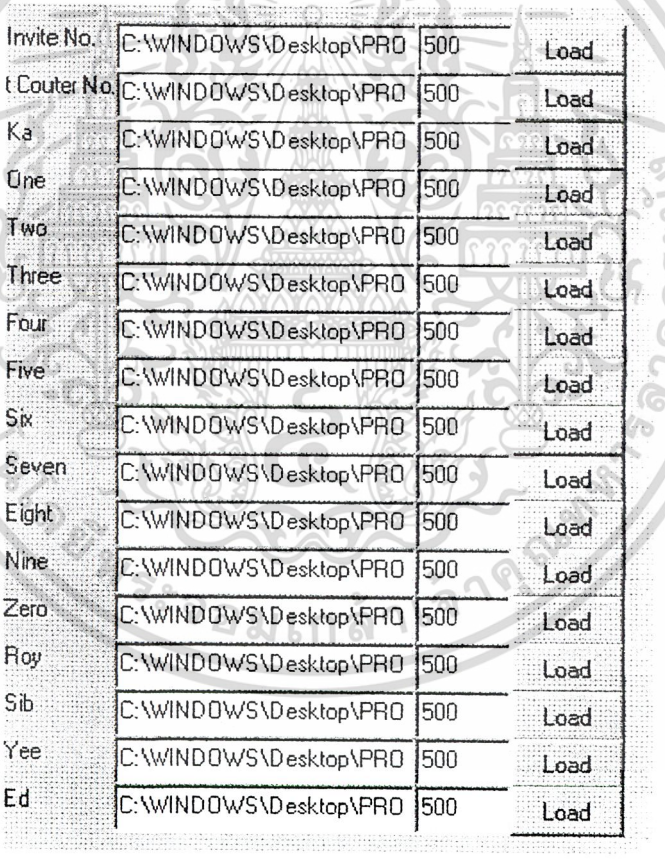
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนที่ใช้ป้อนหมายเลขคิวที่เรียกก่อนหน้านี้



รูปที่ 3.10 แสดงหน้าจอการแก้ไขระบบในส่วนของหมายเลขคิวก่อนหน้า

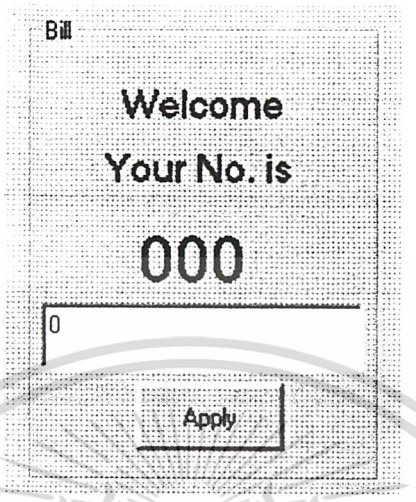
3. ส่วนที่ใช้โหลดไฟล์เสียงที่เก็บเอาไว้



รูปที่ 3.11 แสดงหน้าจอการแก้ไขระบบในส่วนของที่เก็บไฟล์เสียง

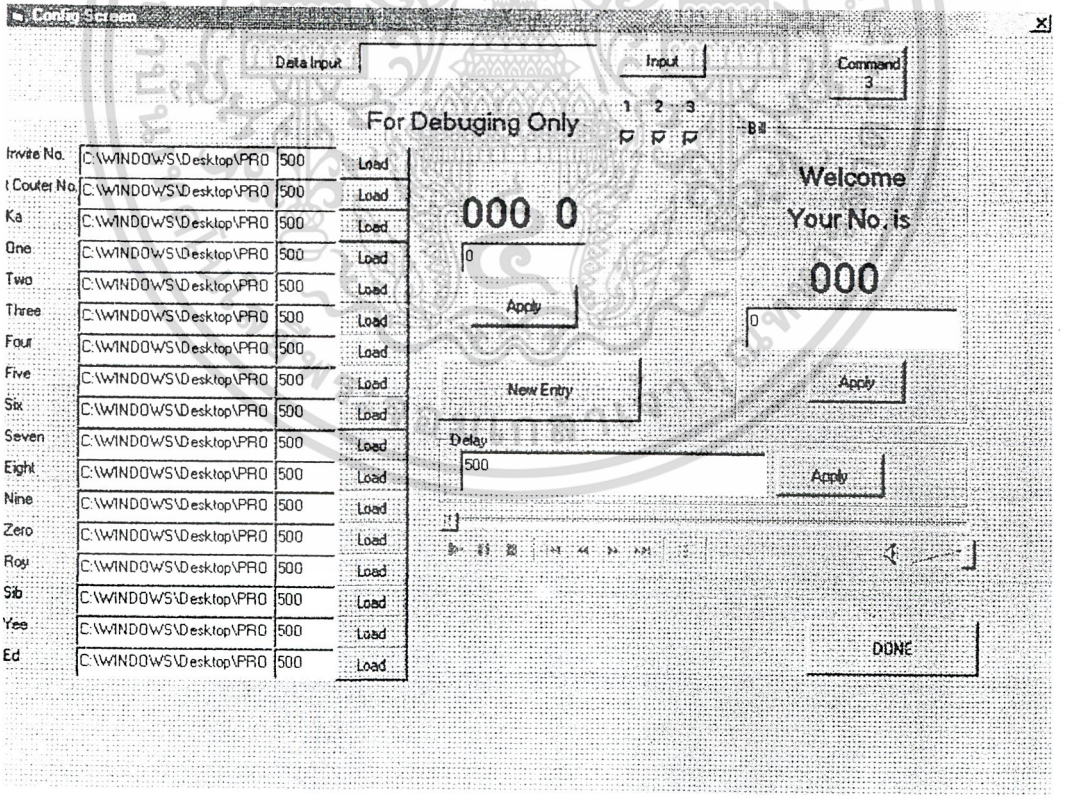
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ส่วนที่ใช้ป้อนคิวเริ่มต้น



รูปที่ 3.12 แสดงหน้าจอการแก้ไขระบบในส่วนของหมายเลขคิว

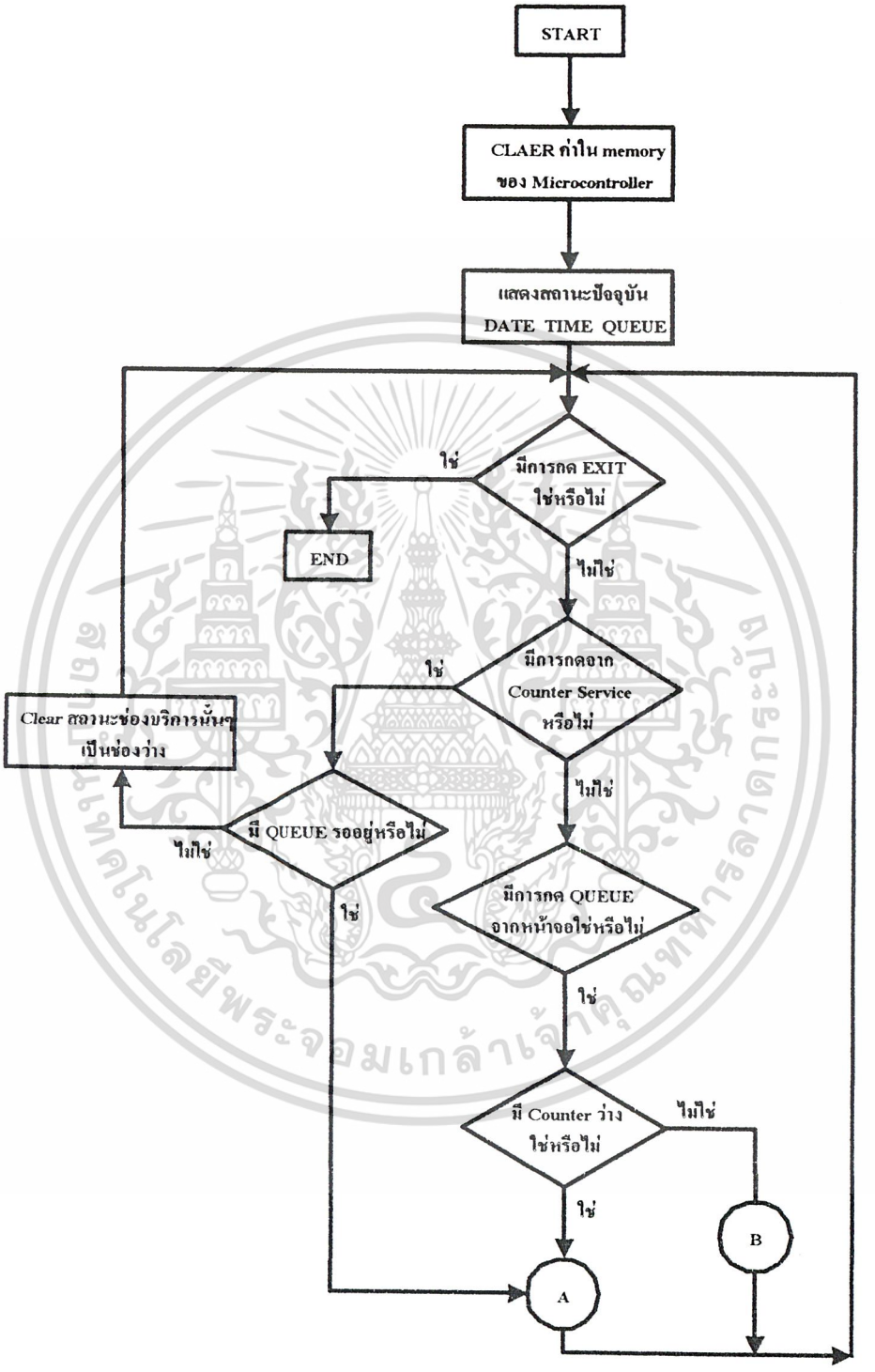
ซึ่งเมื่อทำการออกแบบตามที่กำหนดไว้แล้วจะได้หน้าจอที่มีลักษณะดังนี้



รูปที่ 3.13 แสดงหน้าจอการแก้ไขระบบที่เสร็จสมบูรณ์

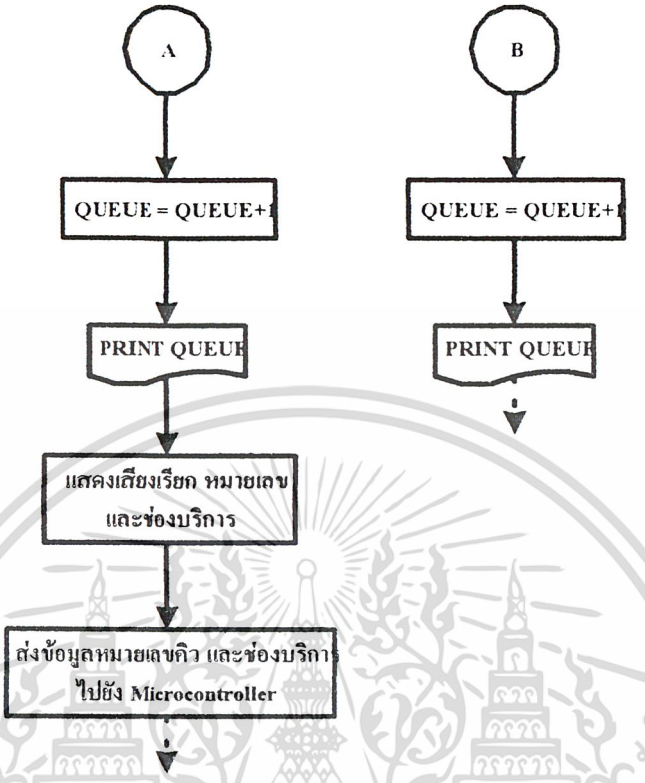
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นจึงทำการเขียนโปรแกรมควบคุมการทำงานและส่งผ่านข้อมูลผ่านทางพอร์ทอนุกรม โดยใช้คำสั่ง MSComm ซึ่งการทำงานของระบบจะแสดงได้ด้วยผังการทำงาน (Flow Chart) ดังนี้



รูปที่ 3.14 แสดงผังการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงผังการทำงานของโปรแกรม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

จากการออกแบบระบบในบทที่ 3 เมื่อนำมาทำการต่อเป็นวงจรจริงและเขียนโปรแกรมควบคุมการทำงานของแต่ละส่วน แล้วนำมาทดสอบการทำงานโดยรวม ให้ผลการทำงานในแต่ละส่วนดังนี้

4.1 การควบคุมเครื่องพิมพ์บัตรคิว

เมื่อทำการรันโปรแกรมแล้วจะได้หน้าจอที่มีลักษณะดังนี้

รูปที่ 4.1 แสดงหน้าจอเมื่อทำการรันโปรแกรม

เมื่อผู้ใช้บริการกดปุ่มที่หน้าจอ โปรแกรมที่คอยควบคุมเครื่องพิมพ์ก็จะสั่งให้เครื่องพิมพ์นั้นพิมพ์บัตรคิวออกมาได้ โดยบัตรคิวที่พิมพ์นั้นจะประกอบไปด้วย หมายเลขลำดับ จำนวนคิวที่รอ วัน เดือน ปี และเวลาในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แสดงตัวอย่างของบัตรคิวที่พิมพ์ออกมา

4.2 การรับข้อมูลจากเคาน์เตอร์รับบริการ

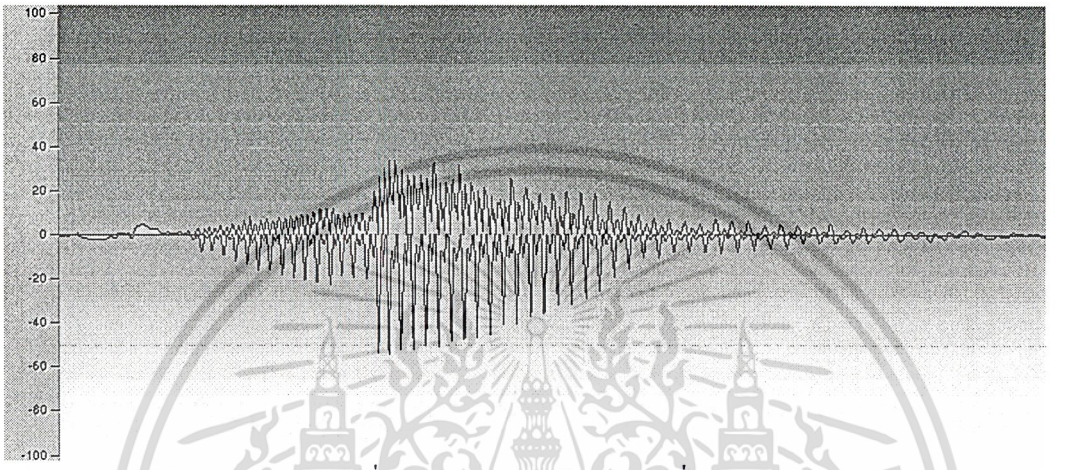
ในส่วนของเคาน์เตอร์รับบริการหลังจากมีการกดสวิตช์ของผู้ให้บริการ ไมโครคอนโทรลเลอร์ที่ควบคุมอยู่ก็จะทำการเก็บค่าสถานะพร้อมกับส่งข้อมูลไปยังโปรแกรมผ่านทางพอร์ทอนุกรม จากนั้นโปรแกรมจะทำการตรวจสอบค่าที่รับได้พร้อมทั้งลดจำนวนคิวที่รอลงตามจำนวนครั้งที่กดเข้ามาของเคาน์เตอร์รับบริการ หลังจากนั้นโปรแกรมจะทำการส่งค่าหมายเลขแรกของคิวที่ยังไม่ได้รับบริการพร้อมทั้งช่องบริการที่สามารถให้บริการได้ไปยังไมโครคอนโทรลเลอร์โดยผ่านทางพอร์ทอนุกรมเพื่อให้ไมโครคอนโทรลเลอร์ประมวลค่าที่ได้เพื่อนำไปแสดงบนจอแสดงผล แล้วจึงเข้าสู่ระบบเสียงพูดต่อไป

รูปที่ 4.3 แสดงหน้าจอของโปรแกรมเมื่อมีการกดเข้ามาของเคาน์เตอร์รับบริการ

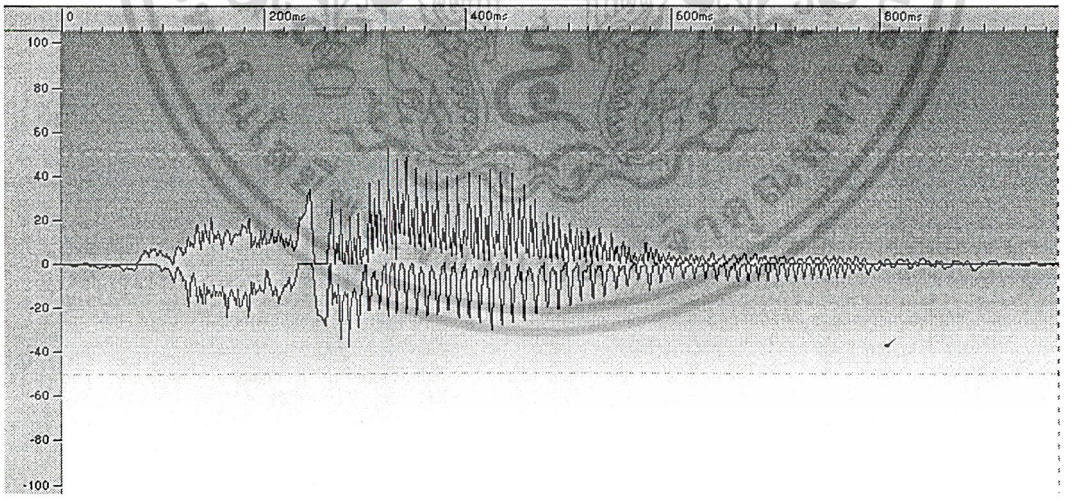
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ระบบเสียงพูด

ในส่วนนี้จะใช้การบันทึกเสียงลงบนฮาร์ดดิสก์ และเล่นเสียงผ่านทางการ์ดเสียง (Sound card) ของคอมพิวเตอร์ การบันทึกเสียงจะบันทึกทีละคำดังนี้ หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า ศูนย์ สิบ ยี่ร้อย ขอเชิญคิวลำดับที่ ที่ช่องบริการหมายเลข ครับ เมื่อโปรแกรมทำการรับค่าที่ส่งมาจากไมโครคอนโทรลเลอร์แล้ว โปรแกรมจะทำการตรวจสอบค่าที่รับได้ว่าเป็นหมายเลขคิวและช่องบริการใด หลังจากนั้นโปรแกรมจะทำการผสมคำและทำการเล่นเสียงออกมา

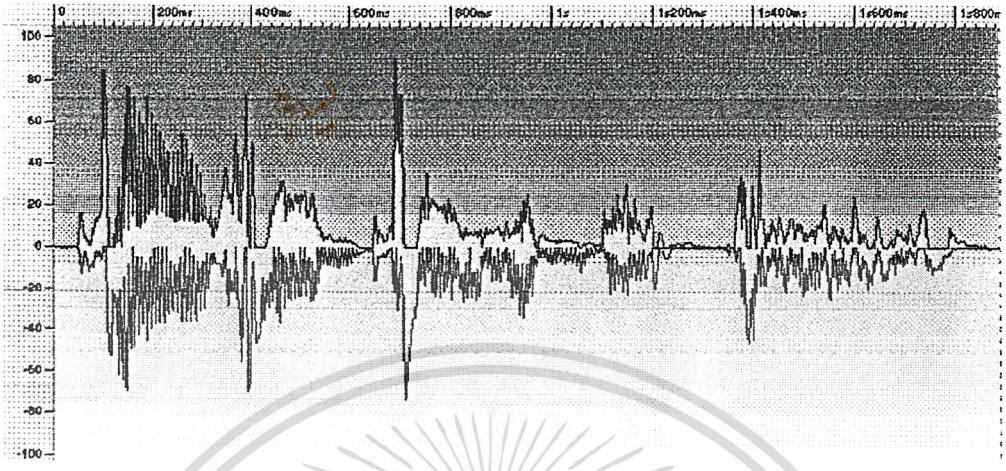


รูปที่ 4.4 แสดงกราฟเสียงพูด “หนึ่ง”

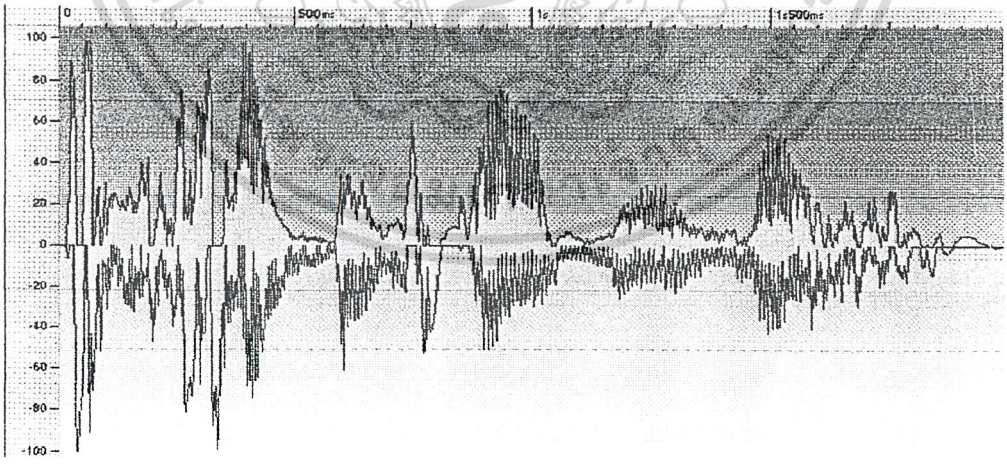


รูปที่ 4.5 แสดงกราฟเสียงพูด “สอง”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

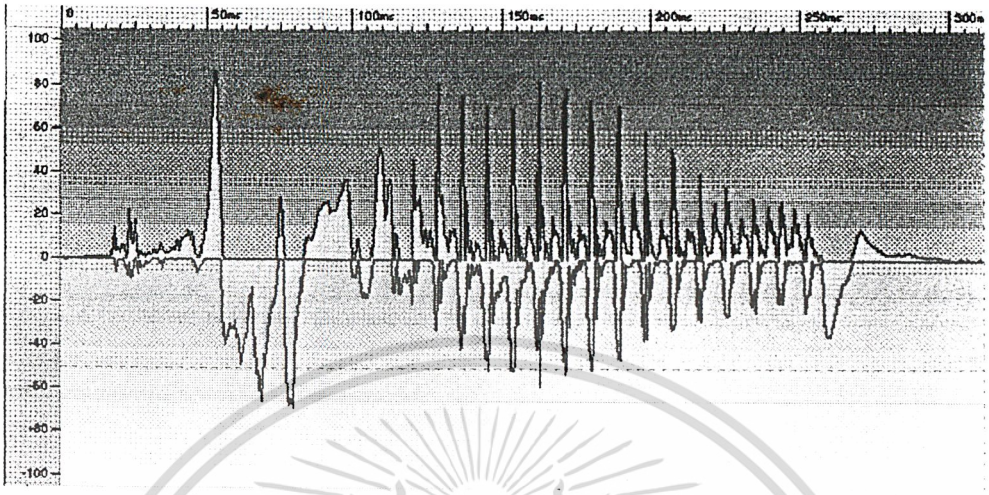


รูปที่ 4.6 แสดงกราฟเสียงพูด “ขอเชิญคิวลำดับที่”

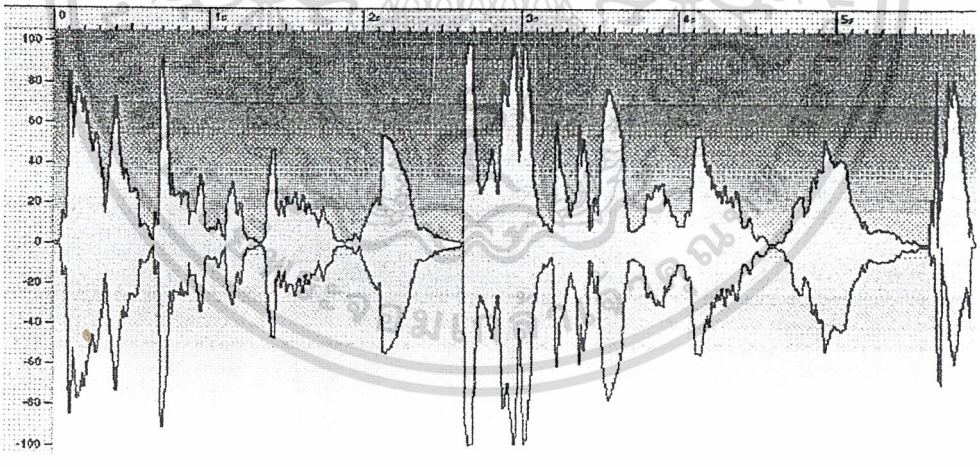


รูปที่ 4.7 แสดงกราฟเสียงพูด “ที่ช่องบริการหมายเลข”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงกราฟเสียงพูด “ครับ”

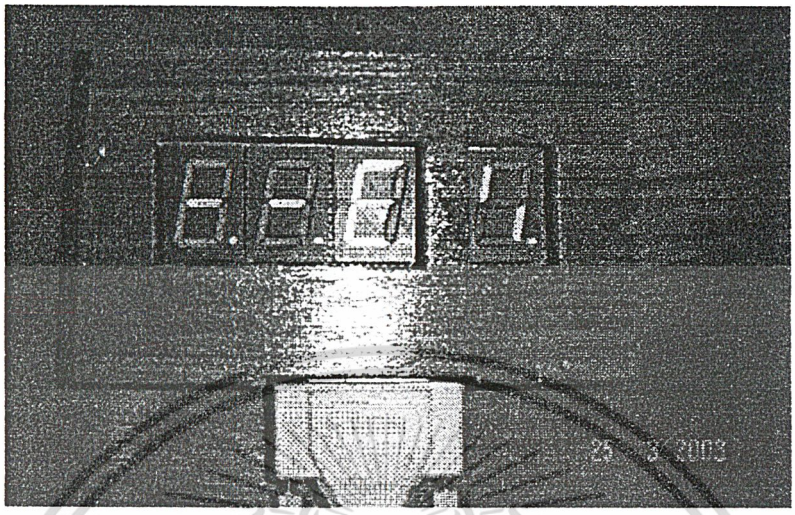


รูปที่ 4.9 แสดงกราฟเสียงพูด “ขอเชิญคิวลำดับที่หนึ่งที่ช่องบริการหมายเลขสองครับ”

4.4 การควบคุมการแสดงผลตัวเลข

หลังจากที่ไมโครคอนโทรลเลอร์รับค่าที่ส่งมาจากโปรแกรมแล้ว ไมโครคอนโทรลเลอร์จะทำการประมวลค่าที่รับได้ แล้วจึงนำมาแสดงผลบนจอแสดงผลซึ่งบนจอแสดงผลจะประกอบไปด้วย หมายเลขคิวที่ได้รับบริการ และหมายเลขช่องบริการ

เอกสารนี้มีประโยชน์ต่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น โดยไม่ได้รับอนุญาตจากผู้เขียนหนังสือและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้าจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

จากผลการทดลองโดยรวมพบว่า สามารถทำงานได้ตามจุดประสงค์ที่ตั้งไว้ ในส่วนของการควบคุมเสียงพูด ความต่อเนื่องของคำยังไม่ดีพอ เนื่องจากกระบวนการอัดเสียง ทำให้เกิดช่องว่างหรือช่วงที่ไม่มีเสียงพูด มีแต่เสียงของสิ่งแวดล้อมจะเกิดในขณะก่อนและหลังการกดปุ่มอัดเสียงหรือในช่วงต้นและช่วงท้ายของคำ และคุณภาพของเสียงยังขึ้นอยู่กับคุณภาพของไมโครโฟนในการอัดเสียง

ในเชิงพาณิชย์ ระบบนี้สามารถใช้เพียงคอมพิวเตอร์เครื่องเดียวควบคุมทุกอย่างเพื่อลดค่าใช้จ่ายได้ แต่ในที่นี้ผู้จัดทำมีความประสงค์ในการเรียนรู้ไมโครคอนโทรลเลอร์จึงได้นำไปประยุกต์ใช้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INCLUDE  " modedefs.bas "
      digit1 VAR BYTE
      digit2 VAR BYTE
      digit3 VAR BYTE
      channel VAR BYTE
      debounce VAR BYTE
      num VAR BYTE
      TRISB = %00000000
      TRISC = %00111000
      digit1 = " 0 "
      digit2 = " 0 "
      digit3 = " 0 "
      channel = " 0 "
      PORTD = 255
      PORTB = 255
      Low PORTC.4
      Low PORTC.5
      Low PORTC.6
      Pause 1000
      High PORTC.3

mainloop:
      IF PORTC.4 = 1 Then
            SerOut PORTD.0,N2400,["1"]
      EndIF
      IF PORTC.5 = 1 Then
            SerOut PORTD.0,N2400,["2"]
      EndIF
      IF PORTC.6 = 1 Then
            SerOut PORTD.0,N2400,["3"]
      EndIF
      For debounce = 0 to 12
            Call seg_show
      Next debounce
      GoTo mainloop

seg_show:
      PORTD = 252
      Low PORTD.7
      num = channel
      Call showdat
      SerIn PORTD.1,N2400,5,CONT1,["U"],digit3,digit2,digit1,channel
  CONT1:
      PORTD = 252
      Low PORTD.6
      num = digit1
      Call showdat
      SerIn PORTD.1,N2400,5,CONT2,["U"],digit3,digit2,digit1,channel
  CONT2:
      PORTD = 252
      Low PORTD.5
      num = digit2
      Call showdat
      SerIn PORTD.1,N2400,5,CONT3,["U"],digit3,digit2,digit1,channel
  CONT3:
      PORTD = 252
      Low PORTD.4
      num = digit3
      Call showdat
      SerIn PORTD.1,N2400,5,CONT4,["U"],digit3,digit2,digit1,channel
  CONT4:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Resume

showdat:
      IF num = 48 Then
            PORTB = %00111111
      Else
            IF num = 49 Then
                  PORTB = %00001001
            Else
            IF num = 50  Then
                  PORTB = %01011110
                  Else
                        IF num = 51 Then
                              PORTB = %01011011
                        Else
                              IF num = 52 Then
                                    PORTB = %01101001
                              Else
                                    IF num = 53 Then
                                          PORTB = %01110011
                                    Else
                                          IF num = 54 Then
                                                PORTB = %01100111
                                          Else
                                                IF num = 55 Then
                                                      PORTB =
%00011001

                                                Else
                                                      IF num = 56
Then

                                                            PORTB =
%01111111

                                                      Else
                                                            PORTB =
%01111001

                                                      EndIF
                                                EndIF
                                          EndIF
                                    EndIF
                              EndIF
                        EndIF
                  EndIF
            EndIF
      EndIF
Resume
End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code 1

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Dim buff As Integer
Dim cnt As Integer
Dim time As Integer
Dim Recent As Integer
Dim Queue As Integer
Dim CounterQ As String
Dim Counter1 As Boolean
Dim Counter2 As Boolean
Dim Counter3 As Boolean
Dim LastService As Integer
Dim word_cnt As Integer
Dim digit1 As String
Dim digit2 As String
Dim digit3 As String
Dim wording As Integer

Private Sub Apply_Click()
Timer1.Interval = Text1.Text
End Sub

Private Sub Command1_Click()
  Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    One.Text = SFile
End Sub

Private Sub Command10_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Five.Text = SFile
End Sub

Private Sub Command11_Click()
Dim SFile As String
    With CommonDialog1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Seven.Text = SFile
End Sub

Private Sub Command12_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Six.Text = SFile
End Sub

Private Sub Command13_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Eight.Text = SFile
End Sub

Private Sub Command14_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End With
    Nine.Text = SFile
End Sub

Private Sub Command15_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Zero.Text = SFile
End Sub


Private Sub Command16_Click()
Recent = Text3.Text
Label11.Caption = Text3.Text
End Sub

Private Sub Command17_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Invite.Text = SFile
End Sub

Private Sub Command18_Click()
Queue = Text4.Text
Label8.Caption = Text4.Text
End Sub

Private Sub Command19_Click()
Form2.Show
Form1.Hide
End Sub

Private Sub Command2_Click()
  Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Two.Text = SFile
End Sub


Private Sub Command20_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Roy.Text = SFile
End Sub

Private Sub Command21_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Sib.Text = SFile
End Sub

Private Sub Command22_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End With
    Yee.Text = SFile
End Sub

Private Sub Command23_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Ed.Text = SFile
End Sub

Private Sub Command4_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Ka.Text = SFile
End Sub

Private Sub Command5_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    At.Text = SFile
End Sub

Private Sub Command6_Click()
Dim val As String
Dim a As Integer
If Text2.Text = "1" Then
    Counter1 = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Check1.Value = False
ElseIf Text2.Text = "2" Then
    Counter2 = False
    Check2.Value = False
ElseIf Text2.Text = "3" Then
    Counter3 = False
    Check3.Value = False
Else
    MsgBox ("Unknow data input")
    Exit Sub
End If
If Recent > Queue Then
    Queue = Queue + 1
    val = Queue
    If Len(val) = 1 Then
        Label8.Caption = "00" & Queue
    ElseIf Len(val) = 2 Then
        Label8.Caption = "0" & Queue
    Else
        Label8.Caption = Queue
    End If
    Form2.Label6.Caption = Form1.Label11.Caption - Form1.Label8.Caption
    If Counter1 = False Then
        Label10.Caption = "1"
        Counter1 = True
        Check1.Value = 1
    ElseIf Counter2 = False Then
        Label10.Caption = "2"
        Counter2 = True
        Check2.Value = 1
    Else
        Label10.Caption = "3"
        Counter3 = True
        Check3.Value = 1
    End If
    If Len(val) = 1 Then
      digit1 = val
      word_cnt = 1
    ElseIf Len(val) = 2 Then
      digit1 = Mid(val, 2, 1)
      digit2 = Mid(val, 1, 1)
      word_cnt = 3
    Else
      digit1 = Mid(val, 3, 1)
      digit2 = Mid(val, 2, 1)
      digit3 = Mid(val, 1, 1)
      word_cnt = 5
    End If
    CounterQ = Label10.Caption
    wording = 1
    MSComm1.Output = "U" + Form1.Label8.Caption + Label10.Caption
End If
End Sub

Private Sub Command7_Click()
Dim val As String
Recent = Recent + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
val = Recent
If Len(val) = 1 Then
    Label11.Caption = "00" & Recent
ElseIf Len(val) = 2 Then
    Label11.Caption = "0" & Recent
Else
    Label11.Caption = Recent
End If
Form2.Label6.Caption = Form1.Label11.Caption - Form1.Label8.Caption
If Counter1 = False Or Counter2 = False Or Counter3 = False Then
    Queue = Queue + 1
    val = Queue
    If Len(val) = 1 Then
        Label8.Caption = "00" & Queue
    ElseIf Len(val) = 2 Then
        Label8.Caption = "0" & Queue
    Else
        Label8.Caption = Queue
    End If
    Form2.Label6.Caption = Form1.Label11.Caption - Form1.Label8.Caption
    If Counter1 = False Then
        Label10.Caption = "1"
        Counter1 = True
        Check1.Value = 1
    ElseIf Counter2 = False Then
        Label10.Caption = "2"
        Counter2 = True
        Check2.Value = 1
    Else
        Label10.Caption = "3"
        Counter3 = True
        Check3.Value = 1
    End If
    If Len(val) = 1 Then
      digit1 = val
      word_cnt = 1
    ElseIf Len(val) = 2 Then
      digit1 = Mid(val, 2, 1)
      digit2 = Mid(val, 1, 1)
      word_cnt = 3
    Else
      digit1 = Mid(val, 3, 1)
      digit2 = Mid(val, 2, 1)
      digit3 = Mid(val, 1, 1)
      word_cnt = 5
    End If
    CounterQ = Label10.Caption
    wording = 1
    MSComm1.Output = "U" + Form1.Label8.Caption + Label10.Caption
End If
    'Print Out
    'Form2.Command1.Visible = False
    'Form2.PrintForm
    'Form2.Command1.Visible = True
End Sub

Private Sub Command8_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Three.Text = SFile
End Sub

Private Sub Command9_Click()
Dim SFile As String
    With CommonDialog1
        .FileName = ""
        .DialogTitle = "Open"
        .CancelError = False
        .Filter = "WAV Files (*.WAV)|*.WAV"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        SFile = .FileName
    End With
    Four.Text = SFile
End Sub

Private Sub Form_load()
MSComm1.PortOpen = True
Media.Volume = 0
buff = 2
Recent = 0
Queue = 0
CounterQ = 0
Counter1 = True
Counter2 = True
Counter3 = True
wording = 6
Text1.Text = Timer1.Interval
Text3.Text = Recent
Text4.Text = Queue
Load Form2
Form2.Show
Form1.Hide
Form1.WindowState = 2
MSComm1.Output = "U0000"
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload Form2
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Label11_Change()
Form2.Label1.Caption = Label11.Caption
End Sub

Private Sub MSComm1_OnComm()
Dim dat As String
dat = MSComm1.Input
If dat >= "1" And dat <= "3" Then
 Text2.Text = dat
 Command6.Value = True
End If
End Sub

Private Sub Timer1_Timer()
Timer1.Enabled = False
If wording < 6 Then
  If wording = 1 Then
     Timer1.Interval = D_invite.Text
     Media.FileName = Invite.Text
     wording = wording + 1
  ElseIf wording = 2 Then
     If word_cnt = 1 Then
       If digit1 = "1" Then
        If digit2 > "0" Then
            Timer1.Interval = D_Ed.Text
            Media.FileName = Ed.Text
        Else
            Timer1.Interval = D_One.Text
            Media.FileName = One.Text
        End If
       ElseIf digit1 = "2" Then
       Timer1.Interval = D_Two.Text
       Media.FileName = Two.Text
       ElseIf digit1 = "3" Then
       Timer1.Interval = D_Three.Text
       Media.FileName = Three.Text
       ElseIf digit1 = "4" Then
       Timer1.Interval = D_Four.Text
       Media.FileName = Four.Text
       ElseIf digit1 = "5" Then
       Timer1.Interval = D_Five.Text
       Media.FileName = Five.Text
       ElseIf digit1 = "6" Then
       Timer1.Interval = D_Six.Text
       Media.FileName = Six.Text
       ElseIf digit1 = "7" Then
       Timer1.Interval = D_Seven.Text
       Media.FileName = Seven.Text
       ElseIf digit1 = "8" Then
       Timer1.Interval = D_Eight.Text
       Media.FileName = Eight.Text
       ElseIf digit1 = "9" Then
       Timer1.Interval = D_Nine.Text
       Media.FileName = Nine.Text
       Else
       End If
       word_cnt = word_cnt - 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  wording = wording + 1
ElseIf word_cnt = 2 Then
   If digit2 > "0" Then
       Timer1.Interval = D_Sib.Text
       Media.FileName = Sib.Text
   End If
   word_cnt = word_cnt - 1
ElseIf word_cnt = 3 Then
  If digit2 = "1" Then
  ElseIf digit2 = "2" Then
  Timer1.Interval = D_Yee.Text
  Media.FileName = Yee.Text
  ElseIf digit2 = "3" Then
  Timer1.Interval = D_Three.Text
  Media.FileName = Three.Text
  ElseIf digit2 = "4" Then
  Timer1.Interval = D_Four.Text
  Media.FileName = Four.Text
  ElseIf digit2 = "5" Then
  Timer1.Interval = D_Five.Text
  Media.FileName = Five.Text
  ElseIf digit2 = "6" Then
  Timer1.Interval = D_Six.Text
  Media.FileName = Six.Text
  ElseIf digit2 = "7" Then
  Timer1.Interval = D_Seven.Text
  Media.FileName = Seven.Text
  ElseIf digit2 = "8" Then
  Timer1.Interval = D_Eight.Text
  Media.FileName = Eight.Text
  ElseIf digit2 = "9" Then
  Timer1.Interval = D_Nine.Text
  Media.FileName = Nine.Text
  Else
  End If
  word_cnt = word_cnt - 1
ElseIf word_cnt = 4 Then
   If digit2 > "0" Then
       Timer1.Interval = D_Roy.Text
       Media.FileName = Roy.Text
   End If
   word_cnt = word_cnt - 1
Else
  If digit3 = "1" Then
  Timer1.Interval = D_One.Text
  Media.FileName = One.Text
  ElseIf digit3 = "2" Then
  Timer1.Interval = D_Two.Text
  Media.FileName = Two.Text
  ElseIf digit3 = "3" Then
  Timer1.Interval = D_Three.Text
  Media.FileName = Three.Text
  ElseIf digit3 = "4" Then
  Timer1.Interval = D_Four.Text
  Media.FileName = Four.Text
  ElseIf digit3 = "5" Then
  Timer1.Interval = D_Five.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Media.FileName = Five.Text
        ElseIf digit3 = "6" Then
        Timer1.Interval = D_Six.Text
        Media.FileName = Six.Text
        ElseIf digit3 = "7" Then
        Timer1.Interval = D_Seven.Text
        Media.FileName = Seven.Text
        ElseIf digit3 = "8" Then
        Timer1.Interval = D_Eight.Text
        Media.FileName = Eight.Text
        ElseIf digit3 = "9" Then
        Timer1.Interval = D_Nine.Text
        Media.FileName = Nine.Text
        Else
        End If
        word_cnt = word_cnt - 1
      End If
   ElseIf wording = 3 Then
      Timer1.Interval = D_At.Text
      Media.FileName = At.Text
      wording = wording + 1
   ElseIf wording = 4 Then
        If CounterQ = "1" Then
        Timer1.Interval = D_One.Text
        Media.FileName = One.Text
        ElseIf CounterQ = "2" Then
        Timer1.Interval = D_Two.Text
        Media.FileName = Two.Text
        ElseIf CounterQ = "3" Then
        Timer1.Interval = D_Three.Text
        Media.FileName = Three.Text
        End If
        wording = wording + 1
   Else
      Timer1.Interval = D_Ka.Text
      Media.FileName = Ka.Text
      wording = wording + 1
   End If
     Timer1.Enabled = True
Else
     Timer1.Enabled = True
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code 2

```
Private Sub Command1_Click()
Form1.Command7.Value = True
End Sub

Private Sub CONFIG_Click(Index As Integer)
Form1.Show
Form2.Hide
End Sub

Private Sub Form_load()
Form2.WindowState = 2
Label4.Caption = Date
Label5.Caption = time
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload Form1
End Sub



Private Sub Timer1_Timer()
Label4.Caption = Date
Label5.Caption = time
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.0 DEVICE OVERVIEW

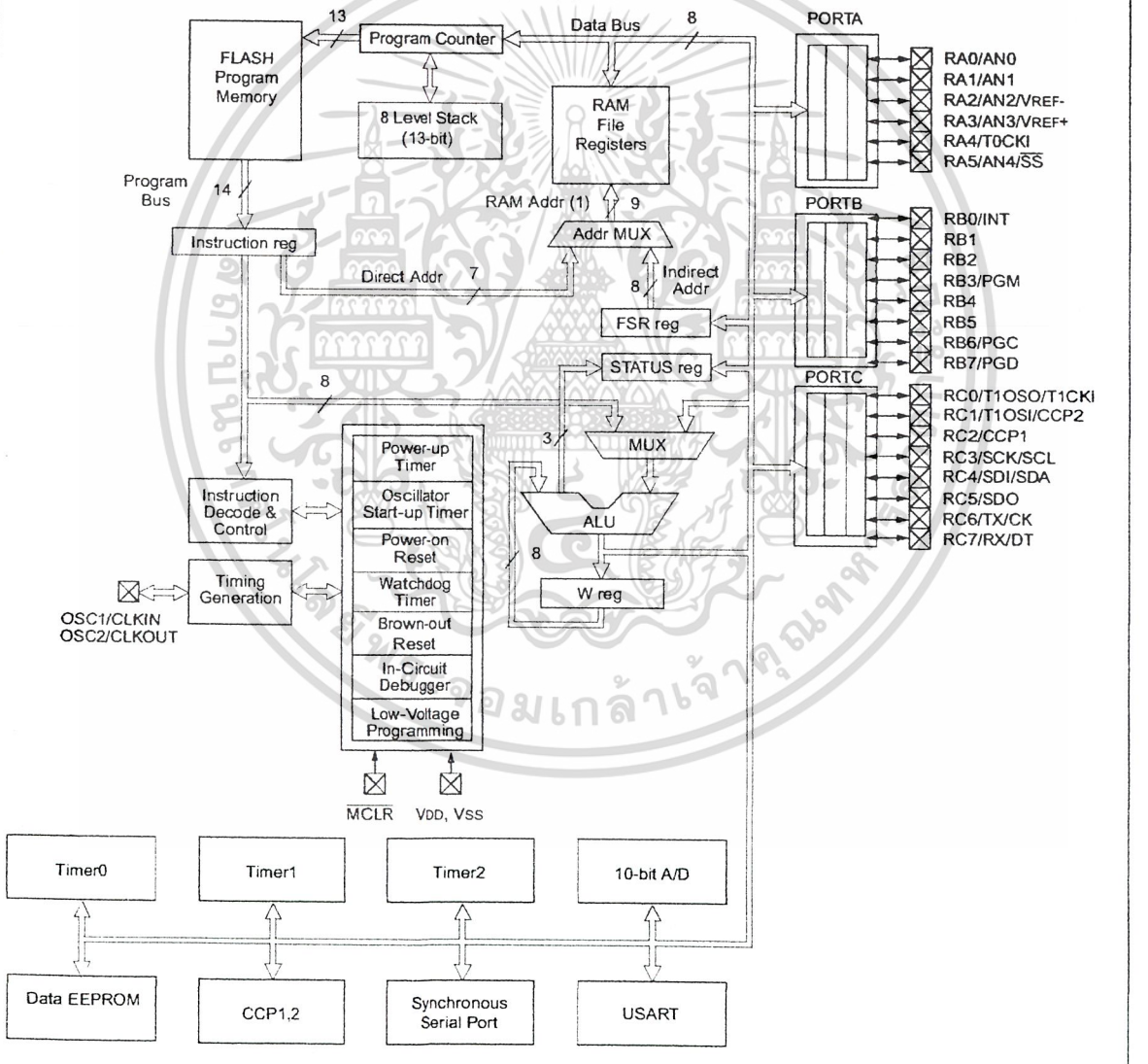
This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The 28-pin devices do not have a Parallel Slave Port implemented.

The following two figures are device block diagrams sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes

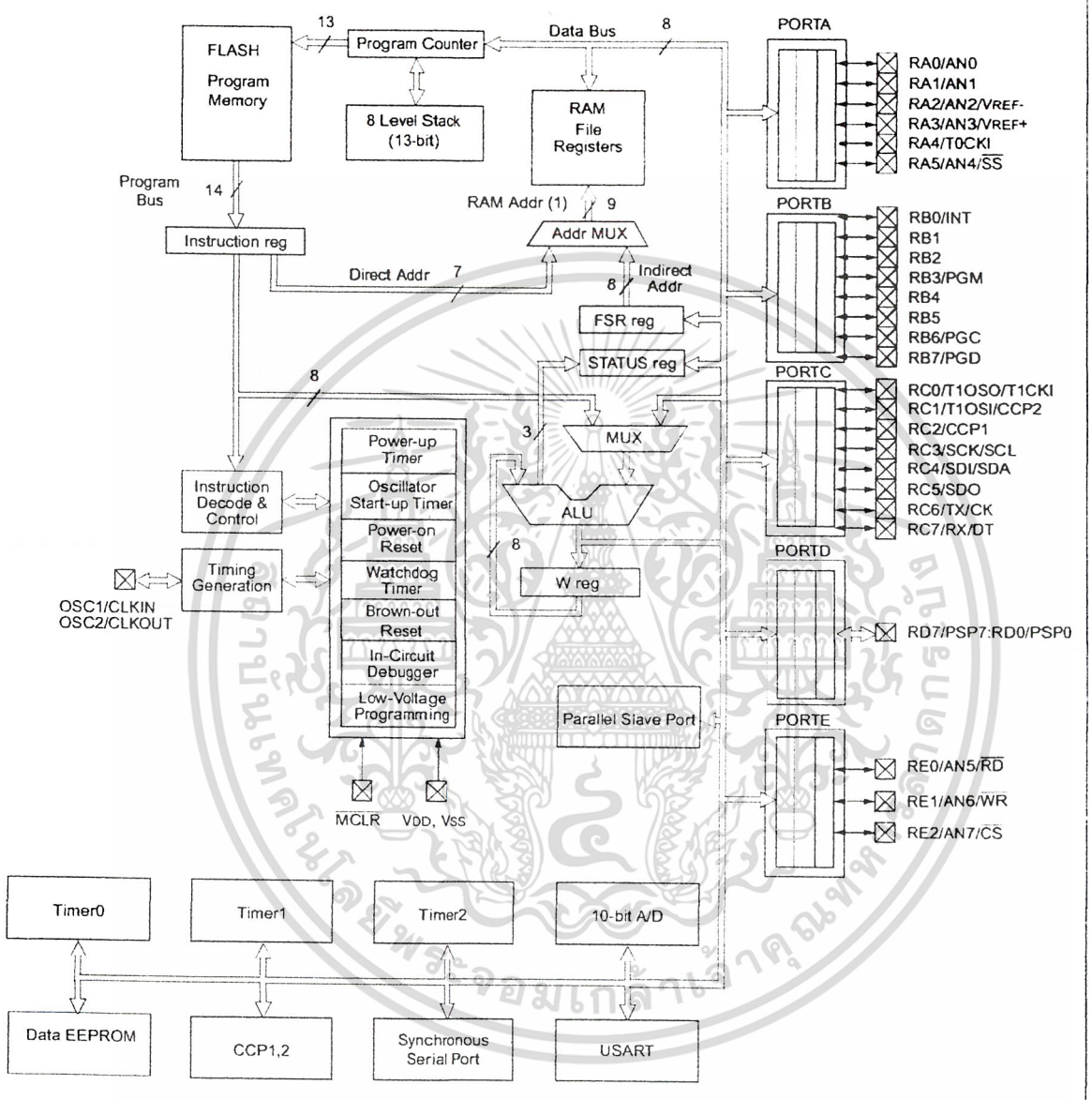


Note 1: Higher order bits are from the STATUS register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

© 1999 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/SS/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input
RB4	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	I/O	TTL	Interrupt on change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS(4)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP/THV	1	2	18	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input
RB4	37	41	14	I/O	TTL	Interrupt on change pin.
RB5	38	42	15	I/O	TTL	Interrupt on change pin.
RB6/PGC	39	43	16	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: *This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.*

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้