สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

มือกล 2

ROBOT ARM 2



นายประมวล แพงทรัพย์

นายไพโรจน์ ศิริกระจาย

เลขหมู่.....
เลขทะเบียน 42317
วัน, เดือน, ปี 16 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT ARM 2

MR.PRAMUAL PHAENGSAP
MR.PHAIROTE SIRIKRAJAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF THE TECHNOLOGY TELECOMMUNICATION

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	มือกล 2 (ROBOT ARM 2)
นักศึกษา	1. นายประมวล แพงทรัพย์ เลขประจำตัว 42015640
	2. นายไพโรจน์ ศิริกระจาย เลขประจำตัว 42015651
อาจารย์ที่ปรึกษา	อ.บุณย์ชนะ ภู่ระหงษ์
ระดับการศึกษา	ปริญญาอุตสาหกรรมศาสตรบัณฑิต
ปีการศึกษา	2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

.....อาจารย์ที่ปรึกษา
(อาจารย์บุณย์ชนะ ภู่ระหงษ์)

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ มือกล 2

โดย 1. นายประมวล แพงทรัพย์ เลขประจำตัว 42015640

2. นายไพโรจน์ ศิริกระจาย เลขประจำตัว 42015651

อาจารย์ที่ปรึกษา อ.บุณย์ชนะ ภู่ระหงษ์

ระดับการศึกษา ปริญญาอุตสาหกรรมศาสตรบัณฑิต

ปีการศึกษา 2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้กล่าวถึงการควบคุมมือกลให้สามารถหยิบหมากฮอสเล่นแข่งกับคนได้โดยอาศัย MICROCONTROLLER MCS-51 เป็นตัวประมวลผลร่วมกับ COMPUTER โดยใช้โปรแกรม VISUAL BASIC 6 เพื่อเป็นการพัฒนามือกลให้สามารถทำงานได้เองโดยอัตโนมัติโดยไม่ต้องมีมนุษย์มาควบคุมและสามารถที่จะนำไปประยุกต์ใช้งานได้ในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT ROBOT ARM 2

NAME 1. MR.PRAMUAL PHAENGSAP ID. 42015640
2. MR.PHAIROTE SIRIKRAJAI ID. 42015651

ADVISOR MR.BOONCHANA PHOORAHONG
LEVEL OF STUDY BACHALOR'S DEGREE IN INDUSTRIAL
ACADEMIC YEAR 2544

ABSTRACT

In this thesis,we purposed the machine for compatible checks with human. The compatible was processed by microcontroller mcs-51 and computer that used visual basic 6 program.For develop the machine hand to working automatic without human and can apply to used in factory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี เกิดจากการร่วมมือร่วมใจของผู้จัดทำ โดยได้รับความช่วยเหลือและคำแนะนำต่างๆเป็นอย่างดีจากอาจารย์ บุณย์ชนะ ภู่ระหงษ์ และอาจารย์อีกหลายท่านรวมทั้งเพื่อนๆที่ได้ให้ความช่วยเหลือในด้านต่างๆ

ท้ายนี้คณะผู้จัดทำ ขอขอบพระคุณ คุณพ่อ คุณแม่ที่ได้ให้ชีวิตลูกมา ขอขอบพระคุณอาจารย์ทุกท่านที่ช่วยประสาทวิชาให้ความรู้ต่างๆจนสามารถทำโครงงานนี้ได้

คณะผู้จัดทำ

12 ตุลาคม 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง **หน้า**

บทคัดย่อ ก

ABSTRACT ข

กิตติกรรมประกาศ ค

สารบัญ ง

สารบัญรูป จ

สารบัญตาราง ฉ

บทที่ 1 บทนำ 1

1.1 แนวความคิดและที่มา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 1

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3

2.1 พอร์ตสื่อสารอนุกรม RS-232 3

2.1.1 การสื่อสารอนุกรม 3

2.1.2 UART 10

2.1.3 วงจรภายในและรีจีสเตอร์ของพอร์ตอนุกรม RS-232 17

2.1.4 การจัดสรรตำแหน่งแอดเดรสบนเครื่อง PC 27

2.2 ไมโครคอนโทรลเลอร์ MCS-51 28

2.2.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51 28

2.2.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ MCS-51 29

2.2.3 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 33

2.2.4 การเชื่อมโยง 8255 กับ MCS-51 38

2.3 Sensor 39

2.4 การต่อใช้งาน MAX232 41

2.5 Visual Basic6 43

2.5.1 ทฤษฎีการทำงานของ Visual Basic6 43

2.5.2 การควบคุม MSComm 55

2.6 หมากฮอส 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 ประวัติหมากฮอส 60
2.6.2 กติกาการเล่นหมากฮอส 61
บทที่ 3 หลักการทำงานและทดลองผล 64
3.1 บล๊อคไดอะแกรมการทำงาน 64
3.2 Flow Chart MCS-51 65
3.3 Flow Chart Visual Basic6 66
3.4 หลักการออกแบบ 67
3.5 การทดลองและผลการทดลอง 68
บทที่ 4 สรุปวิจารณ์และการใช้งาน 69
ภาคผนวก
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 1 รูปแบบอย่างง่ายของข้อมูลแบบอนุกรม	3
รูปที่ 2 รูปแบบอย่างง่ายของข้อมูลแบบอะซิงโครนัส	4
รูปที่ 3 การจัดขาของคอนเน็คเตอร์ของพอร์ตอนุกรม	7
รูปที่ 4 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรม	8
รูปที่ 5 บล็อคไดอะแกรมการทำงานภายในของพอร์ตอนุกรม	13
รูปที่ 6 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม	25
รูปที่ 7 แสดงขาที่ต่อใช้งานของ MCS-51	30
รูปที่ 8 แสดงโครงสร้างภายในของ MCS-51	33
รูปที่ 9 โครงสร้างภายในและการจัดขาของ 8255	34
รูปที่ 10 การเชื่อมโยง 8255 กับ MCS-51	38
รูปที่ 11 วงจรภาครับ-ส่ง SENSOR	39
รูปที่ 12 วงจรใช้งานภาครับของ SENSOR	40
รูปที่ 13 แสดงการต่อ MAX232	41
รูปที่ 14 โครงสร้างภายในและขาต่างๆของ MAX 232	42
รูปที่ 15 หน้าตาของ VISUAL BASIC 6	43
รูปที่ 16 เครื่องมือสร้าง TEXT BOX	47
รูปที่ 17 เครื่องมือสร้าง LABEL	48
รูปที่ 18 เครื่องมือสร้าง COMMAND BUTTON	49
รูปที่ 19 เครื่องมือสร้าง OPTION BUTTON	50
รูปที่ 20 เครื่องมือสร้าง FRAME	50
รูปที่ 21 เครื่องมือสร้าง TIMER	51
รูปที่ 22 เครื่องมือสร้าง IMAGE	51
รูปที่ 23 เครื่องมือสร้าง PICTURE	51
รูปที่ 24 เครื่องมือสร้าง MSCOMM CONTROL	53
รูปที่ 25 บล๊อคไดอะแกรมการทำงาน	64
รูปที่ 26 FLOW CHART การทำงานของ MCS-51	65
รูปที่ 27 FLOW CHART การทำงานของ VISUAL BASIC 6	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1 แสดงพาราตี้ของข้อมูล	5
ตารางที่ 2 เปรียบเทียบ DB9 กับ DB25	7
ตารางที่ 3 แสดงรีจีสเตอร์ ซีเร็ค	13
ตารางที่ 4 แสดงข้อมูลในแอดเดรส 0000:0411H	26
ตารางที่ 5 แสดงการจัดสรรแอดเดรสบนเครื่อง PC	27
ตารางที่ 6 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS 51	29
ตารางที่ 7 แสดงหน้าที่พิเศษของแต่ละขาของ Port P3	31
ตารางที่ 8 การทำงานในโหมดต่างๆ	35
ตารางที่ 9 โหมดการทำงานของ 8255	35
ตารางที่ 10 ตารางความจริงของ 8255	37
ตารางที่ 11 I/O แอดเดรสของ 8255	37
ตารางที่ 12 อุปกรณ์ที่ต้องใช้ของ Max 232	42
ตารางที่ 13 CommEvent เกี่ยวกับสถานะการผิดพลาดในการสื่อสาร	55
ตารางที่ 14 CommEvent เกี่ยวกับสถานะการผิดพลาดเมื่อมีการสื่อสาร	56
ตารางที่ 15 Handshake Property	56
ตารางที่ 16 Input Mode Property	56
ตารางที่ 17 MS Comm Control Property	58
ตารางที่ 18 รายละเอียดที่บ่งชี้ถึงความผิดพลาดในการใช้ MS Comm Control	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวความคิดและที่มา

ในปัจจุบันนี้ความก้าวหน้าทางเทคโนโลยีได้พัฒนาไปอย่างรวดเร็วมนุษย์เราได้มีการนำเครื่องอำนวยความสะดวกมาใช้ในการดำเนินชีวิตประจำวัน ซึ่งส่วนมากจะเป็นการควบคุมอุปกรณ์ต่างๆในลักษณะกึ่งอัตโนมัติ โดยการนำเอาคอมพิวเตอร์มาประยุกต์ใช้กับระบบต่างๆอย่างมากมาย และในกลุ่มของโรงงานอุตสาหกรรมซึ่งก็ได้พัฒนาไปอย่างรวดเร็วได้โดยมีการนำอุปกรณ์หรือเครื่องจักรที่ทันสมัยมาใช้ในโรงงานอุตสาหกรรมในบางโรงงานอาจจะมีความเสี่ยงในการทำงานของพนักงานสูง เช่น ในโรงงานที่เกี่ยวกับพวกสารเคมีต่างๆ หรือบางโรงงานที่ต้องทำงานซ้ำๆ ซากๆ ดังนั้นจึงเกิดโครงการนี้ขึ้นมาเพื่อสร้างหุ่นยนต์ขึ้นมาทำงานแทนคนและหุ่นยนต์จะทำงานได้ต้องมีตัวควบคุมโดยอัตโนมัติ จึงได้คิดค้นพัฒนาโปรแกรมขึ้นเพื่อควบคุมหุ่นยนต์ให้ทำงานตามต้องการได้

การทำงานของโครงการนี้เป็นการนำเอาคอนโทรลเลอร์ MCS-51 มาควบคุมมือกลให้สามารถหยิบหมากฮอสและสามารถที่จะเล่นตอบโต้กับมนุษย์ได้ โดยได้นำเอาคอมพิวเตอร์และโปรแกรม Visual Basic6มาประยุกต์ใช้ด้วยโดยจะเป็นการ อินเตอร์เฟสกันระหว่างคอมพิวเตอร์กับกระดานหมากฮอสซึ่งควบคุมโดยคอนโทรลเลอร์ MCS-51 และโปรแกรม Visual Basic6

1.2 วัตถุประสงค์

- เพื่อศึกษาถึงวิธีการพัฒนามือกลให้สามารถทำงานได้เองโดยอัตโนมัติโดยมีความถูกต้องและแม่นยำ
- เพื่อศึกษาถึงวิธีการเขียนโปรแกรมคอนโทรล MCS-51และการอินเตอร์เฟสกับคอมพิวเตอร์
- เพื่อศึกษาการเขียนโปรแกรมวิชวลเบสิกกับการติดต่อกับอุปกรณ์ภายนอก
- สามารถที่จะประยุกต์ใช้ในโรงงานอุตสาหกรรม

1.3 ขอบเขตของโครงงาน

- เขียนโปรแกรมหมากฮอสให้คอมพิวเตอร์เล่นกับคนได้ในระดับหนึ่งโดยใช้โปรแกรมวิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เขียนโปรแกรมการอิเตอร์เฟสระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกโดยใช้ MCS-51 กับ วิชวลเบสิก และแสดงผลทางหน้าจอคอมพิวเตอร์
- เขียนโปรแกรม MCS-51 ให้ควบคุมมือกลจับหมากฮอสเดินโดยใช้หมากฮอสที่เขียนขึ้นมาเองจากโปรแกรมวิชวลเบสิก



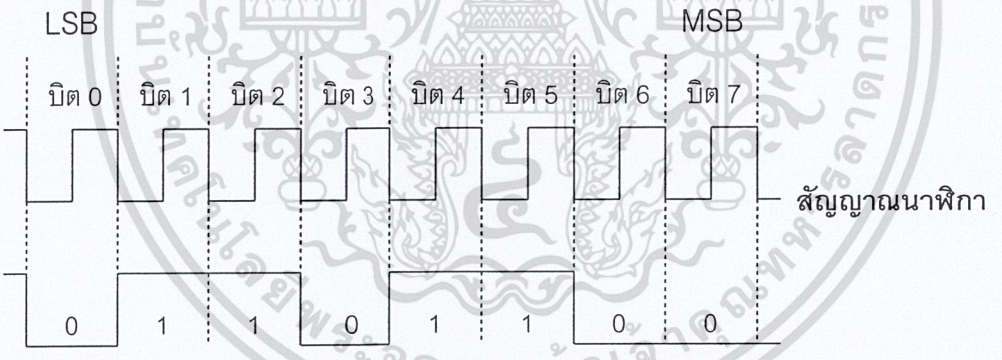
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 พอร์ตสื่อสารอนุกรม RS-232

2.1.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) และการสื่อสารอนุกรมแบบอะซิงโครนัส (Asynchronous) การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา,ข้อมูลและกราวด์ รูปที่ 1 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบ ซิงโครนัส



รูปที่ 1 รูปแบบอย่างง่ายของข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอะซิงโครนัส

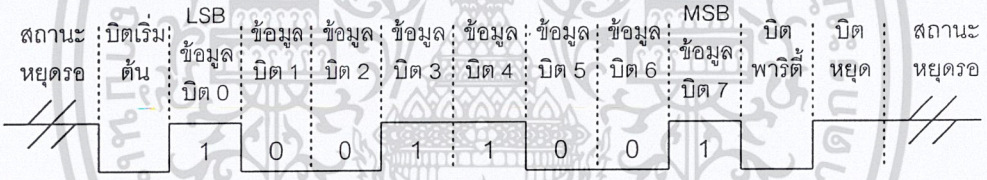
การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณ นาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per second :bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

รูปที่ 2 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATAจะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขา DATA มีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 2 รูปแบบอย่างง่ายของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษ ที่ได้รับการออกแบบมาสำหรับ การรับส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้จะเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 บิต ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

ตารางที่ 1 แสดงบิตพาริตี้ของข้อมูล

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd) ,แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก “1” ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่ หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในบิตพาริตี้ จะต้องมีลอจิก “0” แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้อง เป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิทที่เป็นลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 1 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UART เบอร์ 8250 UART ชิปเหล่านี้มีระดับแรงดันเป็นแบบทีทีแอล (0 และ +5V) แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ที่ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก "0" มีระดับแรงดัน +3V ถึง +12V ในขณะที่ลอจิก "1" มีระดับแรงดัน –3V จนถึง –12V

มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทางโดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดหนึ่งซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association :EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็คเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ –3 ถึง –12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

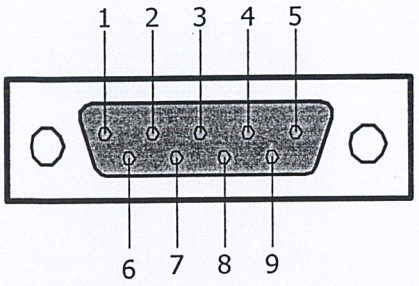
ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็คเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็คเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็คเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

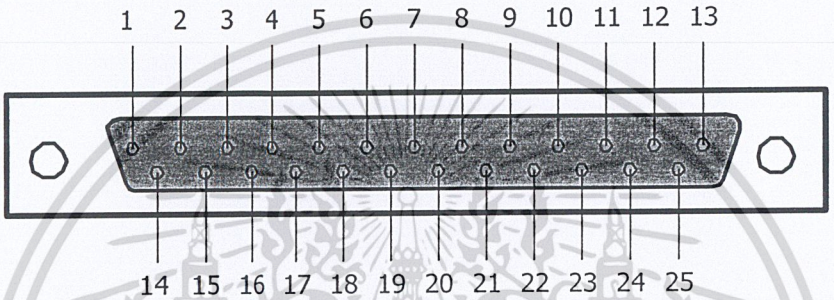
คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็คเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็คเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คอนเน็คเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



คอนเน็คเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

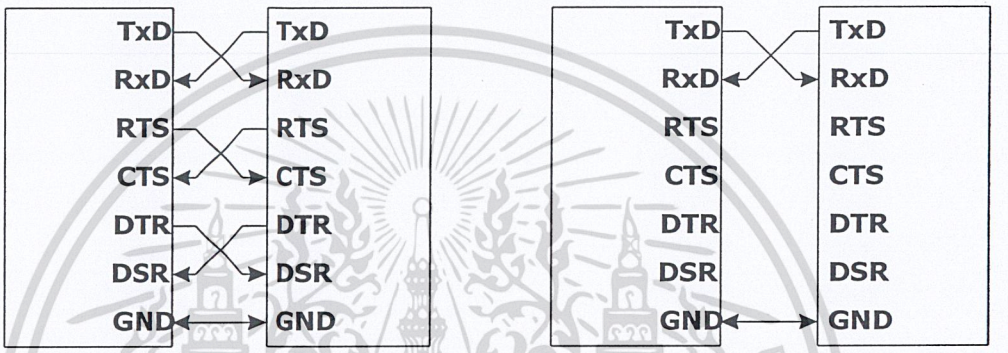
รูปที่ 4 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

คอนเน็คเตอร์ DB-9	คอนเน็คเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect :DCD	อินพุท
2	3	Received Data :RxD	อินพุท
3	2	Transmitted Data : TxD	เอาท์พุท
4	20	Data Terminal Ready:DTR	เอาท์พุท
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุท
7	4	Request To Send : RTS	เอาท์พุท
8	5	Clear To Send : CTS	อินพุท
9	22	Ring Indicator : RI	อินพุท

ตารางที่ 2 เปรียบเทียบ DB9 กับ DB25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังรูปที่ 4 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ (ก) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือ แฮนด์เช็คเต็มรูปแบบ ส่วนในรูปที่ (ข) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem

(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3เส้น

รูปที่ 4 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

<u>รายละเอียดการทำงานในแต่ละขาของพอร์ทอนุกรม RS-232</u>

- **Data Carrier Detect : DCD** หรืออาจเรียกว่า **Carrier Detect : CD** ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
- **Receive Data : RD** หรือ **RxD** ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์
- **Transmitted Data : TD** หรือ **TxD** ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- **Data Terminal Ready : DTR** เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

- **Signal Ground : GND** ขากราวด์ของระบบ
- **Data Set Ready : DSR** ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจาก DTR
- **Request To Send : RTS** เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- **Clear To Send : CTS** ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- **Ring Indicator : RI** ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้,เฟรมข้อมูล,โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1-65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

PC16550 Unversal Asynchronous Receiver / Transmitter with FIFOs

ลักษณะทั่วไป

PC16550 เกิดจากการปรับปรุงและแก้ไข PC16450 โดยสิ่งที่ทำได้ใน PC16450 ก็สามารถทำได้ใน PC16550 เช่นกัน และมี FIFO โหมดเพื่อลดการทำงานในส่วน Software ลง CPU จึงทำงานน้อยลง หน้าที่ของ UART คือแปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนาน เมื่อรับข้อมูลจากภายนอกหรือ MODEM และแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรมเมื่อรับข้อมูลมาจาก CPU และ CPU สามารถอ่านสถานะการทำงานของ UART ได้ตลอดเวลาแม้ในเวลาที่กำลังรับหรือส่งข้อมูล สถานะต่างๆได้แก่ รูปแบบ,สภาวะ ของการส่งและรับ รวมถึงการผิดพลาดต่างๆ (Parity , Overrun ,Framing ,Break interrupt) UART มีโปรแกรมสร้าง Baud rate ที่สามารถหารความถี่จากความถี่อ้างอิงได้ และสามารถควบคุม MODEM ได้เป็นอย่างดี และใช้งานแบบ Interrupt ได้

ลักษณะสำคัญ

- สามารถใช้งานกับ Software ของ 16450 ได้
- ขาต่อใช้งานเหมือนกับ 16450 ยกเว้น ขา CSOUT (24) กับขา NC (29) เปลี่ยนเป็น TXRDY (24) และ RXRDY (29) ตามลำดับ
- เมื่อ RESET ทุก Registers และทุกการแสดงผลจะเหมือนกับ 16450 ทุกอย่าง
- ในโหมด FIFO ทั้งการส่งและรับในแต่ละอันจะมี Buffer 16 Byte เพื่อลดการทำงานของ CPU ลง
- สามารกำหนด Start bit ,Stop bit และ Parity bit ได้
- การ Holding และ Shift Register ใน 16450 ถูกตัดออกไปเพื่อความถูกต้องของการรับและส่งข้อมูลระหว่าง CPU กับข้อมูลแบบอนุกรม
- สามารถควบคุมหรือติดต่อกับ Register ต่างๆในตัว 16550 ได้ตลอดเวลา
- สามารถตั้ง Baud rate ได้ด้วยการหารความถี่จากความถี่อ้างอิงโดยกำหนดตัวหารได้ตั้งแต่ 1 ถึง $(2^{16}-1)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถใช้ความถี่ภายนอกมาเป็นความถี่อ้างอิงได้
- มี Function ที่ใช้ควบคุม MODEM (CTS ,RTS ,DSR ,DTR ,RI และ DCD)
- สามารถโปรแกรมการส่งข้อมูลได้
- ขนาดของข้อมูล 5 ,6 ,7 หรือ 8 bit
- Even ,Odd หรือ no-parity ทั้งรับและส่ง
- 1 , 1.5 หรือ 2 bit
- Baud generation (DC to 1.5M Baud)
- สามารถตรวจสอบ Start bit ที่ผิดพลาดได้
- มีการบอกสถานะการทำงานอย่างละเอียด
- ใช้บัสแบบ 3 สถานะ
- สามารถสร้างและตรวจจับสัญญาณ Break ได้
- มีการเรียงลำดับการ Interrupt ได้

การใช้งานขาต่างๆ

A0, A1, A2 ,Register Select ,ขา 26, 27,28

ใน UART มี Register ที่ใช้แสดงผลและควบคุมการทำงานทั้งหมดอยู่ 8 ตัว โดยใช้ขาทั้ง 3 ขานี้เป็นตัวกำหนดว่า CPU ต้องการติดต่อกับ Register ตัวไหน ดังในตาราง

DLAB	A2	A1	A0	Register
0	0	0	0	Receiver Buffer (read), Transmitter Holding (write)
0	0	0	1	Interrupt Enable
X	0	1	0	Interrupt Identification (read)
X	0	1	0	FIFO Control (Write)
X	0	1	1	Line Control
X	1	0	0	MODEM Control
X	1	0	1	Line Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X	1	1	0	MODEM Status
X	1	1	1	Scratch
1	0	0	0	Divisor Latch (Least)
1	0	0	1	Divisor Latch (most)

ตารางที่ 3 แสดง Register Select

-ADS ,Address Strobe , ขา 25

เมื่อต้องการใช้งานขา A0, A1, A2, CS0, CS1, CS2 จะต้องป้อน Logic "0" ให้ขา –ADS แต่ถ้าขา –ADS มี Logic "1" จะไม่สามารถใช้งานขาทั้ง 6 ขานั้นได้

-BAUDOUT ,Baud out ,ขา 15

เป็นขาที่ส่งค่าความถี่ออกมาจากส่วนส่งข้อมูลอนุกรมของ UART ความถี่ที่ส่งออกมามีค่าเท่ากับ 16 X ความถี่อ้างอิง และความถี่นี้จะถูกหารด้วยค่าที่เราโปรแกรมไว้ใน Register Divisor Latch จึงกลายเป็น Baud rate และขานี้ยังใช้ในส่วนรับข้อมูลได้โดยการต่อขานี้เข้ากับขา RCLK

CS0, CS1, -CS2, Chip Select, ขา 12, 13, 14

เมื่อ CS0, CS1 ได้รับ Logic "1" และ –CS2 ได้รับ Logic "0" CPU จะสามารถติดต่อกับ UART ได้ (แต่ขา –ADS จะต้องได้รับ Logic "0" ด้วย)

-CTS, Clear to Send, ขา 36

เป็นขา Output เมื่อมีสถานะเป็น "0" หมายความว่าพร้อมที่จะรับข้อมูลแล้วเป็นการบอกให้อุปกรณ์ที่ต่ออยู่ด้วยรู้ว่าพร้อมรับข้อมูลแล้วให้ส่งข้อมูลมาได้ และถ้าเป็น "1" หมายถึงยังไม่พร้อมให้หยุดการส่งข้อมูลมาก่อน

D7-D0, Data Bus, ขา 1 – 8

ใช้ Buffer แบบ 3 สถานะ("1", "0" และ High Impedance)ในการเชื่อมต่อกับ Bus ของ CPU เพื่อใช้รับและส่งข้อมูลซึ่งกันและกัน

-DCD, Data Carrier Detect, ขา 38

ถ้าเป็น "0" หมายถึงอุปกรณ์ที่ต่ออยู่ด้วยนั้นสามารถตรวจจับข้อมูลที่ส่งไปได้แล้ว ขานี้จะไม่มีความหมายในขณะรับข้อมูล

DDIS, Driver Disable, ขา 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขานี้จะมี Logic เป็น “0” ก็ต่อเมื่อ CPU มีการอ่านข้อมูลจากตัว UART และมันสามารถกำหนดให้ยกเลิกหรือควบคุมการส่งข้อมูลระหว่าง CPU กับ UART ได้

-DSR, Data Set Ready, ขา 37

เมื่อเป็น “0” หมายความว่าอุปกรณ์ภายนอกสามารถติดต่อสื่อสารกับ UART ได้แล้ว โดย CPU จะตรวจสอบสัญญาณนี้จาก MODEM Status Register ในตัว UART

-DTR, Data Terminal Ready, ขา 33

ปกติเป็น “1” และจะเป็น “0”เมื่อตัว UART พร้อมที่จะติดต่อกับอุปกรณ์ที่ต่ออยู่ เมื่อเกิดการ RESET ขานี้จะกลับเป็น “1”

INTR, Interrupt, ขา 30

ปกติเป็น “0” และเมื่อเกิดการ Interrupt จะมีค่าเป็น “1” การ Interrupt อาจเกิดจากการ Error ของ Register ต่างๆ เช่น FIFO เป็นต้น

MR, Master Reset, ขา 35

เมื่อกำลังใช้งานจะต้องส่ง Logic “0” ให้ แต่เมื่อขา Input นี้มี Logic เป็น “1” มันจะทำการ Clear ค่าใน Register ทุก Register ใหม่หมดและขา Output ทุกขาจะอยู่ในสภาวะเริ่มต้นหมดทุกขา

-OUT1, Output 1, ขา 34

ผู้ใช้สามารถนำขานี้ไปใช้งานได้อย่างอิศระจะนำไปใช้ควบคุมอะไรก็ได้ และขานี้สามารถสั่งให้เป็น “1” หรือ “0” ได้โดยการตั้งที่โปรแกรม Set bit ที่ 2 ใน Register MODEM Control

-OUT2, Output 2, ขา 31

การใช้งานเหมือนกับ Output 1 เพียงแต่เปลี่ยนไป set Bit 3 ของ Register MODEM Control แทน

RCLK, Receiver Clock, ขา 9

ขานี้ เป็นขา Input ใช้ต่อกับขา –Baudout

RD, **-RD**, Read, ขา 22, 21

ถ้าขา RD เป็น Logic “1” หรือ ขา –RD เป็น Logic “0” ในขณะที่ Chip select CPU จะสามารถอ่านค่าจาก Register ต่างๆในตัว UART ที่เลือกโดย Address(A2, A1, A0)

-RI, Ring Indicator, ขา 39

เมื่อมีสถานะเป็น “0” เมื่ออุปกรณ์ที่ต่ออยู่ได้รับสัญญาณเรียกจากโทรศัพท์และ CPU สามารถตรวจดูสภาวะนี้ได้ที่ Bit ที่ 6 ของ Register MODEM Status และเมื่อมีการเปลี่ยนสถานะจาก “1’ ไปเป็น “0” จะทำให้เกิดการ Interrupt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-RTS, Request to Send, ขา 32

เมื่อเป็น “0” แสดงว่า UART ต้องการจะส่งข้อมูลให้แก่อุปกรณ์ที่ต่ออยู่ และสามารถควบคุมสัญญาณนี้ได้โดยการ Set bit ที่ 1 ใน Register MODEM Control

SIN, Serial input, ขา 10

เป็นขารับข้อมูลอนุกรมจากอุปกรณ์ต่างๆที่นำมาต่อ เช่น MODEM เป็นต้น

SOUT, Serial Output, ขา 11

เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรมให้แก่อุปกรณ์ที่นำมาต่อ ปกติจะมีสถานะเป็น Logic “1”

-TXRDY, **-RXRDY**, ขา 24, 29

เมื่อใช้โหมด FIFO การรับการส่งแบบ DMA จะถูกแสดงผ่านสองขานี้ และสามารถเลือกโหมดของ DMA ได้ผ่านทาง FCR3 DMA โหมด 0 จะส่งได้อย่างเดียวใน 1 รอบการทำงาน แต่ ใน DMA โหมด 0 สามารถทำได้มากกว่า ใน 16450 สามารถใช้ DMA โหมด 0 เท่านั้น

RXRDY, โหมด 0

เมื่ออยู่ในโหมด 16450 หรือ ใน FIFO โหมด 0 เมื่อมีข้อมูลเข้ามาใน RCVR FIFO หรือ Register RCVR Holding ขา RXRDY (29) จะเปลี่ยนเป็น Logic “0” และเมื่อไม่มีข้อมูลเข้ามาอีกจะเปลี่ยนเป็น Logic “1”

RXRDY, โหมด 1

ใน FIFO โหมด 1 และ เกิด Timeout ขึ้น ที่ขา READY (26) จะมี Logic “0” และเมื่อไม่มีข้อมูลเข้ามา จะเปลี่ยนเป็น Logic “1”

TXRDY, โหมด 0

เมื่ออยู่ในโหมด 16450 หรือ ใน FIFO โหมด 0 เมื่อไม่มีข้อมูลอยู่ใน XMIT FIFO หรือ Register XMIT Holding ขา TXRDY จะมี Logic “0” และจะกลับเป็น “1” อีกครั้งหลังจากข้อมูลตัวแรกถูกโหลดเข้าไปใน XMIT FIFO หรือ holding register

TXRDY, โหมด 1

ใน FIFO โหมด 1 เมื่อไม่มีข้อมูลใน XMIT FIFO จากนั้น ขา –TXRDY จะมีสถานะเป็น Logic “0” และจะกลับเป็น “1” อีกครั้งเมื่อ XMIT FIFO มีข้อมูลเต็ม

VDD, ขา 40

ต่อเข้ากับไฟ +5V ใช้เลี้ยงวงจร

VSS, ขา 20

เป็นขา Ground ของวงจร

WR, **-WR**, Write, ขา 19, 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าขา WD เป็น Logic “1” หรือ ขา –WD เป็น Logic “0” ในขณะที่ Chip select CPU จะสามารถเขียนค่าลงใน Register ต่างๆในตัว UART ที่เลือกโดย Address(A2, A1, A0)

XIN, External Crystal input, ขา 16

สัญญาณ input นี้ใช้ต่อร่วมกับขา XOUT โดยรับการป้อนกลับจากวงจรกำเนิดความถี่ของ Baud Rate

XOUT, External Crystal Output, ขา 17

สัญญาณ output นี้ใช้ต่อร่วมกับขา XIN โดยรับการป้อนกลับจากวงจรกำเนิดความถี่ของ Baud Rate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.1.3 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM1,COM2,COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

ในรูปที่ แสดงไดอะแกรมการทำงานภายในของพอร์ตอนุกรม ซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวนได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM มีแอดแดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ ใช้ในการเซ็ตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อโมเด็ม เช่น RTS หรือ DTR

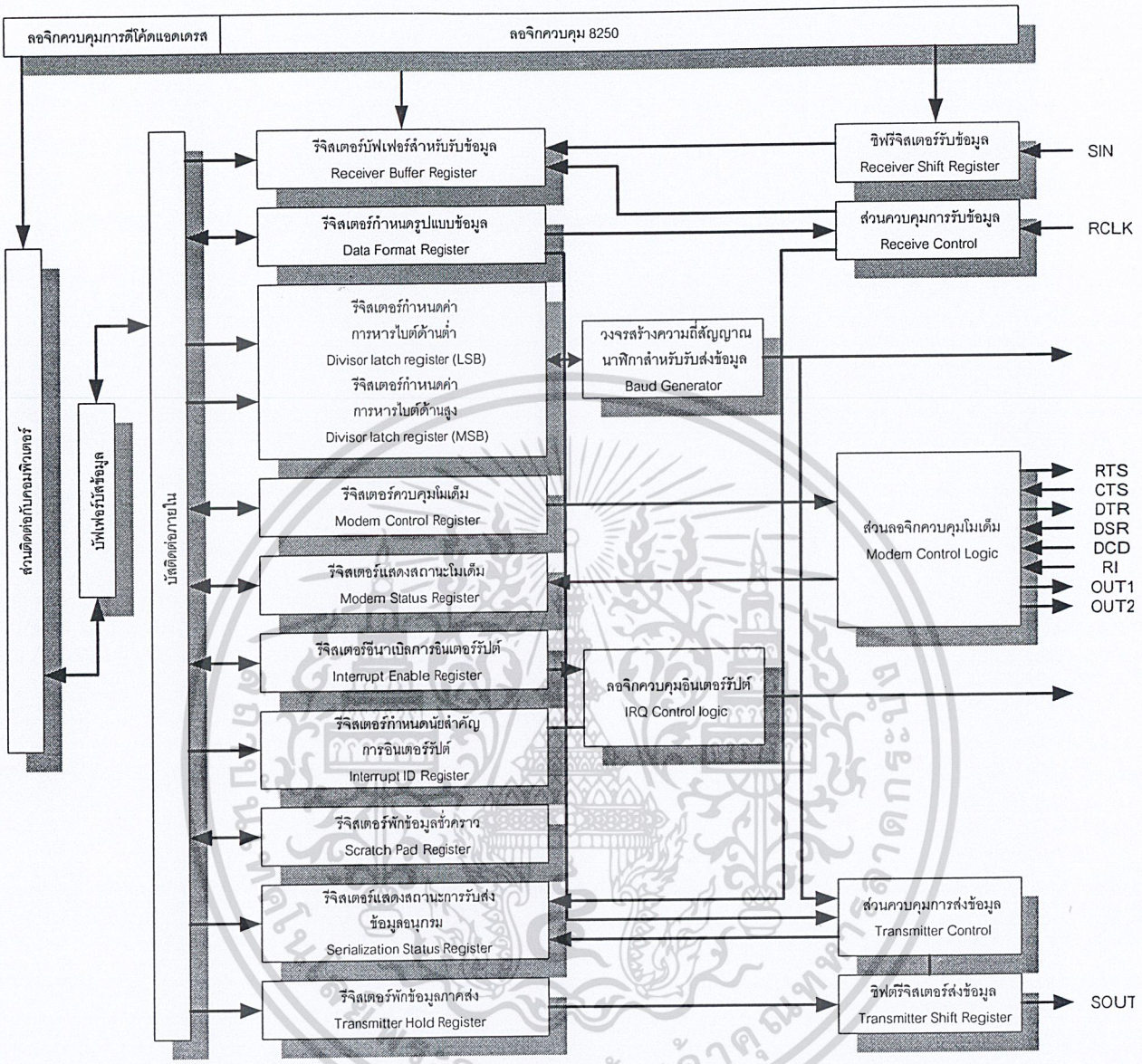
05H รีจิสเตอร์แสดงสถานะ การรับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD ,RI,DSR และ CTS

07H รีจิสเตอร์ทำการเก็บข้อมูลชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42317



รูปที่ 5 ไดอะแกรมการทำงานภายในของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์

รีจิสเตอร์ตำแหน่ง 00H : รีจิสเตอร์บัฟเฟอร์

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาและข้อมูลที่จะส่งออกไป โดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูลที่ต้องการจะส่งจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูลและข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อข้อมูลที่รับเข้ามาเรียบร้อยและแปลงเป็นแบบขนานแล้ว ข้อมูลจะถูกส่งมายังรีจิสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจิสเตอร์นี้ออกไปรีจิสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับการรับข้อมูลในไบต์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ตำแหน่ง 01H : รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์

เป็นรีจิสเตอร์สำหรับการอีนาเบิลการอินเตอร์รัปต์ ซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเตอร์รัปต์ขึ้นมา ฟังก์ชั่นการทำงานในแต่ละบิตของรีจิสเตอร์มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	SINP	ERBK	TBE	RxRD

บิต 4-7 บิตเหล่านี้ไม่ถูกใช้งาน กำหนดให้เท่ากับ “0”

SINP อีนาเบิลการอินเตอร์รัปต์เนื่องจากเกิดการเปลี่ยนสถานะที่ขาอินพุท CTS, DSR, DCD หรือขา RI

ERBK อีนาเบิลการอินเตอร์รัปต์เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจากพาริตี้, โอเวอร์รัน ,เฟรมข้อมูล หรือการเบรกข้อมูล
“1” อีนาเบิลการอินเตอร์รัปต์
“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือ Disable

TBE อีนาเบิลการอินเตอร์รัปต์เมื่อรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง
“1” อีนาเบิลการอินเตอร์รัปต์
“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือ Disable

RxRD อีนาเบิลการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ได้รับข้อมูลเรียบร้อย
“1” อีนาเบิลการอินเตอร์รัปต์
“0” ไม่มีการใช้อินเตอร์รัปต์รูปแบบนี้หรือ Disable

รีจิสเตอร์ตำแหน่ง 02H : รีจิสเตอร์แสดงโหมดและสถานะการอินเตอร์รัปต์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	ID1	ID0	PND

บิต 3-7 ไม่ได้ใช้งาน อ่านค่าได้เท่ากับ “0”

ID1,ID0 ใช้งานร่วมกันเพื่อแจ้งสาเหตุของการเกิดอินเตอร์รัปต์
“00” เกิดการอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุทขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 4

"01" เกิดการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่างขึ้น

การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 3

"10" เกิดการอินเตอร์รัปต์เนื่องจาก ข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว

การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 2

"11" เกิดการอินเตอร์รัปต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรก (break : เกิดการหยุดถ่ายทอดข้อมูลกระทันหัน)

การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 1 หรือมีนัยสำคัญสูงสุด

PND ใช้แสดงสถานะของการเกิดอินเตอร์รัปต์

"1" แสดงว่าไม่มีการอินเตอร์รัปต์

"0" แสดงว่ามีการอินเตอร์รัปต์เกิดขึ้น

เมื่อมีการสร้างสัญญาณอินเตอร์รัปต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดอินเตอร์รัปต์ครั้งต่อไป โดยสามารถทำได้ดังนี้คือ

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุทจะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากบัฟเฟอร์ส่งข้อมูลว่างจะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ ส่งข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเก็บข้อมูลลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อย จะต้องเคลียร์ค่าอินเตอร์รัปต์ โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการเบรก จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านค่ารีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม

รีจิสเตอร์ตำแหน่ง 03H : รีจิสเตอร์กำหนดรูปแบบของข้อมูล

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DLAB ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัฟเฟอร์ (00H)

"1" เป็นการเข้าสู่โหมดการหารค่าบอดเรต

"0" เป็นการเข้าถึงรีจิสเตอร์บัฟเฟอร์ (รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับการอีนาเบิลอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 01H) เมื่อบิต DLAB เป็น "1" รีจิสเตอร์บัฟเฟอร์ (00H) และรีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนดค่าบอดเรต โดยรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหาค่าบอดเรตสามารถเขียนเป็นสมการได้ดังนี้

บอดเรต = 115200/ค่าตัวหาร 16 บิต

ค่าตัวเลข 115200 มาจากความถี่ของคริสตอลในวงจร UART ภายในเครื่องคอมพิวเตอร์ โดยคริสตอลที่ใช้มีความถี่ 1.8432MHz วงจรภายใน UART จะทำการหารค่าความถี่นี้ด้วย 16 ทำให้ได้ค่าความถี่ 115200 Hz ออกมา

ค่าตัวหาร 16 บิต = ข้อมูลในรีจิสเตอร์ 00H + (256 x ข้อมูลในรีจิสเตอร์ 01H)

สมมติว่าต้องการค่าบอดเรตเท่ากับ 9600 ค่าตัวหารที่ใช้จะต้องมีค่าเท่ากับ 12 ซึ่งค่านี้จะต้องถูกโหลดลงในรีจิสเตอร์ 00H และโหลดค่า 0 ลงไปในรีจิสเตอร์ 01H ค่าตัวหารที่ทำให้เกิดค่าบอดเรตสูงสุดที่ 115200 บิตต่อวินาทีคือ ค่า 0001 นั่นคือรีจิสเตอร์ 00H มีค่าเท่ากับ 1 และรีจิสเตอร์ 01H มีค่าเท่ากับ 0

BRK ใช้ควบคุมการหยุดถ่ายทอดข้อมูล

"1" สามารถหยุดหรือเบรกได้

"0" ไม่มีการหยุดหรือเบรกได้

PAR2,PAR1,PAR0 ใช้เพื่อกำหนดบิตพาริตี้

"000" ไม่ใช้บิตพาริตี้

"001" กำหนดพาริตี้คี่

"011" กำหนดพาริตี้คู่

"101" มาร์ค (mark)

"111" ช่องว่าง (space)

STOP ใช้กำหนดจำนวนบิตปิดท้าย

"1" มีบิตปิดท้าย 2 บิต

"0" มีบิตปิดท้าย 1 บิต

DAB1,DAB0 ใช้ร่วมกันในการกำหนดจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"00" จำนวนบิตข้อมูลเท่ากับ 5 บิต

"01" จำนวนบิตข้อมูลเท่ากับ 6บิต

"10" จำนวนบิตข้อมูลเท่ากับ 7 บิต

"11" จำนวนบิตข้อมูลเท่ากับ 8 บิต

รีจิสเตอร์ตำแหน่ง 04H : รีจิสเตอร์ควบคุมโมเด็ม

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

<u>บิต 5-7 ไม่มีการใช้งาน อ่านค่าได้เท่ากับ 0</u>

LOOP "1" อีนาเบิลการส่งค่ากลับ

"0" ดิสเอเบิล (disable)

OUT1,OUT2 "1" อีนาเบิลการใช้งานภายใน

"0" ดิสเอเบิล (disable)

RTS ใช้ควบคุมการทำงานของขา RTS (Ready To Send)

"1" อีนาเบิล

"0" ดิสเอเบิล (disable)

DTR ใช้ควบคุมการทำงานของขา DTR (Data Terminal Ready)

"1" อีนาเบิล

"0" ดิสเอเบิล (disable)

รีจิสเตอร์ตำแหน่ง 05H : รีจิสเตอร์แสดงสถานะการรับส่งข้อมูลอนุกรมของ UART

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อแสดงสาเหตุของการเกิดอินเตอร์รัปต์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	TXE	TBE	BREK	FRME	PARE	OVFE	RxRD

TXE (Transmitter Empty) "1" แสดงว่ารีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" แสดงว่ายังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

TBE (Transmitter Buffer Empty) "1" รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

BREK (Break) "1" UART ตรวจพบการเบรก

"0" ไม่มีการเบรก

FRME (Frame Error) "1" UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล

"0" ไม่พบความผิดพลาดด้านเฟรมข้อมูล

PARE (Parity Error) "1" UART ตรวจพบความผิดพลาดทางพาริตี้

"0" ไม่พบความผิดพลาดทางพาริตี้

OVRE (Overrun Error) "1" UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน

"0" ไม่พบความผิดพลาดแบบโอเวอร์รัน

RxRD (Received Data Ready) "1" มีการรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์

"0" ไม่มีข้อมูล

รีจิสเตอร์ตำแหน่ง 06H : รีจิสเตอร์แสดงสถานะของโมเด็ม

ใช้เพื่อกำหนดสถานะสัญญาณอินพุท ของพอร์ตอนุกรม RS-232 ซึ่งได้แก่ สัญญาณ DCD ,DSR ,CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบเอนกประสงค์

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

DCD ใช้แสดงสภาวะของขา DCD

"1" แสดงว่าที่ขา DCD เป็นลอจิก "1"

"0" แสดงว่าที่ขา DCD เป็นลอจิก "0"

RI ใช้แสดงสภาวะของขา RI

"1" แสดงว่าที่ขา RI เป็นลอจิก "1"

"0" แสดงว่าที่ขา RI เป็นลอจิก "0"

DSR ใช้แสดงสภาวะของขา DSR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"1" แสดงว่าที่ขา DSR เป็นลอจิก "1"

"0" แสดงว่าที่ขา DSR เป็นลอจิก "0"

DCTS (Delta Clear To Send) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

"1" แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDSR (Delta Data Set Ready) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DSR

"1" แสดงว่าบิต DSR (Data Set Ready) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

"1" แสดงว่าบิต RI (Ringing Indicator) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DDCD (Delta Data Carrier Detect) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต DDCD

"1" แสดงว่าบิต CTS (Clear To Send) เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว

DCTS (Delta Clear To Send) ใช้แสดงสภาวะของขา CTS

"1" แสดงว่าที่ขา CTS เป็นลอจิก "1"

"0" แสดงว่าที่ขา CTS เป็นลอจิก "0"

รีจิสเตอร์ตำแหน่ง 07H : รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว

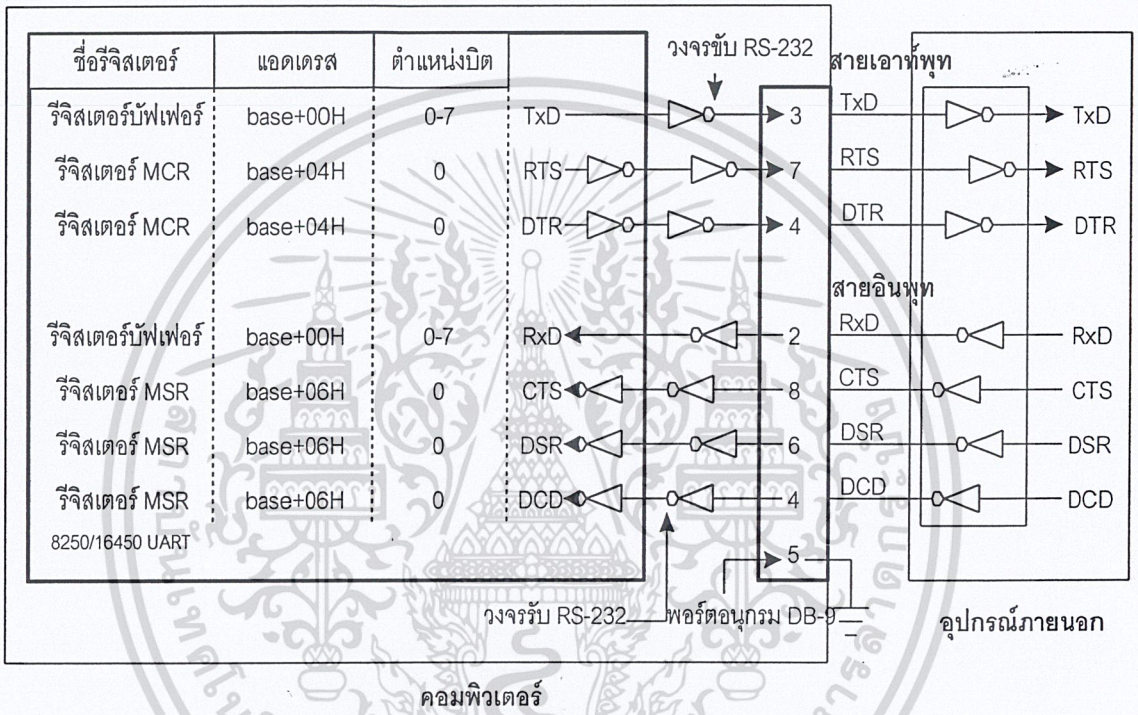
ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใดๆต่อการใช้งาน UART

ลักษณะสัญญาณอินพุทและเอาท์พุทของพอร์ต RS-232

สัญญาณเอาท์พุทที่ใช้ควบคุม (RTS และ DTS) และสัญญาณแสดงสถานะอินพุท (CTS,DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาท์พุทออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมในรูปที่ 6



รูปที่ 6 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออส ภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1 จะเก็บไว้ที่แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM2 = 0000:0402H – 0000:0403H

COM3 = 0000:0404H – 0000:0405H

COM4 = 0000:0406H – 0000:0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วยโดยมีรายละเอียดดังแสดงในตารางที่ 2

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

ตารางที่ 4 แสดงข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การจัดสรรตำแหน่งแอดเดรสบนเครื่อง PC

อินพุทเอาท์พุทแอดเดรส	หน้าที่การทำงาน
000H – 00FH	ส่วนควบคุม DMA
020H – 021H	ส่วนควบคุมการอินเตอร์รัพพ์
040H – 043H	ไทเมอร์/เคาน์เตอร์
060H – 063H	รีจิสเตอร์ระบบ
080H – 083H	รีจิสเตอร์ของ DMA
0A0H – 0BFH	NMI รีจิสเตอร์อินเตอร์รัพพ์
0C0H – 0FFH	สงวนไว้
100H – 1FFH	ส่วนควบคุมฟรอนท์พาเนล
200H – 20FH	สำหรับเกมคอมพิวเตอร์ (พอร์ตเกม)
210H – 217H	ส่วนขยายเพิ่มเติม
220H – 24FH	สงวนไว้
278H – 27FH	เครื่องพิมพ์ที่ 2
2F8H – 2FFH	ซีเรียลอินเตอร์เฟสที่ 2
300H – 31FH	สำหรับพอร์ททดลองเพิ่มเติม
320H – 32FH	ส่วนควบคุมฮาร์ดดิสค์
378H – 37FH	พรินเตอร์อินเตอร์เฟส (พอร์ตขนาน)
380H – 38FH	SDLC อินเตอร์เฟส
3A0H – 3AFH	สงวนไว้
3B0H – 3BFH	สำหรับโมโนโครมอะแดปเตอร์ และเครื่องพิมพ์
3C0H – 3CFH	สงวนไว้
3D0H – 3DFH	สำหรับการ์ดโคเลอร์กราฟฟิก
3E0H – 3E7H	สงวนไว้
3F0H – 3F7H	ส่วนควบคุมฟลอปปี้ดิสค์
3F8H – 3FFH	ซีเรียลอินเตอร์เฟส

ตารางที่ 5 แสดงการจัดสรรตำแหน่งแอดเดรสบนเครื่อง PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไมโครคอนโทรลเลอร์ MCS-51

2.2.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51

คุณสมบัติทั่วไปที่สำคัญของไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
- มีขาสัญญาณอินพุตเอาต์พุตจำนวน 32 บิต
- สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (extemal data memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำโปรแกรมภายในตัว (on-chip program memory) ขนาด 4 K โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับเบอร์ 8031 และ 8032 จะไม่มีหน่วยความจำในส่วนนี้
- มีหน่วยความจำข้อมูลภายในตัว (on-chip data memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์ 8032 และ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น
- มีไทเมอร์/เคาน์เตอร์ (timer/counter) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8032 หรือ 8052 จะมีไทเมอร์/เคาน์เตอร์จำนวน 3 ตัว
- การอินเตอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ 8052 จะทำการอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัปต์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์ (full duplex)
- มีคำสั่งในการคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์
- คำสั่งโดยส่วนใหญ่ใช้เวลาการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
- ต้องการแหล่งจ่ายไป 5 โวลต์ เพียงชุดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเบอร์	หน่วยความจำภายใน		จำนวนไทเมอร์/เคาน์เตอร์	จำนวนอินเตอร์รัปต์
	เก็บโปรแกรม	เก็บข้อมูล		
8052AH	8K × 8ROM	256 × 8 RAM	3 × 16-Bit	6
8051AH	4K × 8ROM	128 × 8 RAM	2 × 16-Bit	5
8051	4K × 8ROM	128 × 8 RAM	2 × 16-Bit	5
8032AH	ไม่มี	256 × 8 RAM	3 × 16-Bit	6
8031AH	ไม่มี	128 × 8 RAM	2 × 16-Bit	5
8031	ไม่มี	128 × 8 RAM	2 × 16-Bit	5
8751H	4K × 8ROM	128 × 8 RAM	2 × 16-Bit	5
8751H-12	4K × 8ROM	128 × 8 RAM	2 × 16-Bit	5

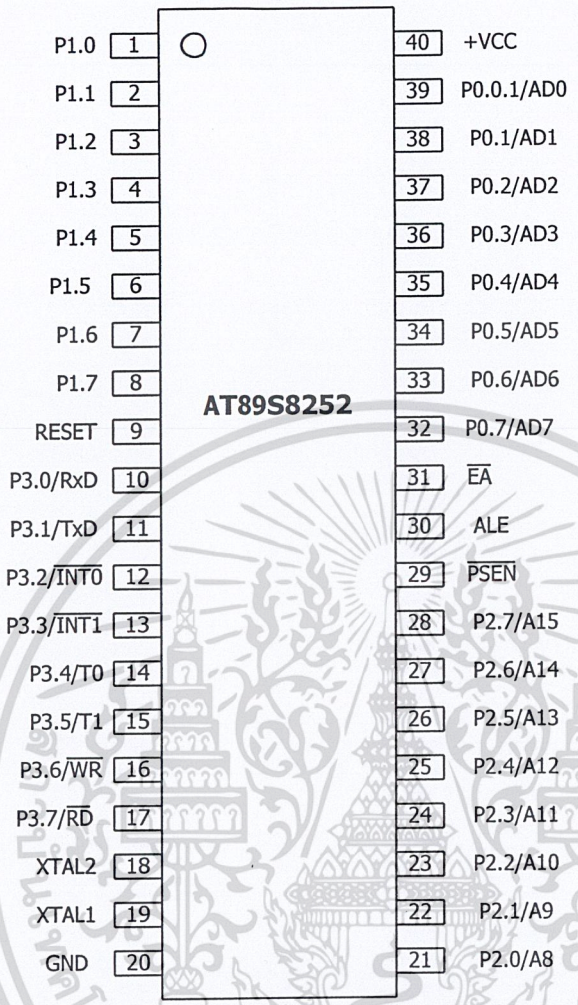
ตารางที่ 6 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูล MCS-51

โครงสร้างภายนอกของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังแสดงในรูปของการจัดตำแหน่งขาต่าง ๆ ของ ไม่โครคอนโทรลเลอร์ตระกูล MCS-51 สำหรับหน้าที่การใช้งานของแต่ละขา มีดังนี้

- ขา V_{cc} เป็นขาป้อนแรงดันไฟเลี้ยง + 5 โวลต์
- ขา V_{ss} เป็นขากราวด์
- ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ขา $P_{0.0}$-$P_{0.7}$ เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำมาใช้เป็นพอร์ตอินพุตอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จำใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์ต่ำ (A_0-A_7) ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการับส่งข้อมูลขนาด 8 บิต (D_0-D_7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 แสดงขาที่ต่อใช้งานของ MCS-51

- ขาพอร์ต 1 (Port 1) มี 8 ขา ได้แก่ขา $P_{1.0}$-$P_{1.7}$ เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับการใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากนี้สำหรับเบอร์ 8032 และ 8052 ขาพอร์ต $P_{1.0}$ และ $P_{1.1}$ จะถูกนำมาใช้งานเป็นขา T2 และ T2EX ตามลำดับด้วย
- ขาพอร์ต 2 (Port 2) มี 8 ขาได้แก่ขา $P_{2.0}$-$P_{2.7}$ เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์สูง (A_8 –A 15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 3 (Port 3) มี 8 ขาได้แก่ขา $P_{3.0} - P_{3.7}$ เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในหน้าที่พิเศษต่าง ๆ ดังแสดงในรูปตารางที่ 2
- ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตต้อง

คงสถานะเป็น 1 อย่างน้อยนาน 2 แมชซีนไซเคิล ในขณะที่ออสซิเลเตอร์ยังทำงานอยู่

- ขา ALE/PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตซ์ (latch) ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่เป็นอินพุตรับพัลส์ในการโปรแกรม (program pulse input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM

ขาพอร์ต	หน้าที่พิเศษ
$P_{3.0}$	RXC (serial input port)
$P_{3.1}$	TXD (serial output port)
$P_{3.2}$	$\overline{\text{INT0}}$ (external interrupt 0)
$P_{3.3}$	$\overline{\text{INT1}}$ (external interrupt 1)
$P_{3.4}$	T0 (Timer 0 external input)
$P_{3.5}$	T1 (Timer 1 external input)
$P_{3.6}$	$\overline{\text{WR}}$ (external data memory write strobe)
$P_{3.7}$	$\overline{\text{RD}}$ (external data memory read stobe)

ตารางที่ 7 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต P_3

- ขา $\overline{\text{PSEN}}$ (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตรบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบจำนวน 2 ครั้ง ในแต่ละแมชซีนไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอกจะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด
- ขา $\overline{\text{EA}}$/VPP (External Access enable/VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจากภายในหรือจากภายนอก โดยถ้ามีสถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส O-OFFFH (0-1FFFH ถ้าเป็นเบอร์ 8052) อย่างไรก็ตามถ้าบิตป้องกัน (security bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไปสำหรับการโปรแกรม (V_{pp}) ขนาด 21 โวลต์ เพื่อใช้ในระหว่างการโปรแกรม EPROM

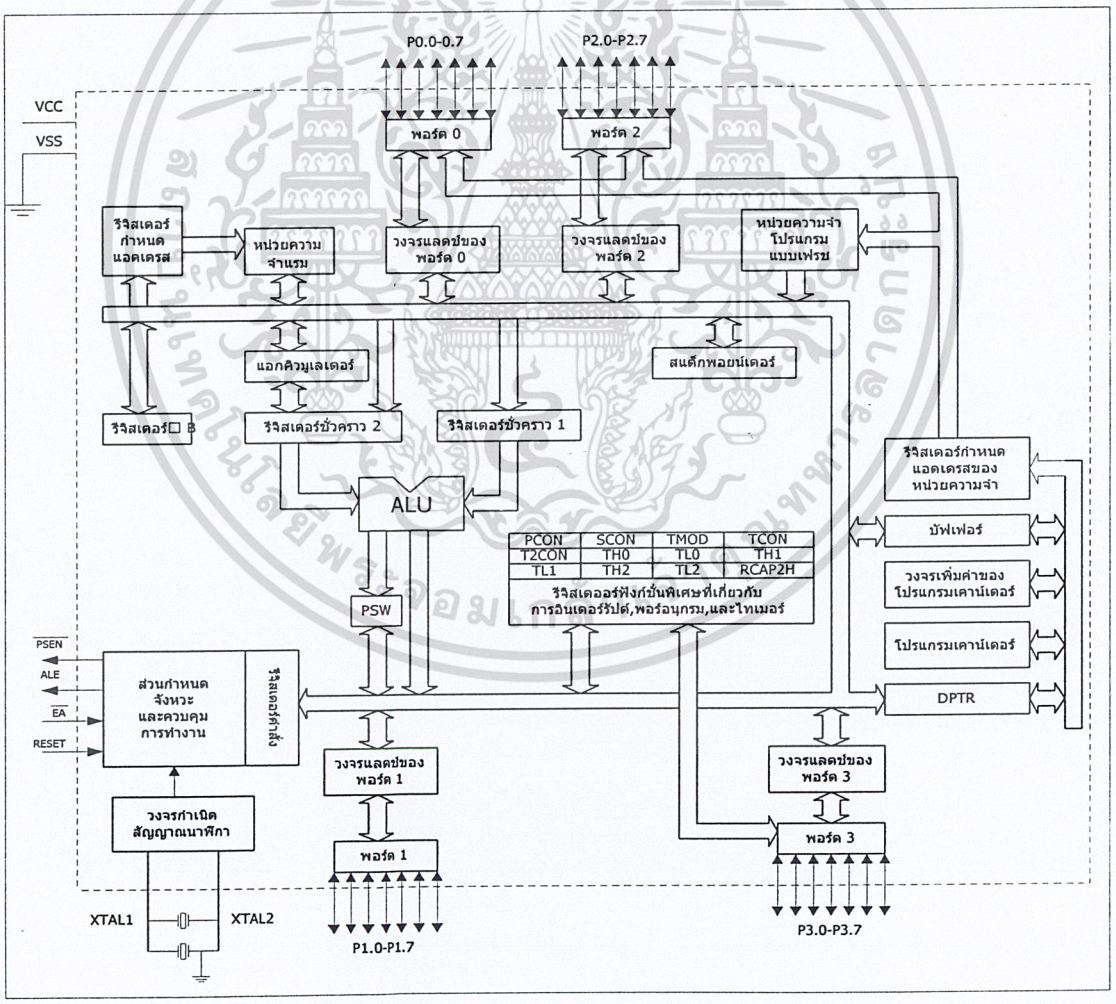
- ขา $XTAL_1$ และขา $XTAL_2$ เป็นขาอินพุตและเอาต์พุตของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (inverting oscillator amplifier) สำหรับใช้ต่อร่วมกับคริสตอลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MSC-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 IC เบอร์ 8051 แสดงในรูปประกอบด้วยอุปกรณ์ต่าง ๆ ดังนี้

1. หน่วยความจำภายในสำหรับเก็บข้อมูลขนาด 128 ไบต์ (Internal data memory 128 byte)
2. หน่วยความจำภายในสำหรับโปรแกรมขนาด 4 กิโลไบต์ (Internal program memory 4Kb)
3. อุปกรณ์ควบคุมการอินเตอร์รัปต์ (Interrupt Control Unit)
4. ตัวตั้งเวลาและตัวนับขนาด 16 บิต 2 ชุด (Timer/Counter and Timer/Counter1)
5. พอร์ตควบคุมการสื่อสารอนุกรมแบบ Full Duplex ซึ่งสามารถรับและส่งข้อมูลพร้อมกัน
6. พอร์ตขนานสำหรับติดต่ออุปกรณ์ภายนอกจำพวก 4 พอร์ต ๆ ละ 8 บิต
7. วงจรผลิตสัญญาณนาฬิกาภายใน

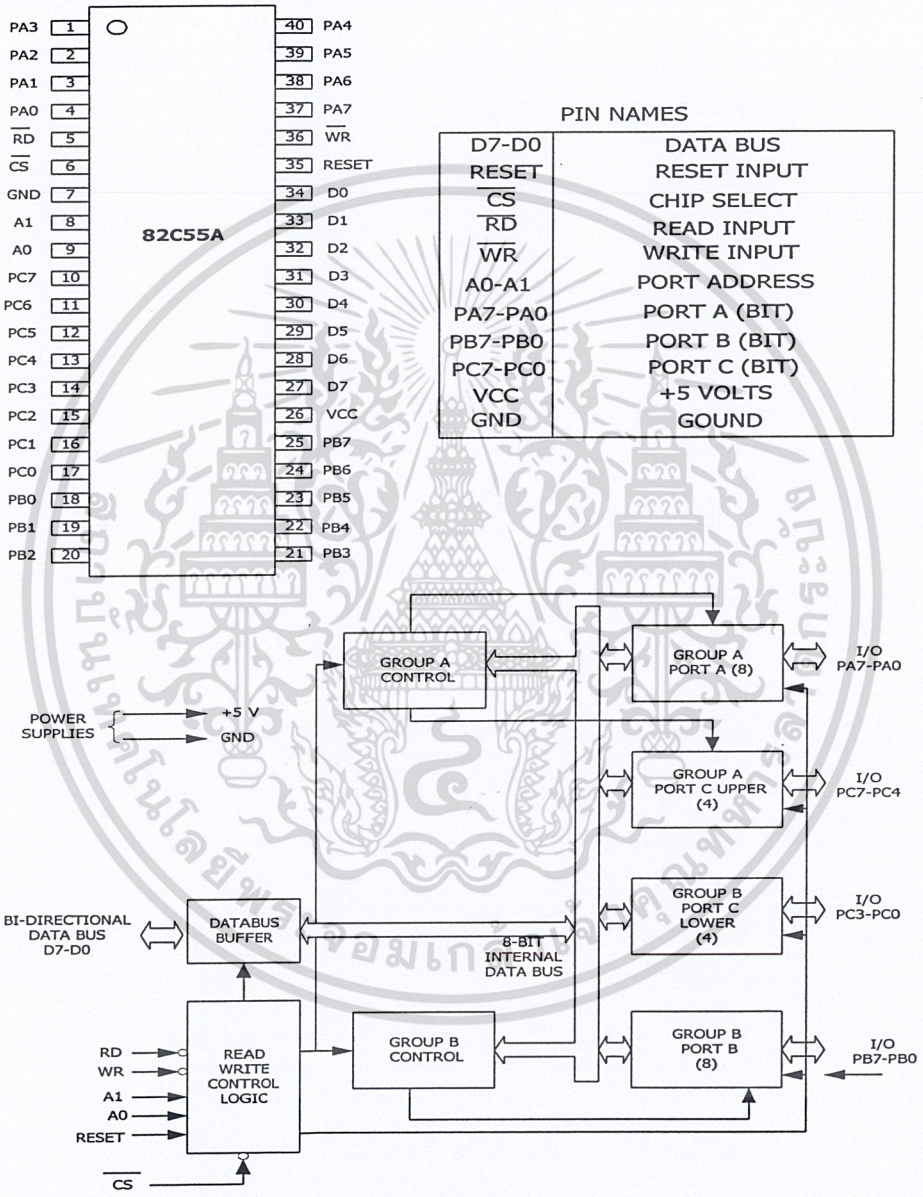


รูปที่ 8 แสดงโครงสร้างภายในของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การเชื่อมโยง 8255 กับ MCS-51

IC8255A เป็นชิพขนาด 40 ขา มีอยู่ 3 พอร์ต คือ A,B,C เป็นพอร์ต 8 บิท ที่สามารถโปรแกรมให้เป็นอินพุทหรือเอาท์พุทก็ได้ โดยที่พอร์ต C ยังแบ่งเป็น 4 บิท ล่างและ 4 บิท บน โดยมีโครงสร้างตามรูป ที่ 3



รูปที่ 9 โครงสร้างภายในและการจัดวางขาของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงาน

การทำงานมีอยู่ด้วยกัน 3 โหมด ดังตารางที่ 8

โหมด 0 มีการทำงานแบบ BASIC I/O ไม่มี handshake

โหมด 1 โหมดนี้ใช้พอร์ต A,B ในการรับหรือส่งข้อมูล และใช้พอร์ต C ในการตรวจสอบสัญญาณ (handshake)

โหมด 2 โหมดนี้ใช้พอร์ต A ในการรับส่งข้อมูล 2 ทิศทางและพอร์ต B ในการรับหรือส่งข้อมูลและใช้พอร์ต C บิต 0,1,2 ในการรับส่งข้อมูลบิตและบิต 3,4,5,6,7 เป็นสัญญาณ handshake

ตารางที่ 8 การทำงานของโหมดต่างๆ

	MODE 0		MODE 1		MODE 2
	IN	OUT	IN	OUT	GROUP A ONLY
PA_0	IN	OUT	IN	OUT	◄——►
PA_1	IN	OUT	IN	OUT	◄——►
PA_2	IN	OUT	IN	OUT	◄——►
PA_3	IN	OUT	IN	OUT	◄——►
PA_4	IN	OUT	IN	OUT	◄——►
PA_5	IN	OUT	IN	OUT	◄——►
PA_6	IN	OUT	IN	OUT	◄——►
PA_7	IN	OUT	IN	OUT	◄——►
PB_0	IN	OUT	IN	OUT	——
PB_1	IN	OUT	IN	OUT	——
PB_2	IN	OUT	IN	OUT	——
PB_3	IN	OUT	IN	OUT	——
PB_4	IN	OUT	IN	OUT	——
PB_5	IN	OUT	IN	OUT	——
PB_6	IN	OUT	IN	OUT	——
PB_7	IN	OUT	IN	OUT	——

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O
PC_1	IN	OUT	IBF_B	$\overline{OBF_B}$	I/O
PC_2	IN	OUT	$\overline{STB_B}$	$\overline{ACK_B}$	I/O
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$
PC_4	IN	OUT	$\overline{STB_A}$	I/O	$\overline{IST_B}$
PC_5	IN	OUT	IBF_A	I/O	IBF_A
PC_6	IN	OUT	I/O	$\overline{ACK_A}$	$\overline{ACK_A}$
PC_7	IN	OUT	I/O	$\overline{OBF_A}$	$\overline{OBF_A}$

ตารางที่ 9 สรุปโหมดต่าง ๆ ของ 8225

ขาสัญญาณต่าง ๆ ของ 8255

D_7-D_0 บัสข้อมูลเชื่อมโยงกับ CPU

A_1-A_0 ใช้เลือกพอร์ต A,B,C และพอร์ตควบคุม

RESET เมื่อขานี้ได้รับสัญญาณกระตุ้นลอจิก 1 จะทำให้ 8255 ถูกรีเซ็ทมีผลทำให้ทุกพอร์ตเป็นอินพุททันที

PA_7-PA_0 เป็นพอร์ตขนาน 8 บิท

PB_7-PB_0 เป็นพอร์ตขนาน 8 บิท

PC_7-PC_0 เป็นพอร์ตขนาน 8 บิท

$\overline{RD}$ ในการอ่านข้อมูลที่พอร์ตของ 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{WR}$ ในการเขียนข้อมูลหรือโปรแกรมลงบน 8255 ต้องทำให้ขานี้ เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{CS}$ เป็นขาเลือกชิพ 8255 ได้ขานี้โดยต่อกับ I/O DECODER

เมื่อขา $\overline{WR}$,$\overline{RD}$,A0,A1,$\overline{CS}$ ทำงานทั้ง 5 ขาจะมีฟังก์ชันการทำงานดัง ตารางที่ 9

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
					Input operation (READ)
0	0	0	1	0	Port A data bus
0	1	0	1	0	Port B data bus
1	0	0	1	0	Port C data bus
					Output operation (WRITE)
0	0	1	0	0	Data bus Port A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0	1	1	0	0	Data bus	Port B
1	0	1	0	0	Data bus	Port C
1	1	1	0	0	Data bus	CONTROL
					Disable function	
x	x	x	x	1	Data bus	3-STATE
1	1	0	1	0	Illegal condition	
x	x	1	1	0	Data bus	3-STATE

Source: Courtesy of Intel Corporation

ตารางที่ 10 ตาราง ความจริงของ 8255

การถอดรหัสตำแหน่งพอร์ตของ 8255 ดังแสดงในรูปที่ 9 ได้เบอร์พอร์ตดังนี้

(8000H-8FFFH),(9000H-9FFFH),(A000H-AFFFH),(B000H-BFFFH),
(C000H-CFFFH),(D000H-DFFFH),(E000H-EFFFH),(F000H-FFFFH)

และใช้เอาท์พุตที่ตำแหน่ง (F000h-FFFFH) มาถอดรหัสร่วมกับ A8,A9,A10 โดยใช้ 74LS 138 ได้พอร์ต

(F800H-F8FFH),(F900H-F9FFH),(FA00H-FAFFH),(FB00H-FBFFH),
(FC00H-FCFFH),(FD00H-FDFFH),(FE00H-FEFFH),(FF00H-FFFFH)

สัญญาณเลือกชิพ 8255 ใช้เบอร์พอร์ต (FC00H-FCFFH) และ A_0,A_1 ต่อเข้า A_0,A_1 ของ 8255 จะได้เบอร์พอร์ตของ 8255 ดังตารางที่11

I/O (ADDRESS)	8255(PORT)
FC00H	A
FC01H	B
FC02H	C
FC03H	CONTROL

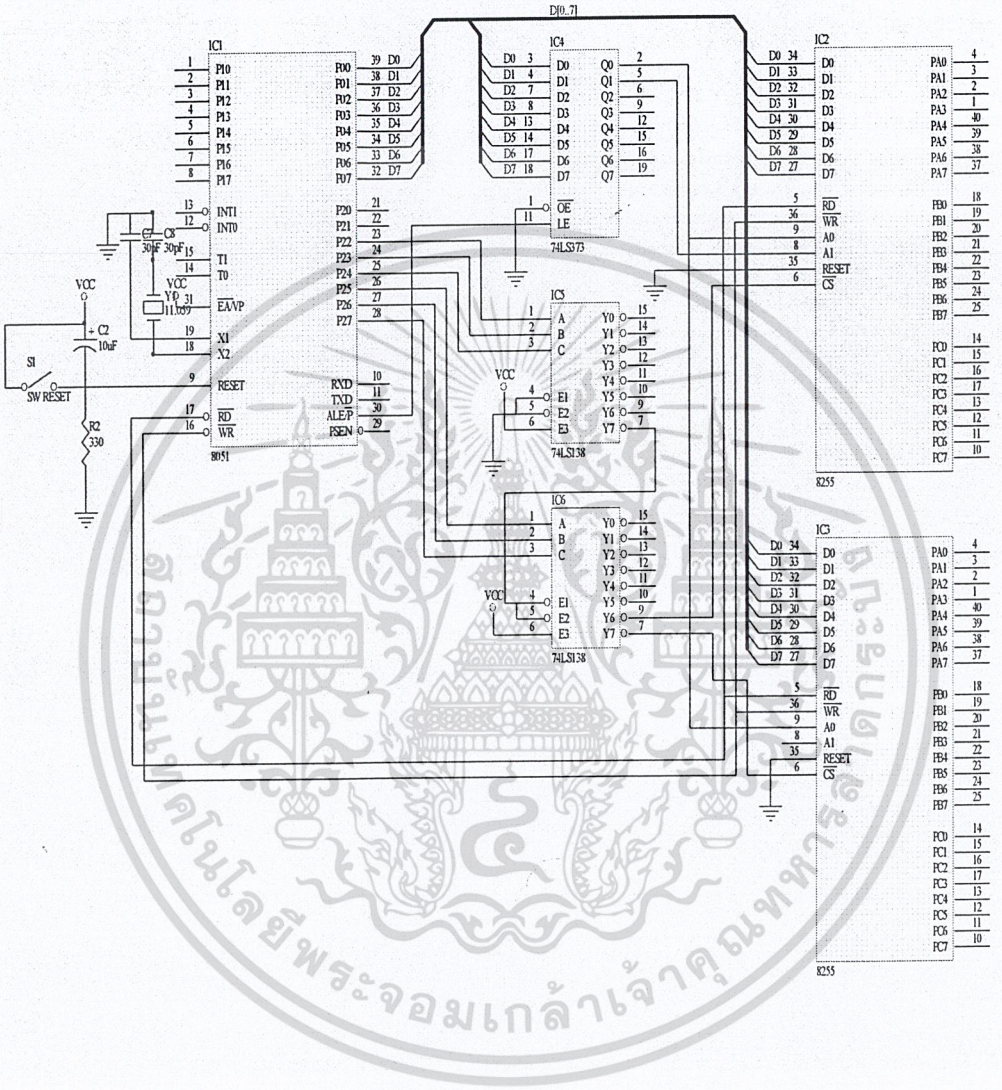
ตารางที่ 11 I/O ADDRESS ของ 8255

ส่วนขา $\overline{RD}$,$\overline{WR}$ ของ 8255 ต่อมาจาก $\overline{RD}$,$\overline{WR}$ ของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การเชื่อมโยง 8255 เข้ากับ MCS-51

การต่อ8255 เข้ากับ MCS-51 แสดงดังรูปที่ 10



รูปที่ 10 การเชื่อมโยง 8255 เข้ากับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Sensor

ในวงจร Sensor นี้จะใช้ อินฟราเรดเป็นตัวรับส่ง ซึ่งตัวรับนั้นเมื่อมีสัญญาณอินฟราเรดมากระทบตัวมัน ตัวรับนี้จะยอมให้กระแสไหลผ่านตัวมันได้ และจะมีกระแสไหลมากขึ้นถ้าสัญญาณที่มากระทบตัวมันมีความแรงขึ้น และในขณะที่ไม่มีแสงอินฟราเรดมากระทบตัวมัน ทำให้ตัวรับอินฟราเรดมีกระแสหรือยอกให้กระแสไหลผ่านได้น้อยมากซึ่งการต่ออินฟราเรดมาใช้งาน ได้แสดงดังรูปที่ 11

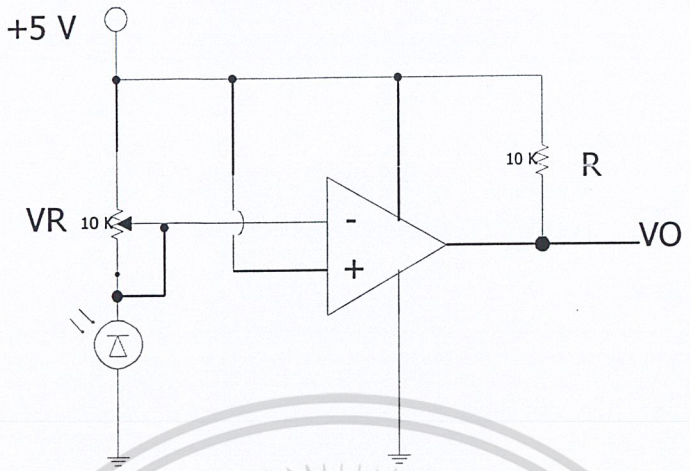


รูปที่ 11 วงจรภาครับ-ส่ง SENSOR

จากรูปที่ 11 เมื่อมีสัญญาณอินฟราเรดจากตัวส่งส่งออกมา กระทบตัวรับ ตัวรับก็จะทำงานทำให้ LED ติดสว่าง

แต่ในทางปฏิบัติเมื่อตัวส่งอินฟราเรดส่งสัญญาณอินฟราเรดมากระทบที่ตัวรับซึ่งน้อยมาก ทำให้กระแสไหลผ่านตัวมันน้อย ทำให้มีแรงดันตกคร่อมน้อยด้วย ซึ่งเมื่อมีแรงดันน้อยทำให้ Output ที่ต้องการที่เป็น Logic (H,L) ไม่ถูกต้อง คือ เมื่อแรงดันน้อย ซึ่งตัวรับก็ทำงานอยู่แต่ Logic ที่ได้ เป็น Logic L ซึ่งจริง ๆ แล้ว เราต้องการ Logic H เมื่อตัวรับทำงาน ดังนั้น จึงใช้ Op Amp เข้ามาช่วย เพื่อแก้ปัญหาดังกล่าว แสดงดังรูปที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 วงจรภาครับของ Sensor

จากรูปที่ 12 ต้องใช้ Op-Amp ที่ต้องการไฟเลี้ยงเป็น + และ GND เพราะว่าหลักการของ Op-Amp เมื่อต่อเป็นลักษณะ Comparator แล้วจะได้ Output คือ

$V_1 > V_2 = V_E$

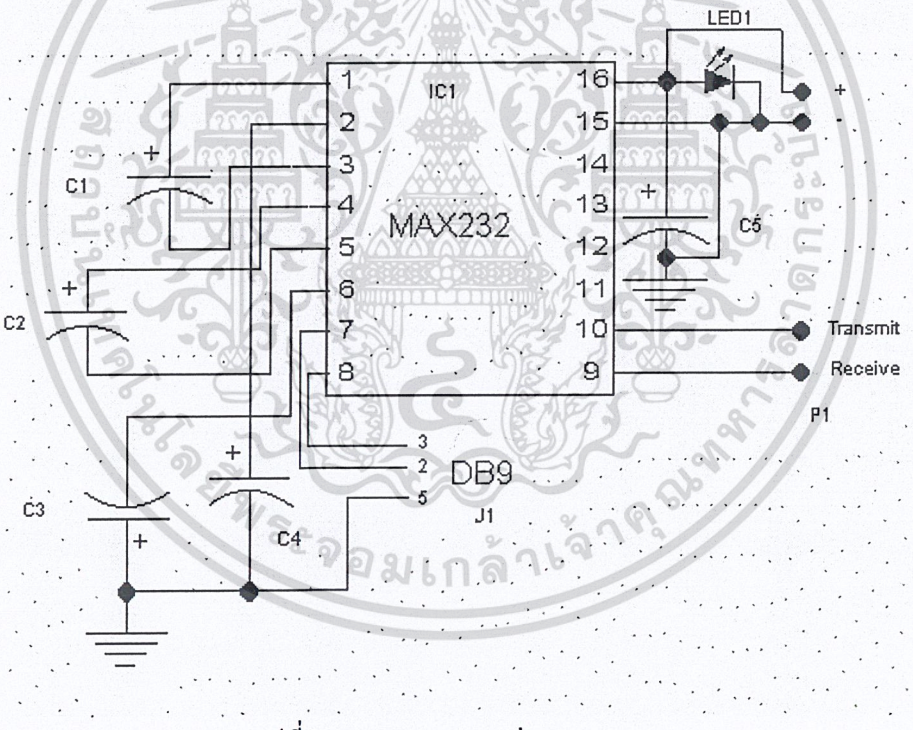
$V_2 > V_1 = G$

จากวงจร V_R มีไว้เพื่อปรับระดับความแรงในการรับอินฟราเรด ส่วน R ที่ต่อกับ V_0 เป็น R pull up การทำงาน ถ้า ตัวรับยังไม่ทำงาน V_1 มีแรงไฟประมาณ V_{cc} และ V_2 ก็ประมาณ V_{cc} ซึ่ง output จะไม่มีการเปรียบเทียบ ถ้าตัวรับทำงานก็เปรียบเสมือน V_R ต่อไฟ + ลงกราวด์ ทำให้ V_1 เปรียบเสมือนลงกราวด์ด้วย ทำให้ V_2 มากกว่า V_1 ได้ Output + V_{cc} ออก Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การต่อใช้งาน MAX232

การต่อใช้งาน RS-232 สำหรับการสื่อสารกับเครื่องคอมพิวเตอร์อย่างเช่นการส่งผ่านข้อมูลกันระหว่างไมโครคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์ การดาวโหลดข้อมูลหรือโปรแกรมไปเก็บไว้ยังตัวซีพียูหรืออีอีพรอม การสื่อสารโดยตรงเช่นการส่งข้อมูลไปแสดงผลยังจอภาพ LCD ผ่านทาง Port RS-232 ซึ่งเป็นที่นิยมใช้กัน จริงๆแล้วการต่อ RS-232 เราสามารถทำได้หลายวิธีอย่างเช่นการใช้ Resistor สองตัวต่อร่วมกันเป็นวงจรง่ายๆ แต่ใช้ได้กับบางลักษณะงานเท่านั้น แต่ในที่นี้ใช้ MAX232 ซึ่งเป็น IC ที่ทำหน้าที่เปลี่ยนจาก TEXT เป็น DIGITAL หรือเปลี่ยนระดับสัญญาณจาก RS-232 เป็น TTL และเปลี่ยนจาก TTL เป็น RS-232 และหาซื้อได้ง่าย สำหรับการต่อก็ไม่ยุ่งยากซับซ้อนมากนัก อุปกรณ์ที่ใช้ก็มีจำนวนน้อย ซึ่งแสดงการต่อวงจรดังรูปที่ 13



รูปที่ 13 แสดงวงจรการต่อ MAX232

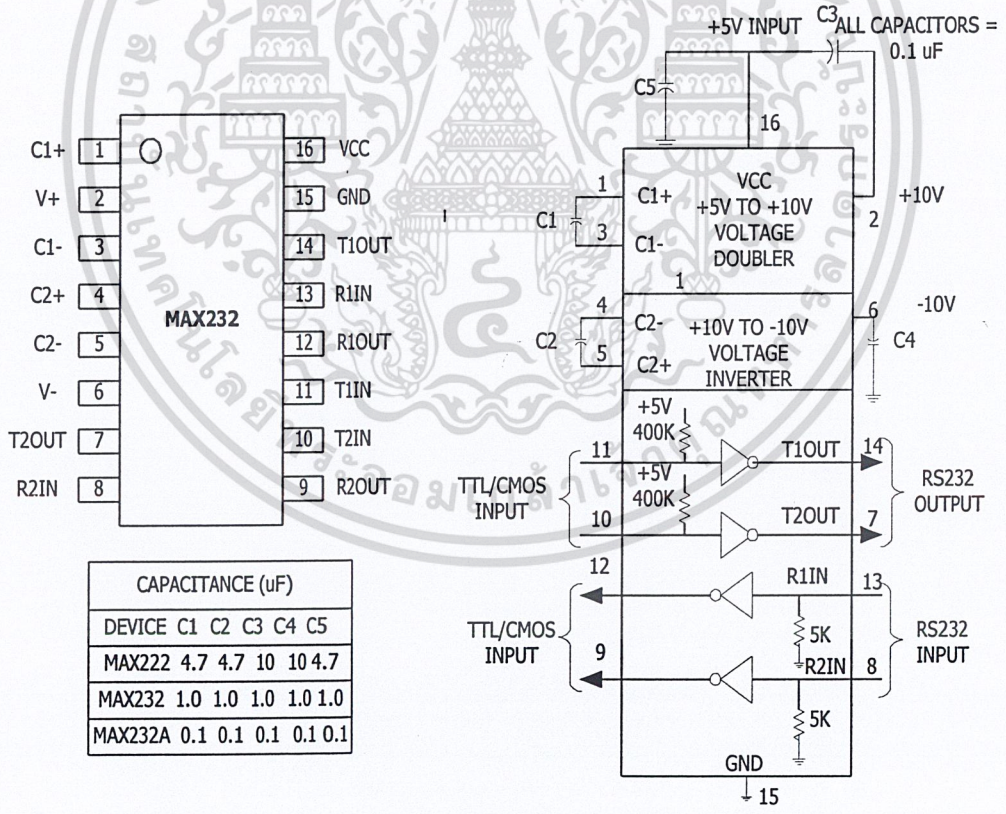
จากรูปที่ 13 เป็นวงจรการต่อMAX232 ออกไปใช้งาน โดยที่มีขา 2,3 ต่อกับคอนเน็กเตอร์ DB9 เพื่อติดต่อกับ COMPUTER และขา 9,10 ต่อไปยัง MCS-51 สำหรับไฟเลี้ยงวงจรใช้ไฟ 5V

สำหรับอุปกรต่างๆ แสดงในตารางที่ 12 ซึ่งเปรียบเทียบระหว่าง MAX232 กับ MAX232A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Part	Description
IC1	MAX232 or MAX232A
C1-C4	1uF 25V electrolytic capacitor (MAX232) or .1uF 25V capacitor (MAX232A)
C5	10uF 25V electrolytic capacitor
J1	DB9 Female connector with straight solder cups
P1	8 pin header on .1" centers
LED1	5V LED **with integral dropping resistor** (Optional)
PC Board	1/2 Radio Shack DIP-1 PC Board

ตารางที่ 12 อุปกรณ์ที่ต้องใช้



รูปที่ 14 แสดงโครงสร้างภายใน และ ขาต่างๆของ MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

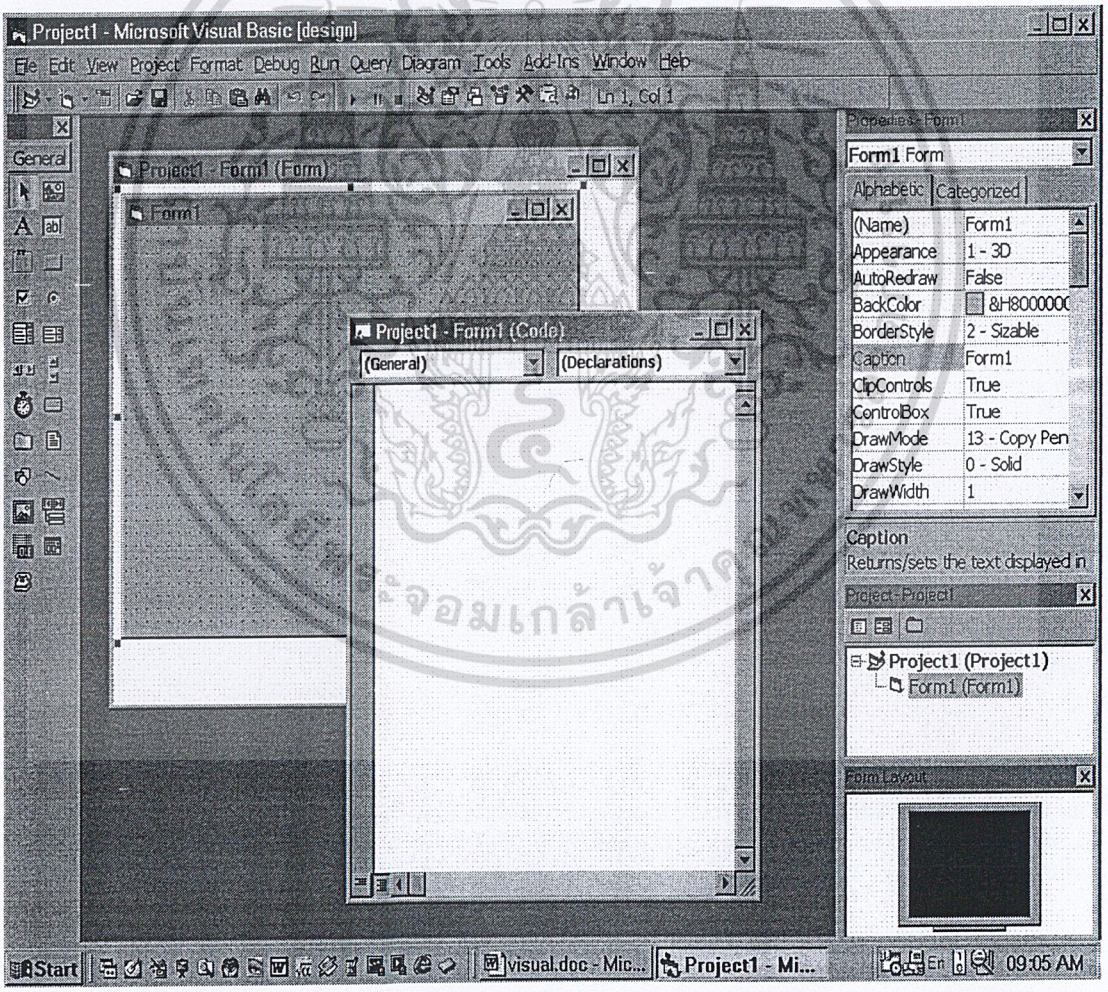
2.5 Visual Basic

2.5.1 ทฤษฎีการทำงาน Visual Basic

การพัฒนาแอพพลิเคชั่นใน VB6 จะไม่เหมือนกับการเขียนโปรแกรมที่เราเคยเขียนมาก่อนในภาษาโปรแกรมแบบอื่น ๆ บน DOS ที่โปรแกรมจะทำงานตั้งแต่คำสั่งแรกจนถึงคำสั่งสุดท้าย ซึ่งอาจจะมีทางเลือกการทำซ้ำ หรือการเรียกโปรแกรมย่อยมาทำงาน

ใน VB6 จะเป็นการเขียนโปรแกรมแบบ Event-driven การเขียนโปรแกรมแบบนี้เป็นการเขียนโปรแกรมที่ต้องตอบสนองต่อเหตุการณ์ (Event) เป็นเหตุการณ์ ๆ ไป

รายละเอียดของส่วนประกอบต่าง ๆของหน้าจอของ VB



รูปที่ 15 หน้าจอของ Visual Basic

เมนูบาร์ (Menu Bar)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งที่เราสามารถใช้งานได้ทั้งหมดใน VB

ทูลบาร์ (Toolbar)

เป็นส่วนที่รวบรวมเอาไอคอนเล็ก ๆ มากมายเอาไว้ ซึ่งแต่ละไอคอน (Icon) เหล่านี้ก็เปรียบเสมือนคำสั่งหนึ่ง ๆ ของเมนู นั่นคือไอคอนเหล่านี้ได้ถูกออกแบบมาเพื่อให้การใช้งานคำสั่งต่าง ๆ ทำได้รวดเร็วและมีลักษณะที่สื่อความหมายกับผู้ใช้มากขึ้น

ทูลบ็อกซ์ (Toolbox)

เป็นแถบเครื่องมือที่ประกอบไปด้วยไอคอนต่าง ๆ ซึ่งเรียกว่า "คอนโทรล" (Control) โดยจะนำมาใช้ร่วมกับฟอร์มเพื่อสร้างจอภาพของโปรเจ็กต์ แต่ละคอนโทรลจะใช้เป็นเครื่องมือในการสร้างส่วนติดต่อกับผู้ใช้ (User Interface) โดยการคลิกเลือกที่คอนโทรลนั้นแล้วนำมาวางลงบนฟอร์มด้วยการลากแล้ววาง (Drag and Drop) หรือโดยการดับเบิ้ลคลิกที่ไอคอน ในทูลบ็อกซ์

หน้าต่างโปรเจ็กต์ (Project Explorer Window)

เป็นหน้าต่างที่รวบรวมรายชื่อของฟอร์มโมดูลไฟล์(Custom Control) โมดูลคลาส(Class Module) หรือ ไฟล์ทรัพยากร (Resource File) สำหรับการสร้างแอพพลิเคชั่นหนึ่ง ๆ ซึ่งการรวมเอาไฟล์เหล่านี้เข้าด้วยกันเพื่อสร้าง แอพพลิเคชั่นภายใต้ VB เรียกว่า โปรเจ็กต์ (Project)

หน้าต่างคุณสมบัติ (Properties Window)

เป็นหน้าต่างที่รวบรวมคุณสมบัติทั้งหมดของฟอร์มหรือคอนโทรลเอาไว้ ซึ่งคุณสมบัติทั้งหมดที่ปรากฏในหน้าต่างนี้จะเป็นคุณสมบัติที่ผู้อ่านสามารถกำหนดค่าได้ในขณะกำลังออกแบบ เมื่อผู้อ่านทำการแก้ไขค่าคุณสมบัติต่าง ๆ ในหน้าต่างคุณสมบัตินี้ก็จะส่งผลต่อคอนโทรลตัวนั้นทันที ซึ่งบางคุณสมบัติสามารถแสดงให้เห็นการเปลี่ยนแปลงคุณลักษณะได้ทันที ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรันแอพพลิเคชั่นเท่านั้น

ฟอร์ม (Form)

เป็นหน้าต่างที่ใช้ในการออกแบบส่วนติดต่กับผู้ใช้ของแอพพลิเคชั่นของเรา เนื่องจาก VB ได้รับการออกแบบให้ผู้ใช้นำมาออกแบบแอพพลิเคชั่นในลักษณะของการสื่อสารด้วยรูป (Visual) ดังนั้นการสื่อการทำงานต่าง ๆ ระหว่างแอพพลิเคชั่นกับผู้ใช้จะต้องกระทำผ่านฟอร์ม โดยในแอพพลิเคชั่นหนึ่งสามารถมีได้หลายฟอร์ม และภายในฟอร์มก็จะถูกใช้ในการบรรจุคอนโทรลต่าง ๆ ฟอร์มจึงทำหน้าที่เป็นตัวบรรจุ (Container) และฟอร์มยังเป็นออบเจ็กต์ (Object) ตัวหนึ่งของ VB ที่อนุญาตให้ ผู้ใช้แก้ไขคุณสมบัติได้ในขณะออกแบบจากหน้าต่างคุณสมบัติและสามารถควบคุมพฤติกรรมต่าง ๆ ของฟอร์มด้วยวิธีเดียวกับออบเจ็กต์อื่น ๆ

หน้าต่างโค้ดอิดิเตอร์ (Code Editor Window)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นหน้าต่างที่ใช้ในการพิมพ์คำสั่งโปรแกรมเข้าไปในแอพลิเคชั่นของเรา

หน้าต่างฟอร์มเลย์เอาท์ (Form Layout Window)

เป็นหน้าต่างแสดงตำแหน่งฟอร์มบนจอ ช่วยให้เรากำหนดตำแหน่งของฟอร์มบนหน้าจอได้สะดวกขึ้น โดยการคลิกเมาส์ ลากฟอร์มที่แสดงบนหน้าต่างให้เคลื่อนไปยังตำแหน่งที่ต้องการ

คุณสมบัติ (Property)

เป็นสิ่งที่ใช้บรรยายลักษณะต่าง ๆ ของคอนโทรล ทำให้คอนโทรลสามารถใช้งานที่หลายรูปแบบมากขึ้น และมีลักษณะแปลกตาไม่ซ้ำกัน การกำหนดค่าคุณสมบัติอาจทำในขณะออกแบบผ่านทางหน้าต่างคุณสมบัติ หรือกำหนดขณะรันแอพพลิเคชั่นโดยเขียนคำสั่งโปรแกรมให้ตั้งคุณสมบัติขณะโปรแกรมทำงาน

เมดถอด (Method)

เป็นโปรแกรมย่อยประเภทหนึ่ง ซึ่งเป็นสมาชิกของคอนโทรลนั้น ๆ เหมือนกับคุณสมบัติของคอนโทรลเมื่อเรียกใช้เมดถอดจะเป็นการสั่งให้คอนโทรลทำงานได้ ซึ่งอาจจะมีการเปลี่ยนแปลงค่าคุณสมบัติต่าง ๆ ของคอนโทรลนั้น

อีเว็นต์ (Event)

เป็นการตอบสนองต่อเหตุการณ์ภายนอกของคอนโทรล และเป็นโปรแกรมย่อยที่ทำงานทันทีที่เกิดเหตุการณ์นั้นขึ้นมา

คอนโทรลของ VB

คอนโทรลเป็นเครื่องมืออย่างหนึ่งที่ช่วยให้การสร้างแอพพลิเคชั่นด้วย VB ทำได้ง่ายและรวดเร็วยิ่งขึ้น แต่เมื่อได้นำไปวางบนฟอร์มแล้ว จะเรียกว่าออบเจ็กต์ (Object) ซึ่งจริง ๆ แล้วคอนโทรลก็คือไฟล์ไดนามิกลิงค์ไลบรารีทั่ว ๆ ไปของวินโดว์ ไฟล์เหล่านี้จะมีนามสกุล.VBX หรือ .OCX ซึ่งเราเรียกว่า Custom Control ซึ่งไฟล์เหล่านี้จะถูกจัดเก็บในไดเร็คทอรี่ย่อย System ของ Window

คอนโทรลพื้นฐานใน VB จะเป็นคอนโทรลติดต่อกับผู้ใช้ในการทำงานพื้นฐานต่าง ๆ เช่น การแสดงข้อความและรับข้อความจากผู้ใช้ การแสดงรูปภาพ การแสดงตัวเลือก เป็นต้น ซึ่งจะกล่าวถึงเฉพาะคอนโทรลที่ได้นำมาใช้งานดังต่อไปนี้

ฟอร์ม (Form)

ฟอร์มเป็นส่วนประกอบพื้นฐานในการติดต่อกับผู้ใช้ ที่ใช้แสดงผลการทำงาน รวมทั้งรับอินพุตต่างๆจากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติที่สำคัญของฟอร์ม

AutoRedraw เป็นคุณสมบัติที่บอกว่าต้องมีการวาดสิ่งที่อยู่บนฟอร์มนั้นใหม่หรือไม่ เมื่อจำเป็นต้องมีการวาดใหม่ เช่น มีหน้าต่างอื่นมาบัง เป็นต้น ถ้าเป็น True เมื่อจำเป็นจะต้องมีการปรับปรุงหน้าจอ VB จะจัดการเรื่องนี้เองโดยอัตโนมัติ แต่ถ้าเป็น Flase เราจำเป็นต้องจัดการเอง

BackColor ใช้กำหนดสีส่วนหลัง (Background) ของฟอร์ม

Caption ใช้กำหนดข้อความที่แสดงบนไตเติ้ลบาร์ของฟอร์ม

ControlBox เป็นคุณสมบัติที่บอกให้มีเมนูที่ใช้ย่อ ขยาย หรือปิดหน้าต่างอยู่ที่มุมด้านบนด้านซ้ายขิงหน้าต่างหรือไม่ ถ้าเป็น True จะมีการแสดงปุ่มเมนู ถ้าเป็น Flase จะไม่มีการแสดงปุ่มเมนู

ForeColor เป็นคุณสมบัติที่ใช้กำหนดสีของข้อความหรือกราฟฟิกบนฟอร์ม

Height,Width เป็นคุณสมบุติที่ใช้กำหนดความสูงและความกว้างของฟอร์มซึ่งวัดอยู่ในหน่วยที่เรียกว่า ทวิพ (1 ทวิพ = 1/1440 นิ้ว)

MaxButton,MinButton เป็นคุณสมบัติมที่กำหนดว่าจะให้มีปุ่มย่อขยายฟอร์มที่มุมบนขวาหรือไม่ซึ่งจะมีผลในขณะรันเท่านั้น

Name เป็นคุณสมบัติที่กำหนดชื่อของฟอร์มในรหัสโปรแกรม ซึ่งเปลี่ยนแปลงได้เฉพาะในขณะออกแบบเท่านั้น

Picture เป็นคุณสมบัติที่ใช้แสดงภาพบนหน้าต่างเสมอ ในการเปลี่ยนแปลงคุณสมบัติในขณะรันต้องใช้ฟังก์ชัน LoadPicture

Visible เป็นคุณสมบัติที่กำหนดว่าจะให้เห็นฟอร์มหรือไม่ในขณะรัน

WindowState เป็นคุณสมบัติที่กำหนดว่าจะให้แสดงฟอร์มในขณะรันในสถานะปกติ (ค่ากำหนดเป็น 0) ในสถานะลดขนาดเป็นไอคอน (ค่ากำหนดเป็น 1) หรือในสถานะขยายขนาด (ค่ากำหนดเป็น 2)

อีเว็นต์ (Event)

Click เป็นอีเว็นต์ที่เกิดขึ้นเมื่อมีการคลิ้กเมาส์บนฟอร์ม โดยถ้าผู้ใช้คลิ้กเมาส์บนออบเจ็กต์ที่ปรากฏอยู่บนฟอร์ม อีเว็นต์ Click ก็จะไม่ถูกส่งไปให้แก่ฟอร์ม

Load,Unload อีเว็นต์ Load จะเกิดขึ้นเมื่อฟอร์มนั้นโหลดขึ้นมา โดยมากอีเว็นต์นี้จะมีคำสั่งเกี่ยวกับการเริ่มต้นต่างๆของคอนโทรล และตัวแปรมในโมดุลฟอร์มนี้ ส่วน Unload จะเป็นอีเว็นต์ที่เกิดขึ้นยกเลิกฟอร์มไปแล้ว

เมดถอด (Method)

Cls เป็นเมดถอดที่ใช้เคลียร์สิ่งที่วาดบนฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hide เป็นเมดถอดที่ใช้ซ่อนฟอร์มไว้ชั่วคราว

Show เป็นเมดถอดที่ใช้ในการแสดงฟอร์มที่ต้องการ

SetFocus เป็นเมดถอดที่ในการโฟกัสมายังฟอร์มนี้เพื่อให้ผู้ใช้สามารถทำงานด้วยได้

เท็กบ็อกซ์ (Text Box)

เป็นคอนโทรลที่ผู้ใช้สามารถเติม แก้ไขข้อความในเท็กซ์บ็อกนี้ได้โดยผ่านคีย์บอร์ด หรือแสดงข้อมูลต่างๆในโปรแกรมของเรา



รูปที่ 16 เครื่องมือสร้างTextBox

คุณสมบัติ

Alignment เป็นคุณสมบัติที่กำหนดรูปแบบของข้อความที่แสดงในคอนโทรลว่าจะให้ข้อความอยู่ชิดซ้าย (LeftJustify),ชิดขวา (rigthJustify) หรืออยู่ตรงกลาง (Center) คุณสมบัตินี้จะใช้ได้ก็ต่อเมื่อ Multiline มีค่าเป็น True

FontBold,FontItalic,FontName,FontSize,FontUnderline เป็นคุณสมบัติที่กำหนดรูปแบบของข้อความที่แสดงในคอนโทรลว่าใช้ตัวอักษรแบบใด (FontBold) ตัวเอียง (FontItalic) หรือขีดเส้นใต้(FontUnderline)

Maxlength เป็นคุณสมบัติที่กำหนดจำนวนอักขระสูงสุดของข้อความ โดยถ้ากำหนดเป็น 0 จะรับข้อความได้ไม่จำกัด

Multiline เป็นคุณสมบัติที่กำหนดให้เท็กซ์บ็อกซ์แสดงข้อความในรูปแบบหลายบรรทัดได้

ScrollBars เป็นคุณสมบัติที่กำหนดรูปแบบของข้อความที่แสดงในคอนโทรล ว่าจะให้แสดงสกรอลบาร์ด้วยหรือไม่อย่างไร เมื่อข้อความยาวเกินความกว้างหรือความยาวของคอนโทรล

SelLength เป็นคุณสมบัติที่บอกว่าความยาวของข้อความที่เลือกอยู่ในขณะนั้นมีค่าเท่าไร

SelStart เป็นคุณสมบัติที่บอกว่าตำแหน่งแรกของข้อความที่เลือกอยู่ในขณะนั้นอยู่ในตำแหน่งที่เท่าไร ถ้าไม่มีการเลือกข้อความอยู่ จะบอกตำแหน่งที่จะแทรกข้อความอยู่ในขณะนั้น

SelText เป็นคุณสมบัติที่บอกว่าข้อความส่วนที่เลือกอยู่ในขณะนั้นคืออะไร

Text เป็นคุณสมบัติที่เก็บข้อความของเท็กซ์บ็อกในปัจจุบันไว้ และผู้ใช้ยังสามารถโปรแกรมดัดแปลงแก้ไขข้อความที่แสดงออกมาได้อีเว็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Change เป็นอีเว็นต์ที่เกิดขึ้นเมื่อมีการเปลี่ยนปลงคุณสมบัติ Text ของ เท็กซ์บ็อกซ์ ไม่ว่าจะเป็นจากผู้ใช้ป้อนข้อความเข้าไปใหม่ หรือ จากการตั้งค่าคุณสมบัติ Text ใหม่ในโปรแกรม

LostFocus เป็นอีเว็นต์ที่เกิดขึ้นเมื่อผู้ใช้ตั้งระยะเคลื่อนออกไปจากเท็กซ์บ็อกซ์หรือใช้เมาส์เลือกออบเจ็กต์อื่น ๆ

เมดถอด

SelFocus เป็นวิธีการที่มีประโยชน์ที่สุดในการชี้ความสนใจไปยังเท็กซ์บ็อกซีที่สำคัญที่สุดในหลาย ๆ เท็กซ์บ็อกซ์ โดยเคอร์เซอร์จะไปอยู่ในเท็กซ์บ็อกซ์ที่เรากำหนด

เลเบล (Label)

เป็นคอนโทรลที่ใช้ในการแสดงข้อความตามที่เราต้องการ เพื่อแสดงข้อมูลบางอย่างแก่ผู้ใช้โดยผู้ใช้ไม่สามารถแก้ไขข้อความนั้นได้ในตอนรันโปรแกรม การแก้ไขข้อความในเลเบลจะต้องทำในขณะออกแบบ หรือในคำสั่งโปรแกรมตอนรันเท่านั้น



รูปที่ 17 เครื่องมือสร้างLabel

คุณสมบัติ

คุณสมบัติดังที่ได้กล่าวไปแล้วที่มีในเลเบลได้แก่ Alignment, BackColor, FontBold, FontItalic, FontName, FontSize, FontUnderline, Name, Visibel คุณสมบัติอื่น ๆ ได้แก่

Autosize เป็นคุณสมบัติที่กำหนดให้คอนโทรลปรับขนาดตามข้อความที่แสดง

Caption เป็นคุณสมบัติที่บอกถึงข้อความที่แสดงบนคอนโทรลในขณะนั้น

WordWarp เป็นคุณสมบัติที่กำหนดให้ข้อความขึ้นบรรทัดใหม่ เมื่อยาวเกินความยาวของคอนโทรล

Enabled ถ้ามีค่าเป็น True จะใช้งานตามปกติ แต่ถ้าเป็น False ข้อความจะเป็นสีเทาและอีเว็นต์ของเมาส์จะใช้งานบนคอนโทรลนี้ไม่ได้

อีเว็นต์

อีเว็นต์ของเลเบลจะเป็น Click และ Db1Click เช่นเดียวกับที่ฟอร์มทำได้

ปุ่มคำสั่ง (Command Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคอนโทรล ที่ใช้เมาส์ในการทำงานอย่างหนึ่งตามที่เราต้องการ โดยเราสามารถกำหนดสิ่งที่เราต้องการให้ทำงานในอีเว็นต์ Click



รูปที่ 18 เครื่องมือสร้างCommand Button

การกำหนดข้อความที่ให้แสดงบนคอนโทรล

ในการกำหนดข้อความที่ต้องการให้แสดงผลบนคอนโทรล ให้เรากำหนดที่คุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติ หรือในคำสั่งโปรแกรมก็ได้

Default เป็นคุณสมบัติที่บ่งบอกว่าเมื่อกดปุ่ม (Enter) เป็นการบ่งบอกว่าทำงานเหมือนกับการคลิ้กปุ่มคำสั่งนี้

Cancel จะเป็นคุณสมบัติที่บ่งบอกว่าเมื่อกดปุ่ม (ESC) จะเป็นการบอกว่าทำงานเหมือนกับการคลิ้กปุ่มคำสั่งนี้

Style จะเป็นคุณสมบัติบ่งบอกว่า ต้องการให้คอนโทรลเป็นแบบธรรมดา (Standard) หรือแบบมีรูปภาพ (Graghical) ด้วยหรือไม่

Picture คุณสมบัตินี้จะเป็นการบ่งบอกว่าให้แสดงรูปใดในคอนโทรล

DownPicture คุณสมบัตินี้จะเป็นการบอกว่า ให้แสดงรูปใดในปุ่มคำสั่งเมื่อปุ่มคำสั่งถูกกดลงไป

DisabledPicture คุณสมบัตินี้จะเป็นการบอกว่า ให้แสดงรูปใดในปุ่มคำสั่งเมื่อปุ่มคำสั่งนั้นไม่สามารถทำงานได้ (Disable อยู่)

การสร้างฮอตคีย์

กำหนดฮอตคีย์ให้กับคอนโทรลตามที่เราต้องการได้โดยการเพิ่มสัญลักษณ์ & ให้อยู่หน้าตัวอักษรในข้อความที่ปรากฏบนคอนโทรลให้เป็นฮอตคีย์ ทำให้ตัวอักษรตัวนั้นมีเส้นใต้ขีดอยู่และเมื่อเรากดปุ่ม (Alt) + ตัวอักษรที่มีเครื่องหมาย & อยู่ข้างหน้า จะทำให้สามารถโฟกัสไปยังคอนโทรลนั้นๆได้

อีเว็นต์

Click จะเป็นอีเว็นต์ที่เกิดขึ้นเมื่อเราได้คลิ้กบนปุ่มคำสั่ง (คอนโทรลอื่นก็มีอีเว็นต์นี้เช่นกัน และเกิดขึ้นเมื่อมีการคลิ้กที่คอนโทรลนั้นเกิดขึ้น

GetFocus เป็นอีเว็นต์ที่เกิดขึ้นเมื่อคอนโทรลได้รับโฟกัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LostFocus เป็นอีเว็นต์ที่เกิดขึ้นเมื่อเลื่อนโฟกัสจากคอนโทรลออกไป

ออปชั่นบัตทอล (Option Button)

เป็นคอนโทรลที่ใช้เป็นตัวเลือกให้ผู้ใช้สามารถเลือกได้ครั้งละหนึ่งตัวเลือกเท่านั้น เราสามารถใช้คอนโทรลเฟรมในการแบ่งกลุ่มคอนโทรลเช่นเดียวกับคอนโทรลเช็คบล็อกซ์



รูปที่ 19 เครื่องมือสร้างOption Button

คุณสมบัติ

Value เป็นคุณสมบัติที่บอกว่าคอนโทรลถูกเลือกในขณะนั้นหรือไม่ โดยะมีค่าเป็น True เมื่อคอนโทรลถูกเลือกและเป็น FIase เมื่อไม่ถูกเลือก

เฟรม (Frame)

เป็นคอนโทรลที่ใช้ในการจัดกลุ่มคอนโทรลที่ต้องใช้ร่วมกัน ทำงานอย่างเดียวกันไว้ด้วยกันซึ่งโดยทั่วไปคอนโทรลเฟรมเป็นคอนโทรลที่ช่วยเพิ่มความเรียบร้อยและความสวยงามของโปรแกรม



รูปที่ 20 เครื่องมือสร้างFrame

คุณสมบัติ

Caption เป็นคุณสมบัติแสดงข้อความบนคอนโทรลนอกเหนือจากนี้เรายังสามารถกำหนดคุณสมบัตินี้ให้ใช้ฮอตคีย์ได้เช่นเดียวกับคอนโทรลที่กล่าวมา โดยการใช้อักษร & อยู่หน้าตัวอักษรที่เราต้องการให้เป็นปุ่มที่ใช้ฮอตคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวจับเวลา (Timer)

เป็นคอนโทรลที่ใช้เพื่อให้ทำงานในทุกๆช่วงเวลาที่กำหนดได้และสามารถทำงานในแบบหลังฉากได้ในตอนรันโปรแกรมคอนโทรลนี้จะไม่สามารถมองเห็นได้ (Non-visual Interface)



รูปที่ 21 เครื่องมือสร้างTimer

คุณสมบัติ

Interval เป็นคุณสมบัติที่ใช้กำหนดช่วงเวลามีหน่วยเป็นมิลลิวินาที

อีเว็นต์

Timer เป็นอีเว็นต์ที่เกิดขึ้นในทุกช่วงเวลาที่กำหนดไว้มนคุณสมบัติ Interval

อิมเมจและพิคเจอร์บ็อกซ์ (Image and Picture Box)

อิมเมจคอนโทรลและ พิกเจอร์บ็อกซ์คอนโทรล ทำให้สามารถวางสารสนเทศทางกราฟฟิกไว้ในตำแหน่งเฉพาะบนฟอร์ม พิกเจอร์บ็อกซ์จะมีความยืดหยุ่นมากกว่า และด้วยเหตุนี้จึงต้องการหน่วยความจำและหน่วยประมวลผลในปริมาณที่มากกว่า พิกเจอร์บ็อกซ์เหมาะอย่างยิ่งสำหรับสภาพแวดล้อมไดนามิก เช่น การวาดกราฟฟิกเข้าไปที่จอภาพโดยตรงในขณะที่โปรแกรมกำลังทำงาน หรือการทำไอคอนให้เคลื่อนไหวได้ ด้วยการเคลื่อนย้ายไอคอนไปรอบ ๆ จอภาพ อิมเมจคอนโทรลเหมาะสำหรับสถานการณ์สถิต นั่นคือ ไม่มีการเปลี่ยนแปลงภาพที่สร้างหรือวางเอาไว้บนจอ



รูปที่ 22 เครื่องมือสร้างImage



รูปที่ 23 เครื่องมือสร้างPicture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ

คุณสมบัติ Enabled, Height, Left, Name, Picture, Top, Visible และ Width ส่วนใหญ่จะคล้ายกับคุณสมบัติของฟอร์ม เราสามารถตั้งคุณสมบัติ Picture ในเวลารันจำเป็นต้องใช้ฟังก์ชัน LoadPicture พิกเจอร์บ็อกซ์ซึ่งทำงานเกือบเหมือนกับฟอร์มเล็ก ๆ ที่อยู่ภายในฟอร์มหลัก มีคุณสมบัติที่ไม่มีอยู่ในอิมเมจคอนโทรล นั่นคือ AutoRedraw, FontBold, FontItalic. FontName, Fontsize, FontUnderline ซึ่งคุณสมบัติเหล่านี้ทำงานในทำนองเดียวกับที่มีอยู่ในฟอร์ม อิมเมจคอนโทรลมีคุณสมบัติอย่างหนึ่งที่ไม่มีอยู่ในพิกเจอร์บ็อกซ์คือ คุณสมบัติ Stretch โดยปกติแล้วจะมีค่าเป็น False ทำให้อิมเมจคอนโทรลปรับขนาดตัวเองเพื่อให้พอดีกับภาพที่บรรจุอยู่ แต่เมื่อกำหนดให้เป็น True จะทำให้อิมเมจคอนดทรลปรับขนาดภาพให้พอดีกับขนาดของคอนโทรลที่เราวาดเอาไว้ จอแสดงผลของฟอร์มนั้นประกอบด้วยชั้น 3 ชั้น สารสนเทศทั้งหมดที่แสดงบนฟอร์มโดยตรงจะปรากฏอยู่ชั้นล่างสุด สารสนเทศจากคอนโทรลทางด้านกราฟฟิก 3 ตัว คือ อิมเมจ (Image), เชพ (Shape) และ ไลน์ (Line) จะปรากฏอยู่ตรงชั้นกลาง และอ็อบเจ็กซ์อื่น ๆ ทั้งหมด จะอยู่ชั้นบนสุด ด้วยเหตุนี้อิมเมจคอนโทรลจึงอาจบดบังข้อความที่พิมพ์ออกมาบนฟอร์มในขณะเดียวกันพิกเจอร์บ็อกซ์ก็อาจบดบังอิมเมจคอนโทรลได้เช่นกัน

อิมเมจคอนโทรลจึงอาจบดบังข้อความที่พิมพ์ออกมาบนฟอร์มในขณะเดียวกันพิกเจอร์บ็อกซ์ก็อาจบดบังอิมเมจคอนโทรลได้เช่นกัน

อีเว็นต์

Click, Db1Click เป็นอีเว็นต์พื้นฐานที่พบง่ายและบ่อยที่สุด

DragDrop เกิดขึ้นเมื่อผู้ใช้ลากแล้ววาง

MouseMove, MouseDown, MouseUp เกิดขึ้นเมื่อมีการใช้งานเมาส์นั้นคือ เมื่อมีการเคลื่อนเมาส์ กดปุ่มและปล่อยปุ่มของเมาส์

GetFocus, LostFocus เป็นอีเว็นต์ของพิกเจอร์บ็อกซ์ที่ไม่มีในอิมเมจคอนโทรล เกิดขึ้นเมื่อคอนโทรลได้รับโฟกัสและเมื่อเลื่อนโฟกัสออกจากคอนโทรล

ฟังก์ชันและเมดถอด

เราสามารถเรียกฟังก์ชัน LoadPicture เพื่อใช้ตั้งคุณสมบัติ Picture ของพิกเจอร์บ็อกซ์หรืออิมเมจคอนโทรล พิกเจอร์บ็อกซ์ยังสนับสนุนเมดถอด Cls, Point และเมดถอดทางกราฟฟิกอื่น ๆ

เครื่องมือกำหนดการสื่อสาร (Communication Control)

การติดต่อสื่อสารระหว่าง Computer และ PC จะกระทำผ่าน control ของ Visual Basic ที่เรียกว่า Microsoft communication control หรือ MSComm ผ่าน port อนุกรม RS-232 ซึ่งเป็น port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใช้กันอย่างแพร่หลาย ซึ่งกำหนดมาตรฐานโดยองค์การอุตสาหกรรมอิเล็กทรอนิกส์แห่งอเมริกา (EIA) โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะ คือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (DTE) เข้ากับโมเด็ม (DCE) เป็นต้น



รูปที่ 24 เครื่องมือสร้าง MSComm Control

พอร์ตอนุกรม RS-232 จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ทั้งประเภท 9 ขา และ 25 ขา ซึ่งเราเรียกว่า พอร์ต Com1 และ Com2 นั่นเอง โดยในทางความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิพ UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่งซึ่ง UART จะทำหน้าที่ในการรับ/ส่งข้อมูลดังนี้

การส่งข้อมูล(Data Transmission)

- รับตัวอักษรจากคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต(เราเรียกว่าเป็นขบวนการ Serialization)
- สร้างเฟรมของข้อมูลโดยเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น start,
- Stop และ parity เป็นต้น
- ส่งผ่านข้อมูลที่สร้างขึ้นมาจากขั้นตอนที่ผ่านมาด้วยความเร็วของ baud rate
- แสดงสถานะพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

การรับข้อมูล (Data Receiver)

รับตัวอักษรจากการอินเทอร์เฟส

ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานที่กำหนด โดยหากเฟรมของข้อมูลมีรูปแบบที่ไม่ถูกต้องจะมีการแจ้งข้อผิดพลาด

ตรวจสอบความถูกต้องของพาริตี้

แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร

ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์

แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเทอร์เฟส

การนำ MSComm มาใช้มีรายละเอียดดังนี้คือ

1. ต้องมี MSComm ซึ่งจะมีอยู่แล้วเมื่อตอนที่ติดตั้ง Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กำหนด Custom Control MSCOMM.OCX (สำหรับ Visual Basic 4 ขึ้นไป) โดยการเลือก MSCOMM.OCX จากเมนู Project/Components/ที่ Control เลือก Microsoft Comm Control 5.0 (กรณีที่ใช้ version 5) ก็จะได้ MSComm รูปโทรศัพท์ปรากฏในทูลบอกซ์
3. นำคอนโทรลไปใส่ไว้ใน Form ที่ต้องการใช้งาน
4. กำหนดคุณสมบัติต่าง ๆ ให้กับ Control เขียนฟังก์ชันที่เกี่ยวข้องกับ MSComm ในการใช้ MSComm ติดต่อกับ Serial Port นั้นมีการติดต่อกัน 2 แบบ คือ
   - การสื่อสารแบบกระตุ้นด้วยเหตุการณ์ (Event-Droven Communication) หรือเป็นการเขียนกระตุ้นด้วยอินเทอร์รัปนั่นเอง โดยเมื่อข้อมูลเข้ามาก็จะเกิด CommEvent ขึ้นกับ Oncomm Event
   - การสื่อสารแบบโพลลิ่ง (Polling) โดยการวนรอบตรวจสอบข้อมูลจาก Serial Port ตลอดเวลา

การกำหนด Property ที่สำคัญ ๆ เช่น

- Comport การกำหนดหมายเลขของ Port ในการติดต่อ RS 232 เช่น Com1, Com2 เป็นต้น
  ตัวอย่างการกำหนด ⟹ MSComm1.ComPort = 1
- Settings เป็นการกำหนดอัตราบอด แพริที่ จำนวนข้อมูลบิตจบ เช่น 9600,n,8,1 สำหรับใน PC Link กำหนดเป็น 9600,e,7,2 ตัวอย่างการกำหนด MSComm1.Settings = “9600,e,7,2”
- PortOpen เป็นการกำหนดสถานะว่าให้เปิด หรือ ปิด พอร์ต ตัวอย่างการกำหนด ⟹ MSComm1.PortOpen = True สำหรับการเปิดพอร์ต และ MSComm1.PortZOpen = False สำหรับการสั่งให้ปิดพอร์ต
- Input เป็นการย้ายข้อมูลจากบัฟเฟอร์การรับข้อมูล
- Output ส่งข้อมูลให้กับบัฟเฟอร์ส่งข้อมูลส่งออก
  ตัวอย่างการกำหนด ⟹ MSComm1.Input = ข้อมูลที่จะรับจากพอร์ต
  MSComm1.Output = ข้อมูลที่จะส่งออกพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การควบคุม MSComm

ชื่อ Property	คำอธิบาย
ComEventBreak	การได้รับสัญญารเบรก
ComEventCDTO	เมื่อเกิดไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ CD(Carrier Detect)
ComEventCTSTO	เมื่อเกิดไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ CTS(CarrIer To Send)
ComEventDSRTO	เมื่อเกิดไทม์เอาต์ ขณะที่กำลังคอยสัญญาณ DSR (Data Set Ready)
ComEventFrame	การที่เกิดความผิดผลาดทางเฟรม เป็นลักษณะที่ไม่พบบิตจบตามที่ควรจะเป็น
ComEventOverrun	การที่เกิดความผิดผลาดโอเวอร์รัน เป็นลัษณะที่รับข้อมูลไม่ทันในการประมวลผล
ComEventRxOver	บัฟเฟอร์ที่รับข้อมูลเกิดโอเวอร์โฟวล์ ก็คือรับตัวอักษรหลังจากการรับ EOF Char
ComEventRxParIty	การที่เกิดความผิดผลาดทางพาริตี้ เป็นลัษณะที่ตัวอักษรที่รับได้มีพาริตี้ไม่ถูกต้อง
ComEventTxFull	ตัวบัฟเฟอร์ที่ส่งข้อมูลเต็ม
ComEventDCB	การที่เกิดความผิดผลาดขึ้นโดยไม่ได้คาดถึง

ตารางที่ 13 CommEvent เกี่ยวกับการเกิดสถานะเมื่อเกิดการผิดผลาดในการสื่อสาร

ชื่อ Property	คำอธิบาย
ComEvCD	CD(CarrIrt Detect) เมื่อเปลี่ยนซึ่งคือสายของสัญญาณ ReceIve LIne Signal Detect(RLSD)
ComEvCTS	RCTS(CarrIer To Send)เมื่อมีการเปลี่ยนสถานะเกิดขึ้น
ComEvDSR	DSR(Data Set Ready) เมื่อมีการเปลี่ยนสถานะเกิดขึ้น
ComEvRIng	เมื่อตรวจจับสัญญาณ Ring IndIcator ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ComEvReceIve	เมื่อได้รับข้อมูลเก็บลงใน InputBuffer
ComEvSend	เมื่อส่งข้อมูลออกจาก OutputBuffer
ComEvEof	เมื่อพบอักขระ EOF(End Of FIle)

ตารางที่ 14 CommEvent เกียวกับเกิดสถานะการเมื่อมีการสื่อสาร

ชื่อ Property	คำอธิบาย
ComNone	ไม่ใช้ให้ตรวจสอบแฮนเช็ก
ComXonZXoff	ให้มีการตรวจสอบแฮนเช็ก ในแบบ Xon/Xoff
ComRTS	ให้มีการตรวจสอบแฮนเช็ก ผ่านขา RTS และ CTS
ComRTSXOnXoff	กำหนดให้มีการตรวจทั้ง2แบบคือ RTS-CTS และXOn/Xo

ตารางที่ 15 Handshake Property

ชื่อ Properties	คำอธิบาย
ComInputModeText	คุณสมบัติในการรับข้อมูลมาเป็นแบบข้อความ ปกติจะเป็นค่านี้อยู่แล้ว
ComInputModeBinary	คุณสมบัติในการรับข้อมูลมาเป็นแบบไบนารี่หรือเลขฐานสองนั้นเอง

ตารางที่ 16 Input Mode Property

ชื่อ Property	คำอธิบาย
Break	ในการที่เรากำหนดหรือเคลียร์สัญญาณเบรก
CDHoldIng	ตรวจสอบสัญญาณ Carrier Detect(CD)ดูว่ายังคงมีสถานะอยู่หรือเปล่า
CDTImeout	การกำหนดค่าหรือว่าให้ค่าของเวลา(หน่วย mmSec) ที่รอสัญญาณ Carrier Detect
CommEvent	จะให้ผลของการเกิด Event ของ Communucation
CommID	จะให้ผลของการเฮดเดิลของ Communucation ที่เปิดใช้อยู่
CommPort	การกำหนดหรือว่าอ้างอิงของหมายเลขคอมพอร์ต ที่เปิดใช้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เช่น Com1=1,Com2=2
CTSHoldIng	เป็นการตรวจสอบสัญญาณของ Clear To Send ว่ายังคงมีสถานะอยู่หรือเปล่า
CTSTImeout	การกำหนดค่าหรือว่าให้ค่าของเวลา(หน่วย mmSec) ที่รอสัญญาณ Data Set Ready
DSRHoldIng	เป็นการตรวจสอบสัญญาณของ Data Set Ready ว่ายังคงมีสถานะอยู่หรือเปล่า
DSRTimeout	การกำหนดค่าหรือว่าให้ค่าของเวลา(หน่วย mmSec) ที่รอสัญญาณ Clear To Send
DTREnable	ให้อีนาเบิล สายของสัญญาณ Data TermInal Ready(DTR)
HandshakIng	กำหนดการแฮนเช็คทางฮาร์ดแวร์ เพื่อที่คอยตรวจสอบการรับส่งข้อมูล
InBufferCount	ให้ค่าของจำนวนข้อมูลที่อยู่ภายในบัฟเฟอร์รับข้อมูล
InBufferSIze	กำหนดหรือว่าให้ค่าของขนาดในบัฟเฟอร์รับข้อมูล
Input	เป็นการให้ค่าหรือว่าเคลื่อนย้ายข้อมูลจากบัฟเฟอร์รับข้อมูล
InoputLen	การกำหนดหรือว่าให้ของจำนวนข้อมูลที่นำมาจากบัฟเฟอร์รับข้อมูล
Interval	เป็นการกำหนดอัตราความเร็วของการใช้งานในโหมดโพลลิ่ง
NullDIscard	เป็นการกำหนดให้มีการรับ Null Character เก็บลงในบัฟเฟอร์รับข้อมูล
OutBufferCount	เป็นจำนวนข้อมูลที่คอยอยู่ในบัฟเฟอร์ส่งข้อมูล
OutBufferSIze	การกำหนดหรือว่าให้ค่าขนาดของบัฟเฟอร์ส่งข้อมูล
Output	เป็นการส่งข้อมูลให้กับบัฟเฟอร์ส่งข้อมูลเพื่อทำการส่งข้อมูลออก
ParItyReplace	เป็นการกำหนดให้ส่งอักขระที่กำหนดนี้แทนหากเกิดการผิดผลาดในข้อมูล
PortOpen	เป็นการกำหนดหรือว่าให้ค่าของสถานะพอร์ตว่าเปิดหรือปิดอยู่
Rthrehold	การกำหนดหรือว่าให้ค่าของจำนวนข้อมูลที่เก็บลงในบัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รับข้อมูลก่อนการเกิด CommEvent ในการรับข้อมูล
RTSEnable	ให้อีนาเบิล สัญญาณ Request To Send(RTS)
SettIngs	เป็นการกำหนดอัตราบอด พาริตี้ ข้อมูล บิตหยุด
Sthreshold	การกำหนดหรือว่าให้ผลของจำนวนข้อมูลที่เก็บลงในของบัฟเฟอร์ส่งข้อมูลก่อนการเกิด CommEvent ในการที่ส่งข้อมูล

ตารางที่ 17 MSComm Control Property

ชื่อ Property	ค่าตัวเลข	คำอธิบาย
ComInvalidPrppertyValue	380	ตั้งค่าไม่ถูกต้อง
ComSetNotSupported	383	กำหนดค่าที่ตั้งไว้สามารถอ่านได้อย่างเดียว เขียนหรือเปลี่ยนข้อมูลไม่ได้
ComGetNotSupported	394	กำหนดค่าที่รับไว้สามารถอ่านได้อย่างเดียว เขียนหรือเปลี่ยนข้อมูลไม่ได้
ComPortOpen	8000	จะอ่านค่าไม่ได้ในขณะที่ Port นั้นยังถูกเปิดใช้อยู่
ComPortOpen	8001	ค่าของเวลาที่หาออกมาได้ต้องมีค่ามากกว่าศูนย์
ComPortOpen	8002	กำหนดหมายเลข Port ไม่ถูกต้อง
ComPortOpen	8003	ผลลัพธ์ของข้อมูลจะเกิดในขณะที่มีการทำงาน
ComPortOpen	8004	Port นั้นจะสามารถอ่านค่าได้ในขณะที่มีการทำงานเท่านั้น
ComPortAlreadyOpen	8005	Port ได้ถูกเปิดไว้เรียบร้อยแล้ว
ComPortAlreadyOpen	8006	อุปกรณ์เกิดความผิดผลาดหรือไม่สามารถรองรับค่าได้
ComPortAlreadyOpen	8007	อุปกรณ์ไม่ยอมรับค่าที่ Baud Rate ถูกตั้งเอาไว้
ComPortAlreadyOpen	8008	ขนานของข้อมูลผิดผลาด
ComPortAlreadyOpen	8009	ค่าของตัวแปรที่แสดงอยู่ผิดผลาด
ComPortAlreadyOpen	8010	อุปกรณ์ภายนอก(Hardware)ยังไม่พร้อมที่จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ComPortAlreadyOpen	8011	ฟังก์ชั่นไม่สามารถกำหนดแถวข้อมูลได้
ComNoOpen	8012	Com Port ยังไม่พร้อมที่จะถูกเปิดใช้งาน
ComNoOpen	8013	Com Port พร้อมที่จะถูกเปิดใช้งาน
ComNoOpen	8014	Com Port ไม่สามารถทำงานได้
ComSetCommStateFailed	8015	ไม่สามารถตั้งค่าสถานะของ Port ได้
ComSetCommStateFailed	8016	ไม่สามารถSet Port ตามเหตุการณ์ที่กำหนดให้ได้
ComPortNotOpen	8018	จะสามารถหาผลลัพธ์ของข้อมูลได้ก็ต่อเมื่อ Port มีการทำงานแล้วเท่านั้น
ComPortNotOpen	8019	Port ไม่มีที่ว่างมีข้อมูลเต็มใน Port
ComReadError	8020	เกิดความผิดลาดขึ้นขณะที่อ่าน
ComDCBError	8021	เกิดความผิดผลาดภายในต้องไปแก้ไขที่ตัวควบคุม Port

ตารางที่ 18 รายละเอียดที่บ่งชี้ถึงความผิดผลาดในการใช้ MS Comm Cntrol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 หมากฮอส

2.6.1 ประวัติหมากฮอส

เชื่อกันว่า หมากฮอส (checkers) มีกำเนิดมาจากหมากรุก (chess) โดยดัดแปลงรูปแบบและกติกาการเล่นให้ง่ายขึ้น แม้แต่เด็กก็สามารถเรียนรู้และเล่นเป็นได้ในเวลาไม่นาน คำว่า checkers หมายถึงตาหมากรุก เบี้ยจะเดินคล้ายการเดินของควีน (queen) ในหมากรุกยุคแรก ๆ (เหมือนเม็ดใน หมากรุกไทย) คือเดินทแยงไปด้านหน้าได้ครั้งละ 1 ตา (ถอยหลังไม่ได้) เมื่อเดินไปจนสุดกระดานอีก ด้านหนึ่งจะมีการเปลี่ยนเป็นฮอส และสามารถเดินถอยหลังได้ จากหลักฐานทางประวัติศาสตร์ ทำให้ทราบว่า หมากรุกมีกำเนิดทางตอนเหนือของประเทศ อินเดียประมาณ พ.ศ.1113 ส่วนหมากฮอสปรากฏครั้งแรกในบันทึกภาษาสเปน เมื่อ พ.ศ.2090 ก่อน จะแพร่หลายไปสู่ส่วนต่าง ๆของโลก ในยุคกลาง (middle ag) หมากฮอส ถูกรังเกียจว่าเป็น "หมากรุกของผู้หญิง" หรือ chess for lady โดยยังมีหลักฐานปรากฏในภาษาปัจจุบันของชาติต่าง ๆ เช่น Jeudes ในภาษาฝรั่งเศษ Damenspiel ในภาษาเยอรมัน และ Giuoco delle dame ในภาษาอิตาลี เป็นต้น กติกาหมากฮอสท้องถิ่นแต่ละประเทศ จะมีส่วนปลีกย่อยต่างกันออกไป เช่น ในสเปนและไทย ฮอสสามารถ เดินได้ยาวเหมือนบิชอพ(Bishop)ของหมากรุก ในอิตาลี เบี้ยตัวเดียวจะกินฮอสไม่ได้ และในโปแลนด์ ใช้เล่นบนตาราง 10 x 10 ช่อง มีเบี้ยข้างละ 20 ตัว เป็นต้น

หมากฮอสสากล

เป็นหมากฮอสที่เป็นที่ยอมรับและใช้แข่งขันทั้งระดับประเทศและระหว่างประเทศ ผู้เล่นทั้ง 2 ฝ่ายจะมีเบี้ยข้างละ 12 ตัว ฝ่ายดำเดินก่อนเมื่อเปลี่ยนเป็นฮอส(Checker King)อีกฝ่ายหนึ่งจะ สวมมงกุฎให้โดยใช้เบี้ยสีเดียวกันวางซ้อนลงไปอีก 1 ตัว ฮอสจะเดินหน้าหรือถอยหลังก็ได้ครั้งละ1ตา การเล่นหมากฮอสสากลในปัจจุบัน ได้ถูกค้นคว้าแง่มุมการเดินต่าง ๆ จนเกือบหมดสิ้น ดังนั้น ถ้าไม่เลือก เดินอย่างอิสระ นักเล่นที่เก่งสามารถเดินเสมอกี่กระดานก็ได้เท่าที่ต้องการเล่น ยกเว้นแต่จะเสี่ยงเดินเพื่อ เอาชนะ ในการแข่งขันจึงเต็มไปด้วยการเสมอ ตัวอย่างเช่นการแข่งขันชิงแชมป์ระหว่าง Anderson กับ James Willie ปรากฏว่าเสมกกันถึง 54 กระดาน หมากฮอสจึงได้ชื่อว่าเป็นกีฬาแห่งการเสมอกัน ภายหลังจึงมีการใช้กติกา 2 ก้าวบังคับ (47 วิธี) และ 3 ก้าวบังคับ (137 วิธี)สำหรับการเปิดหมากซึ่งเราได้ดัดแปลงนำมาใช้ การแข่งขันที่สำคัญของกีฬาหมากฮอส เป็นการแข่งประเภททีมระหว่างประเทศ เริ่มมีครั้งแรกปี พ.ศ. 2427 ระหว่างอังกฤษกับสก๊อตแลนด์ ในปี พ.ศ.2448 อังกฤษส่งทีมข้ามทวีปไปแข่งขันกับสหรัฐอเมริกา จนถึงปัจจุบันนี้ก็ยังมีการจัดการแข่งขันระหว่างประเทศอยู่ แม้จะไม่กว้างขวางและได้รับนิยมเท่าหมากรุกก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมากฮอสไทย

ดัดแปลงมาจากหมากฮอสสากล ความแตกต่างก็คือฮอสสามารถเดินได้ยาวเหมือนบิชอพ (Bishop) และลดเบี้ยเหลือฝ่ายละ 8 ตัว การแปรรูปมีน้อยกว่า หากค้นคว้าและฝึกจนเชี่ยวชาญแล้วจะ แพ้ไม่เป็นเลย เมื่อหมากฮอสไทยพัฒนาถึงจุดหนึ่งแล้ว ก็คงได้ชื่อว่าเป็นกีฬาแห่งการเสมอกัน การใช้กติกา 3 ก้าวบังคับไม่อาจช่วยได้มากนัก และดูไม่สมเหตุสมผลสำหรับกีฬาที่ต้องการความเป็นอิสระ ในการคิดและตัดสินใจ ดังนั้นรูปแบบและกติกาการเล่นหมากฮอสไทยคงจะมีการเปลี่ยนแปลงในอนาคต เพื่อให้คงเป็นกีฬาแห่งการสร้างสรรค์และให้ความเพลิดเพลินต่อไป

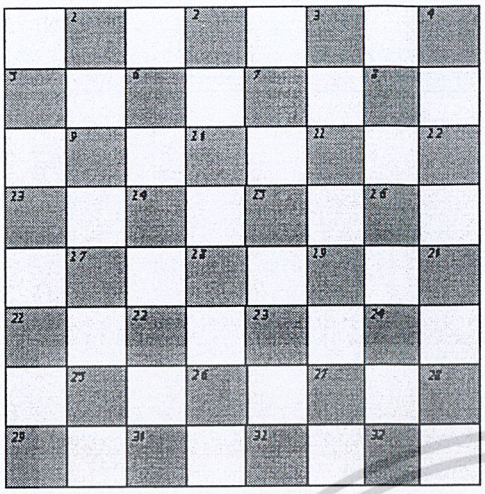
2.6.2 กติกาการเล่นหมากฮอส

หมากฮอส เป็นกีฬาหมากกระดานชนิดหนึ่ง ประกอบด้วยผู้เล่น 2 ฝ่าย และอุปกรณ์ ในที่นี้ จะแสดงเฉพาะกรณีการเล่นทั่วไป หากเป็นการแข่งขัน จะต้องมีรายละเอียดเพิ่มเติมอีกมาก

อุปกรณ์การเล่น ได้แก่ กระดานและตัวหมาก

กระดาน เป็นวัสดุพื้นเรียบ รูปทรงสี่เหลี่ยมจัตุรัส แบ่งแต่ละด้านออกเป็น 8 ช่องเท่ากัน จะได้ตารางย่อย 64 รูป เรียกสี่เหลี่ยมย่อยแต่ละรูปว่า ตา หรือ ช่อง กำหนดเป็นตาสีเข้ม 32 ตาและตาสีอ่อน 32 สลับกันโดยให้ตามุมซ้ายด้านบนของผู้เล่นเป็นตาสีอ่อน ตาที่ใช้เดินจะใช้เฉพาะตาสีเข้มรวม 32 ตา ในที่นี้จะกำหนดตำแหน่งของแต่ละตาด้วยเลขจำนวนนับ เพื่อให้ทราบว่ากล่าวถึงตาใดในกระดาน โดยเริ่มจากตาสีเข้มบนซ้ายสุดเป็น เลข 1 ตาสีเข้มถัดมาในแนวนอนเป็นตา 2 และในแนวนอนถัดไปเป็นตา 3 และ 4 ตามลำดับ จากนั้นในแถวต่อมาเริ่มตาสีเข้มซ้ายมือสุดเป็นตา 5 เรียงลำดับไปเรื่อย ๆ จนถึงแถวที่ 8 ตาสีเข้มขวามือล่างสุด จะเป็นตา 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตัวอย่างกระดานและการกำหนดตำแหน่งตาโดยตัวเลข

ตัวหมาก เมื่อเริ่มต้น ทั้ง 2 ฝ่ายจะมีหมากฝ่ายละ 8 ตัว ในที่นี้กำหนดให้เป็นฝ่ายขาวและแดง และตั้งกระดานโดยให้ตัวหมากของผู้เล่นอยู่บนสองแถวแรกที่ใกล้ตัวผู้เล่นฝ่ายนั้น

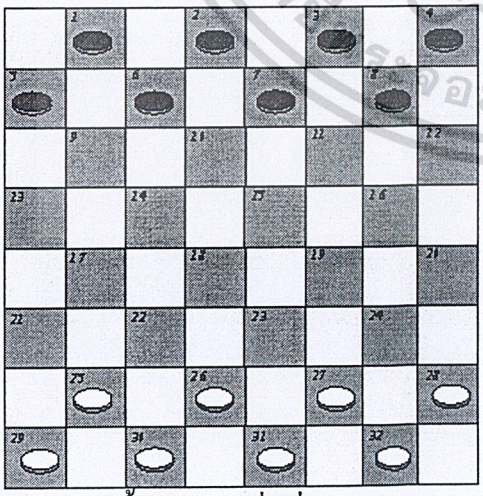
แสดงตัวอย่างสัญลักษณ์ตัวหมากของทั้ง 2 ฝ่าย

= เบี้ยขาว

= ฮอสขาว

= เบี้ยแดง

= ฮอสแดง



แสดงการตั้งกระดานเมื่อเริ่มต้นการเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กติกา

1. ผู้เล่นหมากสีขาวเป็นฝ่ายเดินก่อน
2. ทั้ง 2 ฝ่ายสลับกันเดินจนกว่าการเล่นจะสิ้นสุด
3. การเดินหมาก

3.1 ตัวหมากของแต่ละฝ่ายเมื่อเริ่มเล่นเรียกว่า เบี้ย

3.2 เบี้ยจะเดินได้ครั้งละ 1 ตา ในลักษณะเฉียงไปด้านหน้า เว้นแต่จะมีหมากตัวอื่นขวางทาง หรือเมื่อไปชนริมกระดาน

3.3 หากเบี้ยเดินไปจนถึงแถวที่ 8 นับจากตัวผู้เล่น ตัวเบี้ยจะเปลี่ยนเป็นฮอส (เรียกว่า เข้าฮอส) แสดงให้เห็นว่าแตกต่างจากเบี้ย โดยใช้เบี้ยสีเดียวกันวางซ้อนเบี้ยตัวที่เข้าฮอสอีก 1 ตัว

3.4 ฮอสจะเดินกี่ตาก็ได้ในแนวเฉียง ทั้งด้านหน้าและด้านหลัง

3.5 หากในการเดินของเบี้ยหรือฮอส มีหมากของฝ่ายตรงข้ามวางอยู่บนตาเดิน และมีตาว่างอยู่ตาถัดไปในแนวเดียวกัน เบี้ยหรือฮอสของฝ่ายที่เดิน สามารถเดินข้ามหมากฝ่ายตรงข้ามและยกหมากตัวนั้นออกจากกระดานไป และเรียกการเดินในลักษณะนี้ว่า การกิน

3.6 การกินสามารถทำได้อย่างต่อเนื่อง กล่าวคือ หากเมื่อกินหมากของคู่ต่อสู้แล้ว มีหมากตัวอื่นของอีกฝ่ายหนึ่งวางอยู่บนตากิน เบี้ยหรือฮอสนั้น สามารถกินต่อได้อีก เรียกว่าเป็นการกินหลายต่อโดยหมากตัวเดียว ในการเดินครั้งเดียว

3.7 แม้ผู้เล่นจำมีอิสระในการเลือกเดินตัวหมากตัวใดก็ได้ แต่หากมีตัวหมากที่กินหมากของ อีกฝ่ายหนึ่งได้ ผู้เล่นจะต้องเดินหมากเพื่อกินในการเดินครั้งนั้น จะไปเดินตัวอื่นไม่ได้ (เว้นแต่จะมีตัวหมากที่กินคู่ต่อสู้ได้มากกว่า 1 ตัว ก็สามารถเลือกที่จะกินวิธีใดวิธีหนึ่ง) กล่าวคือ ไม่มีการหักขา หรือเลี่ยงการกินอย่างเด็ดขาด

4. การตัดสิน

ผลของการเล่นแต่ละกระดาน จะสิ้นสุดลงโดยมีการแพ้-ชนะ หรือเสมอ ดังนี้

4.1 ฝ่ายที่สามารพกินหมากคู่ต่อสู้จนหมดจากกระดานเป็นฝ่ายชนะ

4.2 ฝ่ายที่ไม่สามารถเดินหมากตัวใดได้เลย แม้จะมีตัวหมากเหลืออยู่ เป็นฝ่ายแพ้

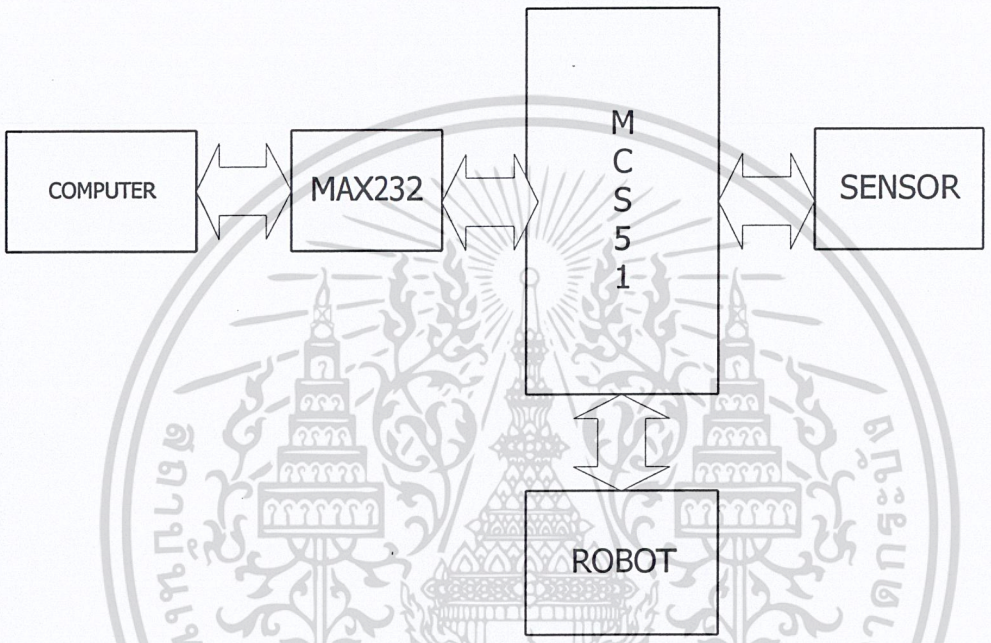
4.3 ผลการสิ้นสุดโดยการยินยอมทั้ง 2 ฝ่าย เช่น ตกลงยอมเสมอ หรือฝ่ายใดฝ่ายหนึ่งยอมแพ้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงานและทดลองผล

3.1 บล๊อคไดอะแกรมการทำงาน

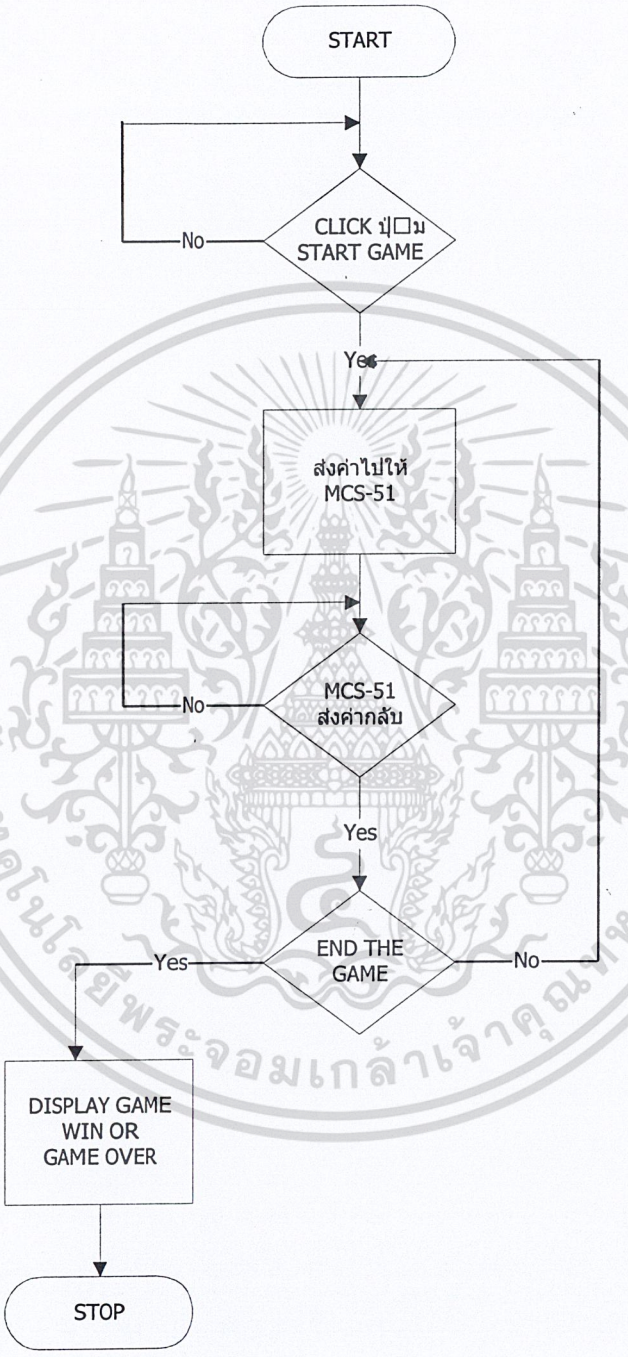


บล๊อคไดอะแกรมการทำงาน

รูปที่ 25 บล๊อคไดอะแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

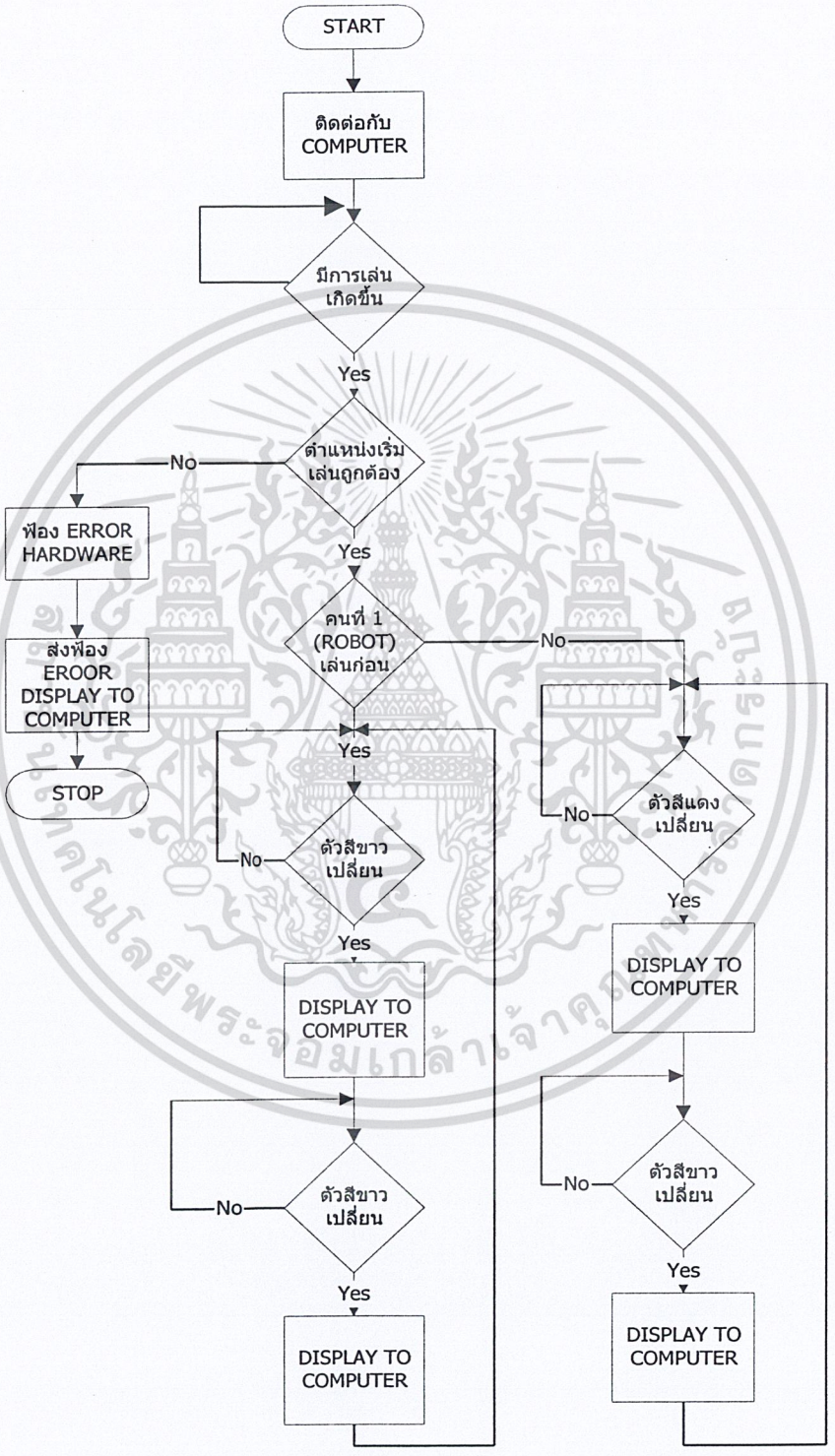
3.2 Flow Chart MCS-51



รูปที่ 26 Flow Chart การทำงาน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Flow Chart Visual Basic6



รูปที่ 27 Flow Chart การทำงาน Visual Basic6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 หลักการออกแบบ

3.4.1 Hardware

ใช้ MCS-51 เป็นตัวควบคุมมือกลให้จับหมากฮอส ต้องกำหนดพอร์ตที่ต้องใช้ ซึ่งในการทำนี้ต้องใช้พอร์ตเป็นจำนวนมาก โดยเฉพาะในวงจร SENSOR ต้องใช้ 32 ตัว และยังมีที่ต้องต่อไปยังมือกล เพื่อควบคุมมือกลให้จับหมากฮอส ดังนั้น ต้องทำการขยายพอร์ตออกไปอีก โดยใช้ IC 8255 จำนวน 2 ตัว เพื่อขยายพอร์ตให้พอกับที่ต้องการใช้

ในการขยายพอร์ตโดยใช้ IC 8255 จะต้องออกแบบวงจรเพื่อที่จะให้วงจรทำงานตามตำแหน่งที่อยู่ที่กำหนด คือ ถ้าต้องการ ADDRESS ที่ 8C00h-8C03h ของตัวที่ 1 และ ADDRESS ที่ 8F00h-8F03h ของตัวที่ 2 จะต้องต่อสาย ADDRESS,DATA,RD,WR และสายอื่นอีกแสดงดังรูปที่ 10 ใช้ 4 พอร์ต ต่อเข้ากับวงจร SENSOR และที่เหลือไว้สำหรับต่อเข้ากับมือกล

ในส่วนของวงจร SENSOR จะใช้การตรวจจับแสงสว่าง กล่าวคือ ถ้ามีแสงสว่างพอประมาณมาตกกระทบที่ SENSOR แสดงว่าที่ตำแหน่งนั้นไม่มีตัวหมากฮอสอยู่ โดยวงจร SENSOR ใช้ LDR เป็นตัวตรวจจับ ซึ่งหลักการของ LDR นั้นคือ เมื่อมีแสงมาตกกระทบที่ตัวมัน ก็จะทำให้ตัวมันมีความต้านทานน้อยลง และทำให้มีกระแสไหลผ่านตัวมันมากขึ้น จากนั้นต่อผ่าน OP-AMP ทำให้เป็นลอจิก “0” , “1” เพื่อนำไปต่อเข้ากับ MCS-51 ได้ แสดงดังรูปที่ 12

LDR แต่ละตัวต้องติดอยู่ที่ตารางหมากฮอสของแต่ละช่อง โดยที่ ตารางหมากฮอสนั้นมีด้วยกัน 64 ช่อง แต่ใช้เพียง 32 ช่องเท่านั้น ดังนั้นจึงติด LDR ไว้ 32 ช่องเท่านั้น

3.4.2 Software

นอกจากใช้แอสเซมบี้ที่ใช้เขียนMCS-51แล้วยังใช้โปรแกรมวิชวลเบสิกเขียนโปรแกรมหมากฮอสเพื่อที่ส่งต่อไปยัง MCS-51 ควบคุมมือกลอีกทีหนึ่ง และยังใช้เขียนเพื่อติดต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก เขียนให้แสดงออกที่หน้าจอคอมพิวเตอร์อีกด้วย

ในส่วนของ MCS-51 คือ เมื่อคอมพิวเตอร์ส่งค่าที่เป็นการบอกว่าหมากฮอสเล่นแล้ว ใน MCS-51 จะรับค่านั้นมาแล้วตรวจสอบดูว่าเป็นตัวไหนที่เปลี่ยนตำแหน่งไป แล้วทำการควบคุมมือกลให้จับตัวหมากจากกระดานจริงเปลี่ยนตำแหน่งให้ถูกตามคอมพิวเตอร์ และเมื่อมีผู้เล่นนั้นเล่นแล้ว MCS-51 จะส่งค่าไปยังคอมพิวเตอร์อีกทีหนึ่งด้วย แสดงดังรูปที่ 26

ในส่วนของวิชวลเบสิก ทำการเขียนโปรแกรมหมากฮอสขึ้นมาเพื่อให้เป็นสมองของมือกลและทำการแสดงออกที่หน้าจอคอมพิวเตอร์ด้วย แสดงดังรูปที่ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การทดลองและผลการทดลอง

การทดลอง แบ่งออกเป็น 2 ทาง คือ การทดลองทางด้าน HARDWARE และ SOFTWARE

HARDWARE

ในการทดลองแบ่งออกเป็น 2 ส่วนคือ พอร์ตขยายอนุกรม 8255 และ วงจร LDR

-พอร์ตขยายอนุกรม 8255 การทดลองในส่วนนี้เป็นการทดสอบ พอร์ตของ 8255 (A,B,C) ทำงานที่ตำแหน่ง แอดเดรส ที่กำหนดไว้ว่าตรงจริงหรือไม่ โดยในวงจรมี 8255 อยู่ 2 ตัวทำงานที่ตำแหน่ง DC00-DC03 และ FC00-FC03 ทดสอบโดยให้พอร์ต A,B,C ของตัวแรกเป็นอินพุท และตัวที่สองเป็นเอาท์พุท แล้วสลับกันบ้างโดยที่ให้ตัวแรกเป็นเอาท์พุท และตัวที่สองเป็นอินพุท การทดลอง ป้อนลอจิกเข้าที่อินพุท แล้วดูผลที่เอาท์พุท

ผลการทดลอง ผลที่เอาท์พุทออกตรงตามที่ป้อนอินพุทเข้ามาไม่ว่าจะสลับตัวใดแล้วก็ตาม

-วงจร LDR ในการทดลองชุด LDR ต่อวงจรดังรูปที่ 12 แต่ R ใช้ ความต้านทานปรับค่าได้ เพื่อหาค่าของ resister ที่ต้องการ คือเมื่อต่อแล้วทำให้ LDR ทำงานที่ระยะพอดีตามความเหมาะสม โดยการให้ LDR รับแสง และปรับความต้านทานให้วงจรทำงานที่ความสว่างพอเหมาะสม เมื่อได้แล้ว ทำการทดลอง LDR ให้ครบ 32 ตัว แล้วนำมาติดลงตารางหมากฮอส เพื่อเป็นตัวตรวจจับตำแหน่งของหมากฮอส

ผลการทดลอง ได้ค่าความต้านทาน 33 กิโลโอห์ม

SOFTWARE

ในการทดลองแบ่งออกเป็น 2 ส่วนคือ MCS-51 และ วิชวลเบสิก

-MCS-51 การทดลองโปรแกรม MCS-51 นั้น จะทดสอบพอร์ตของแต่ละพอร์ตของ MCS-51 โดยการเขียนโปรแกรม ทั้งรับค่าจากพอร์ต และออกจากพอร์ต ของแต่ละพอร์ต

ผลการทดลอง รับและส่งจากพอร์ตได้

-วิชวลเบสิก ในส่วนของวิชวล จะทดลองการรับ-ส่งข้อความ จากคอมพิวเตอร์กับMCS-51

ผลการทดลอง รับส่งข้อความได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สรุปวิจารณ์และการใช้งาน

ปริญญานิพน์ฉบับนี้ ได้ทำการเขียนโปรแกรมควบคุมมือกลให้สามารถหยิบหมากฮอสเล่นแข่งกับคนซึ่งเป็นพื้นฐานอุปกรณ์และเครื่องจักรที่ทำงานโดยอัตโนมัติในโรงงานอุตสาหกรรมโดยการควบคุมจะใช้ MICROCONTROLLER MCS-51 เป็นตัวควบคุมโดยทำงานร่วมกันกับ COMPUTERที่ใช้โปรแกรม VISUAL BASIC6 ทำการอินเตอร์เฟสกันผ่านพอร์ตอนุกรมด้วยไอซี MAX232 จากการทดลองมือกลสามารถที่จะจับหมากฮอสเล่นแข่งกับคนได้ ซึ่งถ้านำไปประยุกต์ใช้ในโรงงานอุตสาหกรรมจะสามารถลดต้นทุนค่าแรงงานในการจ้างคน และสามารถทำงานได้ตลอดเวลาทำให้ได้งานที่มีปริมาณมากขึ้น

แนวทางในการพัฒนาและประยุกต์ใช้งาน

ควรประยุกต์ใช้งานกับโรงงานอุตสาหกรรมที่มีการทำงานตลอดเวลา

ปัญหาที่เกิดจากการทำโครงงาน

ในการทดลองปัญหาที่เกิดขึ้นจะอยู่กับอุปกรณ์การทดลองและการเขียนโปรแกรม ASSEMBLYเพื่อควบคุมมือกลและโปรแกรม VISUAL BASIC6 จะต้องศึกษาคุณสมบัติและคำสั่งต่างๆให้เข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
    Dim counter As Integer
    Dim table(8, 8)
    Dim chek(8, 8)
    Dim keep As String
    Dim rote(5) As String
    Dim a0 As Integer
    Dim a1 As Integer
    Dim a2 As String
    Dim a3 As String
    Dim a4 As String
    Dim a5 As String
    Dim a6 As String
    Dim a7 As String
    Dim a9 As Integer
    Dim b1 As Integer
    Dim b2 As Integer
    Dim b3 As Integer
    Dim b4 As Integer
    Dim b5 As Integer
    Dim b6 As Integer
    Dim bb As Integer
    Dim bbb As Integer
    Dim abcd As Integer
    Dim abc As Integer
    Dim ab As Integer
    Dim a As Integer
    Dim az As Variant
    Dim z As Integer
    Dim z1 As Integer
    Dim z2 As Integer
    Dim z3 As Integer
    Dim ch As Integer
    Dim ch0 As Integer
    Dim ck As Integer
    Dim ct As Integer
    Dim che As Integer
    Dim chk As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim ww As Integer
    Dim w As Integer
    Dim rr As Integer
    Dim walk As Integer
    Dim l As Integer
    Dim ton0(32) As Integer
    Dim ton1(32) As Integer
    Dim ton As Integer
    Dim y0 As Integer
    Dim y1 As Integer
    Dim y2 As Integer
    Dim str0(32) As String
    Dim str1(32) As String
    Dim two As Integer
    Dim m As Integer
    Dim rot As Integer

Private Sub Command1_Click()
    Dim Sr As String
    Dim sr2 As String
    Dim q As Integer
    a0 = 0
    ab = 0
    ch = 0
    ck = 0
    ct = 0
    abc = 0
    che = 0
    chk = 0
    two = 0
    y0 = 16
    y1 = 0
    y2 = 16
    rr = 0
    rot = 1
    ton0(0) = 1
    ton0(1) = 1
    ton0(2) = 1
    ton0(3) = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ton0(4) = 1
    ton0(5) = 1
    ton0(6) = 1
    ton0(7) = 1
    ton0(8) = 0
    ton0(9) = 0
    ton0(10) = 0
    ton0(11) = 0
    ton0(12) = 0
    ton0(13) = 0
    ton0(14) = 0
    ton0(15) = 0
    ton0(16) = 0
    ton0(17) = 0
    ton0(18) = 0
    ton0(19) = 0
    ton0(20) = 0
    ton0(21) = 0
    ton0(22) = 0
    ton0(23) = 0
    ton0(24) = 1
    ton0(25) = 1
    ton0(26) = 1
    ton0(27) = 1
    ton0(28) = 1
    ton0(29) = 1
    ton0(30) = 1
    ton0(31) = 1
    If Option3.Value = True Then
       sr2 = "ใช้ Mouse เล่น Game"
    ElseIf Option4.Value = True Then
       sr2 = "ใช้หน้าจอเป็น Display จากกระดานจริง"
    End If
    If Option5.Value = True Then
       Sr = MsgBox(sr2 & Chr(13) + Chr(10) & "โดยที่ " & Option5.Caption, vbQuestion +
vbYesNo)
    ElseIf Option6.Value = True Then
       Sr = MsgBox(sr2 & Chr(13) + Chr(10) & "โดยที่ " & Option6.Caption, vbQuestion +
vbYesNo)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    If Sr = vbYes Then
        Timer4.Enabled = True
        Timer4.Interval = 100
        Timer2.Enabled = True
        mnuStop.Enabled = True
        Command8.Enabled = True
        Command8.Caption = "ห&ยุดเล่นชั่วคราว"
        If Option1.Value = True Then
            MsgBox "เล่นหมากฮอสกันระหว่าง :>>" & " [ " & "ROBOT" & " กับ " & Text1.Text & " ]
", vbOKOnly
            Text2 = Text1
            Text1 = "Robot"
        ElseIf Option2.Value = True Then
            MsgBox "เล่นหมากฮอสกันระหว่าง :>>" & " [ " & Text1.Text & " กับ " & Text2.Text & "
] ", vbOKOnly
        End If
        Label65.Caption = Text1
        Label67.Caption = Text2
        Label71.Caption = Text1
        Label72.Caption = Text2
        Option5.Caption = Text1
        Option6.Caption = Text2
        If Option5.Value = True Then
            Label74.Caption = Text1 & " เริ่มเล่นก่อน"
        ElseIf Option6.Value = True Then
            Label74.Caption = Text2 & " เริ่มเล่นก่อน"
        End If
        Option1.Enabled = False
        Option2.Enabled = False
        Option3.Enabled = False
        Option4.Enabled = False
        Option5.Enabled = False
        Option6.Enabled = False
        Text1.BackColor = &H8000000B
        Text1.Enabled = False
        Text2.BackColor = &H8000000B
        Text2.Enabled = False
        Command1.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Command6.Enabled = True
    mnuStart.Enabled = False
    mnuNew.Enabled = True
    If Option4.Value = True Then
    For q = 0 To 31 Step 1
       If Label4(q).Visible = True Then
          y1 = y1 + 1
       End If
    Next q
    If y1 <> 16 Then
          MsgBox "โปรดตรวจสอบตำแหน่งหมากฮอสให้ถูกต้อง !!!"
          Command6_Click
    End If
    y1 = 0
    End If
'*************************************************************************
    Image1(0).Visible = True
    Image1(1).Visible = True
    Image1(2).Visible = True
    Image1(3).Visible = True
    Image1(4).Visible = True
    Image1(5).Visible = True
    Image1(6).Visible = True
    Image1(7).Visible = True
    Image2(0).Visible = True
    Image2(1).Visible = True
    Image2(2).Visible = True
    Image2(3).Visible = True
    Image2(4).Visible = True
    Image2(5).Visible = True
    Image2(6).Visible = True
    Image2(7).Visible = True
    Image3(0).Visible = False
    Image3(1).Visible = False
    Image3(2).Visible = False
    Image3(3).Visible = False
    Image3(4).Visible = False
    Image3(5).Visible = False
    Image3(6).Visible = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Image3(7).Visible = False
    Image4(0).Visible = False
    Image4(1).Visible = False
    Image4(2).Visible = False
    Image4(3).Visible = False
    Image4(4).Visible = False
    Image4(5).Visible = False
    Image4(6).Visible = False
    Image4(7).Visible = False
'*************************************************************************
  table(0, 0) = 5
  table(0, 1) = 1
  table(0, 2) = 5
  table(0, 3) = 1
  table(0, 4) = 5
  table(0, 5) = 1
  table(0, 6) = 5
  table(0, 7) = 1
  table(1, 0) = 1
  table(1, 1) = 5
  table(1, 2) = 1
  table(1, 3) = 5
  table(1, 4) = 1
  table(1, 5) = 5
  table(1, 6) = 1
  table(1, 7) = 5
  table(2, 0) = 5
  table(2, 1) = 0
  table(2, 2) = 5
  table(2, 3) = 0
  table(2, 4) = 5
  table(2, 5) = 0
  table(2, 6) = 5
  table(2, 7) = 0
  table(3, 0) = 0
  table(3, 1) = 5
  table(3, 2) = 0
  table(3, 3) = 5
  table(3, 4) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

table(3, 5) = 5
table(3, 6) = 0
table(3, 7) = 5
table(4, 0) = 5
table(4, 1) = 0
table(4, 2) = 5
table(4, 3) = 0
table(4, 4) = 5
table(4, 5) = 0
table(4, 6) = 5
table(4, 7) = 0
table(5, 0) = 0
table(5, 1) = 5
table(5, 2) = 0
table(5, 3) = 5
table(5, 4) = 0
table(5, 5) = 5
table(5, 6) = 0
table(5, 7) = 5
table(6, 0) = 5
table(6, 1) = 2
table(6, 2) = 5
table(6, 3) = 2
table(6, 4) = 5
table(6, 5) = 2
table(6, 6) = 5
table(6, 7) = 2
table(7, 0) = 2
table(7, 1) = 5
table(7, 2) = 2
table(7, 3) = 5
table(7, 4) = 2
table(7, 5) = 5
table(7, 6) = 2
table(7, 7) = 5
chek(0, 0) = 5
chek(0, 1) = 10
chek(0, 2) = 5
chek(0, 3) = 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

chek(0, 4) = 5
chek(0, 5) = 12
chek(0, 6) = 5
chek(0, 7) = 13
chek(1, 0) = 14
chek(1, 1) = 5
chek(1, 2) = 15
chek(1, 3) = 5
chek(1, 4) = 16
chek(1, 5) = 5
chek(1, 6) = 17
chek(1, 7) = 5
chek(2, 0) = 5
chek(2, 1) = 0
chek(2, 2) = 5
chek(2, 3) = 0
chek(2, 4) = 5
chek(2, 5) = 0
chek(2, 6) = 5
chek(2, 7) = 0
chek(3, 0) = 0
chek(3, 1) = 5
chek(3, 2) = 0
chek(3, 3) = 5
chek(3, 4) = 0
chek(3, 5) = 5
chek(3, 6) = 0
chek(3, 7) = 5
chek(4, 0) = 5
chek(4, 1) = 0
chek(4, 2) = 5
chek(4, 3) = 0
chek(4, 4) = 5
chek(4, 5) = 0
chek(4, 6) = 5
chek(4, 7) = 0
chek(5, 0) = 0
chek(5, 1) = 5
chek(5, 2) = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    chek(5, 3) = 5
    chek(5, 4) = 0
    chek(5, 5) = 5
    chek(5, 6) = 0
    chek(5, 7) = 5
    chek(6, 0) = 5
    chek(6, 1) = 24
    chek(6, 2) = 5
    chek(6, 3) = 25
    chek(6, 4) = 5
    chek(6, 5) = 26
    chek(6, 6) = 5
    chek(6, 7) = 27
    chek(7, 0) = 20
    chek(7, 1) = 5
    chek(7, 2) = 21
    chek(7, 3) = 5
    chek(7, 4) = 22
    chek(7, 5) = 5
    chek(7, 6) = 23
    chek(7, 7) = 5
    End If
End Sub

Private Sub Command5_Click()
    Dim str As Variant
    str = MsgBox("คุณต้องการที่จะออกจากโปรแกรมนี้?", vbQuestion + vbYesNo)
    If str = vbYes Then
        MsgBox "ขอบคุณทุกๆท่านที่เข้ามาเล่น โอกาสต่อไปเชิญเข้ามาเล่นอีกนะครับ"
        MSComm1.PortOpen = False
        End
    End If
End Sub

Private Sub Command6_Click()
    Timer2.Enabled = False
    Timer4.Enabled = False
    counter = 0
    Text5.Text = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Text6.Text = ""
    mnuStop.Enabled = False
    Command8.Enabled = False
    Label2.Caption = " "
    Form1.Text1 = "คนเล่นที่ 1"
    Form1.Text2 = "คนเล่นที่ 2"
    Form1.Label65 = "คนเล่นที่ 1"
    Form1.Label67 = "คนเล่นที่ 2"
    Form1.Label78 = ""
    Form1.Label74 = ""
    Form1.Label71 = "NAME 1"
    Form1.Label72 = "NAME 2"
       Option1.Enabled = True
       Option1.Value = True
       Option2.Enabled = True
       Timer3.Enabled = False
       Option3.Enabled = True
       Option4.Value = True
       Option4.Enabled = True
       Option5.Enabled = False
       Option5.Value = True
       Option6.Enabled = False
       Text1.Enabled = True
       Text1.BackColor = &H80000005
       Label78.Enabled = True
       Label78.BackColor = &H8000000B
       Command1.Enabled = True
       Command6.Enabled = False
       mnuStart.Enabled = True
       mnuNew.Enabled = False
       Option5.Caption = "ROBOT เริ่มเล่นก่อน"
       Option6.Caption = "คนเล่นที่ 1  เริ่มเล่นก่อน"
    Function_new
End Sub

Private Sub Command8_Click()
    If Timer2.Enabled = True Then
       Timer2.Enabled = False
       Command8.Caption = "เ&ริ่มเล่นต่อ"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mnuStop.Caption = "เ&ริ่มเล่นต่อ"
    Else
        Timer2.Enabled = True
        Command8.Caption = "ห&ยุดเล่นชั่วคราว"
         mnuStop.Caption = "ห&ยุดเล่นชั่วคราว"
    End If
End Sub

Private Sub Form_Load()
    Dim START As String
    START = MsgBox("ตอนนี้คุณใช้หน้าจอขนาด 1024*768 pixels หรือมากกว่านี้",
vbExclamation + vbYesNo)
    If START = vbNo Then
        MsgBox ("ขออภัยครับ!!! คุณต้องปรับหน้าจอของคุณให้มีขนาด 1024*768 pixels หรือ
มากกว่านี้นะครับ")
        End
    End If
    abcd = 960
    Text5.Text = ""
    Text6.Text = ""
    mnuStop.Enabled = False
    mnuNew.Enabled = False
    MSComm1.Settings = "9600,n,8,1"
    MSComm1.CommPort = 1
    If Not MSComm1.PortOpen Then
        MSComm1.PortOpen = True
    End If
    Label69.Caption = "ติดต่อกับ PortCom " & MSComm1.CommPort & " ตั้งแต่เวลา " &
 Format(Time(), "hh:mm:ss")
    For I = 0 To 31 Step 1
        str1(I) = str0(I)
    Next I
    str1(0) = 0
    str1(1) = 0
    str1(2) = 0
    str1(3) = 0
    str1(4) = 0
    str1(5) = 0
    str1(6) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    str1(7) = 0
    str1(24) = 0
    str1(25) = 0
    str1(26) = 0
    str1(27) = 0
    str1(28) = 0
    str1(29) = 0
    str1(30) = 0
    str1(31) = 0
End Sub

Private Sub Label1_DragDrop(Index As Integer, Source As Control, X As Single, Y As
Single)
    Dim zz As String
    z2 = Index
    If (Option6.Value = True) And (Option3.Value = True) Then
    If two = 2 Then
       bb = a1 + 40
       If z2 = 1 Then
          a7 = 1560
          a6 = 2280
       ElseIf z2 = 3 Then
          a7 = 1560
          a6 = 4200
       ElseIf z2 = 5 Then
          a7 = 1560
          a6 = 6120
       ElseIf z2 = 7 Then
          a7 = 1560
          a6 = 8040
       ElseIf z2 = 8 Then
          a7 = 2520
          a6 = 1320
       ElseIf z2 = 10 Then
          a7 = 2520
          a6 = 3240
       ElseIf z2 = 12 Then
          a7 = 2520
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a6 = 5160
  ElseIf z2 = 14 Then
    a7 = 2520
    a6 = 7080
  ElseIf z2 = 17 Then
    a7 = 3480
    a6 = 2280
  ElseIf z2 = 19 Then
    a7 = 3480
    a6 = 4200
  ElseIf z2 = 21 Then
    a7 = 3480
    a6 = 6120
  ElseIf z2 = 23 Then
    a7 = 3480
    a6 = 8040
  ElseIf z2 = 24 Then
    a7 = 4440
    a6 = 1320
  ElseIf z2 = 26 Then
    a7 = 4440
    a6 = 3240
  ElseIf z2 = 28 Then
    a7 = 4440
    a6 = 5160
  ElseIf z2 = 30 Then
    a7 = 4440
    a6 = 7080
  ElseIf z2 = 33 Then
    a7 = 5400
    a6 = 2280
  ElseIf z2 = 35 Then
    a7 = 5400
    a6 = 4200
  ElseIf z2 = 37 Then
    a7 = 5400
    a6 = 6120
  ElseIf z2 = 39 Then
    a7 = 5400
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    a6 = 8040
  ElseIf z2 = 40 Then
    a7 = 6360
    a6 = 1320
  ElseIf z2 = 42 Then
    a7 = 6360
    a6 = 3240
  ElseIf z2 = 44 Then
    a7 = 6360
    a6 = 5160
  ElseIf z2 = 46 Then
    a7 = 6360
    a6 = 7080
  ElseIf z2 = 49 Then
    a7 = 7320
    a6 = 2280
  ElseIf z2 = 51 Then
    a7 = 7320
    a6 = 4200
  ElseIf z2 = 53 Then
    a7 = 7320
    a6 = 6120
  ElseIf z2 = 55 Then
    a7 = 7320
    a6 = 8040
  ElseIf z2 = 56 Then
    a7 = 8280
    a6 = 1320
  ElseIf z2 = 58 Then
    a7 = 8280
    a6 = 3240
  ElseIf z2 = 60 Then
    a7 = 8280
    a6 = 5160
  ElseIf z2 = 62 Then
    a7 = 8280
    a6 = 7080
  End If
  a2 = Image4(a1).Left
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        a3 = Image4(a1).Top
        Function_chek
        If z2 >= z1 Then
           z3 = z2 - z1
        if z3 = 7 Or z3 = 14 Or z3 = 21 Or z3 = 28 Or z3 = 35 Or z3 = 42 Or z3 = 49 Or z3
= 0 Or z3 = 9 Or z3 = 18 Or z3 = 27 Or z3 = 36 Or z3 = 45 Or z3 = 54 Then
           If z3 = 7 Then
             If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) Then
                Image4(a1).Move a6, a7
                table(b1, b2) = 0
                table(b1 + 1, b2 - 1) = 4
                chek(b1, b2) = 0
                chek(b1 + 1, b2 - 1) = bb
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 + 1 & b2 - 1
                m = b1 + 1 & b2 - 1
             End If
           ElseIf z3 = 14 Then
             If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) Then
                Image4(a1).Move a6, a7
                table(b1, b2) = 0
                table(b1 + 2, b2 - 2) = 4
                chek(b1, b2) = 0
                chek(b1 + 2, b2 - 2) = bb
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 - 2
                m = b1 + 2 & b2 - 2
             ElseIf ((table(b1 + 1, b2 - 1) = 1) Or (table(b1 + 1, b2 - 1) = 3)) And (table(b1
+ 2, b2 - 2) = 0) Then
                Image4(a1).Move a6, a7
                If table(b1 + 1, b2 - 1) = 1 Then
                   bbb = chek(b1 + 1, b2 - 1) - 10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Image1(bbb).Visible = False
        ElseIf table(b1 + 1, b2 - 1) = 3 Then
            bbb = chek(b1 + 1, b2 - 1) - 30
            Image3(bbb).Visible = False
        End If
        table(b1, b2) = 0
        chek(b1, b2) = 0
        table(b1 + 2, b2 - 2) = 4
        table(b1 + 1, b2 - 1) = 0
        chek(b1 + 1, b2 - 1) = 0
        chek(b1 + 2, b2 - 2) = bb
        Function_chnext
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 - 2
        m = b1 + 2 & b2 - 2
    End If
ElseIf z3 = 21 Then
    If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) And (table(b1 + 3, b2 - 3) = 0) Then
        Image4(a1).Move a6, a7
        table(b1, b2) = 0
        table(b1 + 3, b2 - 3) = 4
        chek(b1, b2) = 0
        chek(b1 + 3, b2 - 3) = bb
        Option5.Value = True
        Timer3.Enabled = True
        ab = 0
        ct = 0
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + 3 & b2 - 3
        m = b1 + 3 & b2 - 3
    ElseIf (table(b1 + 1, b2 - 1) = 0) And ((table(b1 + 2, b2 - 2) = 1) Or (table(b1 +
2, b2 - 2) = 1)) And (table(b1 + 3, b2 - 3) = 0) Then
        Image4(a1).Move a6, a7
        If table(b1 + 2, b2 - 2) = 1 Then
            bbb = chek(b1 + 2, b2 - 2) - 10
            Image1(bbb).Visible = False
        ElseIf table(b1 + 2, b2 - 2) = 3 Then
            bbb = chek(b1 + 2, b2 - 2) - 30
            Image3(bbb).Visible = False
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            table(b1, b2) = 0
            table(b1 + 3, b2 - 3) = 4
            chek(b1, b2) = 0
            table(b1 + 2, b2 - 2) = 0
            chek(b1 + 2, b2 - 2) = 0
            chek(b1 + 3, b2 - 3) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 3 & b2 - 3
            m = b1 + 3 & b2 - 3
          End If
        ElseIf z3 = 28 Then
          If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) And (table(b1 + 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 4, b2 - 4) = 4
            chek(b1, b2) = 0
            chek(b1 + 4, b2 - 4) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 4 & b2 - 4
            m = b1 + 4 & b2 - 4
          ElseIf (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 - 2) = 0) And ((table(b1
+ 3, b2 - 3) = 1) Or (table(b1 + 3, b2 - 3) = 3)) And (table(b1 + 4, b2 - 4) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 3, b2 - 3) = 1 Then
              bbb = chek(b1 + 3, b2 - 3) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 3, b2 - 3) = 3 Then
              bbb = chek(b1 + 3, b2 - 3) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 4, b2 - 4) = 4
            chek(b1, b2) = 0
            table(b1 + 3, b2 - 3) = 0
            chek(b1 + 3, b2 - 3) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chek(b1 + 4, b2 - 4) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 4 & b2 - 4
            m = b1 + 4 & b2 - 4
          End If
        ElseIf z3 = 35 Then
          If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) And (table(b1 + 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) And (table(b1 + 5,
b2 - 5) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 5, b2 - 5) = 4
            chek(b1, b2) = 0
            chek(b1 + 5, b2 - 5) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 5 & b2 - 5
            m = b1 + 5 & b2 - 5
          ElseIf (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 - 2) = 0) And (table(b1
+ 3, b2 - 3) = 0) And ((table(b1 + 4, b2 - 4) = 1) Or (table(b1 + 4, b2 - 4) = 3)) And
(table(b1 + 5, b2 - 5) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 4, b2 - 4) = 1 Then
              bbb = chek(b1 + 4, b2 - 4) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 4, b2 - 4) = 3 Then
              bbb = chek(b1 + 4, b2 - 4) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 5, b2 - 5) = 4
            chek(b1, b2) = 0
            table(b1 + 4, b2 - 4) = 0
            chek(b1 + 4, b2 - 4) = 0
            chek(b1 + 5, b2 - 5) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 5 & b2 - 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            m = b1 + 5 & b2 - 5
          End If
        ElseIf z3 = 42 Then
          If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) And (table(b1 + 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) And (table(b1 + 5,
b2 - 5) = 0) And (table(b1 + 6, b2 - 6) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 6, b2 - 6) = 4
            chek(b1, b2) = 0
            chek(b1 + 6, b2 - 6) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 6 & b2 - 6
            m = b1 + 6 & b2 - 6
          ElseIf (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 - 2) = 0) And (table(b1
+ 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) And ((table(b1 + 5, b2 - 5) = 1) Or (table
(b1 + 5, b2 - 5) = 3)) And (table(b1 + 6, b2 - 6) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 5, b2 - 5) = 1 Then
              bbb = chek(b1 + 5, b2 - 5) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 5, b2 - 5) = 3 Then
              bbb = chek(b1 + 5, b2 - 5) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 6, b2 - 6) = 4
            chek(b1, b2) = 0
            table(b1 + 5, b2 - 5) = 0
            chek(b1 + 5, b2 - 5) = 0
            chek(b1 + 6, b2 - 6) = bb
             Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 6 & b2 - 6
            m = b1 + 6 & b2 - 6
          End If
        ElseIf z3 = 49 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 -
2) = 0) And (table(b1 + 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) And (table(b1 + 5,
b2 - 5) = 0) And (table(b1 + 6, b2 - 6) = 0) And (table(b1 + 7, b2 - 7) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 7, b2 - 7) = 4
            chek(b1, b2) = 0
            chek(b1 + 7, b2 - 7) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 7 & b2 - 7
            m = b1 + 7 & b2 - 7
        ElseIf (table(b1 + 1, b2 - 1) = 0) And (table(b1 + 2, b2 - 2) = 0) And (table(b1
+ 3, b2 - 3) = 0) And (table(b1 + 4, b2 - 4) = 0) And (table(b1 + 5, b2 - 5) = 0) And
((table(b1 + 6, b2 - 6) = 1) Or (table(b1 + 6, b2 - 6) = 3)) And (table(b1 + 7, b2 - 7) = 0)
Then
            Image4(a1).Move a6, a7
            If table(b1 + 6, b2 - 6) = 1 Then
              bbb = chek(b1 + 6, b2 - 6) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 6, b2 - 6) = 3 Then
              bbb = chek(b1 + 6, b2 - 6) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 7, b2 - 7) = 4
            chek(b1, b2) = 0
            table(b1 + 6, b2 - 6) = 0
            chek(b1 + 6, b2 - 6) = 0
            chek(b1 + 7, b2 - 7) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 7 & b2 - 7
            m = b1 + 7 & b2 - 7
        End If
    ElseIf z3 = 9 Then
        If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) Then
            Image4(a1).Move a6, a7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            table(b1, b2) = 0
            table(b1 + 1, b2 + 1) = 4
            chek(b1, b2) = 0
            chek(b1 + 1, b2 + 1) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 1 & b2 + 1
            m = b1 + 1 & b2 + 1
          End If
        ElseIf z3 = 18 Then
          If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 2, b2 + 2) = 4
            chek(b1, b2) = 0
            chek(b1 + 2, b2 + 2) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 + 2
            m = b1 + 2 & b2 + 2
          ElseIf ((table(b1 + 1, b2 + 1) = 1) Or (table(b1 + 1, b2 + 1) = 3)) And (table(b1
+ 2, b2 + 2) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 1, b2 + 1) = 1 Then
              bbb = chek(b1 + 1, b2 + 1) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 1, b2 + 1) = 3 Then
              bbb = chek(b1 + 1, b2 + 1) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 2, b2 + 2) = 4
            chek(b1, b2) = 0
            table(b1 + 1, b2 + 1) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chek(b1 + 1, b2 + 1) = 0
            chek(b1 + 2, b2 + 2) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 + 2
            m = b1 + 2 & b2 + 2
          End If
        ElseIf z3 = 27 Then
          If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) And (table(b1 + 3, b2 + 3) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 3, b2 + 3) = 4
            chek(b1, b2) = 0
            chek(b1 + 3, b2 + 3) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 3 & b2 + 3
            m = b1 + 3 & b2 + 3
          ElseIf (table(b1 + 1, b2 + 1) = 0) And ((table(b1 + 2, b2 + 2) = 1) Or (table(b1
+ 2, b2 + 2) = 3)) And (table(b1 + 3, b2 + 3) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 2, b2 + 2) = 1 Then
              bbb = chek(b1 + 2, b2 + 2) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 2, b2 + 2) = 3 Then
              bbb = chek(b1 + 2, b2 + 2) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 3, b2 + 3) = 4
            chek(b1, b2) = 0
            table(b1 + 2, b2 + 2) = 0
            chek(b1 + 2, b2 + 2) = 0
            chek(b1 + 3, b2 + 3) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 3 & b2 + 3
            m = b1 + 3 & b2 + 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
        ElseIf z3 = 36 Then
            If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) And (table(b1 + 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) Then
                Image4(a1).Move a6, a7
                table(b1, b2) = 0
                table(b1 + 4, b2 + 4) = 4
                chek(b1, b2) = 0
                chek(b1 + 4, b2 + 4) = bb
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 + 4 & b2 + 4
                m = b1 + 4 & b2 + 4
            ElseIf (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2 + 2) = 0) And ((table
(b1 + 3, b2 + 3) = 1) Or (table(b1 + 3, b2 + 3) = 3)) And (table(b1 + 4, b2 + 4) = 0)
Then
                Image4(a1).Move a6, a7
                If table(b1 + 3, b2 + 3) = 1 Then
                  bbb = chek(b1 + 3, b2 + 3) - 10
                  Image1(bbb).Visible = False
                ElseIf table(b1 + 3, b2 + 3) = 3 Then
                  bbb = chek(b1 + 3, b2 + 3) - 30
                  Image3(bbb).Visible = False
                End If
                table(b1, b2) = 0
                table(b1 + 4, b2 + 4) = 4
                chek(b1, b2) = 0
                table(b1 + 3, b2 + 3) = 0
                chek(b1 + 3, b2 + 3) = 0
                chek(b1 + 4, b2 + 4) = bb
                Function_chnext
                Label2.Caption = b1 & b2 & "-" & ">" & b1 + 4 & b2 + 4
                m = b1 + 4 & b2 + 4
            End If
        ElseIf z3 = 45 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) And (table(b1 + 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) And (table(b1
+ 5, b2 + 5) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 5, b2 + 5) = 4
            chek(b1, b2) = 0
            chek(b1 + 5, b2 + 5) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 5 & b2 + 5
            m = b1 + 5 & b2 + 5
        ElseIf (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2 + 2) = 0) And (table(b1
+ 3, b2 + 3) = 0) And ((table(b1 + 4, b2 + 4) = 1) Or (table(b1 + 4, b2 + 4) = 3)) And
(table(b1 + 5, b2 + 5) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 4, b2 + 4) = 1 Then
              bbb = chek(b1 + 4, b2 + 4) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 4, b2 + 4) = 3 Then
              bbb = chek(b1 + 4, b2 + 4) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 5, b2 + 5) = 4
            chek(b1, b2) = 0
            table(b1 + 4, b2 + 4) = 0
            chek(b1 + 4, b2 + 4) = 0
            chek(b1 + 5, b2 + 5) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 5 & b2 + 5
            m = b1 + 5 & b2 + 5
        End If
    ElseIf z3 = 54 Then
        If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) And (table(b1 + 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) And (table(b1
+ 5, b2 + 5) = 0) And (table(b1 + 6, b2 + 6) = 0) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 6, b2 + 6) = 4
            chek(b1, b2) = 0
            chek(b1 + 6, b2 + 6) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 6 & b2 + 6
            m = b1 + 6 & b2 + 6
        ElseIf (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2 + 2) = 0) And (table(b1
+ 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) And ((table(b1 + 5, b2 + 5) = 1) Or
(table(b1 + 5, b2 + 5) = 3)) And (table(b1 + 6, b2 + 6) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 5, b2 + 5) = 1 Then
              bbb = chek(b1 + 5, b2 + 5) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 5, b2 + 5) = 3 Then
              bbb = chek(b1 + 5, b2 + 5) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 6, b2 + 6) = 4
            chek(b1, b2) = 0
            table(b1 + 5, b2 + 5) = 0
            chek(b1 + 5, b2 + 5) = 0
            chek(b1 + 6, b2 + 6) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 6 & b2 + 6
            m = b1 + 6 & b2 + 6
        End If
    ElseIf z3 = 63 Then
        If (ch = 0 And ch0 = 0) And (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2
+ 2) = 0) And (table(b1 + 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) And (table(b1
+ 5, b2 + 5) = 0) And (table(b1 + 6, b2 + 6) = 0) And (table(b1 + 7, b2 + 7) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 + 7, b2 + 7) = 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chek(b1, b2) = 0
            chek(b1 + 7, b2 + 7) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 7 & b2 + 7
            m = b1 + 7 & b2 + 7
        ElseIf (table(b1 + 1, b2 + 1) = 0) And (table(b1 + 2, b2 + 2) = 0) And (table(b1
+ 3, b2 + 3) = 0) And (table(b1 + 4, b2 + 4) = 0) And (table(b1 + 5, b2 + 5) = 0) And
((table(b1 + 6, b2 + 6) = 1) Or (table(b1 + 6, b2 + 6) = 3)) And (table(b1 + 7, b2 + 7) =
0) Then
            Image4(a1).Move a6, a7
            If table(b1 + 6, b2 + 6) = 1 Then
              bbb = chek(b1 + 6, b2 + 6) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 + 6, b2 + 6) = 3 Then
              bbb = chek(b1 + 6, b2 + 6) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 + 7, b2 + 7) = 4
            chek(b1, b2) = 0
            table(b1 + 6, b2 + 6) = 0
            chek(b1 + 6, b2 + 6) = 0
            chek(b1 + 7, b2 + 7) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 7 & b2 + 7
            m = b1 + 7 & b2 + 7
          End If
        End If
      End If
      ElseIf z2 <= z1 Then          'เดินขึ้นบน
        z3 = z1 - z2
      If z3 = 7 Or z3 = 14 Or z3 = 21 Or z3 = 28 Or z3 = 35 Or z3 = 42 Or z3 = 49 Or z3
= 0 Or z3 = 9 Or z3 = 18 Or z3 = 27 Or z3 = 36 Or z3 = 45 Or z3 = 54 Then
        If z3 = 7 Then
          If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) Then
            Image4(a1).Move a6, a7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            table(b1, b2) = 0
            table(b1 - 1, b2 + 1) = 4
            chek(b1, b2) = 0
            chek(b1 - 1, b2 + 1) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 1 & b2 + 1
            m = b1 - 1 & b2 + 1
          End If
        ElseIf z3 = 14 Then
          If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 2, b2 + 2) = 4
            chek(b1, b2) = 0
            chek(b1 - 2, b2 + 2) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
          ElseIf ((table(b1 - 1, b2 + 1) = 1) Or (table(b1 - 1, b2 + 1) = 3)) And (table(b1 -
2, b2 + 2) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 1, b2 + 1) = 1 Then
              bbb = chek(b1 - 1, b2 + 1) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 1, b2 + 1) = 3 Then
              bbb = chek(b1 - 1, b2 + 1) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 1, b2 + 1) = 0
            table(b1 - 2, b2 + 2) = 4
            chek(b1, b2) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chek(b1 - 1, b2 + 1) = 0
            chek(b1 - 2, b2 + 2) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
          End If
        ElseIf z3 = 21 Then
          If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) And (table(b1 - 3, b2 + 3) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 3, b2 + 3) = 4
            chek(b1, b2) = 0
            chek(b1 - 3, b2 + 3) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 3 & b2 + 3
            m = b1 - 3 & b2 + 3
          ElseIf (table(b1 - 1, b2 + 1) = 0) And ((table(b1 - 2, b2 + 2) = 1) Or (table(b1 -
2, b2 + 2) = 1)) And (table(b1 - 3, b2 + 3) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 2, b2 + 2) = 1 Then
              bbb = chek(b1 - 2, b2 + 2) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 2, b2 + 2) = 3 Then
              bbb = chek(b1 - 2, b2 + 2) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 3, b2 + 3) = 4
            chek(b1, b2) = 0
            table(b1 - 2, b2 + 2) = 0
            chek(b1 - 2, b2 + 2) = 0
            chek(b1 - 3, b2 + 3) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 3 & b2 + 3
            m = b1 - 3 & b2 + 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
    ElseIf z3 = 28 Then
        If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) And (table(b1 - 3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 4, b2 + 4) = 4
            chek(b1, b2) = 0
            chek(b1 - 4, b2 + 4) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2 + 4
            m = b1 - 4 & b2 + 4
        ElseIf (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 + 2) = 0) And ((table(b1
- 3, b2 + 3) = 1) Or (table(b1 - 3, b2 + 3) = 3)) And (table(b1 - 4, b2 + 4) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 3, b2 + 3) = 1 Then
              bbb = chek(b1 - 3, b2 + 3) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 3, b2 + 3) = 3 Then
              bbb = chek(b1 - 3, b2 + 3) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 4, b2 + 4) = 4
            chek(b1, b2) = 0
            table(b1 - 3, b2 + 3) = 0
            chek(b1 - 3, b2 + 3) = 0
            chek(b1 - 4, b2 + 4) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2 + 4
            m = b1 - 4 & b2 + 4
        End If
    ElseIf z3 = 35 Then
        If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) And (table(b1 - 3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) And (table(b1 - 5,
b2 + 5) = 0) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 5, b2 + 5) = 4
            chek(b1, b2) = 0
            chek(b1 - 5, b2 + 5) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 5 & b2 + 5
            m = b1 - 5 & b2 + 5
        ElseIf (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 + 2) = 0) And (table(b1 -
3, b2 + 3) = 0) And ((table(b1 - 4, b2 + 4) = 1) Or (table(b1 - 4, b2 + 4) = 3)) And (table
(b1 - 5, b2 + 5) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 4, b2 + 4) = 1 Then
              bbb = chek(b1 - 4, b2 + 4) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 4, b2 + 4) = 3 Then
              bbb = chek(b1 - 4, b2 + 4) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 5, b2 + 5) = 4
            chek(b1, b2) = 0
            table(b1 - 4, b2 + 4) = 0
            chek(b1 - 4, b2 + 4) = 0
            chek(b1 - 5, b2 + 5) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 5 & b2 + 5
            m = b1 - 5 & b2 + 5
        End If
    ElseIf z3 = 42 Then
        If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) And (table(b1 - 3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) And (table(b1 - 5,
b2 + 5) = 0) And (table(b1 - 6, b2 + 6) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 6, b2 + 6) = 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chek(b1, b2) = 0
            chek(b1 - 6, b2 + 6) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 6 & b2 + 6
            m = b1 - 6 & b2 + 6
        ElseIf (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 + 2) = 0) And (table(b1 -
3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) And ((table(b1 - 0, b2 + 5) = 1) Or (table
(b1 - 5, b2 + 5) = 3)) And (table(b1 - 6, b2 + 6) = 0) Then
            Image4(a1).Move a3, a2
            If table(b1 - 5, b2 + 5) = 1 Then
              bbb = chek(b1 - 5, b2 + 5) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 5, b2 + 5) = 3 Then
              bbb = chek(b1 - 5, b2 + 5) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 6, b2 + 6) = 4
            chek(b1, b2) = 0
            table(b1 - 5, b2 + 5) = 0
            chek(b1 - 5, b2 + 5) = 0
            chek(b1 - 6, b2 + 6) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 6 & b2 + 6
            m = b1 - 6 & b2 + 6
        End If
    ElseIf z3 = 49 Then
        If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 +
2) = 0) And (table(b1 - 3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) And (table(b1 - 5,
b2 + 5) = 0) And (table(b1 - 6, b2 + 6) = 0) And (table(b1 - 7, b2 + 7) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 7, b2 + 7) = 4
            chek(b1, b2) = 0
            chek(b1 - 7, b2 + 7) = bb
            Option5.Value = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 7 & b2 + 7
            m = b1 - 7 & b2 + 7
        ElseIf (table(b1 - 1, b2 + 1) = 0) And (table(b1 - 2, b2 + 2) = 0) And (table(b1 -
3, b2 + 3) = 0) And (table(b1 - 4, b2 + 4) = 0) And (table(b1 - 5, b2 + 5) = 0) And ((table
(b1 - 6, b2 + 6) = 1) Or (table(b1 - 6, b2 + 6) = 3)) And (table(b1 - 7, b2 + 7) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 6, b2 + 6) = 1 Then
               bbb = chek(b1 - 6, b2 + 6) - 10
               Image1(bbb).Visible = False
            ElseIf table(b1 - 6, b2 + 6) = 3 Then
               bbb = chek(b1 - 6, b2 + 6) - 30
               Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 7, b2 + 7) = 4
            chek(b1, b2) = 0
            table(b1 - 6, b2 + 6) = 0
            chek(b1 - 6, b2 + 6) = 0
            chek(b1 - 7, b2 + 7) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 7 & b2 + 7
            m = b1 - 7 & b2 + 7
         End If
      ElseIf z3 = 9 Then
         If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 1, b2 - 1) = 4
            chek(b1, b2) = 0
            chek(b1 - 1, b2 - 1) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 1 & b2 - 1
            m = b1 - 1 & b2 - 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
        ElseIf z3 = 18 Then
            If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) Then
                Image4(a1).Move a6, a7
                table(b1, b2) = 0
                table(b1 - 2, b2 - 2) = 4
                chek(b1, b2) = 0
                chek(b1 - 2, b2 - 2) = bb
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                m = b1 - 2 & b2 - 2
            ElseIf ((table(b1 - 1, b2 - 1) = 1) Or (table(b1 - 1, b2 - 1) = 3)) And (table(b1 -
2, b2 - 2) = 0) Then
                Image4(a1).Move a6, a7
                If table(b1 - 1, b2 - 1) = 1 Then
                  bbb = chek(b1 - 1, b2 - 1) - 10
                  Image1(bbb).Visible = False
                ElseIf table(b1 - 1, b2 - 1) = 3 Then
                  bbb = chek(b1 - 1, b2 - 1) - 30
                  Image3(bbb).Visible = False
                End If
                table(b1, b2) = 0
                table(b1 - 2, b2 - 2) = 4
                chek(b1, b2) = 0
                table(b1 - 1, b2 - 1) = 0
                chek(b1 - 1, b2 - 1) = 0
                chek(b1 - 2, b2 - 2) = bb
                Function_chnext
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                m = b1 - 2 & b2 - 2
            End If
        ElseIf z3 = 27 Then
            If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) And (table(b1 - 3, b2 - 3) = 0) Then
                Image4(a1).Move a6, a7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            table(b1, b2) = 0
            table(b1 - 3, b2 - 3) = 4
            chek(b1, b2) = 0
            chek(b1 - 3, b2 - 3) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 3 & b2 - 3
            m = b1 - 3 & b2 - 3
        ElseIf (table(b1 - 1, b2 - 1) = 0) And ((table(b1 - 2, b2 - 2) = 1) Or (table(b1 - 2,
b2 - 2) = 3)) And (table(b1 - 3, b2 - 3) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 2, b2 - 2) = 1 Then
              bbb = chek(b1 - 2, b2 - 2) - 10
              Image1(bbb).Visible = False
            ElseIf table(b1 - 2, b2 - 2) = 3 Then
              bbb = chek(b1 - 2, b2 - 2) - 30
              Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 3, b2 - 3) = 4
            chek(b1, b2) = 0
            table(b1 - 2, b2 - 2) = 0
            chek(b1 - 2, b2 - 2) = 0
            chek(b1 - 3, b2 - 3) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 3 & b2 - 3
            m = b1 - 3 & b2 - 3
        End If
    ElseIf z3 = 36 Then
        If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) And (table(b1 - 3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 4, b2 - 4) = 4
            chek(b1, b2) = 0
            chek(b1 - 4, b2 - 4) = bb
            Option5.Value = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2 - 4
            m = b1 - 4 & b2 - 4
        ElseIf (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2) = 0) And ((table(b1 -
3, b2 - 3) = 1) Or (table(b1 - 3, b2 - 3) = 3)) And (table(b1 - 4, b2 - 4) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 3, b2 - 3) = 1 Then
               bbb = chek(b1 - 3, b2 - 3) - 10
               Image1(bbb).Visible = False
            ElseIf table(b1 - 3, b2 - 3) = 3 Then
               bbb = chek(b1 - 3, b2 - 3) - 30
               Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 4, b2 - 4) = 4
            chek(b1, b2) = 0
            table(b1 - 3, b2 - 3) = 0
            chek(b1 - 3, b2 - 3) = 0
            chek(b1 - 4, b2 - 4) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2 - 4
            m = b1 - 4 & b2 - 4
         End If
      ElseIf z3 = 45 Then
         If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) And (table(b1 - 3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) And (table(b1 - 5, b2 -
5) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 5, b2 - 5) = 4
            chek(b1, b2) = 0
            chek(b1 - 5, b2 - 5) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 5 & b2 - 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            m = b1 - 5 & b2 - 5
        ElseIf (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2) = 0) And (table(b1 -
3, b2 - 3) = 0) And ((table(b1 - 4, b2 - 4) = 1) Or (table(b1 - 4, b2 - 4) = 3)) And (table(b1
- 5, b2 - 5) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 4, b2 - 4) = 1 Then
               bbb = chek(b1 - 4, b2 - 4) - 10
               Image1(bbb).Visible = False
            ElseIf table(b1 - 4, b2 - 4) = 3 Then
               bbb = chek(b1 - 4, b2 - 4) - 30
               Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 5, b2 - 5) = 4
            chek(b1, b2) = 0
            table(b1 - 4, b2 - 4) = 0
            chek(b1 - 4, b2 - 4) = 0
            chek(b1 - 5, b2 - 5) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 5 & b2 - 5
            m = b1 - 5 & b2 - 5
         End If
      ElseIf z3 = 54 Then
         If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) And (table(b1 - 3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) And (table(b1 - 5, b2 -
5) = 0) And (table(b1 - 6, b2 - 6) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 6, b2 - 6) = 4
            chek(b1, b2) = 0
            chek(b1 - 6, b2 - 6) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 6 & b2 - 6
            m = b1 - 6 & b2 - 6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ElseIf (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2) = 0) And (table(b1 -
3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) And ((table(b1 - 5, b2 - 5) = 1) Or (table(b1
- 5, b2 - 5) = 3)) And (table(b1 - 6, b2 - 6) = 0) Then
            Image4(a1).Move a6, a7
            If table(b1 - 5, b2 - 5) = 1 Then
               bbb = chek(b1 - 5, b2 - 5) - 10
               Image1(bbb).Visible = False
            ElseIf table(b1 - 5, b2 - 5) = 3 Then
               bbb = chek(b1 - 5, b2 - 5) - 30
               Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 6, b2 - 6) = 4
            chek(b1, b2) = 0
            table(b1 - 5, b2 - 5) = 0
            chek(b1 - 5, b2 - 5) = 0
            chek(b1 - 6, b2 - 6) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 6 & b2 - 6
            m = b1 - 6 & b2 - 6
         End If
      ElseIf z3 = 63 Then
         If (ch = 0 And ch0 = 0) And (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2)
= 0) And (table(b1 - 3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) And (table(b1 - 5, b2 -
5) = 0) And (table(b1 - 6, b2 - 6) = 0) And (table(b1 - 7, b2 - 7) = 0) Then
            Image4(a1).Move a6, a7
            table(b1, b2) = 0
            table(b1 - 7, b2 - 7) = 4
            chek(b1, b2) = 0
            chek(b1 - 7, b2 - 7) = bb
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 7 & b2 - 7
            m = b1 - 7 & b2 - 7
         ElseIf (table(b1 - 1, b2 - 1) = 0) And (table(b1 - 2, b2 - 2) = 0) And (table(b1 -
3, b2 - 3) = 0) And (table(b1 - 4, b2 - 4) = 0) And (table(b1 - 5, b2 - 5) = 0) And ((table
(b1 - 6, b2 - 6) = 1) Or (table(b1 - 6, b2 - 6) = 3)) And (table(b1 - 7, b2 - 7) = 0) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Image4(a1).Move a6, a7
            If table(b1 - 6, b2 - 6) = 1 Then
               bbb = chek(b1 - 6, b2 - 6) - 10
               Image1(bbb).Visible = False
            ElseIf table(b1 - 6, b2 - 6) = 3 Then
               bbb = chek(b1 - 6, b2 - 6) - 30
               Image3(bbb).Visible = False
            End If
            table(b1, b2) = 0
            table(b1 - 7, b2 - 7) = 4
            chek(b1, b2) = 0
            table(b1 - 6, b2 - 6) = 0
            chek(b1 - 6, b2 - 6) = 0
            chek(b1 - 7, b2 - 7) = bb
            Function_chnext
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 7 & b2 - 7
            m = b1 - 7 & b2 - 7
         End If
      End If
   End If
   End If
   two = 0
ElseIf two = 1 Then
bb = a1 + 20
a2 = Image2(a1).Left
a3 = Image2(a1).Top
Function_chek
   If z1 > z2 Then
      z3 = z1 - z2
      Select Case z3
      Case 7
         If ch <> 1 Then                    'เดินธรรมดา
            If (table(b1 - 1, b2 + 1) = 0) Then
               Image2(a1).Move a2 + abcd, a3 - abcd
               table(b1, b2) = 0
               table(b1 - 1, b2 + 1) = 2
               chek(b1, b2) = 0
               chek(b1 - 1, b2 + 1) = bb
               Option5.Value = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 1 & b2 + 1
            m = b1 - 1 & b2 + 1
          End If
        End If
        Case 9
          If ch <> 1 Then                    'เดินธรรมดา
            If (table(b1 - 1, b2 - 1) = 0) Then
              Image2(a1).Move a2 - abcd, a3 - abcd
              table(b1, b2) = 0
              table(b1 - 1, b2 - 1) = 2
              chek(b1, b2) = 0
              chek(b1 - 1, b2 - 1) = bb
              Option5.Value = True
              Timer3.Enabled = True
              ab = 0
              ct = 0
              Label2.Caption = b1 & b2 & "-" & ">" & b1 - 1 & b2 - 1
              m = b1 - 1 & b2 - 1
            End If
          End If
          Case 14
            If b1 > 2 And b2 <= 5 Then
              If table(b1 - 1, b2 + 1) <> 2 And table(b1 - 1, b2 + 1) <> 0 And table
(b1 - 1, b2 + 1) <> 4 Then
                Function_topright                    'กินขวา
                Function_toprighthot
                If ((b2 + 2) = 5) And ((b1 - 2) = 2) Then
                  If ((table(b1 - 3, b2 + 3) <> 1) And (table(b1 - 3, b2 + 3) <> 3)) Or
(table(b1 - 4, b2 + 4) <> 0) Then      'เช็คขวาต่อ
                    If ((table(b1 - 3, b2 + 1) <> 1) And (table(b1 - 3, b2 + 1) <> 3))
Or (table(b1 - 4, b2) <> 0) Then       'เช็คซ้ายต่อ
                      Option5.Value = True
                      Timer3.Enabled = True
                      ab = 0
                      ct = 0
                      Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                m = b1 - 4 & b2
            End If
        End If
    ElseIf (b2 + 2) >= 6 Or (b1 - 2) = 1 Then
        If (b1 - 2 = 1) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
        ElseIf (table(b1 - 3, b2 + 1) <> 1) And (table(b1 - 3, b2 + 1) <> 3)
Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
        ElseIf b1 > 3 And (table(b1 - 4, b2) <> 0) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
        End If
    ElseIf (b1 > 3) Then
        If (((table(b1 - 3, b2 + 3) <> 1) And (table(b1 - 3, b2 + 3) <> 3))
Or (table(b1 - 4, b2 + 4) <> 0)) And ((table(b1 - 3, b2 + 1) <> 1) Or (table(b1 - 3, b2 +
1) <> 3) Or (table(b1 - 4, b2) <> 0)) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
            m = b1 - 2 & b2 + 2
        End If
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
          Else
            Function_topright
            Function_toprighthot
            If (b1 = 2) Then
              If ((table(b1 - 1, b2 + 1) <> 1) And (table(b1 - 1, b2 + 1) <> 3)) Or
table(b1 - 2, b2 + 2) <> 0 Then
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
                m = b1 - 2 & b2 + 2
              ElseIf (b2 >= 2) Then
                If ((table(b1 - 1, b2 - 1) <> 1) And (table(b1 - 1, b2 - 1) <> 3)) Or
table(b1 - 2, b2 - 2) <> 0 Then
                  Option5.Value = True
                  Timer3.Enabled = True
                  ab = 0
                  ct = 0
                  Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                  m = b1 - 2 & b2 - 2
                End If
              End If
            End If
          End If

        Case 18
          If b1 > 2 And b2 >= 2 Then
            If table(b1 - 1, b2 - 1) <> 2 And table(b1 - 1, b2 - 1) <> 0 And table(b1
- 1, b2 - 1) <> 4 Then
              Function_topleft
              Function_toplefthot
              If ((b2 - 2) = 2) And ((b1 - 2) = 2) Then
                If ((table(b1 - 3, b2 - 3) <> 1) And (table(b1 - 3, b2 - 3) <> 3)) Or
(table(b1 - 4, b2 - 4) <> 0) Then
                  If ((table(b1 - 3, b2 - 1) <> 1) And (table(b1 - 3, b2 - 1) <> 3))
Or (table(b1 - 4, b2) <> 0) Then
                    Option5.Value = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2
                    m = b1 - 4 & b2
                End If
            End If
        ElseIf (b2 - 2) <= 1 Or (b1 - 2) = 1 Then
            If (b1 - 2 = 1) Then
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                m = b1 - 2 & b2 - 2
            ElseIf (table(b1 - 3, b2 - 1) <> 1) And (table(b1 - 3, b2 - 1) <> 3)
Then
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
            ElseIf b1 > 3 And (table(b1 - 4, b2) <> 0) Then
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                m = b1 - 2 & b2 - 2
            End If
        ElseIf (b1 > 3) Then
            If (((table(b1 - 3, b2 - 3) <> 1) And (table(b1 - 3, b2 - 3) <> 3)) Or
(table(b1 - 4, b2 - 4) <> 0)) And ((table(b1 - 3, b2 - 1) <> 1) Or (table(b1 - 4, b2) <> 0))
Then
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
                Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m = b1 - 2 & b2 - 2
                    End If
                  End If
                End If
              Else
                Function_topleft
                Function_toplefthot
                If (b1 = 2) Then
                  If (table(b1 - 1, b2 - 1) <> 1 And table(b1 - 1, b2 - 1) <> 3) Or table
(b1 - 2, b2 - 2) <> 0 Then
                    Option5.Value = True
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                    m = b1 - 2 & b2 - 2
                  ElseIf (b2 <= 5) Then
                    If (table(b1 - 1, b2 + 1) <> 1 And table(b1 - 1, b2 + 1) <> 3) Or
table(b1 - 2, b2 + 2) <> 0 Then
                      Option5.Value = True
                      Timer3.Enabled = True
                      ab = 0
                      ct = 0
                      Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
                      m = b1 - 2 & b2 + 2
                    End If
                  End If
                End If
              End If
            End Select
            a2 = Image2(a1).Left
            a3 = Image2(a1).Top
            Function_chek
            If (b1 = 0) Then
              Image2(a1).Visible = False
              Image4(a1).Move a2, a3
              Image4(a1).Visible = True
              table(b1, b2) = 4
              chek(b1, b2) = bb + 20
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Option5.Value = True
                Timer3.Enabled = True
                ab = 0
                ct = 0
              End If
            End If
          End If
          two = 0
          End If
    End Sub

Private Sub Image2_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
    a1 = Index
    a2 = Image2(a1).Left
    a3 = Image2(a1).Top
    Function_chek
    Function_z1
    z1 = z1 + b2
    two = 1
    Image2(a1).Drag vbBeginDrag
End Sub

Private Sub Image2_MouseUp(Index As Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
    Image2(a1).Drag vbEndDrag
End Sub

Private Sub Image4_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
    a1 = Index
    a2 = Image4(a1).Left
    a3 = Image4(a1).Top
    Function_chek
    Function_z1
    z1 = z1 + b2
    two = 2
    Image4(a1).Drag vbBeginDrag
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Image4_MouseUp(Index As Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
   Image4(a1).Drag vbEndDrag
End Sub

Private Sub mnuAbout_Click()
   Form2.Show
End Sub

Private Sub mnuExit_Click()
   Command5_Click
End Sub

Private Sub mnuNew_Click()
   Command6_Click
End Sub

Private Sub mnuPlay_Click()
   Form3.Show
End Sub

Private Sub mnuPra_Click()
   Form4.Show
End Sub

Private Sub mnuStart_Click()
      Command1_Click
End Sub

Private Sub mnuStop_Click()
   Command8_Click
End Sub

Private Sub MSComm1_OnComm()
   Select Case MSComm1.CommEvent
      Case comEventBreak
       Text5.Text = "Error" & comEventBreak
      Case comEventCDTO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Text5.Text = "Error" & comEventCDTO
      Case comEventCTSTO
        Text5.Text = "Error" & comEventCTSTO
      Case comEventDSRTO
        Text5.Text = "Error" & comEventDSRTO
      Case comEventFrame
        Text5.Text = "Error" & comEventFrame
      Case comEventOverrun
        Text5.Text = "Error" & comEventOverrun
      Case comEventRxOver
        Text5.Text = "Error" & comEventRxOver
      Case comEventRxParity
        Text5.Text = "Error" & comEventRxParity
      Case comEventTxFull
        Text5.Text = "Error" & comEventTxFull
      Case comEvCD
        Text5.Text = "Error" & comEvCD
      Case comEvCTS
        Text5.Text = "Error" & comEvCTS
      Case comEvDSR
        Text5.Text = "Error" & comEvDSR
      Case comEvRing
        Text5.Text = "Error" & comEvRing
      Case comEvReceive
        Text5.Text = "Error" & comEvReceive
      Case comEvSend
        Text5.Text = "Error" & comEvSend
      Case comEvEOF
        Text5.Text = "Error" & comEvEOF
    End Select
End Sub

Private Sub Option1_Click()
    Text2.BackColor = &H8000000B
    Text2.Enabled = False
    Form1.Text1 = "คนเล่นที่ 1"
    Form1.Text2 = "คนเล่นที่ 2"
    Option5.Value = True
    Option5.Caption = "ROBOT เริ่มเล่นก่อน"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Option6.Caption = "คนเล่นที่ 1  เริ่มเล่นก่อน"
End Sub

Private Sub Option2_Click()
    Text2.BackColor = &H80000005
    Text2.Enabled = True
    Form1.Text1 = "คนเล่นที่ 1"
    Form1.Text2 = "คนเล่นที่ 2"
    Option5.Value = True
    Option5.Caption = "คนเล่นที่ 1 เริ่มเล่นก่อน"
    Option6.Caption = "คนเล่นที่ 2 เริ่มเล่นก่อน"
End Sub

Private Sub Text1_Change()
    If Text1.Text <> "" Then
        Command1.Enabled = True
    Else
        Command1.Enabled = False
    End If
    If Option1.Value = True Then
    Option6.Caption = Text1.Text & "  เริ่มเล่นก่อน"
    ElseIf Option2.Value = True Then
    Option5.Caption = Text1.Text & "  เริ่มเล่นก่อน"
    Option6.Caption = Text2.Text & "  เริ่มเล่นก่อน"
    End If
End Sub

    Private Sub Text2_Change()
        If Text2.Text <> "" Then
            Command1.Enabled = True
        Else
            Command1.Enabled = False
        End If
        Option6.Caption = Text2.Text & "  เริ่มเล่นก่อน"
    End Sub

Private Sub Timer1_Timer()
    Label68.Caption = "วันนี้ " & Format(Date, "วันdddd ที่ d MMMM yyyy ") & " ขณะนี้เวลา "
& Format(Time(), "hh:mm:ss")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Option2.Enabled = False
End Sub

Private Sub Timer2_Timer()
Static Count
Count = Count + 1
Dim str As String
  If Timer3.Enabled = True Then
      Timer3.Enabled = False
      Count = 0
   End If
   If Option5.Value = True Then
      Label78.Caption = Text1 & " เป็นฝ่ายกำลังเล่น ใช้เวลาไปแล้ว " & Count & " วินาที"
      str = Text1
   ElseIf Option6.Value = True Then
      Label78.Caption = Text2 & " เป็นฝ่ายกำลังเล่น ใช้เวลาไปแล้ว " & Count & " วินาที"
      str = Text2
   End If
   If Count >= 2000 Then
      MsgBox str & " : คุณเป็นผู้แพ้แล้วครับ เนื่องจากคุณใช้เวลานานเกินไป !", vbOKOnly +
vbExclamation
      Count = 0
      Command6_Click
   End If
End Sub

Private Sub Function_chek()
   If a2 = 1320 Then
      b2 = 0
    End If
    If a2 = 2280 Then
      b2 = 1
    End If
    If a2 = 3240 Then
      b2 = 2
    End If
    If a2 = 4200 Then
      b2 = 3
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If a2 = 5160 Then
      b2 = 4
    End If
    If a2 = 6120 Then
      b2 = 5
    End If
    If a2 = 7080 Then
      b2 = 6
    End If
    If a2 = 8040 Then
      b2 = 7
    End If
    If a3 = 1560 Then
      b1 = 0
    End If
    If a3 = 2520 Then
      b1 = 1
    End If
    If a3 = 3480 Then
      b1 = 2
    End If
    If a3 = 4440 Then
      b1 = 3
    End If
    If a3 = 5400 Then
      b1 = 4
    End If
    If a3 = 6360 Then
      b1 = 5
    End If
    If a3 = 7320 Then
      b1 = 6
    End If
    If a3 = 8280 Then
      b1 = 7
   End If
End Sub

Private Sub Function_z1()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (b1 >= 0) And (b1 <= 7) Then
        If b1 = 0 Then
            z1 = 0
        ElseIf b1 = 1 Then
            z1 = 8
        ElseIf b1 = 2 Then
            z1 = 16
        ElseIf b1 = 3 Then
            z1 = 24
        ElseIf b1 = 4 Then
            z1 = 32
        ElseIf b1 = 5 Then
            z1 = 40
        ElseIf b1 = 6 Then
            z1 = 48
        ElseIf b1 = 7 Then
            z1 = 56
        End If
    End If
End Sub

Private Sub Function_new()
        Image1(0).Move 2280, 1560
        Image1(0).Visible = False
        Image1(1).Move 4200, 1560
        Image1(1).Visible = False
        Image1(2).Move 6120, 1560
        Image1(2).Visible = False
        Image1(3).Move 8040, 1560
        Image1(3).Visible = False
        Image1(4).Move 1320, 2520
        Image1(4).Visible = False
        Image1(5).Move 3240, 2520
        Image1(5).Visible = False
        Image1(6).Move 5160, 2520
        Image1(6).Visible = False
        Image1(7).Move 7080, 2520
        Image1(7).Visible = False
        Image2(0).Move 1320, 8280
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Image2(0).Visible = False
    Image2(1).Move 3240, 8280
    Image2(1).Visible = False
    Image2(2).Move 5160, 8280
    Image2(2).Visible = False
    Image2(3).Move 7080, 8280
    Image2(3).Visible = False
    Image2(4).Move 2280, 7320
    Image2(4).Visible = False
    Image2(5).Move 4200, 7320
    Image2(5).Visible = False
    Image2(6).Move 6120, 7320
    Image2(6).Visible = False
    Image2(7).Move 8040, 7320
    Image2(7).Visible = False
    Image3(0).Visible = False
    Image3(1).Visible = False
    Image3(2).Visible = False
    Image3(3).Visible = False
    Image3(4).Visible = False
    Image3(5).Visible = False
    Image3(6).Visible = False
    Image3(7).Visible = False
    Image4(0).Visible = False
    Image4(1).Visible = False
    Image4(2).Visible = False
    Image4(3).Visible = False
    Image4(4).Visible = False
    Image4(5).Visible = False
    Image4(6).Visible = False
    Image4(7).Visible = False
End Sub

Private Sub Function_left()
    bb = a1 + 10
    a2 = Image1(a1).Left
    a3 = Image1(a1).Top
    Function_chek
    If Image1(a1).Visible = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (a2 >= 2280) And (a3 <= 7320) Then
      If (table(b1 + 1, b2 - 1) <> 1) And (table(b1 + 1, b2 - 1) <> 2) And (table(b1 + 1, b2
- 1) <> 3) And (table(b1 + 1, b2 - 1) <> 4) Then
        Image1(a1).Move a2 - abcd, a3 + abcd
        table(b1, b2) = 0
        table(b1 + 1, b2 - 1) = 1
        chek(b1, b2) = 0
        chek(b1 + 1, b2 - 1) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + 1
        rote(3) = b2 - 1
        rote(4) = 9
        az = b1 & b2 & b1 + 1 & b2 - 1
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + 1 & b2 - 1
      Else: Function_cheking2
      End If
    End If
    a2 = Image1(a1).Left
    a3 = Image1(a1).Top
    Function_chek
    If b1 = 7 Then
      Image1(a1).Visible = False
      Image3(a1).Move a2, a3
      Image3(a1).Visible = True
      table(b1, b2) = 3
      chek(b1, b2) = bb + 20
    End If
  End If
End Sub

Private Sub Function_right()
    bb = a1 + 10
    a2 = Image1(a1).Left
    a3 = Image1(a1).Top
    Function_chek
    If Image1(a1).Visible = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (a2 <= 7080) And (a3 <= 7320) Then
     If (table(b1 + 1, b2 + 1) <> 1) And (table(b1 + 1, b2 + 1) <> 2) And (table(b1 + 1,
b2 + 1) <> 3) And (table(b1 + 1, b2 + 1) <> 4) Then
        Image1(a1).Move a2 + abcd, a3 + abcd
        table(b1, b2) = 0
        table(b1 + 1, b2 + 1) = 1
        chek(b1, b2) = 0
        chek(b1 + 1, b2 + 1) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + 1
        rote(3) = b2 + 1
        rote(4) = 9
        az = b1 & b2 & b1 + 1 & b2 + 1
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + 1 & b2 + 1
      Else: Function_cheking2
      End If
    End If
     a2 = Image1(a1).Left
     a3 = Image1(a1).Top
     Function_chek
     If b1 = 7 Then
      Image1(a1).Visible = False
      Image3(a1).Move a2, a3
      Image3(a1).Visible = True
      table(b1, b2) = 3
      chek(b1, b2) = bb + 20
     End If
    End If
End Sub

Private Sub Function_cheking1()
    bb = a1 + 10
    a2 = Image1(a1).Left
    a3 = Image1(a1).Top
    Function_chek
    If (b1 = 7) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Image1(a1).Visible = False
        Image3(a1).Move a2, a3
        Image3(a1).Visible = True
        table(b1, b2) = 3
        chek(b1, b2) = bb + 20
    End If
    If (a2 >= 2280) And (a3 <= 7320) Then
        If ((chek(b1 + 1, b2 - 1) >= 20) And (chek(b1 + 1, b2 - 1) <= 27)) Or ((chek(b1 + 1,
b2 - 1) >= 40) And (chek(b1 + 1, b2 - 1) <= 47)) Then
            If (b2 > 1) And (b1 < 6) Then
                If (table(b1 + 2, b2 - 2) = 0) Then
                    z = 1
                    Label1(((b1) * 8) + (b2)).BackColor = &H0
                        Image1(a1).Move a2 - (2 * abcd), a3 + (2 * abcd)
                        table(b1, b2) = 0
                        table(b1 + 2, b2 - 2) = 1
                        chek(b1, b2) = 0
                        chek(b1 + 2, b2 - 2) = bb
                    If ((chek(b1 + 1, b2 - 1) >= 20) And (chek(b1 + 1, b2 - 1) <= 27)) Then
                        bbb = chek(b1 + 1, b2 - 1) - 20
                        Image2(bbb).Visible = False
                        Image2(bbb).Enabled = False
                    ElseIf ((chek(b1 + 1, b2 - 1) >= 40) And (chek(b1 + 1, b2 - 1) <= 47)) Then
                        bbb = chek(b1 + 1, b2 - 1) - 40
                        Image4(bbb).Visible = False
                        Image4(bbb).Enabled = False
                    End If
                    table(b1 + 1, b2 - 1) = 0
                    chek(b1 + 1, b2 - 1) = 0
                    chk = chk + 1
                    ct = ct + 1
                    If chk = 8 Then
                        MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
                        'Option6.Value = True
                        Command6_Click
                    End If
                    walk = 1
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 - 2
                    Function_cheking2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Function_cheking1
                rr = 1
              End If
            End If
          End If
        End If
      End Sub

      Private Sub Function_cheking2()
        bb = a1 + 10
        a2 = Image1(a1).Left
        a3 = Image1(a1).Top
        Function_chek
        If (b1 = 7) Then
          Image1(a1).Visible = False
          Image3(a1).Move a2, a3
          Image3(a1).Visible = True
          table(b1, b2) = 3
          chek(b1, b2) = bb + 20
        End If
        If (a2 <= 7080) And (a3 <= 7320) Then
          If ((chek(b1 + 1, b2 + 1) >= 20) And (chek(b1 + 1, b2 + 1) <= 27)) Or ((chek(b1 +
      1, b2 + 1) >= 40) And (chek(b1 + 1, b2 + 1) <= 47)) Then
            If (b2 < 6) And (b1 < 6) Then
              If (table(b1 + 2, b2 + 2) = 0) Then
                z = 1
                Label1(((b1) * 8) + (b2)).BackColor = &H0
                Image1(a1).Move a2 + (2 * abcd), a3 + (2 * abcd)
                table(b1, b2) = 0
                table(b1 + 2, b2 + 2) = 1
                chek(b1, b2) = 0
                chek(b1 + 2, b2 + 2) = bb
                If ((chek(b1 + 1, b2 + 1) >= 20) And (chek(b1 + 1, b2 + 1) <= 27)) Then
                  bbb = chek(b1 + 1, b2 + 1) - 20
                  Image2(bbb).Visible = False
                  Image2(bbb).Enabled = False
                ElseIf ((chek(b1 + 1, b2 + 1) >= 40) And (chek(b1 + 1, b2 + 1) <= 47))
       Then
                  bbb = chek(b1 + 1, b2 + 1) - 40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Image4(bbb).Visible = False
                Image4(bbb).Enabled = False
            End If
            table(b1 + 1, b2 + 1) = 0
            chek(b1 + 1, b2 + 1) = 0
            chk = chk + 1
            ct = ct + 1
            If chk = 8 Then
                MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
                'Option6.Value = True
                Command6_Click
            End If
            walk = 1
            Label2.Caption = b1 & b2 & "-" & ">" & b1 + 2 & b2 + 2
            Function_cheking1
            Function_cheking2
            rr = 1
        End If
    End If
End If
End If
End Sub

Public Sub Function_toplefthot()
        If ((chek(b1 - 1, b2 - 1) >= 30) And (chek(b1 - 1, b2 - 1) <= 37)) Then
            If (b2 > 1) And (b1 > 1) Then
                If (table(b1 - 2, b2 - 2) = 0) Then
                    Image2(a1).Move a2 - (2 * abcd), a3 - (2 * abcd)
                    table(b1, b2) = 0
                    table(b1 - 2, b2 - 2) = 2
                    chek(b1, b2) = 0
                    chek(b1 - 2, b2 - 2) = bb
                    bbb = chek(b1 - 1, b2 - 1) - 30
                    Image3(bbb).Visible = False
                    Image3(bbb).Enabled = False
                    table(b1 - 1, b2 - 1) = 0
                    chek(b1 - 1, b2 - 1) = 0
                    che = che + 1
                    If che = 8 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MsgBox ("คุณ " & Text2.Text & " เป็นผู้ชนะครับ")
                    'Option6.Value = True
                    Command6_Click
                End If
                ch = 0
            End If
        End If
    End If
End Sub

Public Sub Function_topleft()
    If ((chek(b1 - 1, b2 - 1) >= 10) And (chek(b1 - 1, b2 - 1) <= 17)) Then
        If (b2 > 1) And (b1 > 1) Then
            If (table(b1 - 2, b2 - 2) = 0) Then
                Image2(a1).Move a2 - (2 * abcd), a3 - (2 * abcd)
                table(b1, b2) = 0
                table(b1 - 2, b2 - 2) = 2
                chek(b1, b2) = 0
                chek(b1 - 2, b2 - 2) = bb
                bbb = chek(b1 - 1, b2 - 1) - 10
                Image1(bbb).Visible = False
                Image1(bbb).Enabled = False
                table(b1 - 1, b2 - 1) = 0
                chek(b1 - 1, b2 - 1) = 0
                che = che + 1
                If che = 8 Then
                    MsgBox ("คุณ " & Text2.Text & " เป็นผู้ชนะครับ")
                    'Option6.Value = True
                    Command6_Click
                End If
                ch = 0
            End If
        End If
    End If
End Sub

Public Sub Function_toprighthot()
    If (chek(b1 - 1, b2 + 1) >= 30) And (chek(b1 - 1, b2 + 1) <= 37) Then
        If (b2 < 6) And (b1 > 1) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                If (table(b1 - 2, b2 + 2) = 0) Then
                    Image2(a1).Move a2 + (2 * abcd), a3 - (2 * abcd)
                    table(b1, b2) = 0
                    table(b1 - 2, b2 + 2) = 2
                    chek(b1, b2) = 0
                    chek(b1 - 2, b2 + 2) = bb
                    bbb = chek(b1 - 1, b2 + 1) - 30
                    Image3(bbb).Visible = False
                    Image3(bbb).Enabled = False
                    table(b1 - 1, b2 + 1) = 0
                    chek(b1 - 1, b2 + 1) = 0
                    che = che + 1
                    If che = 8 Then
                        MsgBox ("คุณ " & Text2.Text & " เป็นผู้ชนะครับ")
                        'Option6.Value = True
                        Command6_Click
                    End If
                    ch = 0
                End If
            End If
        End If
End Sub

Public Sub Function_topright()
        If (chek(b1 - 1, b2 + 1) >= 10) And (chek(b1 - 1, b2 + 1) <= 17) Then
            If (b2 < 6) And (b1 > 1) Then
                If (table(b1 - 2, b2 + 2) = 0) Then
                    Image2(a1).Move a2 + (2 * abcd), a3 - (2 * abcd)
                    table(b1, b2) = 0
                    table(b1 - 2, b2 + 2) = 2
                    chek(b1, b2) = 0
                    chek(b1 - 2, b2 + 2) = bb
                    bbb = chek(b1 - 1, b2 + 1) - 10
                    Image1(bbb).Visible = False
                    Image1(bbb).Enabled = False
                    table(b1 - 1, b2 + 1) = 0
                    chek(b1 - 1, b2 + 1) = 0
                    che = che + 1
                    If che = 8 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MsgBox ("คุณ " & Text2.Text & " เป็นผู้ชนะครับ")
                    'Option6.Value = True
                    Command6_Click
                End If
                ch = 0
            End If
        End If
    End If
End Sub

Private Sub Function_b3b4()
        If ton = 0 Then
            b3 = 0
            b4 = 1
        ElseIf ton = 1 Then
            b3 = 0
            b4 = 3
        ElseIf ton = 2 Then
            b3 = 0
            b4 = 5
        ElseIf ton = 3 Then
            b3 = 0
            b4 = 7
        ElseIf ton = 4 Then
            b3 = 1
            b4 = 0
        ElseIf ton = 5 Then
            b3 = 1
            b4 = 2
        ElseIf ton = 6 Then
            b3 = 1
            b4 = 4
        ElseIf ton = 7 Then
            b3 = 1
            b4 = 6
        ElseIf ton = 8 Then
            b3 = 2
            b4 = 1
        ElseIf ton = 9 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    b3 = 2
    b4 = 3
  ElseIf ton = 10 Then
    b3 = 2
    b4 = 5
  ElseIf ton = 11 Then
    b3 = 2
    b4 = 7
  ElseIf ton = 12 Then
    b3 = 3
    b4 = 0
  ElseIf ton = 13 Then
    b3 = 3
    b4 = 2
  ElseIf ton = 14 Then
    b3 = 3
    b4 = 4
  ElseIf ton = 15 Then
    b3 = 3
    b4 = 6
  ElseIf ton = 16 Then
    b3 = 4
    b4 = 1
  ElseIf ton = 17 Then
    b3 = 4
    b4 = 3
  ElseIf ton = 18 Then
    b3 = 4
    b4 = 5
  ElseIf ton = 19 Then
    b3 = 4
    b4 = 7
  ElseIf ton = 20 Then
    b3 = 5
    b4 = 0
  ElseIf ton = 21 Then
    b3 = 5
    b4 = 2
  ElseIf ton = 22 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            b3 = 5
            b4 = 4
        ElseIf ton = 23 Then
            b3 = 5
            b4 = 6
        ElseIf ton = 24 Then
            b3 = 6
            b4 = 1
        ElseIf ton = 25 Then
            b3 = 6
            b4 = 3
        ElseIf ton = 26 Then
            b3 = 6
            b4 = 5
        ElseIf ton = 27 Then
            b3 = 6
            b4 = 7
        ElseIf ton = 28 Then
            b3 = 7
            b4 = 0
        ElseIf ton = 29 Then
            b3 = 7
            b4 = 2
        ElseIf ton = 30 Then
            b3 = 7
            b4 = 4
        ElseIf ton = 31 Then
            b3 = 7
            b4 = 6
        End If
End Sub

Public Sub Function_b1b2()
    m = b1 & b2
    If m = 1 Then
        m = 0
    ElseIf m = 3 Then
        m = 1
    ElseIf m = 5 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    m = 2
  Elself m = 7 Then
    m = 3
  Elself m = 10 Then
    m = 4
  Elself m = 12 Then
    m = 5
  Elself m = 14 Then
    m = 6
  Elself m = 16 Then
    m = 7
  Elself m = 21 Then
    m = 8
  Elself m = 23 Then
    m = 9
  Elself m = 25 Then
    m = 10
  Elself m = 27 Then
    m = 11
  Elself m = 30 Then
    m = 12
  Elself m = 32 Then
    m = 13
  Elself m = 34 Then
    m = 14
  Elself m = 36 Then
    m = 15
  Elself m = 41 Then
    m = 16
  Elself m = 43 Then
    m = 17
  Elself m = 45 Then
    m = 18
  Elself m = 47 Then
    m = 19
  Elself m = 50 Then
    m = 20
  Elself m = 52 Then
    m = 21
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ElseIf m = 54 Then
      m = 22
    ElseIf m = 56 Then
      m = 23
    ElseIf m = 61 Then
      m = 24
    ElseIf m = 63 Then
      m = 25
    ElseIf m = 65 Then
      m = 26
    ElseIf m = 67 Then
      m = 27
    ElseIf m = 70 Then
      m = 28
    ElseIf m = 72 Then
      m = 29
    ElseIf m = 74 Then
      m = 30
    ElseIf m = 76 Then
      m = 31
    End If
End Sub

Private Sub Timer4_Timer()
    Dim s As Integer
    Dim zz As Integer
    Dim buf As Variant
    Dim k As Integer
    Function_checker
    counter = counter + 1
    If ab = 0 Then
      If Option5.Value = True Then
        abc = 0
      Else:
        abc = 1
      End If
        ab = 1
    End If
    If Option4.Value = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    For s = 0 To 7 Step 1
       Image2(s).Enabled = False
       Image4(s).Enabled = False
    Next s
'*****************************************************************
    For k = 0 To 6000 Step 1
       DoEvents
    Next k
    For s = 0 To 31 Step 1
       If Label4(s).Visible = True Then
          y1 = y1 + 1
       End If
    Next s
       y2 = y1
    If y2 < y0 Then
       If (y2 + 1) < y0 Then
          MsgBox ("มีการผิดพลาดเกิดขึ้น เนื่องจากมีตัวหมากฮอสหายไป")
          Command6_Click
       Else
          If (y2 + 1) = y0 Then
             For s = 0 To 31 Step 1
                If Label4(s).Visible = True Then
                   ton1(s) = 1
                ElseIf Label4(s).Visible = False Then
                   ton1(s) = 0
                End If
             Next s
             For s = 0 To 31 Step 1
                If ton0(s) <> ton1(s) Then
                   ton = s
                   ton0(s) = ton1(s)
                End If
             Next s
             Function_b3b4
             For s = 0 To 7 Step 1
                a2 = Image2(s).Left
                a3 = Image2(s).Top
                Function_chek
                If b1 = b3 And b2 = b4 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                a1 = s
                b5 = b1
                b6 = b2
                a4 = a2
                a5 = a3
            End If
        Next s
        y0 = y2
    End If
    y1 = 0
  End If
ElseIf y2 > y0 And Image2(a1).Visible = True Then
  If (y2 - 1) = y0 Then
    For s = 0 To 31 Step 1
      If Label4(s).Visible = True Then
        ton1(s) = 1
      ElseIf Label4(s).Visible = False Then
        ton1(s) = 0
      End If
    Next s
    For s = 0 To 31 Step 1
      If ton0(s) <> ton1(s) Then
        ton = s
        ton0(s) = ton1(s)
      End If
    Next s
    Function_b3b4
    If b3 < b5 And b4 > b6 Then
      bb = a1 + 20
      If (b5 - b3) = 1 And (b4 - b6) = 1 Then
        Image2(a1).Move a4 + abcd, a5 - abcd
        table(b5, b6) = 0
        table(b5 - 1, b6 + 1) = 2
        chek(b5, b6) = 0
        chek(b5 - 1, b6 + 1) = bb
        Option5.Value = True
        Timer3.Enabled = True
        ab = 0
        ct = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label2.Caption = b5 & b6 & "-" & ">" & b5 - 1 & b6 + 1
  ElseIf (b5 - b3) = 2 And (b4 - b6) = 2 Then
    b1 = b5
    b2 = b6
    a2 = a4
    a3 = a5
    If b1 > 2 And b2 <= 5 Then
      If table(b1 - 1, b2 + 1) <> 2 And table(b1 - 1, b2 + 1) <> 0 Then
        Function_topright
        If ((b2 + 2) = 5) And ((b1 - 2) = 2) Then
          If (table(b1 - 3, b2 + 3) <> 1 Or table(b1 - 4, b2 + 4) <> 0) Then
            If (table(b1 - 3, b2 + 1) <> 1 Or table(b1 - 4, b2) <> 0) Then
              Option5.Value = True
              Timer3.Enabled = True
              ab = 0
              ct = 0
              Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2
            End If
          End If
        ElseIf (b2 + 2) >= 6 Or (b1 - 2) = 1 Then
          If (b1 - 2 = 1) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
          ElseIf (table(b1 - 3, b2 + 1) <> 1) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
          ElseIf b1 > 3 And (table(b1 - 4, b2) <> 0) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
          End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ElseIf (b1 > 3) Then
                  If ((table(b1 - 3, b2 + 3) <> 1) Or (table(b1 - 4, b2 + 4) <> 0)) And
((table(b1 - 3, b2 + 1) <> 1) Or (table(b1 - 4, b2) <> 0)) Then
                  Option5.Value = True
                  Timer3.Enabled = True
                  ab = 0
                  ct = 0
                  Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
                  End If
                End If
              End If
            Else
              Function_topright
              If (b1 = 2) Then
                If table(b1 - 1, b2 + 1) <> 1 And table(b1 - 2, b2 + 2) <> 0 Then
                  Option5.Value = True
                  Timer3.Enabled = True
                  ab = 0
                  ct = 0
                  Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
                ElseIf (b2 >= 2) Then
                  If table(b1 - 1, b2 - 1) <> 1 And table(b1 - 2, b2 - 2) <> 0 Then
                    Option5.Value = True
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                  End If
                End If
              End If
            End If

          ElseIf b3 < b5 And b4 < b6 Then
            bb = a1 + 20
            If (b5 - b3) = 1 And (b6 - b4) = 1 Then
              Image2(a1).Move a4 - abcd, a5 - abcd
              table(b5, b6) = 0
              table(b5 - 1, b6 - 1) = 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    chek(b5, b6) = 0
    chek(b5 - 1, b6 - 1) = bb
    Option5.Value = True
    Timer3.Enabled = True
    ab = 0
    ct = 0
    Label2.Caption = b5 & b6 & "-" & ">" & b5 - 1 & b6 - 1
  ElseIf (b5 - b3) = 2 And (b6 - b4) = 2 Then
    b1 = b5
    b2 = b6
    a2 = a4
    a3 = a5
    If b1 > 2 And b2 >= 2 Then
      If table(b1 - 1, b2 - 1) <> 2 And table(b1 - 1, b2 - 1) <> 0 Then
        Function_topleft
        If ((b2 - 2) = 2) And ((b1 - 2) = 2) Then
          If (table(b1 - 3, b2 - 3) <> 1 Or table(b1 - 4, b2 - 4) <> 0) Then
            If (table(b1 - 3, b2 - 1) <> 1 Or table(b1 - 4, b2) <> 0) Then
              Option5.Value = True
              Timer3.Enabled = True
              ab = 0
              ct = 0
              Label2.Caption = b1 & b2 & "-" & ">" & b1 - 4 & b2
            End If
          End If
        ElseIf (b2 - 2) <= 1 Or (b1 - 2) = 1 Then
          If (b1 - 2 = 1) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
          ElseIf (table(b1 - 3, b2 - 1) <> 1) Then
            Option5.Value = True
            Timer3.Enabled = True
            ab = 0
            ct = 0
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
          ElseIf b1 > 3 And (table(b1 - 4, b2) <> 0) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Option5.Value = True
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                  End If
                ElseIf (b1 > 3) Then
                  If ((table(b1 - 3, b2 - 3) <> 1) Or (table(b1 - 4, b2 - 4) <> 0)) And
((table(b1 - 3, b2 - 1) <> 1) Or (table(b1 - 4, b2) <> 0)) Then
                    Option5.Value = True
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                    End If
                  End If
                End If
              Else
                Function_topleft
                If (b1 = 2) Then
                  If table(b1 - 1, b2 - 1) <> 1 And table(b1 - 2, b2 - 2) <> 0 Then
                    Option5.Value = True
                    Timer3.Enabled = True
                    ab = 0
                    ct = 0
                    Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 - 2
                  ElseIf (b2 <= 5) Then
                    If table(b1 - 1, b2 + 1) <> 1 And table(b1 - 2, b2 + 2) <> 0 Then
                      Option5.Value = True
                      Timer3.Enabled = True
                      ab = 0
                      ct = 0
                      Label2.Caption = b1 & b2 & "-" & ">" & b1 - 2 & b2 + 2
                    End If
                  End If
                End If
              End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
            y0 = y2
          End If
          y1 = 0
        Else: y1 = 0
        End If
    '*****************************************************************************
    Else
    End If
End Sub

Private Sub Timer5_Timer()
    Dim InString As String
    Dim MyStr As String
    Dim k As Integer
    MSComm1.Output = "9"
    For k = 0 To 9000 Step 1
        DoEvents
    Next k
    MSComm1.OutBufferCount = 0
    MSComm1.InputMode = comInputModeText
    MSComm1.InputLen = 33
        If MSComm1.InBufferCount > 0 Then
            InString = MSComm1.Input
            MSComm1.InBufferCount = 0
        End If
    For I = 0 To 31 Step 1
        MyStr = Mid(InString, I + 1, 1)
        If MyStr = "0" Then
            Label4(I).Visible = True
        Else: Label4(I).Visible = False
        End If
    str0(I) = MyStr
    Next I
End Sub

Private Sub Function_send()
    Dim Buffer As Variant
    Dim i As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim j As Integer
    If abc = 0 Then
      Text6.Text = "ส่งตำแหน่ง : " & rote(0) & rote(1) & rote(2) & rote(3)
      For i = 0 To 4 Step 1
        MSComm1.Output = rote(i)
        For j = 0 To 2000 Step 1
          DoEvents
        Next j
      Next i
      MSComm1.OutBufferCount = 0
    End If
End Sub


Private Sub Function_ch()
    bb = a1 + 10
    Function_cheking2
    If ct = 0 Then
      Function_cheking1
    End If
End Sub

Private Sub Function_checker()
    Dim q As Integer
    Dim qq As Integer
    Dim stri As String
    If abc = 0 Then
      a0 = a0 + 1
      Option5.Value = True
'-------------------------------------------------------------------------
'หมากฮอส Fig
keep = ""
Text3.Text = ""
Text8.Text = ""
For q = 0 To 7 Step 1
    For qq = 0 To 7 Step 1
      If table(q, qq) = 0 Or table(q, qq) = 1 Or table(q, qq) = 2 Or table(q, qq) = 3 Or
table(q, qq) = 4 Then
        keep = Text3.Text & table(q, qq)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Text3.Text = keep
        End If
    Next qq
Next q
    For q = 0 To 7 Step 1        'กิน
     If Image1(q).Visible = True Then
        a1 = q
        If rr = 0 Then
            Function_ch
        End If
     End If
    Next q
    Randomize
    a0 = (2 * Rnd) + 1
    Randomize
    a1 = (7 * Rnd) + 1
    a1 = a1 - 1
    Function_a0
    Text8.Text = "สุ่ม"
    rr = 0
End If
'-------------------------------------------------------------------
   End If
   Function_kin1
   Function_kin2
   Function_kin3
End Sub


Public Sub Function_a0()
   If a1 <= 10 Then
      Select Case a0
         Case 0
            If Image1(a1).Visible = True Then
               z = 0
               If walk <> 0 Then
                  If Option4.Value = True Then
                     Function_send
                     Function_b1b2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Do
              DoEvents
            Loop Until Label4(m).Visible = True
          End If
          ab = 0
          Option6.Value = True
          Timer3.Enabled = True
          abc = 1
          walk = 0
        End If
        a0 = 0
      Else: a0 = a0 - 0
      End If
    Case 1
      If Image1(a1).Visible = True Then
        'Function_ch
        If z <> 1 Then
          Function_left
        End If
        z = 0
        If walk <> 0 Then
          If Option4.Value = True Then
            Function_send
            Function_b1b2
            Do
              DoEvents
            Loop Until Label4(m).Visible = True
          End If
          ab = 0
          Option6.Value = True
          Timer3.Enabled = True
          abc = 1
          walk = 0
        End If
        a0 = 0
      Else: a0 = a0 - 1
      End If
    Case 2
      If Image1(a1).Visible = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If z <> 1 Then
            Function_right
        End If
        z = 0
        If walk <> 0 Then
            If Option4.Value = True Then
                Function_send
                Function_b1b2
                Do
                    DoEvents
                Loop Until Label4(m).Visible = True
            End If
            ab = 0
            Option6.Value = True
            Timer3.Enabled = True
            abc = 1
            walk = 0
        End If
        a0 = 0
    Else: a0 = a0 - 2
    End If
Case 3
    If Image3(a1).Visible = True Then
        If z <> 1 Then
            Function_lefthot
        End If
        z = 0
        If walk <> 0 Then
            If Option4.Value = True Then
                Function_send
                Function_b1b2
                Do
                    DoEvents
                Loop Until Label4(m).Visible = True
            End If
            ab = 0
            Option6.Value = True
            Timer3.Enabled = True
            abc = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            walk = 0
          End If
          a0 = 0
        Else: a0 = a0 - 3
        End If
    Case 4
        If Image3(a1).Visible = True Then
          If z <> 1 Then
            Function_righthot
          End If
          z = 0
          If walk <> 0 Then
            If Option4.Value = True Then
              Function_send
              Function_b1b2
              Do
                DoEvents
              Loop Until Label4(m).Visible = True
            End If
            ab = 0
            Option6.Value = True
            Timer3.Enabled = True
            abc = 1
            walk = 0
          End If
          a0 = 0
        Else: a0 = a0 - 4
        End If
    Case 5
        If Image3(a1).Visible = True Then
          If z <> 1 Then
            Function_lefthot1
          End If
          z = 0
          If walk <> 0 Then
            If Option4.Value = True Then
              Function_send
              Function_b1b2
              Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DoEvents
            Loop Until Label4(m).Visible = True
          End If
          ab = 0
          Option6.Value = True
          Timer3.Enabled = True
          abc = 1
          walk = 0
        End If
        a0 = 0
      Else: a0 = a0 - 5
      End If
  Case 6
      If Image3(a1).Visible = True Then
        If z <> 1 Then
          Function_righthot1
        End If
        z = 0
        If walk <> 0 Then
          If Option4.Value = True Then
            Function_send
            Function_b1b2
            Do
              DoEvents
            Loop Until Label4(m).Visible = True
          End If
          ab = 0
          Option6.Value = True
          Timer3.Enabled = True
          abc = 1
          walk = 0
        End If
        a0 = 0
      Else: a0 = a0 - 6
      End If
  Case 7
      If Image3(a1).Visible = True Then
        z = 0
        If walk <> 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                If Option4.Value = True Then
                    Function_send
                    Function_b1b2
                    Do
                        DoEvents
                    Loop Until Label4(m).Visible = True
                End If
                ab = 0
                Option6.Value = True
                Timer3.Enabled = True
                abc = 1
                walk = 0
            End If
            a0 = 0
        Else: a0 = a0 - 7
        End If
    End Select
    a9 = 0
  ElseIf a1 > 10 Then
    a1 = a1 - 10
  End If
End Sub

Public Sub Function_kin1()
    Dim s As Integer
    ch = 0
    If Option6.Value = True Then
      For s = 0 To 7 Step 1
        If Image2(s).Visible = True Then
        Image2(s).Enabled = False
        a2 = Image2(s).Left
        a3 = Image2(s).Top
        Function_chek
        If (b2 < 6) And (b1 > 1) Then
          If (table(b1 - 1, b2 + 1) = 1) Or (table(b1 - 1, b2 + 1) = 3) Then
            If (table(b1 - 2, b2 + 2) = 0) Then
                ch = 1
                Image2(s).Enabled = True
            End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
          End If
          If (b2 > 1) And (b1 > 1) Then
            If (table(b1 - 1, b2 - 1) = 1) Or (table(b1 - 1, b2 - 1) = 3) Then
              If (table(b1 - 2, b2 - 2) = 0) Then
                  ch = 1
                  Image2(s).Enabled = True
              End If
            End If
          End If
          End If
        Next s
      End If
End Sub

Public Sub Function_kin2()
      Dim s As Integer
      Dim ss As Integer
      Dim b7 As Integer
      Dim b8 As Integer
      b7 = b1
      b8 = b2
      ch0 = 0
      If Option6.Value = True Then
        For s = 0 To 7 Step 1
          If Image4(s).Visible = True Then
            Image4(s).Enabled = False
            a2 = Image4(s).Left
            a3 = Image4(s).Top
            Function_chek
            Do While (b1 >= 2 And b1 <= 8 And b2 >= 2 And b2 <= 8)
              If (table(b1 - 1, b2 - 1) = 1) Or (table(b1 - 1, b2 - 1) = 2) Or (table(b1 - 1,
b2 - 1) = 3) Or (table(b1 - 1, b2 - 1) = 4) Then
                If ((table(b1 - 1, b2 - 1) = 1) Or (table(b1 - 1, b2 - 1) = 3)) And (table(b1
- 2, b2 - 2) = 0) Then
                  ch0 = 1
                  Image4(s).Enabled = True
                End If
                Exit Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
            b1 = b1 - 1
            b2 = b2 - 1
          Loop
          a2 = Image4(s).Left
          a3 = Image4(s).Top
          Function_chek
          Do While (b1 >= 2 And b1 <= 8 And b2 >= 0 And b2 <= 5)
            If (table(b1 - 1, b2 + 1) = 1) Or (table(b1 - 1, b2 + 1) = 2) Or (table(b1 - 1,
b2 + 1) = 3) Or (table(b1 - 1, b2 + 1) = 4) Then
              If ((table(b1 - 1, b2 + 1) = 1) Or (table(b1 - 1, b2 + 1) = 3)) And (table
(b1 - 2, b2 + 2) = 0) Then
                ch0 = 1
                Image4(s).Enabled = True
              End If
              Exit Do
            End If
            b1 = b1 - 1
            b2 = b2 + 1
          Loop
          a2 = Image4(s).Left
          a3 = Image4(s).Top
          Function_chek
          Do While (b1 >= 0 And b1 <= 5 And b2 >= 2 And b2 <= 8)
            If (table(b1 + 1, b2 - 1) = 1) Or (table(b1 + 1, b2 - 1) = 2) Or (table(b1 +
1, b2 - 1) = 3) Or (table(b1 + 1, b2 - 1) = 4) Then
              If ((table(b1 + 1, b2 - 1) = 1) Or (table(b1 + 1, b2 - 1) = 3)) And (table
(b1 + 2, b2 - 2) = 0) Then
                ch0 = 1
                Image4(s).Enabled = True
              End If
              Exit Do
            End If
            b1 = b1 + 1
            b2 = b2 - 1
          Loop
          a2 = Image4(s).Left
          a3 = Image4(s).Top
          Function_chek
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Do While (b1 >= 0 And b1 <= 5 And b2 >= 0 And b2 <= 5)
              If (table(b1 + 1, b2 + 1) = 1) Or (table(b1 + 1, b2 + 1) = 2) Or (table(b1 +
1, b2 + 1) = 3) Or (table(b1 + 1, b2 + 1) = 4) Then
                If ((table(b1 + 1, b2 + 1) = 1) Or (table(b1 + 1, b2 + 1) = 3)) And (table
(b1 + 2, b2 + 2) = 0) Then
                  ch0 = 1
                  Image4(s).Enabled = True
                End If
                Exit Do
              End If
              b1 = b1 + 1
              b2 = b2 + 1
            Loop
          End If
        Next s
      End If
      b1 = b7
      b2 = b8
End Sub

Public Sub Function_chnext()
  ch = 0
  ch0 = 0
  Function_kin2
  If ch0 = 0 Then
    Option5.Value = True
    Timer3.Enabled = True
    ab = 0
    ct = 0
  End If
  che = che + 1
  If che = 8 Then
    MsgBox ("คุณ " & Text2.Text & " เป็นผู้ชนะครับ")
    'Option6.Value = True
    Command6_Click
  End If
End Sub

Public Sub Function_kin3()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim s As Integer
    If ch = 0 And ch0 = 0 Then
       For s = 0 To 7 Step 1
          If Image2(s).Visible = True Then
             Image2(s).Enabled = True
          ElseIf Image4(s).Visible = True Then
             Image4(s).Enabled = True
          End If
       Next s
    End If
End Sub

Public Sub Function_kinlefthot1()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
    If (table(b1 + (a9 - 1), (b2 - (a9 - 1))) = 2) Or (table(b1 + (a9 - 1), (b2 - (a9 - 1))) = 4)
Then
          Image3(a1).Move a2 - (abcd * a9), a3 + (abcd * a9)
          table(b1, b2) = 0
          table(b1 + a9, b2 - a9) = 3
          table(b1 + (a9 - 1), b2 - (a9 - 1)) = 0
          chek(b1, b2) = 0
          chek(b1 + a9, b2 - a9) = bb
          bbb = chek(b1 + (a9 - 1), b2 - (a9 - 1))
          If bbb >= 40 And bbb <= 47 Then
             bbb = bbb - 40
             Image4(bbb).Enabled = False
             Image4(bbb).Visible = False
          ElseIf bbb >= 20 And bbb <= 27 Then
             bbb = bbb - 20
             Image2(bbb).Enabled = False
             Image2(bbb).Visible = False
          End If
          chek(b1 + (a9 - 1), b2 - (a9 - 1)) = 0
          chk = chk + 1
          ct = ct + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If chk = 8 Then
           MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
           'Option6.Value = True
           Command6_Click
        End If
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + a9
        rote(3) = b2 - a9
        rote(4) = 9
        az = b1 & b2 & b1 + a9 & b2 - a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + a9 & b2 - a9
    End If
    End If
End Sub

Public Sub Function_lefthot1()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
        Image3(a1).Move a2 - (abcd * a9), a3 + (abcd * a9)
        table(b1, b2) = 0
        table(b1 + a9, b2 - a9) = 3
        chek(b1, b2) = 0
        chek(b1 + a9, b2 - a9) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + a9
        rote(3) = b2 - a9
        rote(4) = 9
        az = b1 & b2 & b1 + a9 & b2 - a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + a9 & b2 - a9
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub


Public Sub Function_kinlefthot()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
     If (table(b1 - (a9 - 1), (b2 - (a9 - 1))) = 2) Or (table(b1 - (a9 - 1), (b2 - (a9 - 1))) = 4)
Then
        Image3(a1).Move a2 - (abcd * a9), a3 - (abcd * a9)
        table(b1, b2) = 0
        table(b1 - a9, b2 - a9) = 3
        table(b1 - (a9 - 1), b2 - (a9 - 1)) = 0
        chek(b1, b2) = 0
        chek(b1 - a9, b2 - a9) = bb
        bbb = chek(b1 - (a9 - 1), b2 - (a9 - 1))
        If bbb >= 40 And bbb <= 47 Then
           bbb = bbb - 40
           Image4(bbb).Enabled = False
           Image4(bbb).Visible = False
        ElseIf bbb >= 20 And bbb <= 27 Then
           bbb = bbb - 20
           Image2(bbb).Enabled = False
           Image2(bbb).Visible = False
        End If
        chek(b1 - (a9 - 1), b2 - (a9 - 1)) = 0
        chk = chk + 1
        ct = ct + 1
        If chk = 8 Then
           MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
           'Option6.Value = True
           Command6_Click
        End If
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            rote(2) = b1 - a9
            rote(3) = b2 - a9
            rote(4) = 9
            az = b1 & b2 & b1 - a9 & b2 - a9
            Label2.Caption = b1 & b2 & "-" & ">" & b1 - a9 & b2 - a9
        End If
    End If
End Sub

Public Sub Function_lefthot()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
        Image3(a1).Move a2 - (abcd * a9), a3 - (abcd * a9)
        table(b1, b2) = 0
        table(b1 - a9, b2 - a9) = 3
        chek(b1, b2) = 0
        chek(b1 - a9, b2 - a9) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 - a9
        rote(3) = b2 - a9
        rote(4) = 9
        az = b1 & b2 & b1 - a9 & b2 - a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 - a9 & b2 - a9
    End If
End Sub

Public Sub Function_kinrighthot1()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (table(b1 + (a9 - 1), (b2 + (a9 - 1))) = 2) Or (table(b1 + (a9 - 1), (b2 + (a9 - 1))) =
4) Then
        Image3(a1).Move a2 + (abcd * a9), a3 + (abcd * a9)
        table(b1, b2) = 0
        table(b1 + a9, b2 + a9) = 3
        table(b1 + (a9 - 1), b2 + (a9 - 1)) = 0
        chek(b1, b2) = 0
        chek(b1 + a9, b2 + a9) = bb
        bbb = chek(b1 + (a9 - 1), b2 + (a9 - 1))
        If bbb >= 40 And bbb <= 47 Then
           bbb = bbb - 40
           Image4(bbb).Enabled = False
           Image4(bbb).Visible = False
        ElseIf bbb >= 20 And bbb <= 27 Then
           bbb = bbb - 20
           Image2(bbb).Enabled = False
           Image2(bbb).Visible = False
        End If
        chek(b1 + (a9 - 1), b2 + (a9 - 1)) = 0
        chk = chk + 1
        ct = ct + 1
        If chk = 8 Then
           MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
           'Option6.Value = True
           Command6_Click
        End If
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + a9
        rote(3) = b2 + a9
        rote(4) = 9
        az = b1 & b2 & b1 + a9 & b2 + a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + a9 & b2 + a9
   End If
   End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Sub Function_righthot1()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
        Image3(a1).Move a2 + (abcd * a9), a3 + (abcd * a9)
        table(b1, b2) = 0
        table(b1 + a9, b2 + a9) = 3
        chek(b1, b2) = 0
        chek(b1 + a9, b2 + a9) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 + a9
        rote(3) = b2 + a9
        rote(4) = 9
        az = b1 & b2 & b1 + a9 & b2 + a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 + a9 & b2 + a9
   End If
End Sub


Public Sub Function_kinrighthot()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
    If (table(b1 - (a9 - 1), (b2 + (a9 - 1))) = 2) Or (table(b1 - (a9 - 1), (b2 + (a9 - 1))) = 4)
Then
        Image3(a1).Move a2 + (abcd * a9), a3 - (abcd * a9)
        table(b1, b2) = 0
        table(b1 - a9, b2 + a9) = 3
        table(b1 - (a9 - 1), b2 + (a9 - 1)) = 0
        chek(b1, b2) = 0
        chek(b1 - a9, b2 + a9) = bb
        bbb = chek(b1 - (a9 - 1), b2 + (a9 - 1))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If bbb >= 40 And bbb <= 47 Then
           bbb = bbb - 40
           Image4(bbb).Enabled = False
           Image4(bbb).Visible = False
        ElseIf bbb >= 20 And bbb <= 27 Then
           bbb = bbb - 20
           Image2(bbb).Enabled = False
           Image2(bbb).Visible = False
        End If
        chek(b1 - (a9 - 1), b2 + (a9 - 1)) = 0
        chk = chk + 1
        ct = ct + 1
        If chk = 8 Then
           MsgBox ("คุณ " & Text1.Text & " เป็นผู้ชนะครับ")
           'Option6.Value = True
           Command6_Click
        End If
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 - a9
        rote(3) = b2 + a9
        rote(4) = 9
        az = b1 & b2 & b1 - a9 & b2 + a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 - a9 & b2 + a9
   End If
   End If
End Sub

Public Sub Function_righthot()
    bb = a1 + 30
    a2 = Image3(a1).Left
    a3 = Image3(a1).Top
    Function_chek
    If Image3(a1).Visible = True Then
        Image3(a1).Move a2 + (abcd * a9), a3 - (abcd * a9)
        table(b1, b2) = 0
        table(b1 - a9, b2 + a9) = 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        chek(b1, b2) = 0
        chek(b1 - a9, b2 + a9) = bb
        walk = 1
        ck = 0
        rote(0) = b1
        rote(1) = b2
        rote(2) = b1 - a9
        rote(3) = b2 + a9
        rote(4) = 9
        az = b1 & b2 & b1 - a9 & b2 + a9
        Label2.Caption = b1 & b2 & "-" & ">" & b1 - a9 & b2 + a9
    End If
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PORTA1      EQU     0DC00H
PORTB1      EQU     0DC01H
PORTC1      EQU     0DC02H
PORTCON1    EQU     0DC03H
PORTA2      EQU     0FC00H
PORTB2      EQU     0FC01H
PORTC2      EQU     0FC02H
PORTCON2    EQU     0FC03H

START:      ORG     0000H
            LCALL   MAIN
            LJMP    START

MAIN:       MOV     A,#092H
            MOV     DPTR,#PORTCON1
            MOVX    @DPTR,A
            MOV     A,#092H
            MOV     DPTR,#PORTCON2
            MOVX    @DPTR,A

            MOV     TMOD,#021H
            MOV     TH1,#0FDH
            MOV     TL1,#0FDH           ;
            SETB    TR1

            MOV     SCON,#040H
            LCALL   KEEP
            LCALL   SEND
            RET

DELAY:      PUSH    04          ; TIME DELAY
            PUSH    05
            PUSH    06
            MOV     R4,#0FH
LLLL:       MOV     R5,#0FFH
LLL:        MOV     R6,#0FFH
LL1:        DJNZ    R6,LL1
            DJNZ    R5,LLL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ    R4,LLLL
        POP     06
        POP     05
        POP     04
        RET

;*****************************************************************
KEEP:   PUSH    00
        PUSH    01
        PUSH    02
        PUSH    03
        PUSH    ACC
        MOV     R1,#30H
        MOV     DPTR,#PORTA1
        MOVX    A,@DPTR
        MOV     R0,A
        LCALL   CHECK
        MOV     DPTR,#PORTB1
        MOVX    A,@DPTR
        MOV     R0,A
        LCALL   CHECK
        MOV     DPTR,#PORTA2
        MOVX    A,@DPTR
        MOV     R0,A
        LCALL   CHECK
        MOV     DPTR,#PORTB2
        MOVX    A,@DPTR
        MOV     R0,A
        LCALL   CHECK
        DEC     R1
        POP     ACC
        POP     03
        POP     02
        POP     01
        POP     00
        RET

CHECK:  MOV     A,R0
        ANL     A,#00000001B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNZ     ZA1
        LCALL   ZE
        LJMP    ZB1
ZA1:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB1:    MOV     A,R0
        ANL     A,#00000010B
        JNZ     ZA2
        LCALL   ZE
        LJMP    ZB2
ZA2:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB2:    MOV     A,R0
        ANL     A,#00000100B
        JNZ     ZA3
        LCALL   ZE
        LJMP    ZB3
ZA3:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB3:    MOV     A,R0
        ANL     A,#00001000B
        JNZ     ZA4
        LCALL   ZE
        LJMP    ZB4
ZA4:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB4:    MOV     A,R0
        ANL     A,#00010000B
        JNZ     ZA5
        LCALL   ZE
        LJMP    ZB5
ZA5:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB5:    MOV     A,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ANL     A,#00100000B
        JNZ     ZA6
        LCALL   ZE
        LJMP    ZB6
ZA6:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB6:    MOV     A,R0
        ANL     A,#01000000B
        JNZ     ZA7
        LCALL   ZE
        LJMP    ZB7
ZA7:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
ZB7:    MOV     A,R0
        ANL     A,#10000000B
        JNZ     ZA8
        LCALL   ZE
        RET
ZA8:    MOV     A,#31H
        MOV     @R1,A
        INC     R1
        RET
ZE:     MOV     A,#30H
        MOV     @R1,A
        INC     R1
        RET

SEND:   PUSH    ACC
        PUSH    01
        PUSH    02
        MOV     R2,#00
        MOV     R1,#30H

LOOP:   CLR     TI
        MOV     SBUF,@R1
        JNB     TI,$                    ;
        CLR     TI                      ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        INC     R1
        MOV     A,R2
        INC     R2
        CJNE    A,#31,LOOP
        CLR     TI
        MOV     SBUF,#39H
        JNB     TI,$
        CLR     TI
        POP     02
        POP     01
        POP     ACC
        RET

RECEIVED:
        SETB    REN
        JNB     RI,$
        CLR     RI
        MOV     A,SBUF
        RET

EEE:    END
```

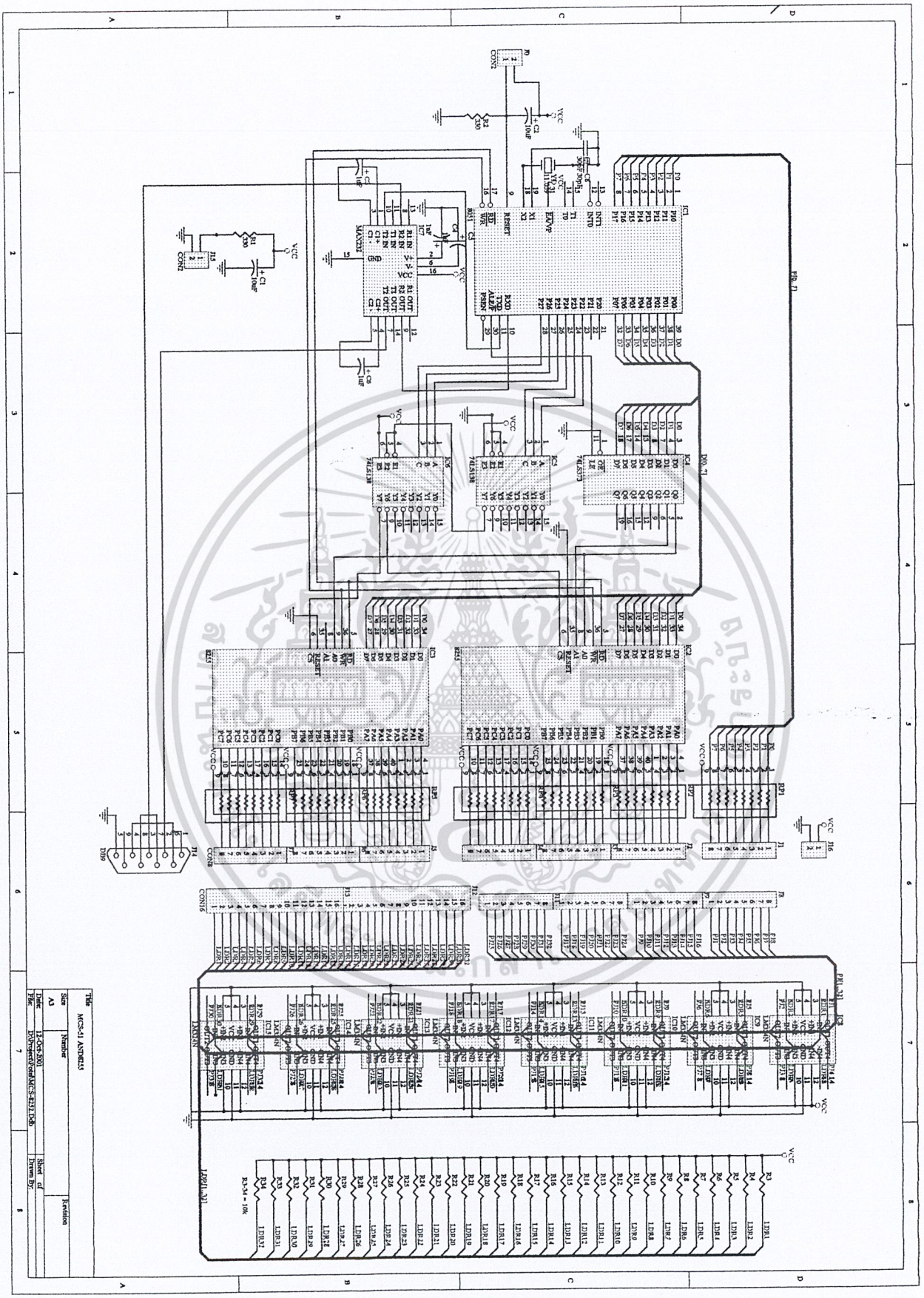


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

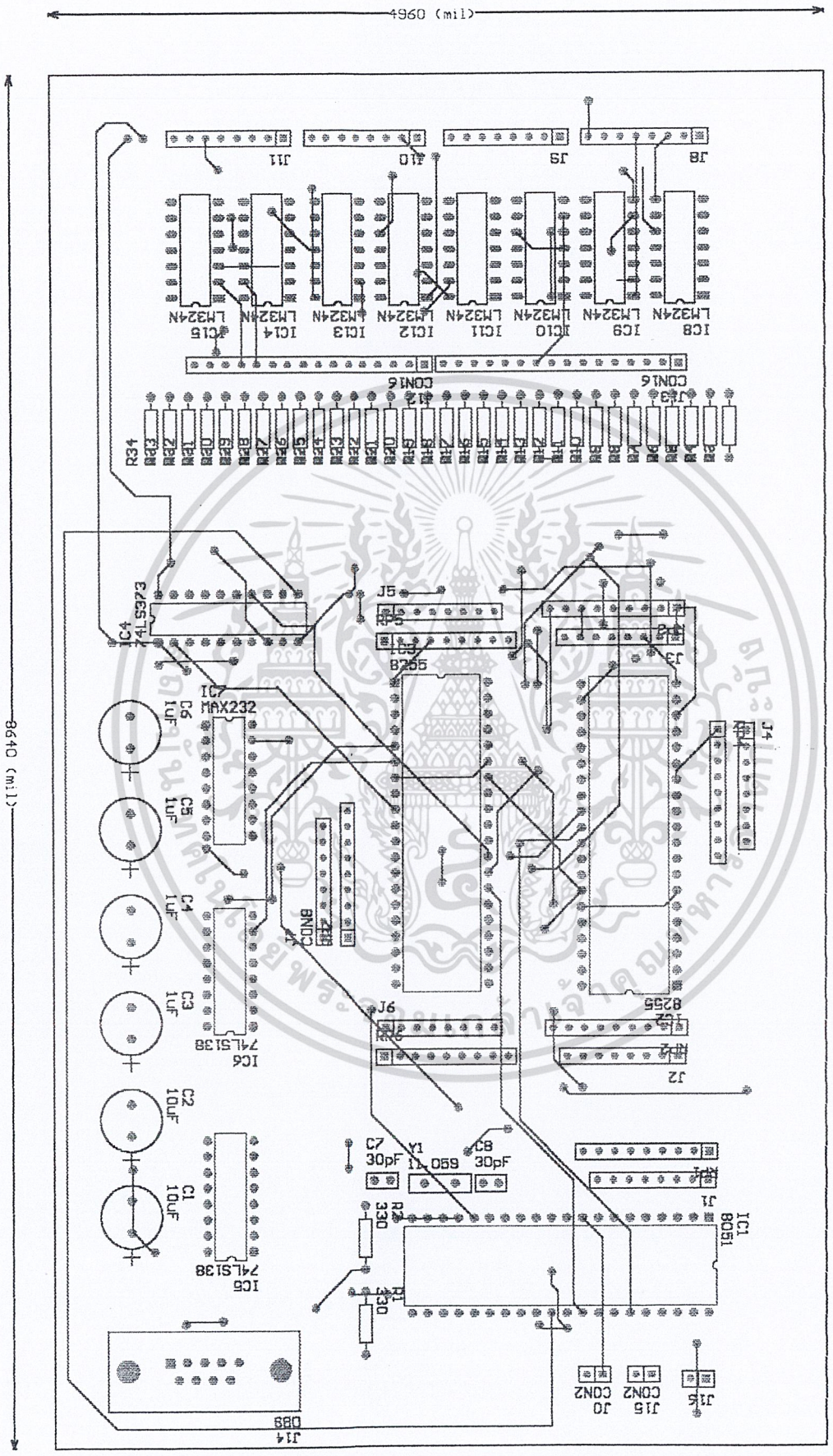


ภาคผนวก ข.

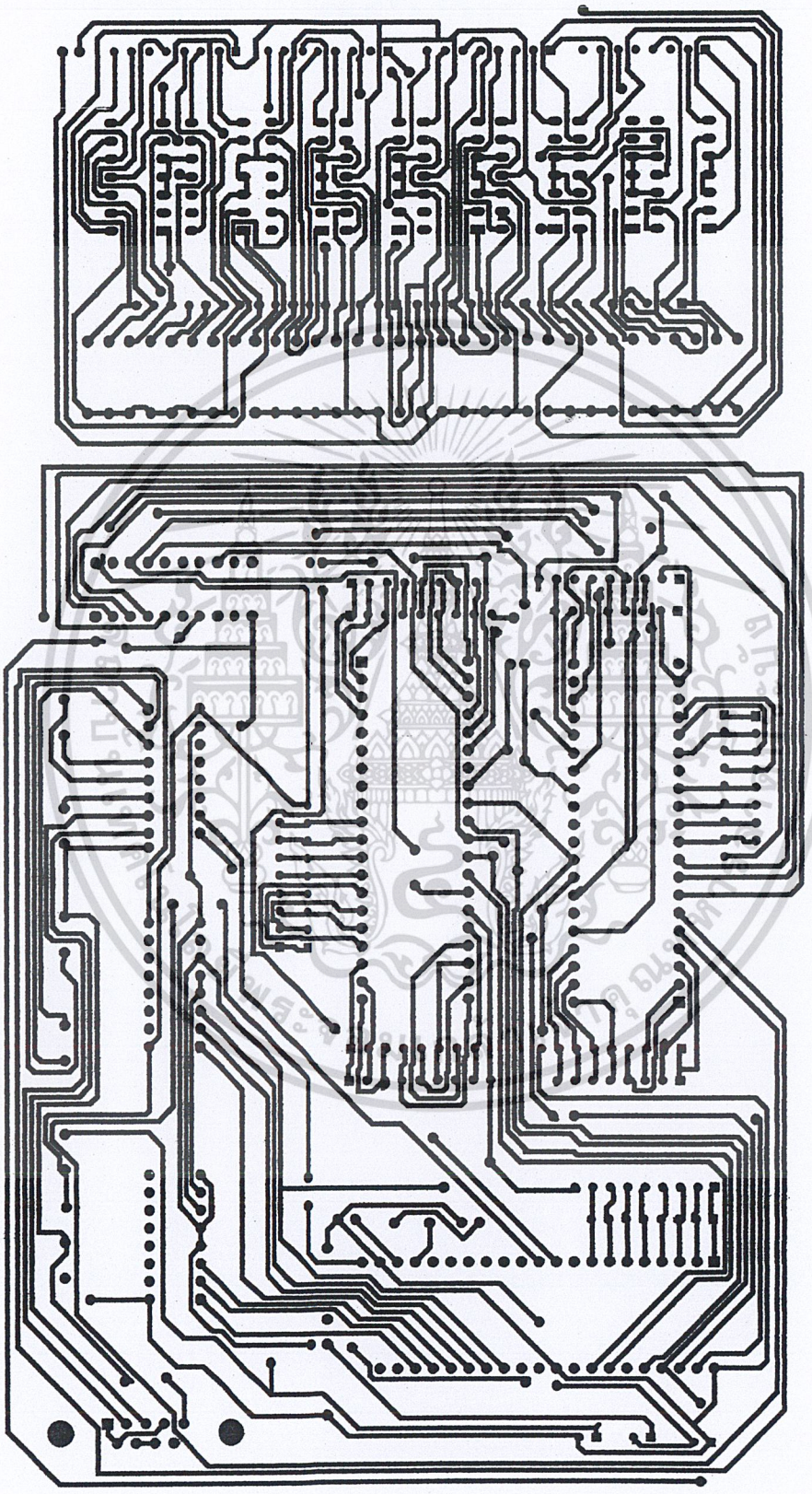
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



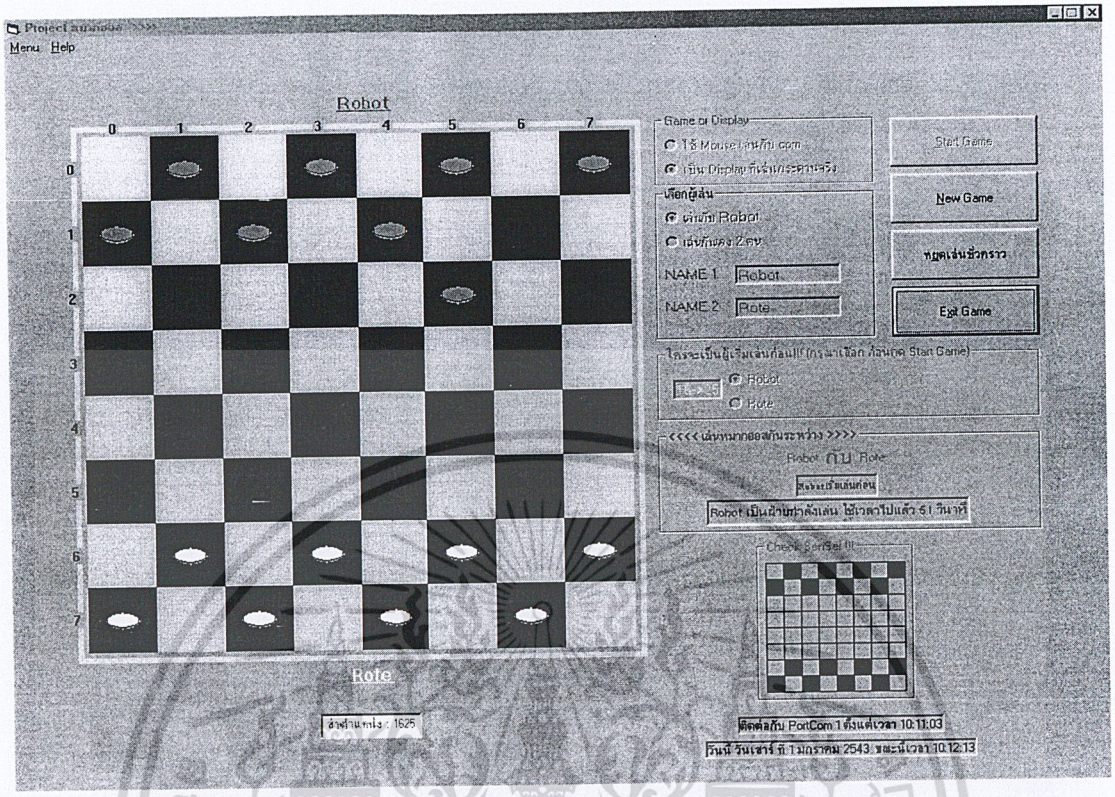
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



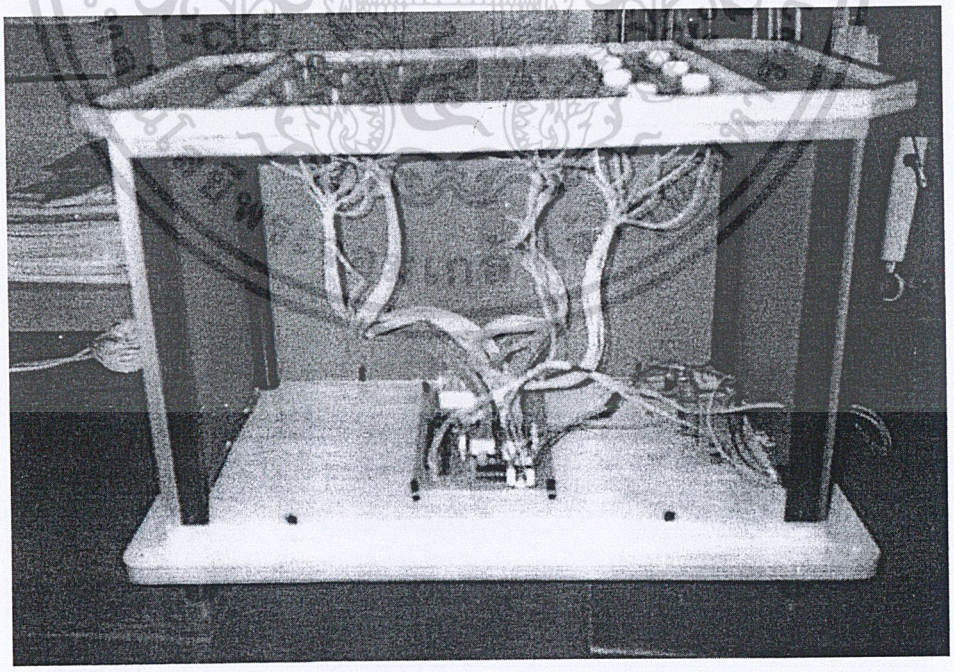
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

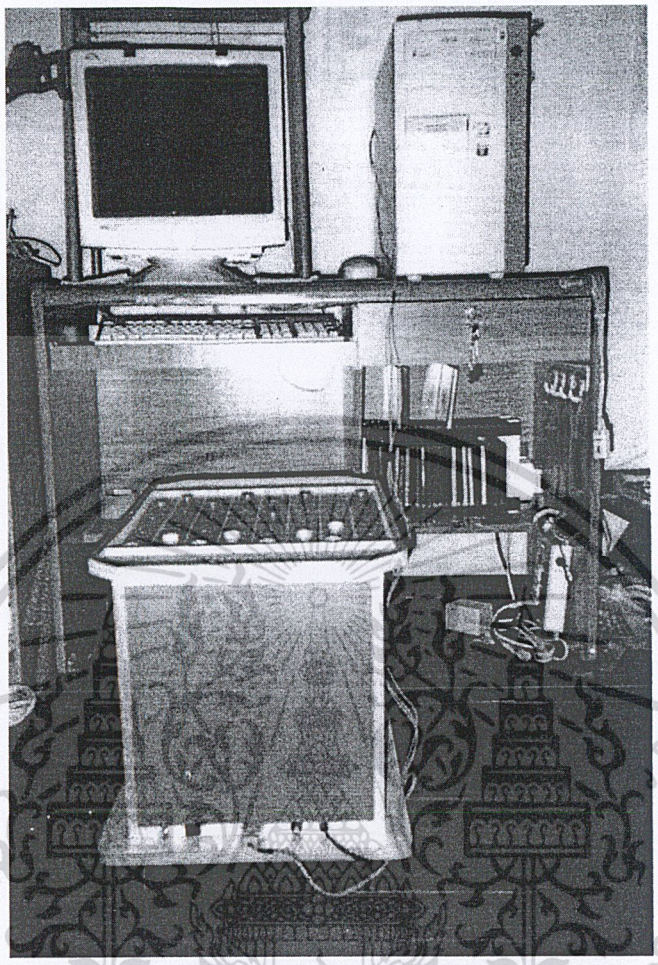


รูปแสดงผลของโปรแกรมที่ออกทางจอคอมพิวเตอร์



รูปแสดงส่วนของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการทำงานระหว่างฮาร์ดแวร์กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS373
SN54/74LS374

OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

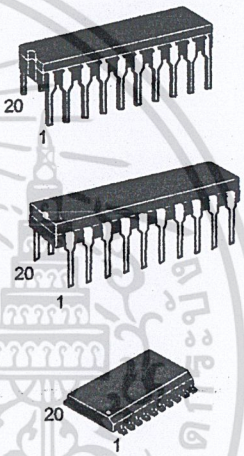
OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

LOW POWER SCHOTTKY

The SN54/74LS373 consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE is LOW, the data that meets the setup times is latched. Data appears on the bus when the Output Enable (OE) is LOW. When OE is HIGH the bus output is in the high impedance state.

The SN54/74LS374 is a high-speed, low-power Octal D-type Flip-Flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. A buffered Clock (CP) and Output Enable (OE) is common to all flip-flops. The SN54/74LS374 is manufactured using advanced Low Power Schottky technology and is compatible with all Motorola TTL families.

- Eight Latches in a Single Package
- 3-State Outputs for Bus Interfacing
- Hysteresis on Latch Enable
- Edge-Triggered D-Type Inputs
- Buffered Positive Edge-Triggered Clock
- Hysteresis on Clock Input to Improve Noise Margin
- Input Clamp Diodes Limit High Speed Termination Effects



J SUFFIX
CERAMIC
CASE 732-03

N SUFFIX
PLASTIC
CASE 738-03

DW SUFFIX
SOIC
CASE 751D-03

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

PIN NAMES

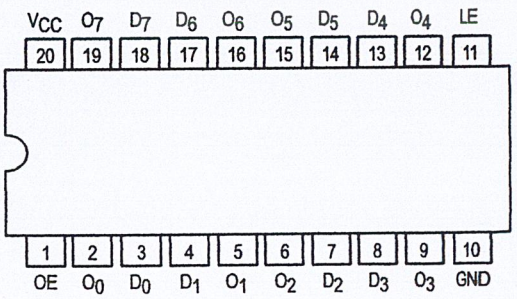
D_0-D_7	Data Inputs
LE	Latch Enable (Active HIGH) Input
$\overline{CP}$	Clock (Active HIGH going edge) Input
$\overline{OE}$	Output Enable (Active LOW) Input
O_0-O_7	Outputs (Note b)

	LOADING (Note a)	
	HIGH	LOW
D_0-D_7	0.5 U.L.	0.25 U.L.
LE	0.5 U.L.	0.25 U.L.
$\overline{CP}$	0.5 U.L.	0.25 U.L.
$\overline{OE}$	0.5 U.L.	0.25 U.L.
O_0-O_7	65 (25) U.L.	15 (7.5) U.L.

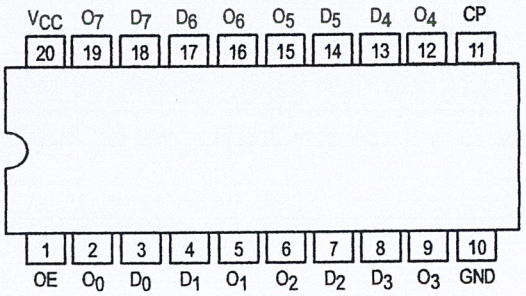
NOTES:
a) 1 TTL Units Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b) The Output LOW drive factor is 7.5 U.L. for Military (54) and 25 U.L. for Commercial (74) Temperature Ranges. The Output HIGH drive factor is 25 U.L. for Military (54) and 65 U.L. for Commercial (74) Temperature Ranges.

CONNECTION DIAGRAM DIP (TOP VIEW)

SN54/74LS373



SN54/74LS374



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54/74LS373 • SN54/74LS374

TRUTH TABLE

LS373

D_n	LE	$\overline{OE}$	O_n
H	H	L	H
L	H	L	L
X	L	L	Q_0
X	X	H	Z*

LS374

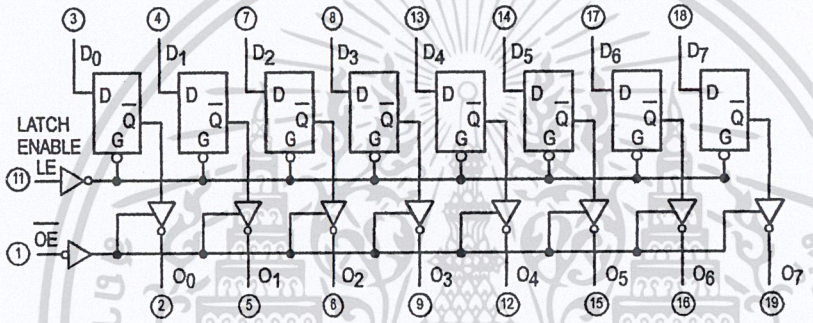
D_n	LE	$\overline{OE}$	O_n
H	⌐	L	H
L	⌐	L	L
X	X	H	Z*

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance

* Note: Contents of flip-flops unaffected by the state of the Output Enable input ($\overline{OE}$).

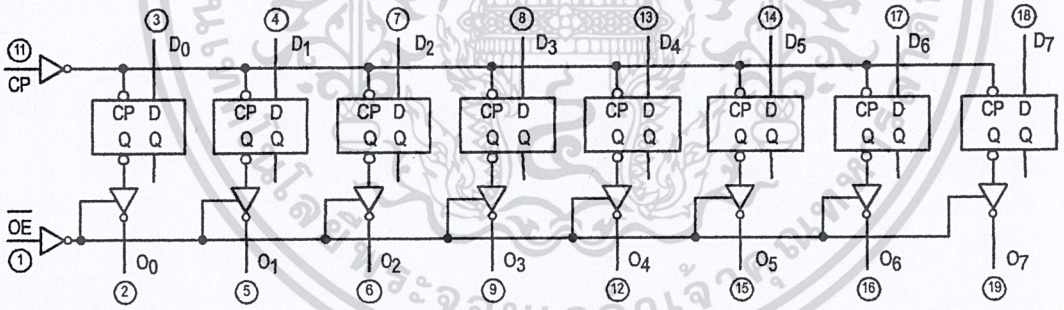
LOGIC DIAGRAMS

SN54LS/74LS373



V_{CC} = PIN 20
GND = PIN 10
◯ = PIN NUMBERS

SN54LS/74LS374



GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54 74			−1.0 −2.6	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Typ	Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.4	3.4		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.4	3.1		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	µA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−20	µA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current				20	µA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current (Note 1)		−30		−130	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				40	mA	V_{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits LS373 Min	LS373 Typ	LS373 Max	LS374 Min	LS374 Typ	LS374 Max	Unit	Test Conditions
f_{MAX}	Maximum Clock Frequency				35	50		MHz	C_L = 45 pF, R_L = 667 Ω
t_{PLH} t_{PHL}	Propagation Delay, Data to Output		12 12	18 18				ns	
t_{PLH} t_{PHL}	Clock or Enable to Output		20 18	30 30		15 19	28 28	ns	
t_{PZH} t_{PZL}	Output Enable Time		15 25	28 36		20 21	28 28	ns	
t_{PHZ} t_{PLZ}	Output Disable Time		12 15	20 25		12 15	20 25	ns	C_L = 5.0 pF

AC SETUP REQUIREMENTS (T_A = 25°C, V_{CC} = 5.0 V)

Symbol	Parameter	Limits LS373 Min	LS373 Max	LS374 Min	LS374 Max	Unit
t_W	Clock Pulse Width	15		15		ns
t_s	Setup Time	5.0		20		ns
t_h	Hold Time	20		0		ns

DEFINITION OF TERMS

SETUP TIME (t_s) — is defined as the minimum time required for the correct logic level to be present at the logic input prior to LE transition from HIGH-to-LOW in order to be recognized and transferred to the outputs.

HOLD TIME (t_h) — is defined as the minimum time following the LE transition from HIGH-to-LOW that the logic level must be maintained at the input in order to ensure continued recognition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC WAVEFORMS

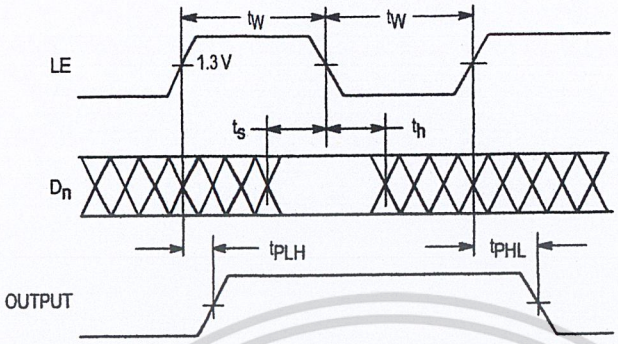


Figure 1

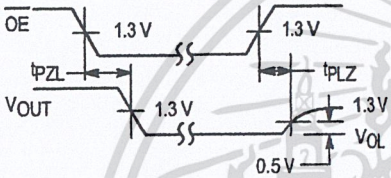


Figure 2

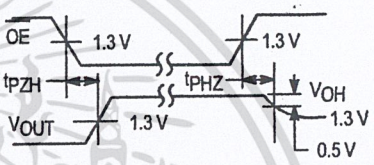
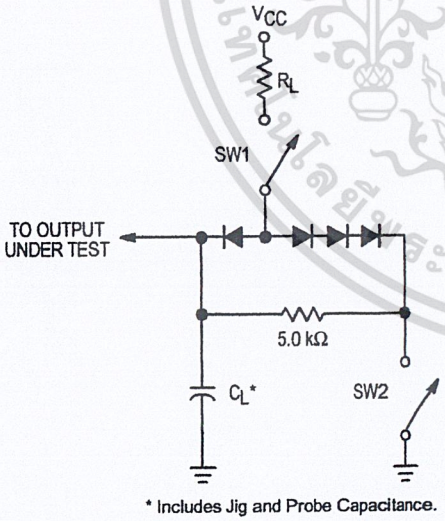


Figure 3

AC LOAD CIRCUIT



SWITCH POSITIONS

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed
t_{PHZ}	Closed	Closed

Figure 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC WAVEFORMS

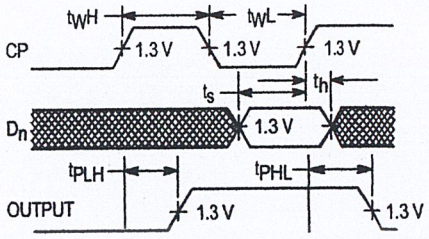


Figure 5

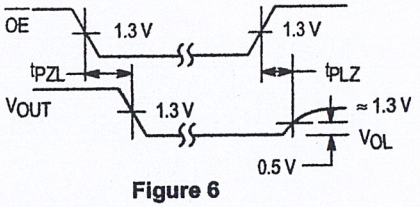


Figure 6

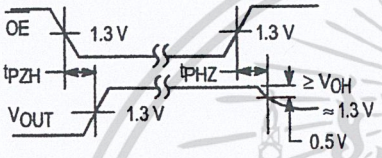
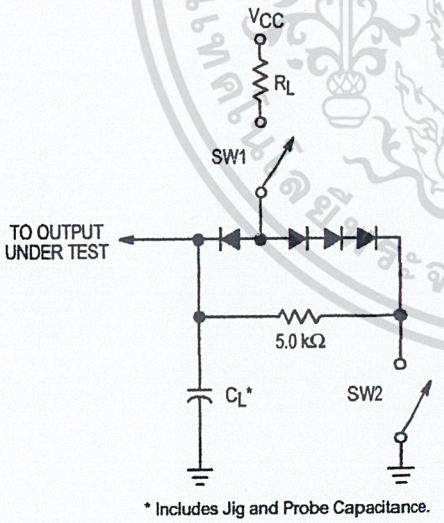


Figure 7

AC LOAD CIRCUIT



SWITCH POSITIONS

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed
t_{PHZ}	Closed	Closed

Figure 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55A
CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- **Compatible with all Intel and Most Other Microprocessors**
- **High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188**
- **24 Programmable I/O Pins**
- **Low Power CHMOS**
- **Completely TTL Compatible**
- **Control Word Read-Back Capability**
- **Direct Bit Set/Reset Capability**
- **2.5 mA DC Drive Capability on all I/O Port Outputs**
- **Available in 40-Pin DIP and 44-Pin PLCC**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

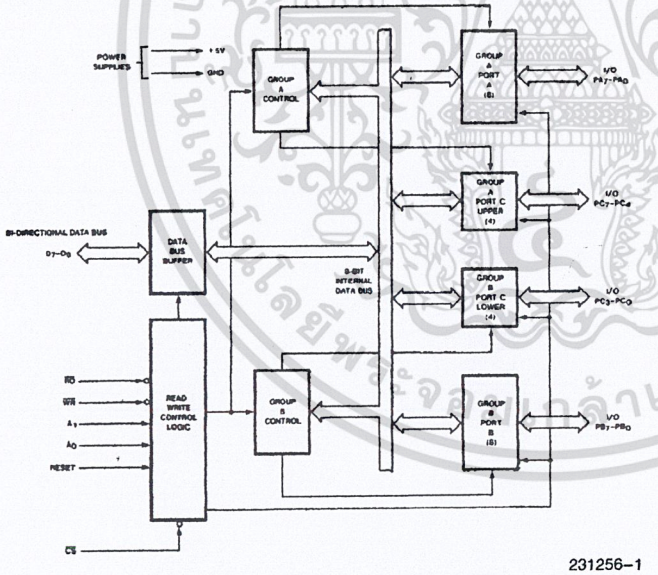
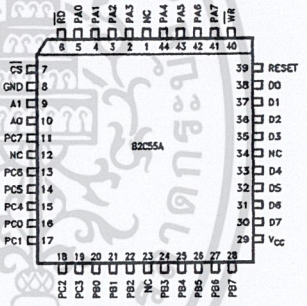


Figure 1. 82C55A Block Diagram



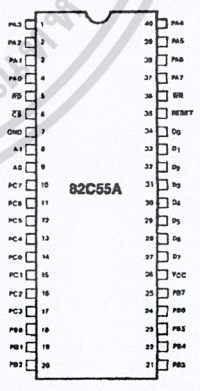


Figure 2. 82C55A Pinout
Diagrams are for pin reference only. Package sizes are not to scale.

October 1995

Order Number: 231256-004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description

Symbol	Pin Number Dip	Pin Number PLCC	Type	Name and Function
PA_{3-0}	1–4	2–5	I/O	**PORT A, PINS 0–3:** Lower nibble of an 8-bit data output latch/ buffer and an 8-bit data input latch.
$\overline{RD}$	5	6	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	6	7	I	**CHIP SELECT:** A low on this input enables the 82C55A to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and WR are ignored otherwise.
GND	7	8		**System Ground**
A_{1-0}	8–9	9–10	I	**ADDRESS:** These input signals, in conjunction $\overline{RD}$ and $\overline{WR}$, control the selection of one of the three ports or the control word registers.
PC_{7-4}	10–13	11,13–15	I/O	**PORT C, PINS 4–7:** Upper nibble of an 8-bit data output latch/ buffer and an 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.
PC_{0-3}	14–17	16–19	I/O	**PORT C, PINS 0–3:** Lower nibble of Port C.
PB_{0-7}	18–25	20–22, 24–28	I/O	**PORT B, PINS 0–7:** An 8-bit data output latch/buffer and an 8-bit data input buffer.
V_{CC}	26	29		**SYSTEM POWER:** + 5V Power Supply.
D_{7-0}	27–34	30–33, 35–38	I/O	**DATA BUS:** Bi-directional, tri-state data bus lines, connected to system data bus.
RESET	35	39	I	**RESET:** A high on this input clears the control register and all ports are set to the input mode.
$\overline{WR}$	36	40	I	**WRITE CONTROL:** This input is low during CPU write operations.
PA_{7-4}	37–40	41–44	I/O	**PORT A, PINS 4–7:** Upper nibble of an 8-bit data output latch/ buffer and an 8-bit data input latch.
NC		1, 12, 23, 34		No Connect

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A - Data Bus
0	1	0	1	0	Port B - Data Bus
1	0	0	1	0	Port C - Data Bus
1	1	0	1	0	Control Word - Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus - Port A
0	1	1	0	0	Data Bus - Port B
1	0	1	0	0	Data Bus - Port C
1	1	1	0	0	Data Bus - Control
					Disable Function
X	X	X	X	1	Data Bus - 3 - State
X	X	1	1	0	Data Bus - 3 - State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55A FUNCTIONAL DESCRIPTION

General

The 82C55A is a programmable peripheral interface device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 82C55A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7–C4)
Control Group B - Port B and Port C lower (C3–C0)

The control word register can be both written and read as shown in the address decode table in the pin descriptions. Figure 6 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A. One 8-bit data output latch/buffer and one 8-bit input latch buffer. Both "pull-up" and "pull-down" bus hold devices are present on Port A.

Port B. One 8-bit data input/output latch/buffer. Only "pull-up" bus hold devices are present on Port B.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B. Only "pull-up" bus hold devices are present on Port C.

See Figure 4 for the bus-hold circuit configuration for Port A, B, and C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

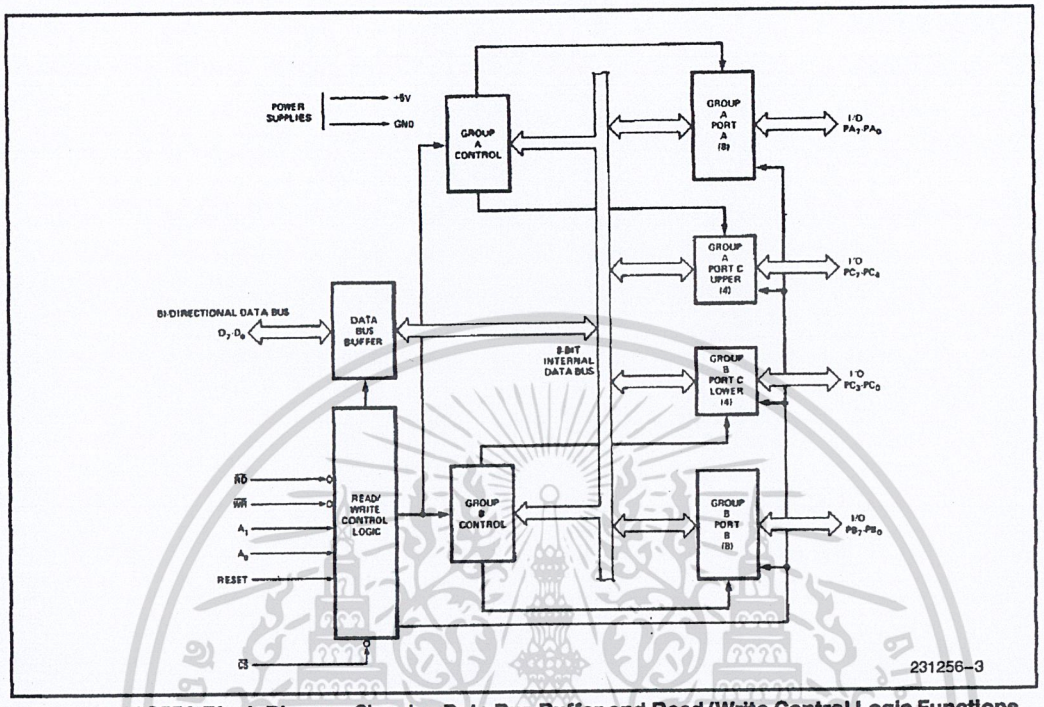
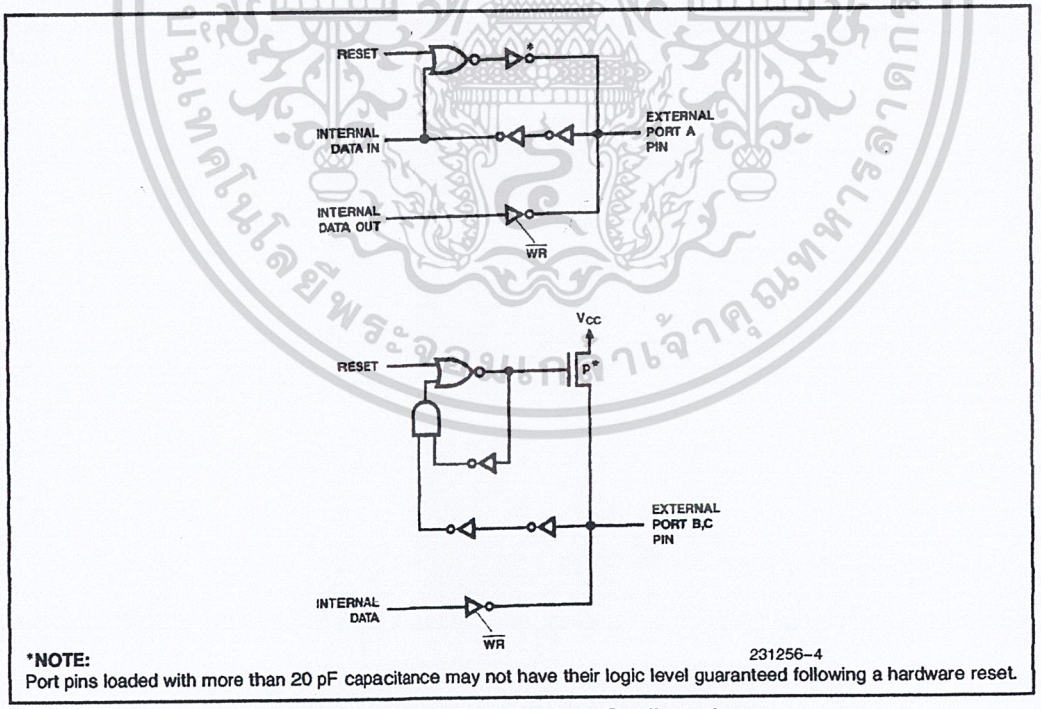


Figure 3. 82C55A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions



***NOTE:**
Port pins loaded with more than 20 pF capacitance may not have their logic level guaranteed following a hardware reset.

Figure 4. Port A, B, C, Bus-hold Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 — Basic input/output
Mode 1 — Strobed Input/output
Mode 2 — Bi-directional Bus

When the reset input goes "high" all ports will be set to the input mode with all 24 port lines held at a logic "one" level by the internal bus hold devices (see Figure 4 Note). After the reset is removed the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown devices in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected by using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

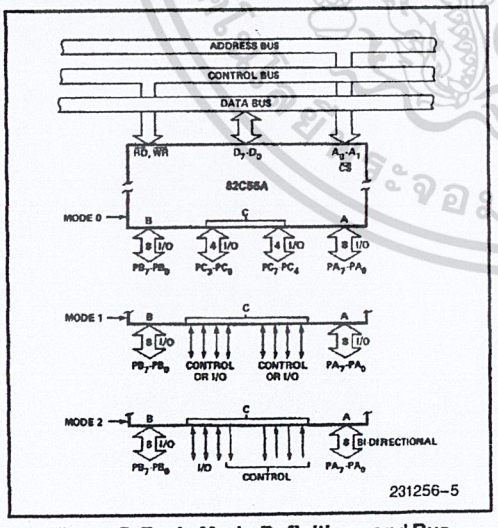


Figure 5. Basic Mode Definitions and Bus Interface

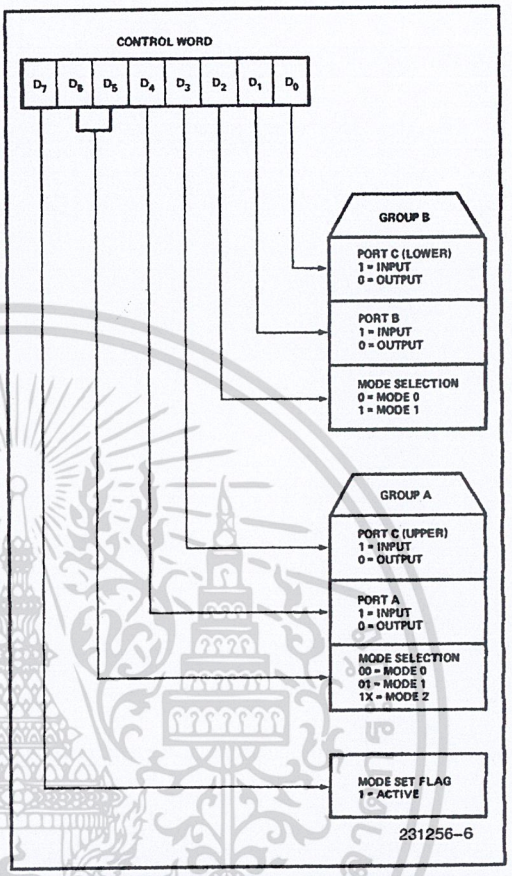


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

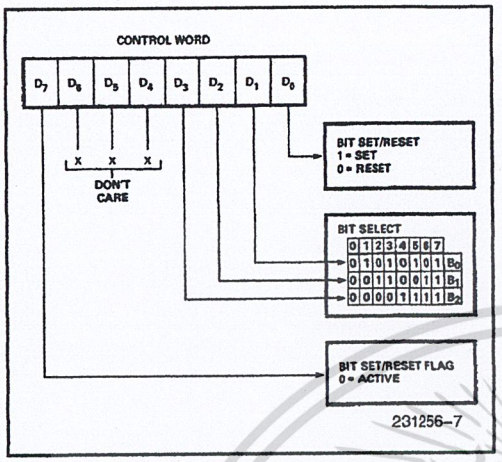


Figure 7. Bit Set/Reset Format

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is SET—Interrupt enable
(BIT-RESET)—INTE is RESET—Interrupt disable

Note:
All Mask flip-flops are automatically reset during mode selection and device Reset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

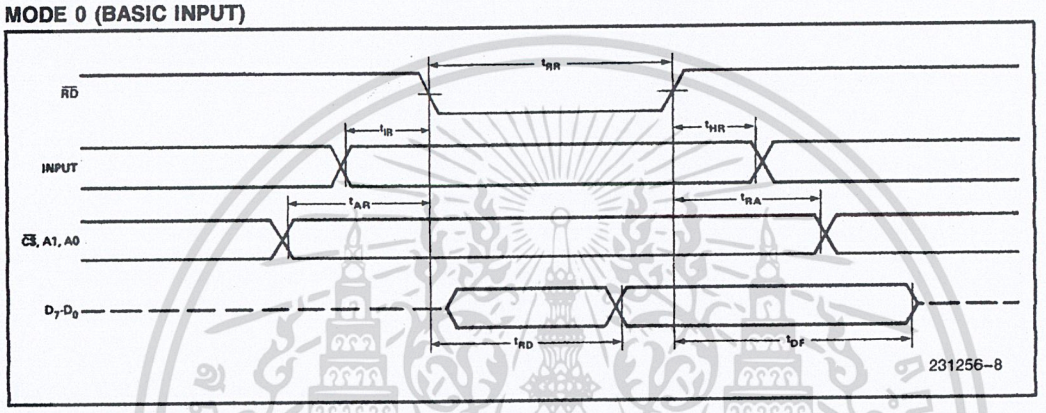
Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

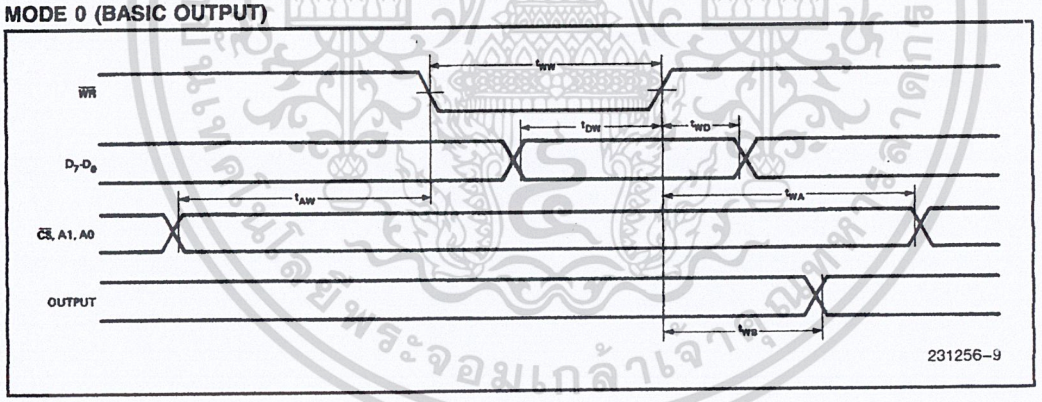
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)

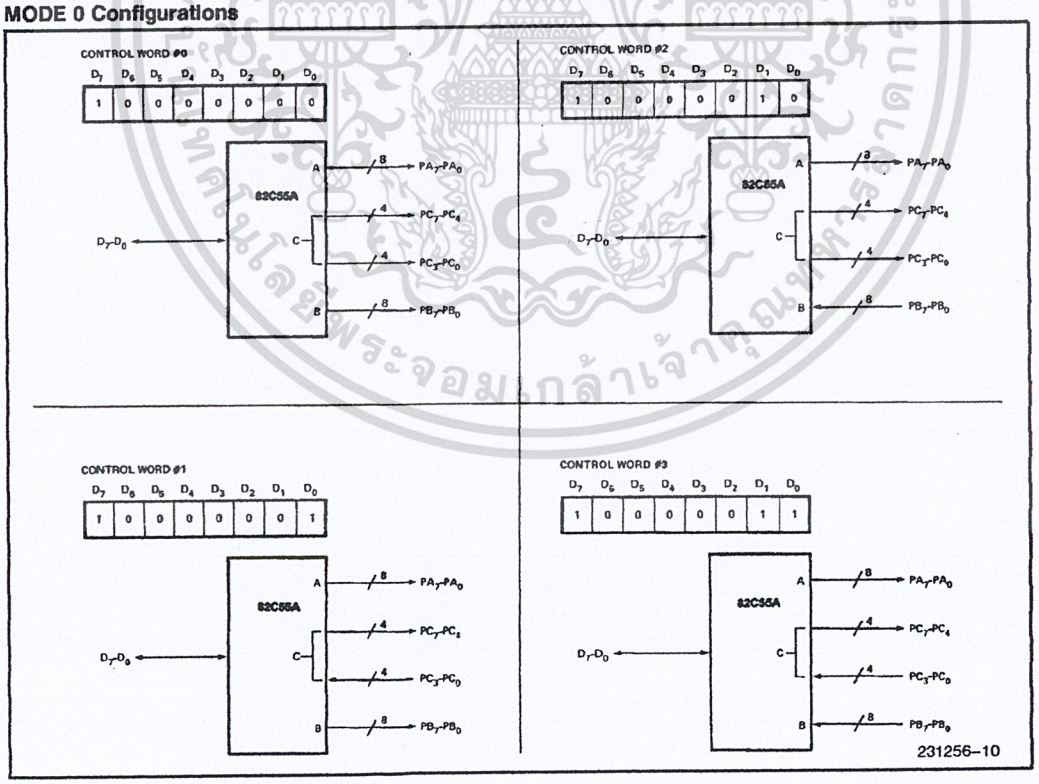


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 Port Definition

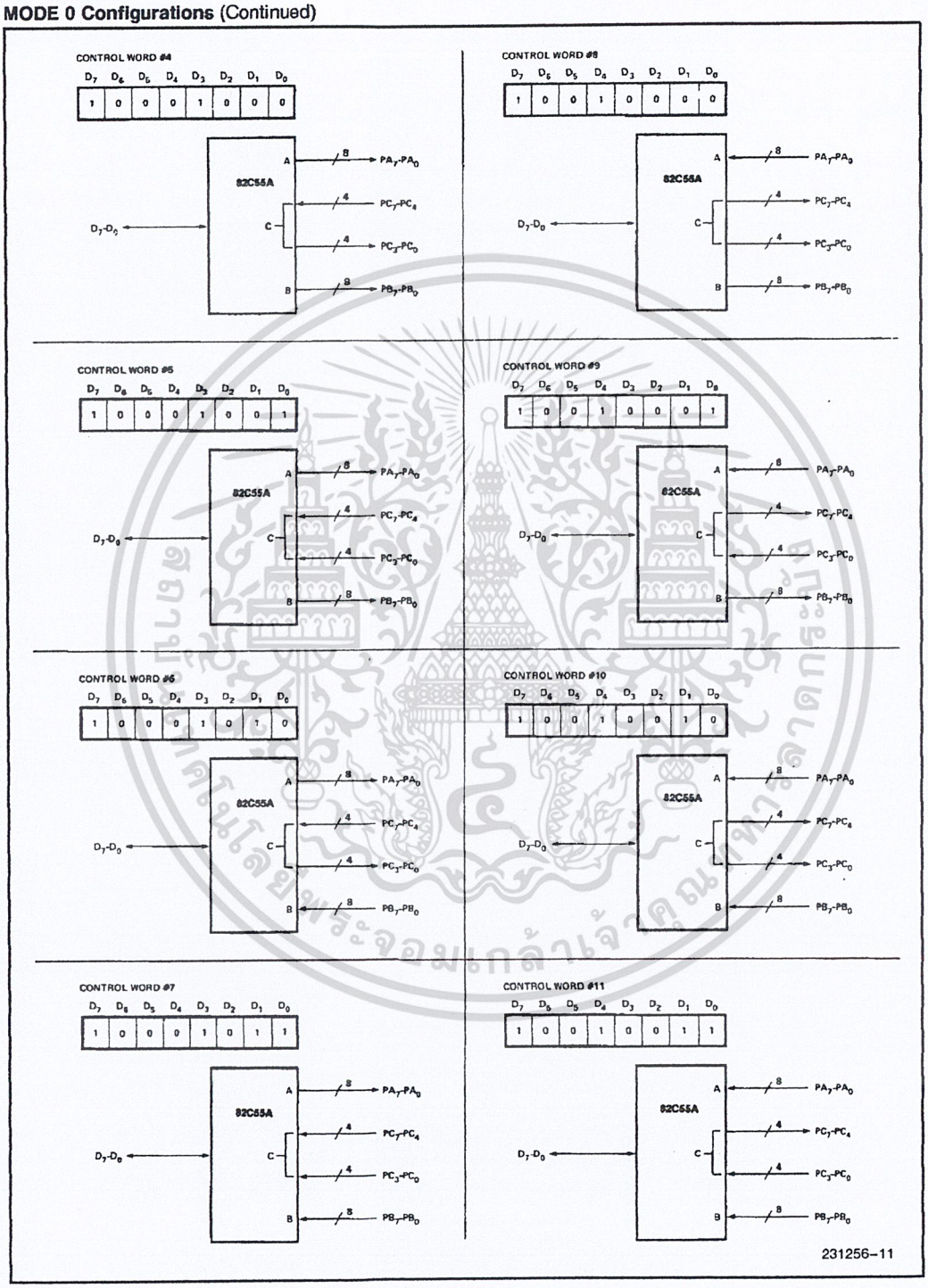
A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations



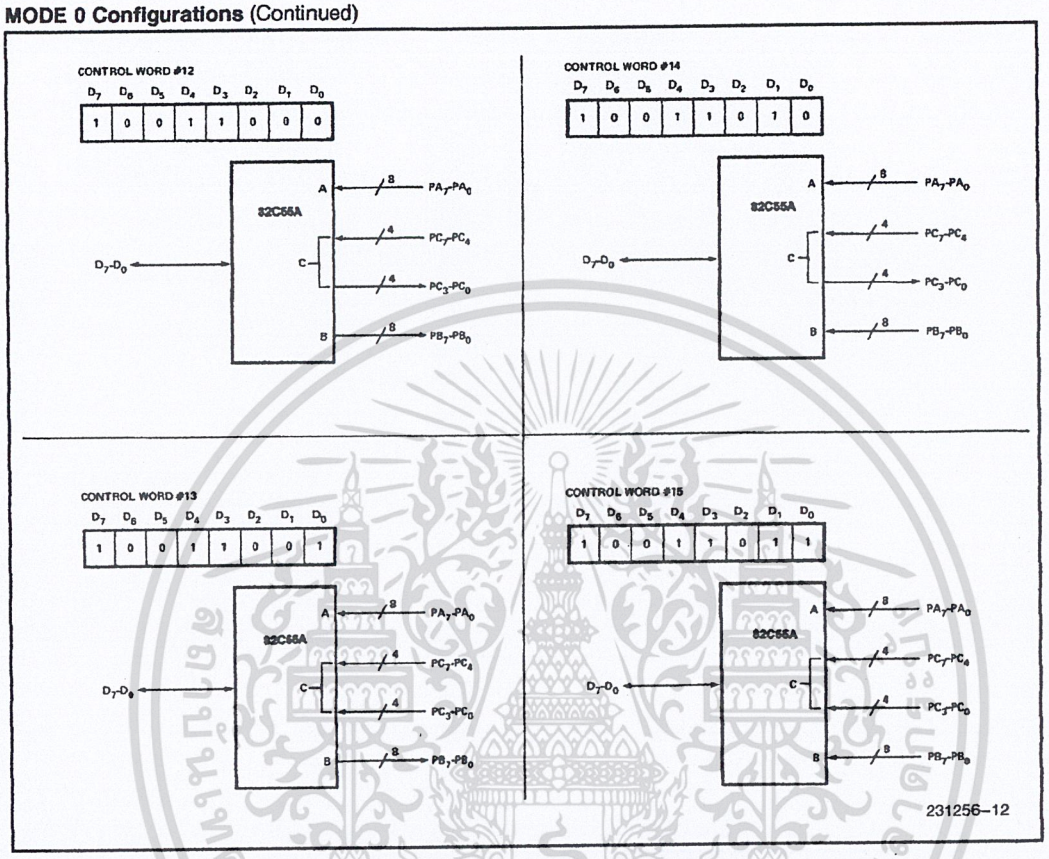
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 Configurations (Continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 Configurations (Continued)



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic functional Definitions:

- Two Groups (Group A and Group B).
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

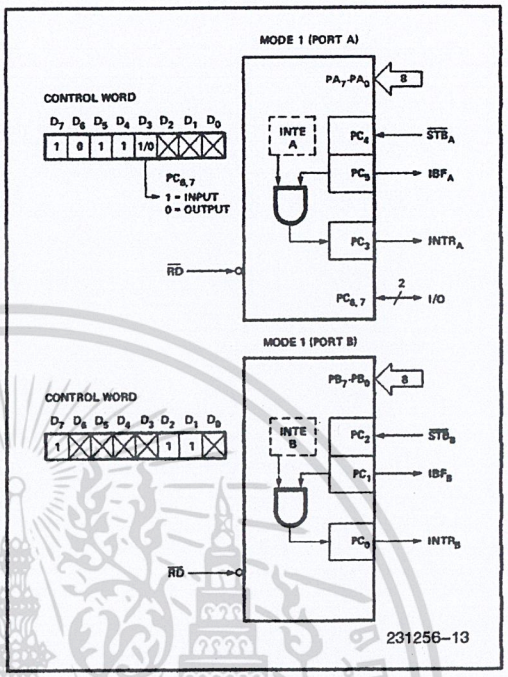


Figure 8. MODE 1 Input

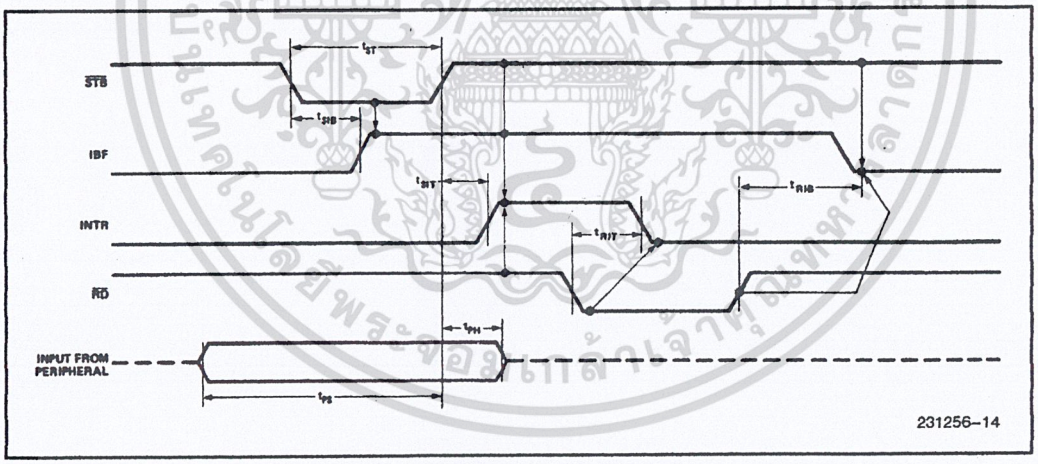


Figure 9. MODE 1 (Strobed Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

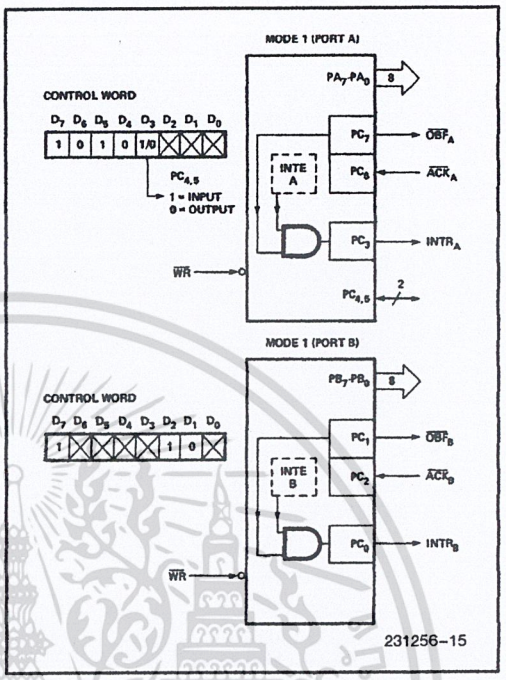


Figure 10. MODE 1 Output

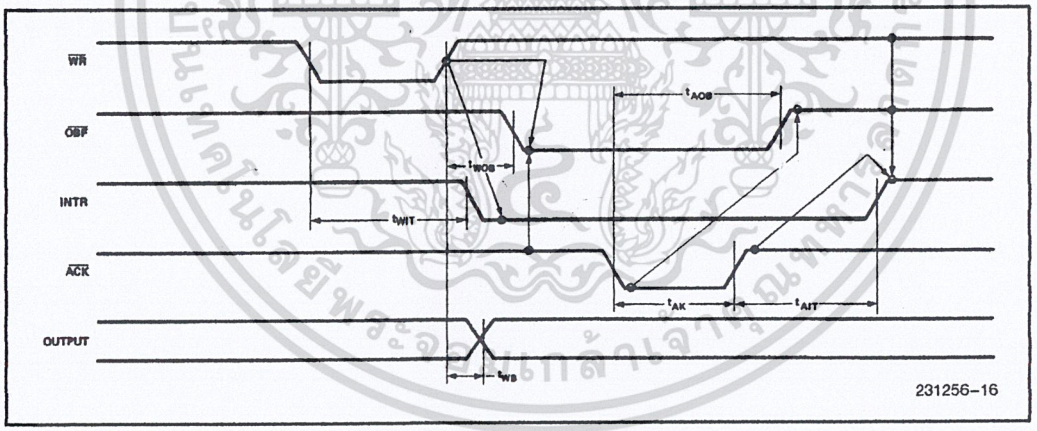


Figure 11. MODE 1 (Strobed Output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

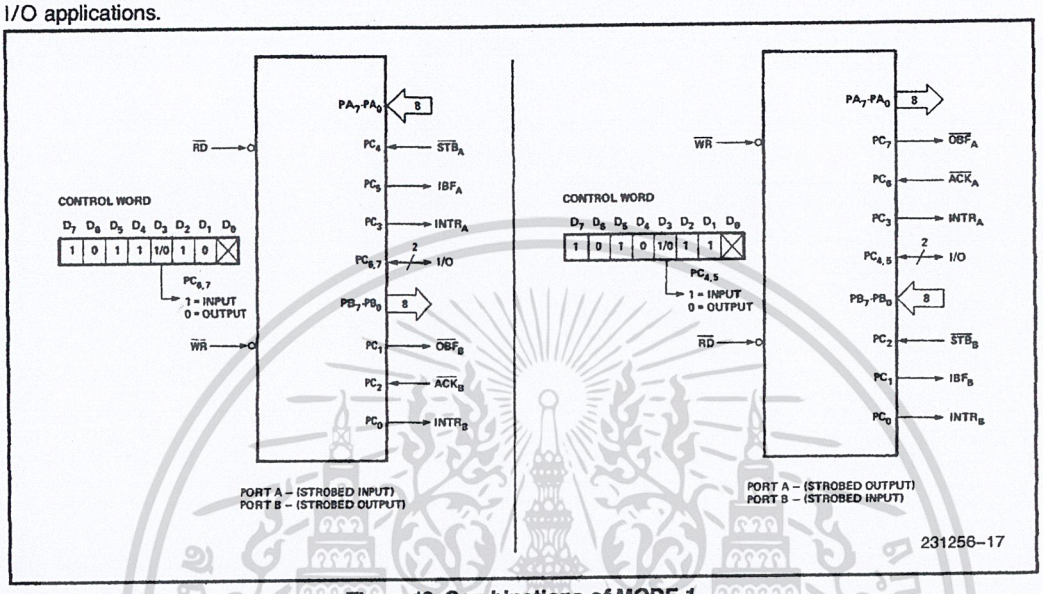


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A **only**.
- One 8-bit, bi-directional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

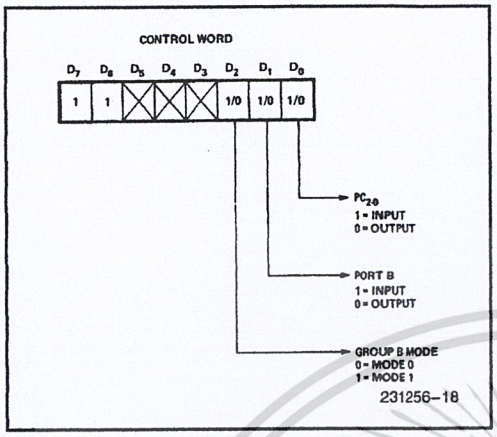


Figure 13. MODE Control Word

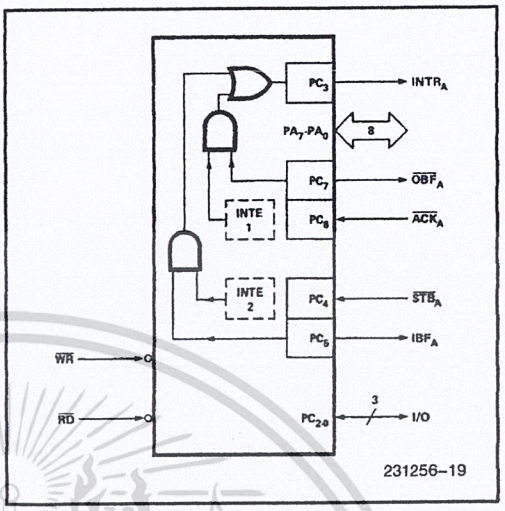


Figure 14. MODE 2

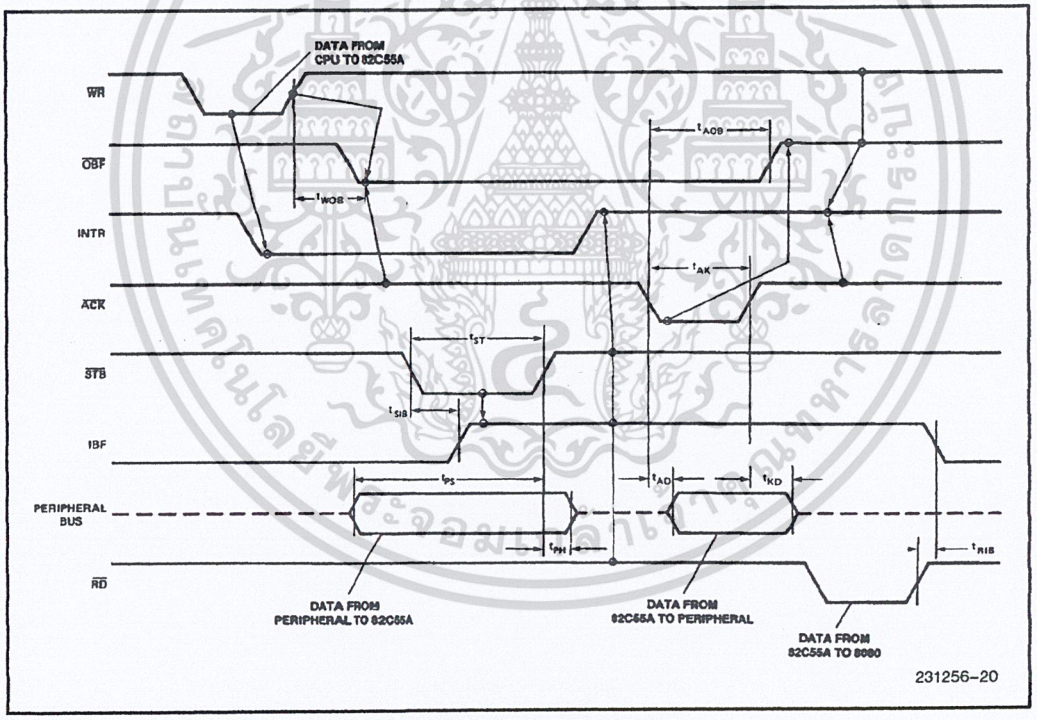


Figure 15. MODE 2 (Bidirectional)

NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$, and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \bullet \overline{MASK} \bullet \overline{STB} \bullet \overline{RD} + \overline{OBF} \bullet \overline{MASK} \bullet \overline{ACK} \bullet \overline{WR}$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

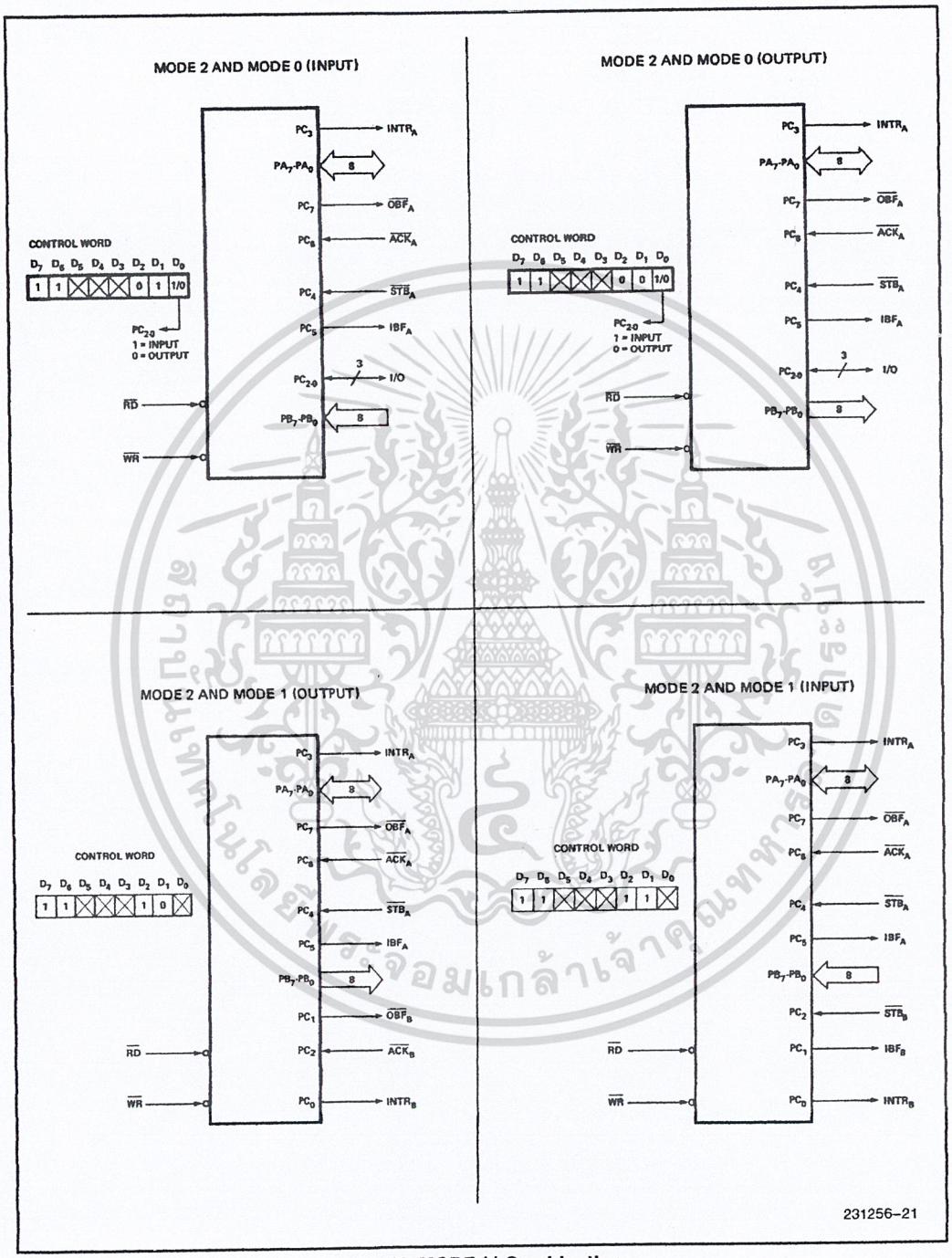


Figure 16. MODE ¼ Combinations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 18.

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5 mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

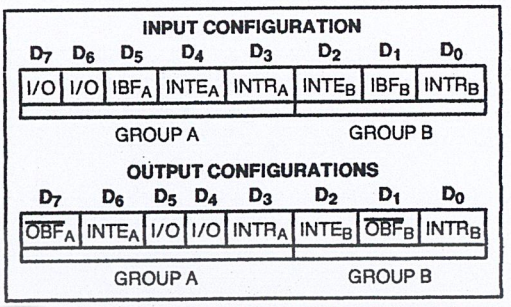


Figure 17a. MODE 1 Status Word Format

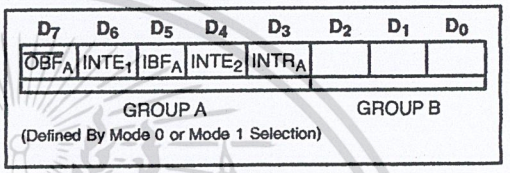


Figure 17b. MODE 2 Status Word Format

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2

Figure 18. Interrupt Enable Flags in Modes 1 and 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias....0°C to + 70°C
Storage Temperature .....− 65°C to + 150°C
Supply Voltage .....− 0.5 to + 8.0V
Operating Voltage .....+ 4V to + 7V
Voltage on any Input.....GND−2V to + 6.5V
Voltage on any Output ..GND−0.5V to V_{CC} + 0.5V
Power Dissipation .....1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V (T_A = −40°C to +85°C for Extended Temperture)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 µA
I_{IL}	Input Leakage Current		±1	µA	V_{IN} = V_{CC} to 0V (Note 1)
I_{OFL}	Output Float Leakage Current		±10	µA	V_{IN} = V_{CC} to 0V (Note 2)
I_{DAR}	Darlington Drive Current	±2.5	(Note 4)	mA	Ports A, B, C R_{ext} = 500Ω V_{ext} = 1.7V
I_{PHL}	Port Hold Low Leakage Current	+50	+300	µA	V_{OUT} = 1.0V Port A only
I_{PHH}	Port Hold High Leakage Current	−50	−300	µA	V_{OUT} = 3.0V Ports A, B, C
I_{PHLO}	Port Hold Low Overdrive Current	−350		µA	V_{OUT} = 0.8V
I_{PHHO}	Port Hold High Overdrive Current	+350		µA	V_{OUT} = 3.0V
I_{CC}	V_{CC} Supply Current		10	mA	(Note 3)
I_{CCSB}	V_{CC} Supply Current-Standby		10	µA	V_{CC} = 5.5V V_{IN} = V_{CC} or GND Port Conditions If I/P = Open/High O/P = Open Only With Data Bus = High/Low $\overline{CS}$ = High Reset = Low Pure Inputs = Low/High

NOTES:
1. Pins A_1, A_0, $\overline{CS}$, $\overline{WR}$, $\overline{RD}$, Reset.
2. Data Bus; Ports B, C.
3. Outputs open.
4. Limit output current to 4.0 mA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAPACITANCE

T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	Unmeasured pins returned to GND f_c = 1 MHz(5)
$C_{I/O}$	I/O Capacitance		20	pF	

NOTE:
5. Sampled not 100% tested.

A.C. CHARACTERISTICS

T_A = 0° to 70°C, V_{CC} = +5V ±10%, GND = 0V

T_A = −40°C to +85°C for Extended Temperature

BUS PARAMETERS

READ CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AR}	Address Stable Before $\overline{RD}\downarrow$	0		ns	
t_{RA}	Address Hold Time After $\overline{RD}\uparrow$	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	150		ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$		120	ns	
t_{DF}	$\overline{RD}\uparrow$ to Data Floating	10	75	ns	
t_{RV}	Recovery Time between $\overline{RD}/\overline{WR}$	200		ns	

WRITE CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AW}	Address Stable Before $\overline{WR}\downarrow$	0		ns	
t_{WA}	Address Hold Time After $\overline{WR}\uparrow$	20		ns	Ports A & B
		20		ns	Port C
t_{WW}	$\overline{WR}$ Pulse Width	100		ns	
t_{DW}	Data Setup Time Before $\overline{WR}\uparrow$	100		ns	
t_{WD}	Data Hold Time After $\overline{WR}\uparrow$	30		ns	Ports A & B
		30		ns	Port C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OTHER TIMINGS

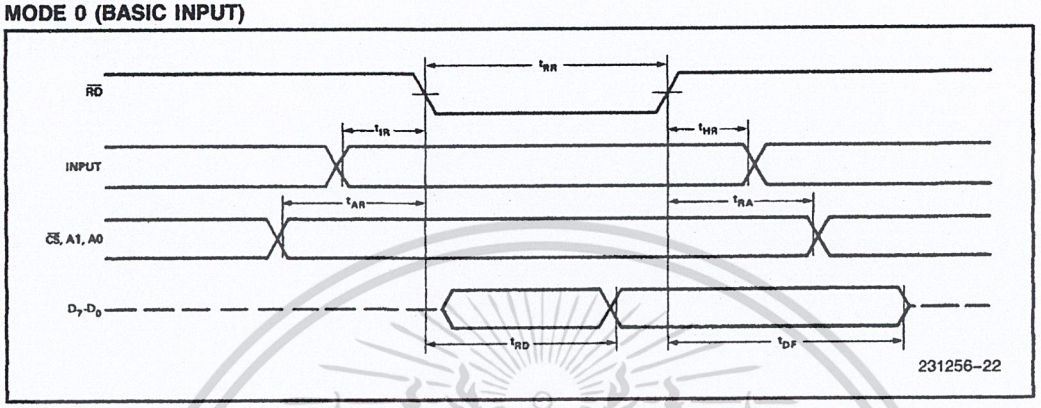
Symbol	Parameter	82C55A-2		Units Conditions	Test
		Min	Max		
t_{WB}	$\overline{WR}$ = 1 to Output		350	ns	
t_{IR}	Peripheral Data Before $\overline{RD}$	0		ns	
t_{HR}	Peripheral Data After $\overline{RD}$	0		ns	
t_{AK}	$\overline{ACK}$ Pulse Width	200		ns	
t_{ST}	$\overline{STB}$ Pulse Width	100		ns	
t_{PS}	Per. Data Before $\overline{STB}$ High	20		ns	
t_{PH}	Per. Data After $\overline{STB}$ High	50		ns	
t_{AD}	$\overline{ACK}$ = 0 to Output		175	ns	
t_{KD}	$\overline{ACK}$ = 1 to Output Float	20	250	ns	
t_{WOB}	$\overline{WR}$ = 1 to $\overline{OBF}$ = 0		150	ns	
t_{AOB}	$\overline{ACK}$ = 0 to $\overline{OBF}$ = 1		150	ns	
t_{SIB}	$\overline{STB}$ = 0 to IBF = 1		150	ns	
t_{RIB}	$\overline{RD}$ = 1 to IBF = 0		150	ns	
t_{RIT}	$\overline{RD}$ = 0 to INTR = 0		200	ns	
t_{SIT}	$\overline{STB}$ = 1 to INTR = 1		150	ns	
t_{AIT}	$\overline{ACK}$ = 1 to INTR = 1		150	ns	
t_{WIT}	$\overline{WR}$ = 0 to INTR = 0		200	ns	see note 1
t_{RES}	Reset Pulse Width	500		ns	see note 2

NOTE:
1. INTR ↑ may occur as early as $\overline{WR}$ ↓.
2. Pulse width of initial Reset pulse after power on must be at least 50 μSec. Subsequent Reset pulses may be 500 ns minimum. The output Ports A, B, or C may glitch low during the reset pulse but all port pins will be held at a logic "one" level after the reset pulse.

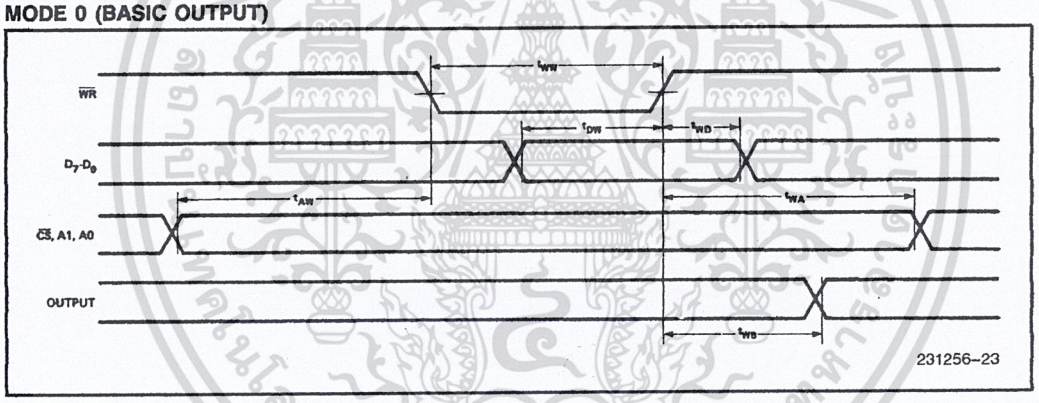
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

MODE 0 (BASIC INPUT)



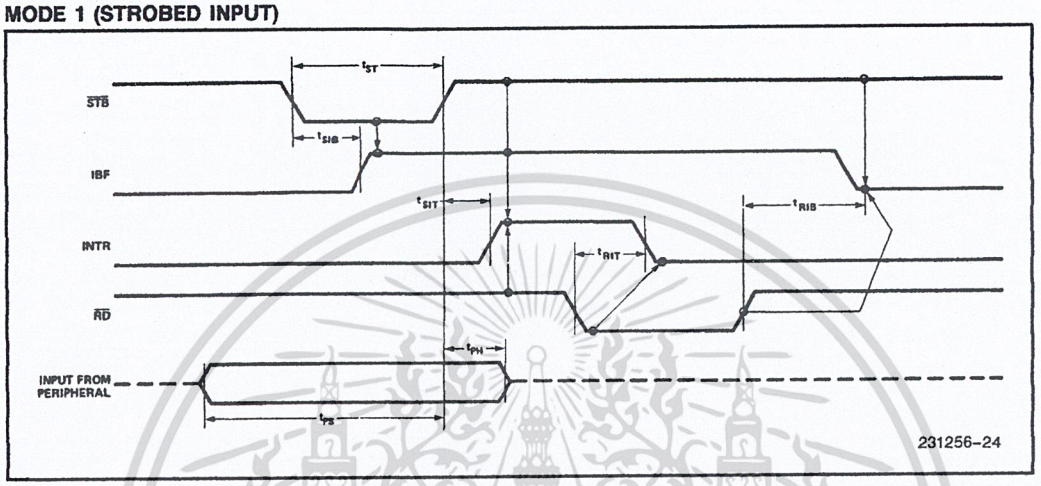
MODE 0 (BASIC OUTPUT)



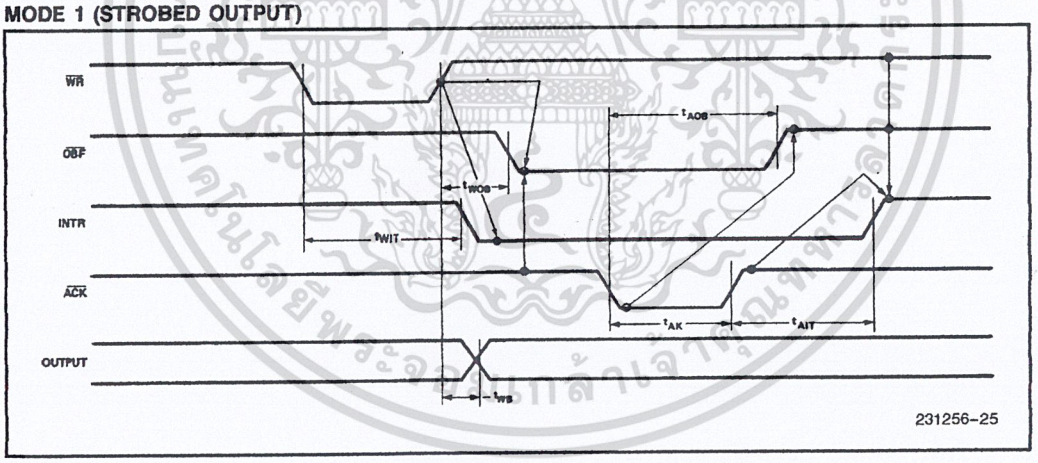
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



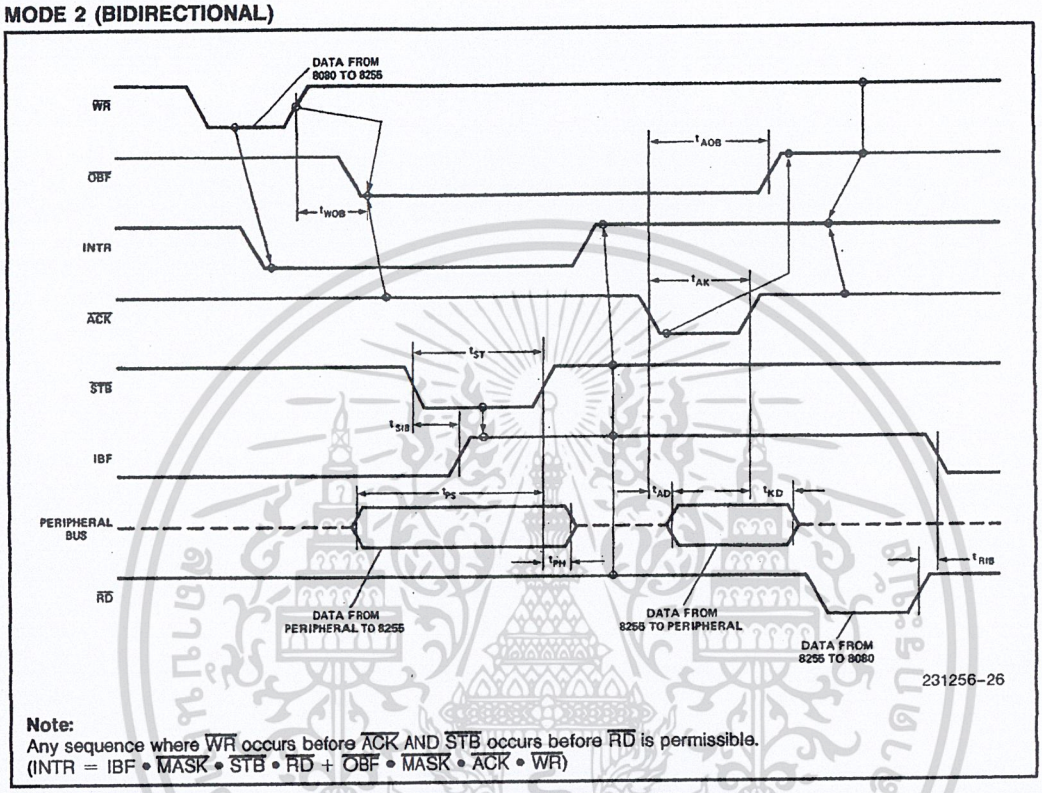
MODE 1 (STROBED OUTPUT)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

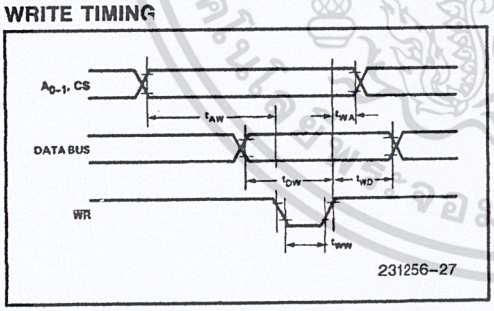
WAVEFORMS (Continued)

MODE 2 (BIDIRECTIONAL)

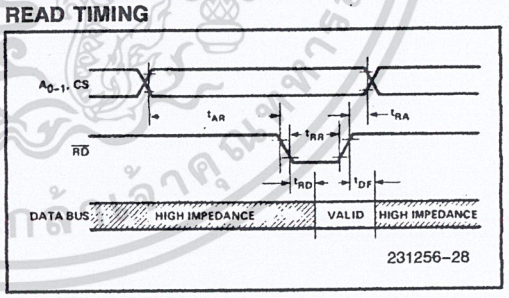


Note:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ AND $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

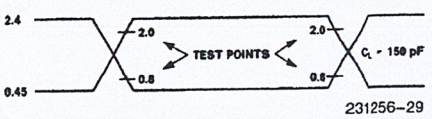
WRITE TIMING



READ TIMING

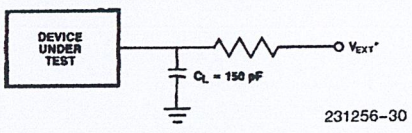


A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing Inputs Are Driven At 2.4V For A Logic 1 And 0.45V For A Logic 0 Timing Measurements Are Made At 2.0V For A Logic 1 And 0.8 For A Logic 0.

A.C. TESTING LOAD CIRCUIT



*V_{EXT} Is Set At Various Voltages During Testing To Guarantee The Specification. C_L Includes Jig Capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

SN54/74LS138

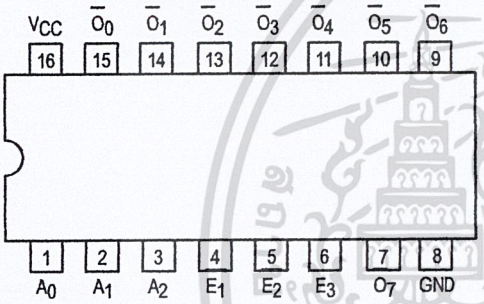
1-OF-8 DECODER/ DEMULTIPLEXER

LOW POWER SCHOTTKY

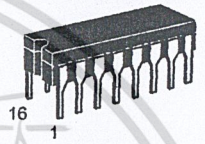
The LSTTL/MSI SN54/74LS138 is a high speed 1-of-8 Decoder/ Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Demultiplexing Capability
- Multiple Input Enable for Easy Expansion
- Typical Power Dissipation of 32 mW
- Active Low Mutually Exclusive Outputs
- Input Clamp Diodes Limit High Speed Termination Effects

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08



D SUFFIX
SOIC
CASE 751B-03

PIN NAMES

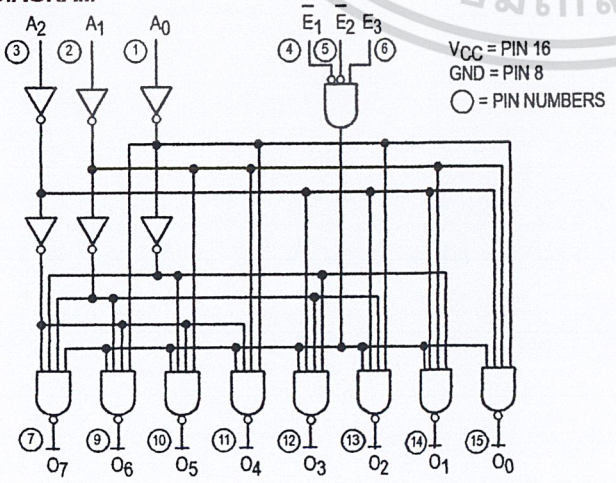
		LOADING (Note a) HIGH	LOW
$\overline{A}_0$–$\overline{A}_2$	Address Inputs	0.5 U.L.	0.25 U.L.
E_1, E_2	Enable (Active LOW) Inputs	0.5 U.L.	0.25 U.L.
E_3	Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
$\overline{O}_0$–$\overline{O}_7$	Active LOW Outputs (Note b)	10 U.L.	5 (2.5) U.L.

NOTES:
a) 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

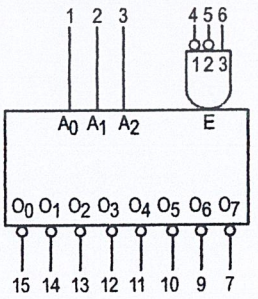
ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

LOGIC DIAGRAM



LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONAL DESCRIPTION

The LS138 is a high speed 1-of-8 Decoder/Demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs (A_0, A_1, A_2) and when enabled provides eight mutually exclusive active LOW Outputs ($\overline{O_0}$–$\overline{O_7}$). The LS138 features three Enable inputs, two active LOW ($\overline{E_1}$, $\overline{E_2}$) and one active HIGH (E_3). All outputs will be HIGH unless $\overline{E_1}$ and $\overline{E_2}$ are LOW and E_3 is HIGH. This multiple enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four LS138s and one inverter. (See Figure a.)

The LS138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
$\overline{E_1}$	$\overline{E_2}$	E_3	A_0	A_1	A_2	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$	$\overline{O_4}$	$\overline{O_5}$	$\overline{O_6}$	$\overline{O_7}$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

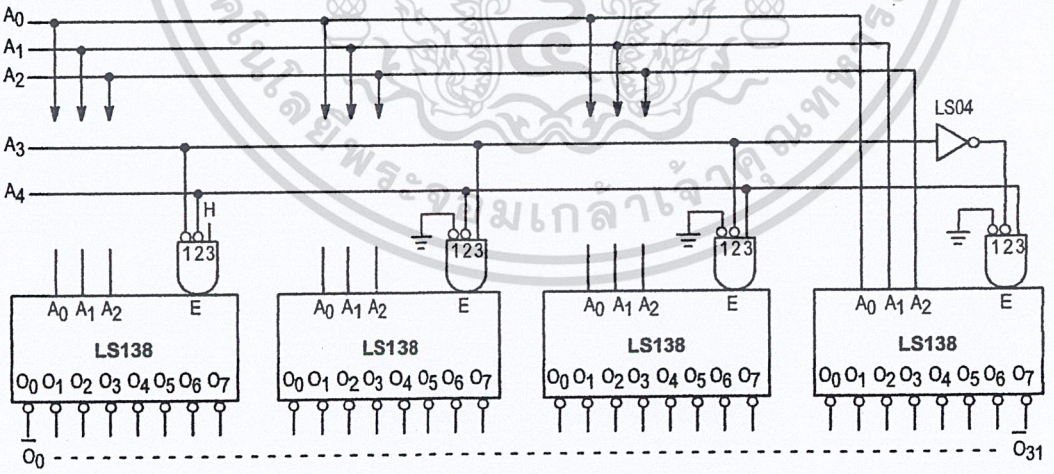


Figure a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Typ	Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	µA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current (Note 1)		−20		−100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				10	mA	V_{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Levels of Delay	Limits Min	Typ	Max	Unit	Test Conditions
t_{PLH} t_{PHL}	Propagation Delay Address to Output	2 2		13 27	20 41	ns	V_{CC} = 5.0 V C_L = 15 pF
t_{PLH} t_{PHL}	Propagation Delay Address to Output	3 3		18 26	27 39	ns	
t_{PLH} t_{PHL}	Propagation Delay E_1 or E_2 Enable to Output	2 2		12 21	18 32	ns	
t_{PLH} t_{PHL}	Propagation Delay E_3 Enable to Output	3 3		17 25	26 38	ns	

AC WAVEFORMS

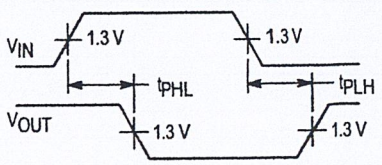


Figure 1

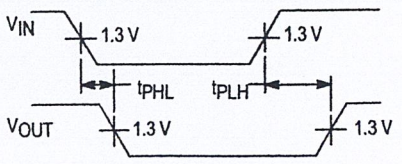


Figure 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 8K bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8 bit Internal RAM
- 32 Programmable I/O Lines
- Three 16 bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

8 bit Microcontroller with 8K bytes Flash

AT89S8252
Preliminary

Description

The AT89S8252 is a low-power, high-performance CMOS 8 bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8 bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

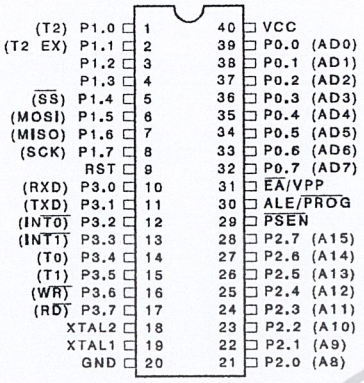
The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16 bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a

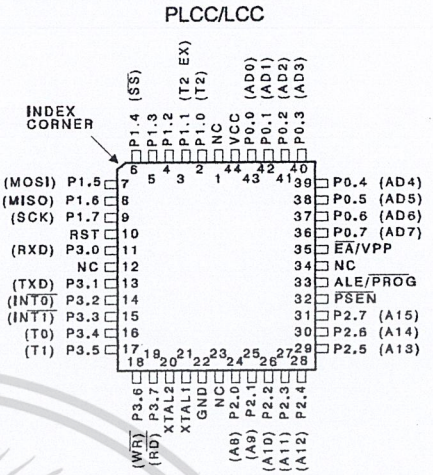
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

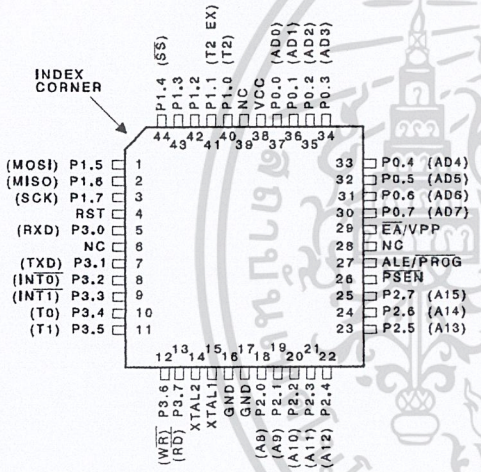
PDIP/Cerdip



PLCC/LCC



PQFP/TQFP



Pin Description

Vcc

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8 bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

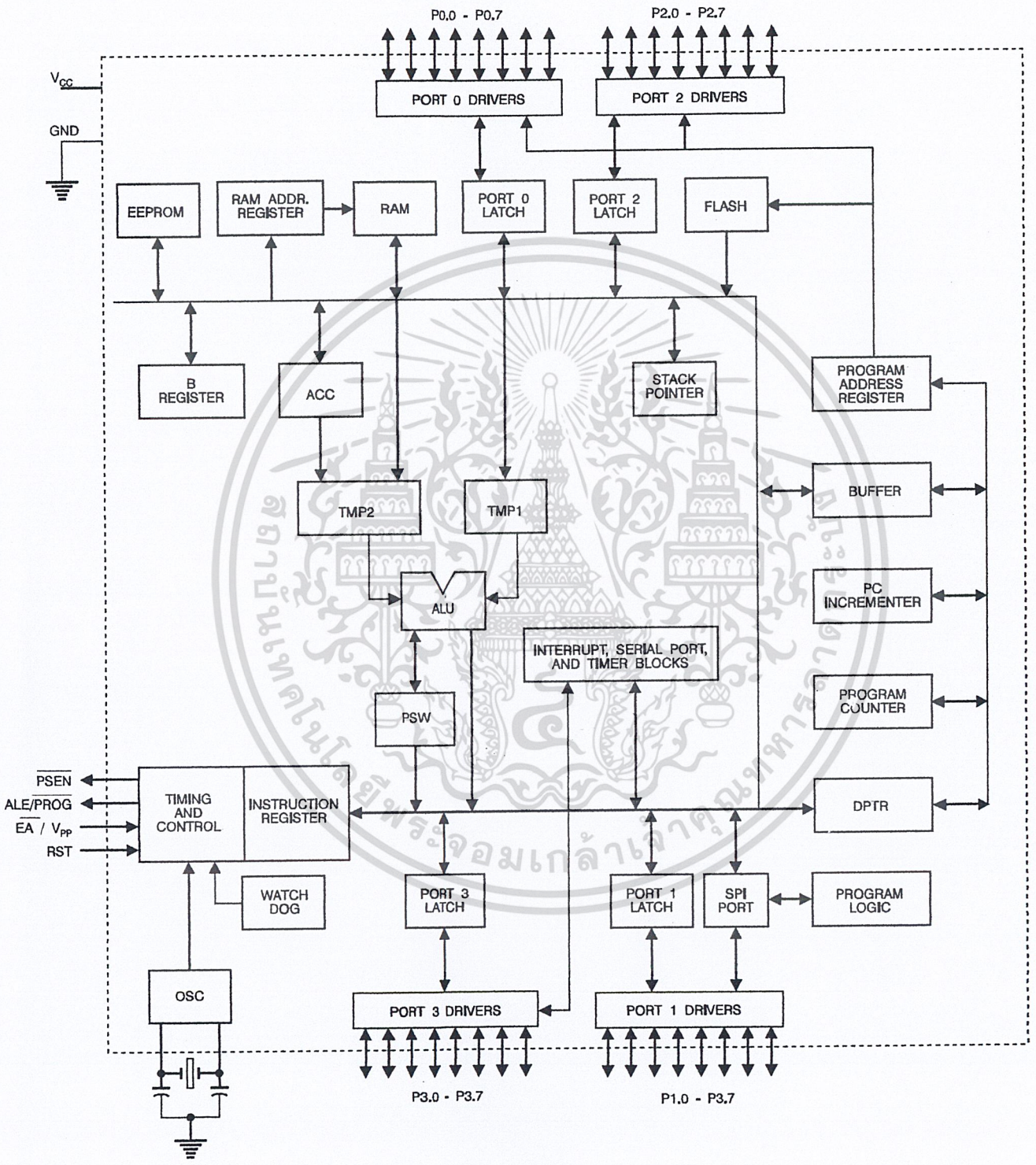
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8 bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups. *(continued)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8 bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16 bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8 bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8 bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcrontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke

(continued)

Table 1. AT89S8252 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXXX0	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00000000						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers (Continued)

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16 bit auto-reload mode.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3) . The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WOCL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H — Reset Value = 0000 0000B

Bit	7	6	5	4	3	2	1	0
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to '0', the watchdog timer has a nominal period of 16 ms. When all three bits are set to '1', the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to '1' before initiating byte write to on-chip EEPROPM with the MOVX instruction. User software should set this bit to '0' after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.
WDTRST RDY/$\overline{BSY}$	Watchdog Timer Reset and EEPROM Ready/$\overline{Busy}$ Flag. Each time this bit is set to '1' by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to '0' in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/$\overline{BSY}$ flag in a Read-Only mode during EEPROM write. RDY/$\overline{BSY}$ = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/$\overline{BSY}$ bit equals '0' and is automatically reset to '1' when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4. SPCR—SPI Control Register

SPCR Address = D5H — Reset Value = 0000 01XXB

Bit	7	6	5	4	3	2	1	0
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

Symbol	Function
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, Fosc., is as follows: SPR1 SPR0 SCK = Fosc. divided by 0 0 4 0 1 16 1 0 64 1 1 128

Table 5. SPSR—SPI Status Register

SPCR Address = AAH — Reset Value = 0000 0000B

Bit	7	6	5	4	3	2	1	0
	SPIF	WCOL	—	—	—	—	—	—

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6. SPDR—SPI Data Register

SPDR Address = 86H — Reset Value = unchanged

Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	7	6	5	4	3	2	1	0

Data Memory—EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

MOV 0A0H, #data

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to '0'.

The EEMWE bit in the WMCON register needs to be set to '1' before any byte location in the EEPROM can be written. User software should reset EEMWE bit to '0' if no further EEPROM write is required. EEPROM write cycles are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/$\overline{BSY}$ bit (read-only) in SFR WMCON. RDY/$\overline{BSY}$ = 0 means programming is still in progress and RDY/$\overline{BSY}$ = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period
PS2	PS1	PS0	(nominal)
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8 bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the os-

Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16 bit Auto-Reload
0	1	1	16 bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit,

Auto-Reload (Up or Down Counter)

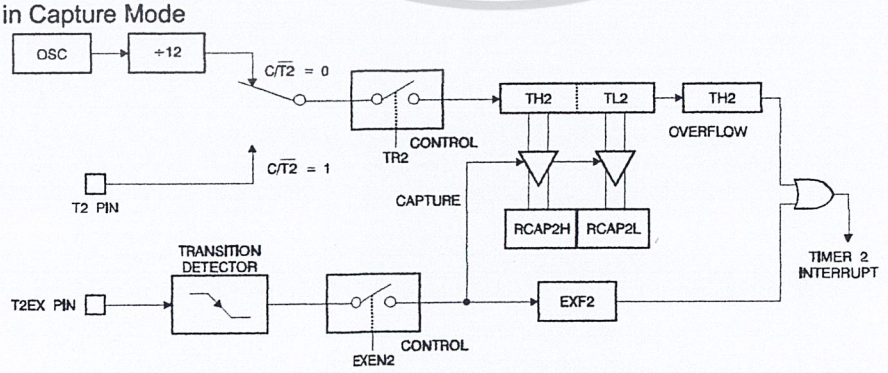
Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in

(continued)

Figure 1. Timer 2 in Capture Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto-Reload (Up or Down Counter) (Continued)

RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

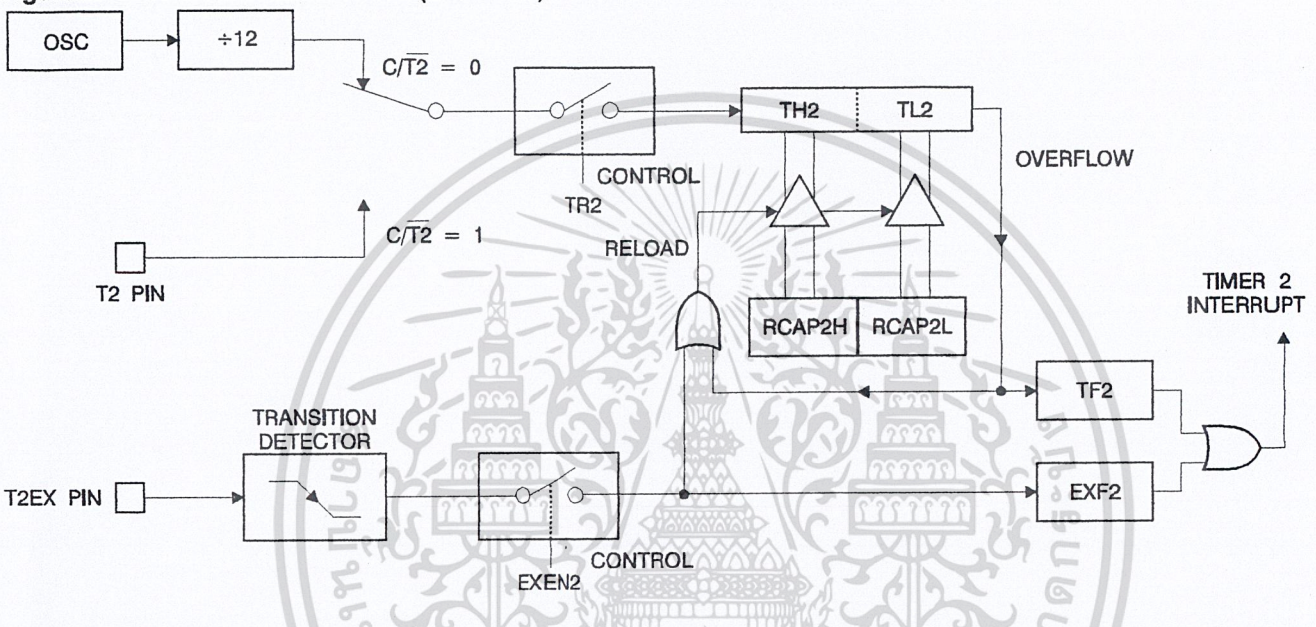


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

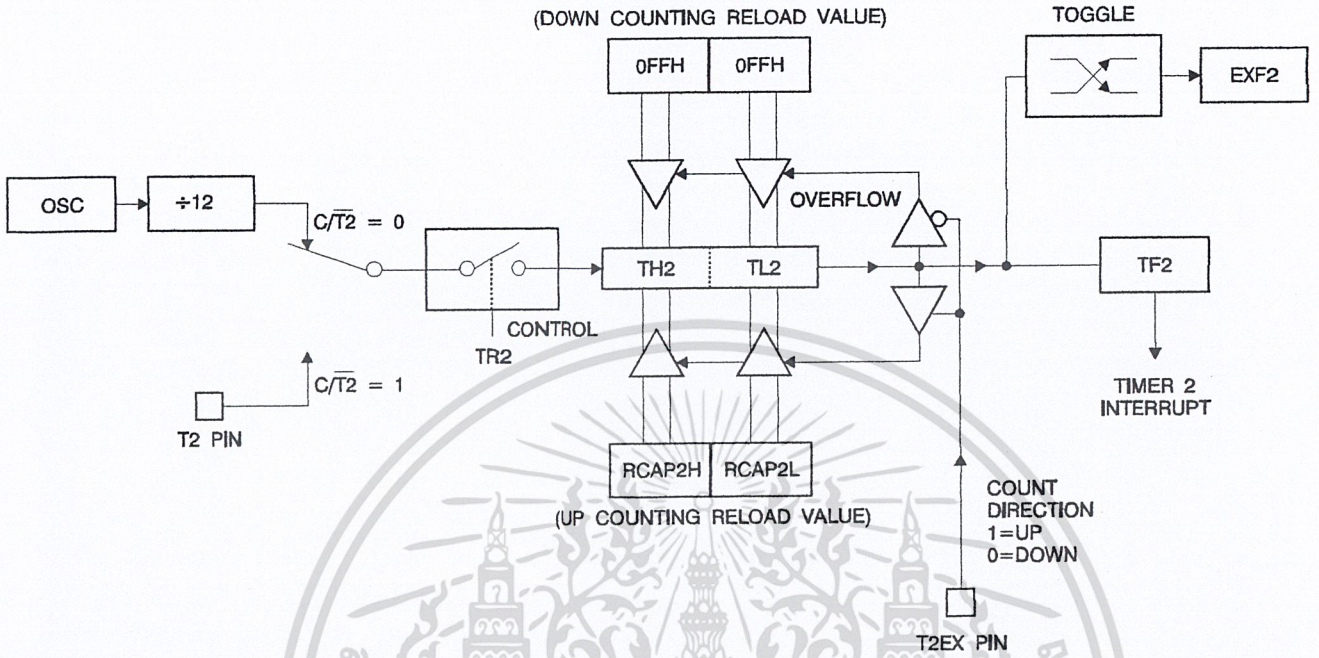
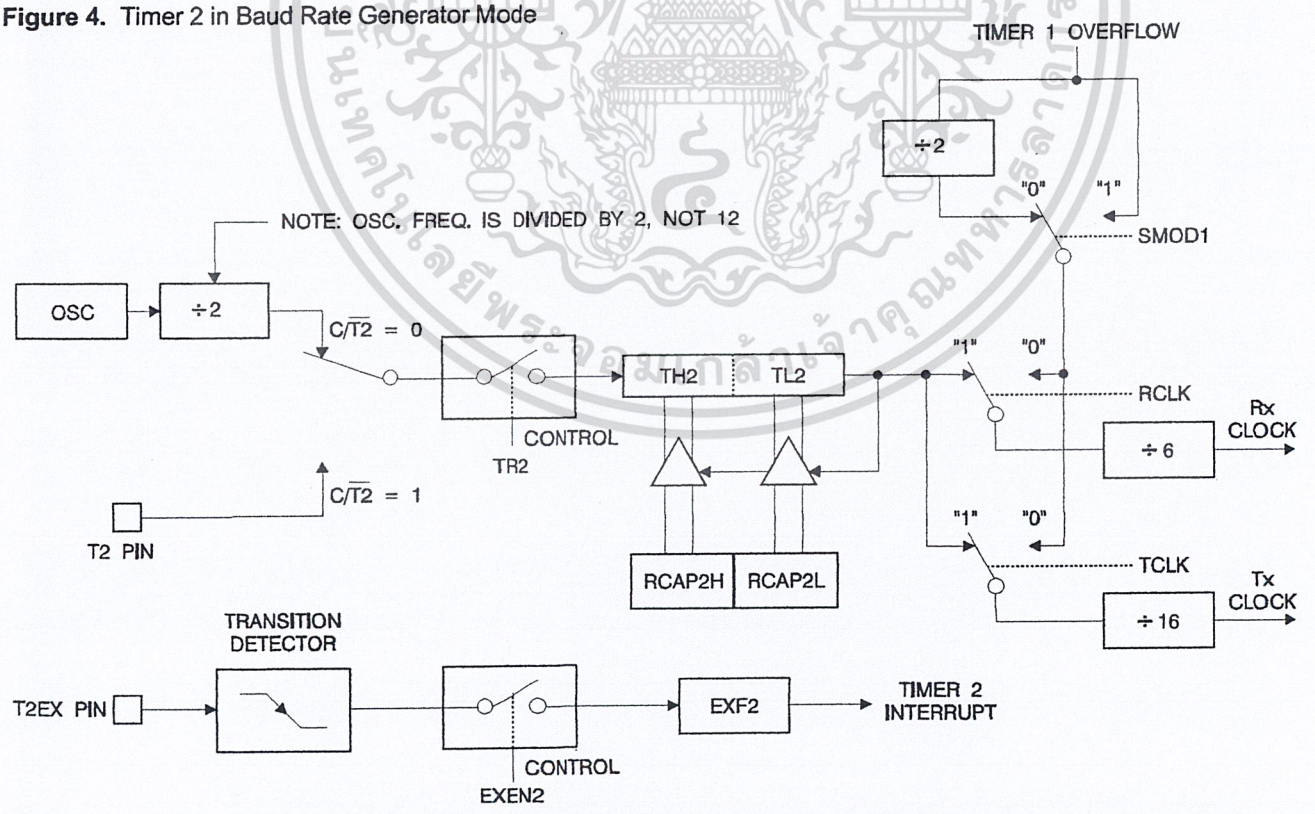


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes I and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

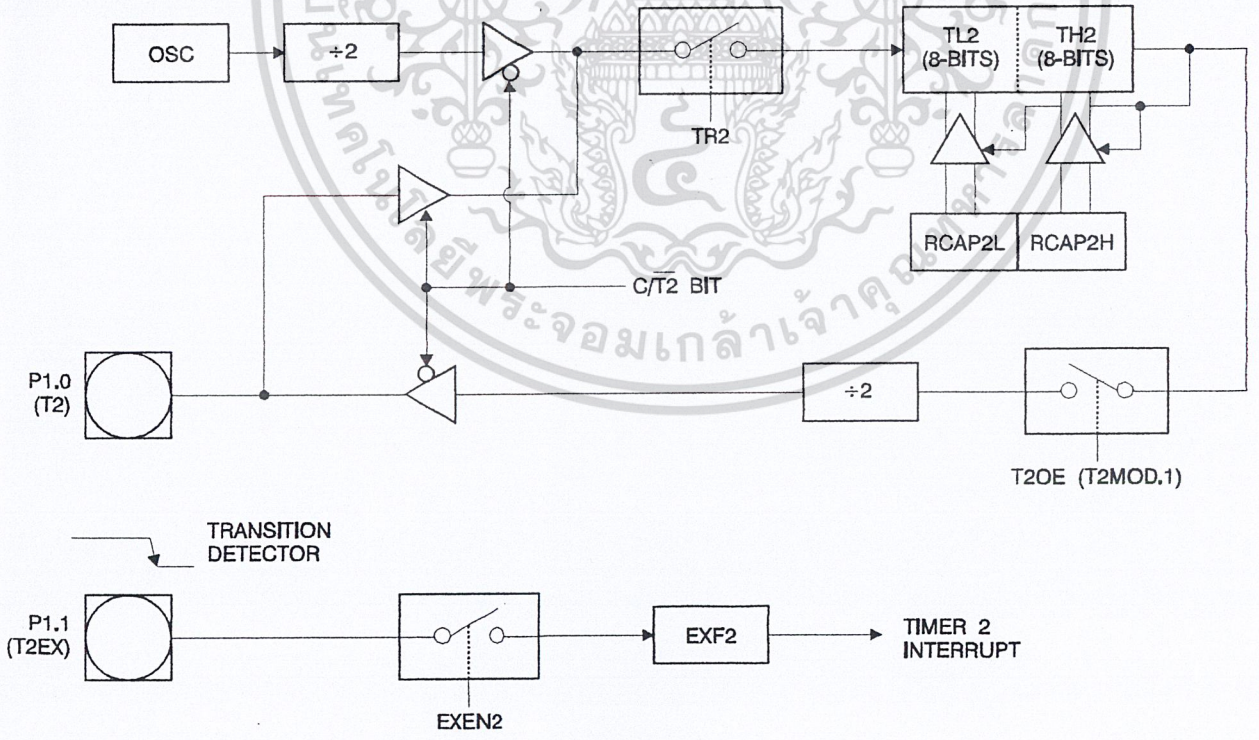
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might

Figure 5. Timer 2 in Clock-Out Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock–Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H, RCAP2L}]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out

UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

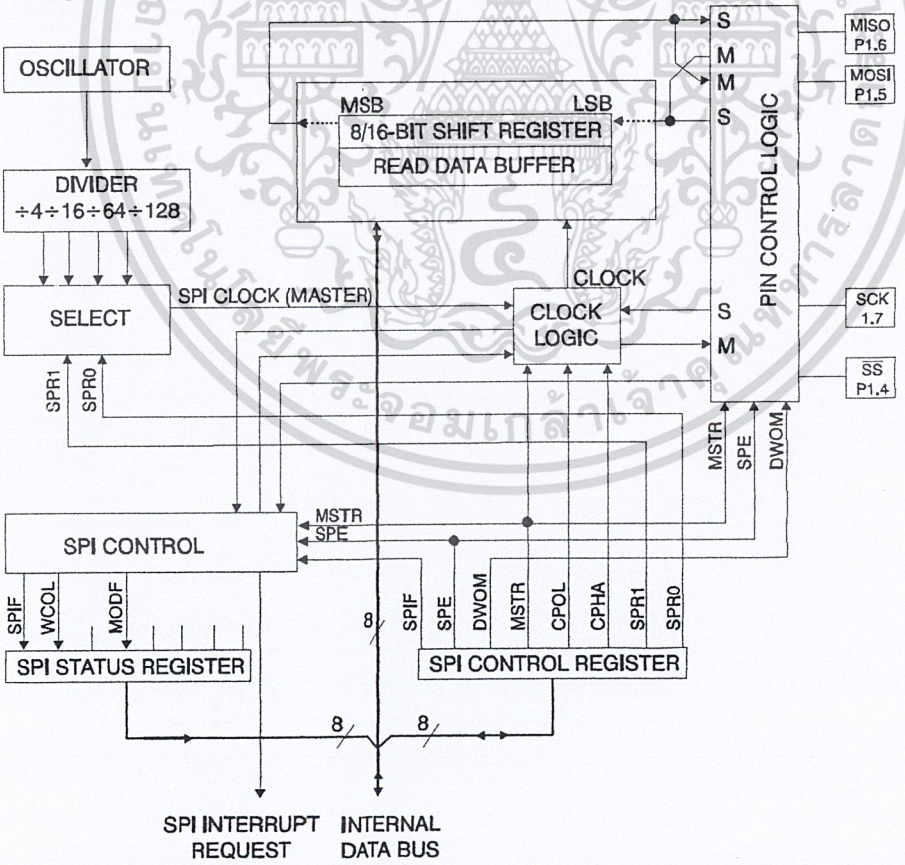
Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 6 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

(continued)

Figure 6. SPI Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Peripheral Interface (Continued)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}$/P1.4, is set low to select an individual SPI device as a slave. When $\overline{SS}$/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.6 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

Figure 7. SPI Master -Slave Interconnection

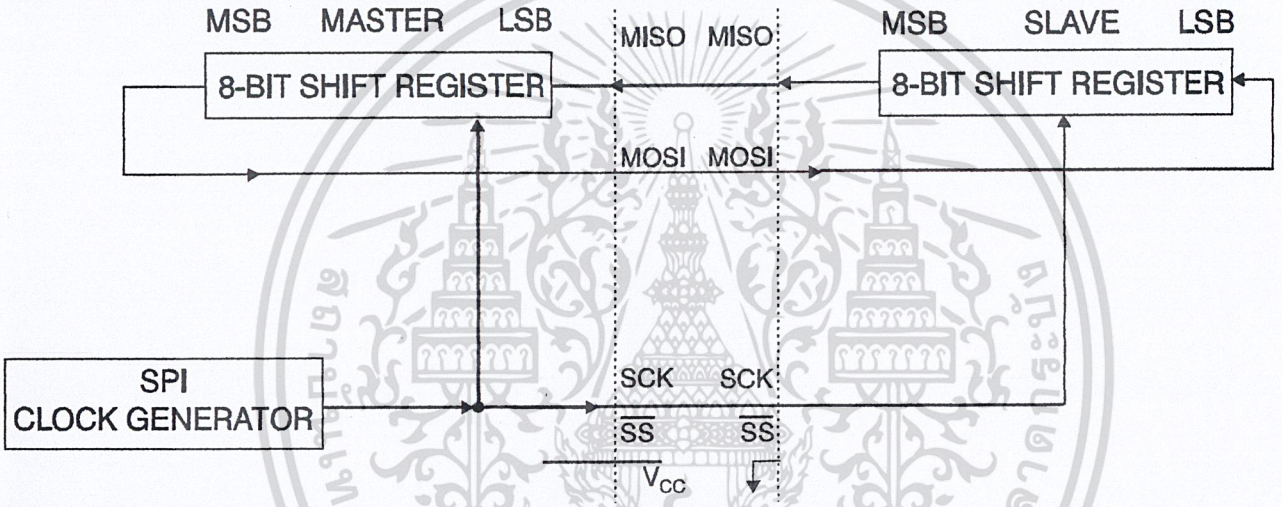
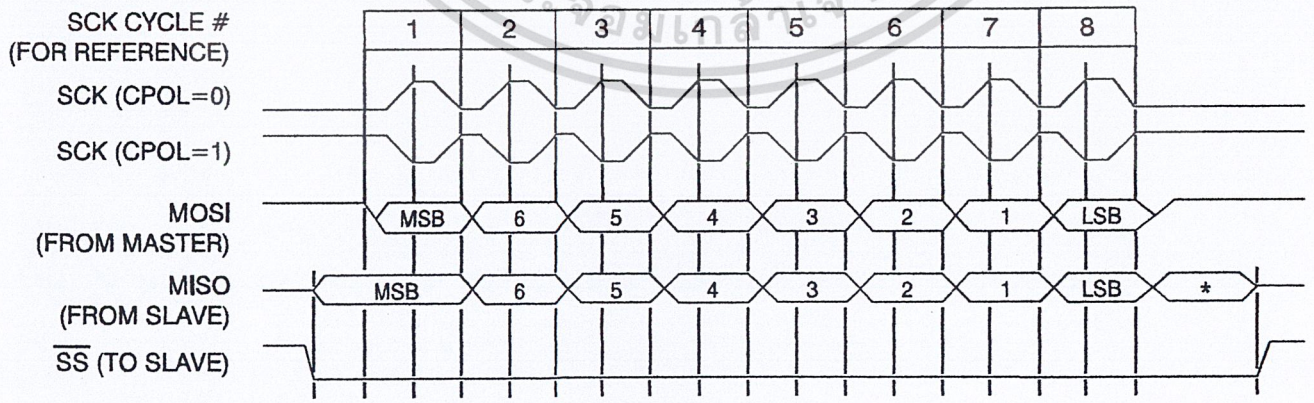


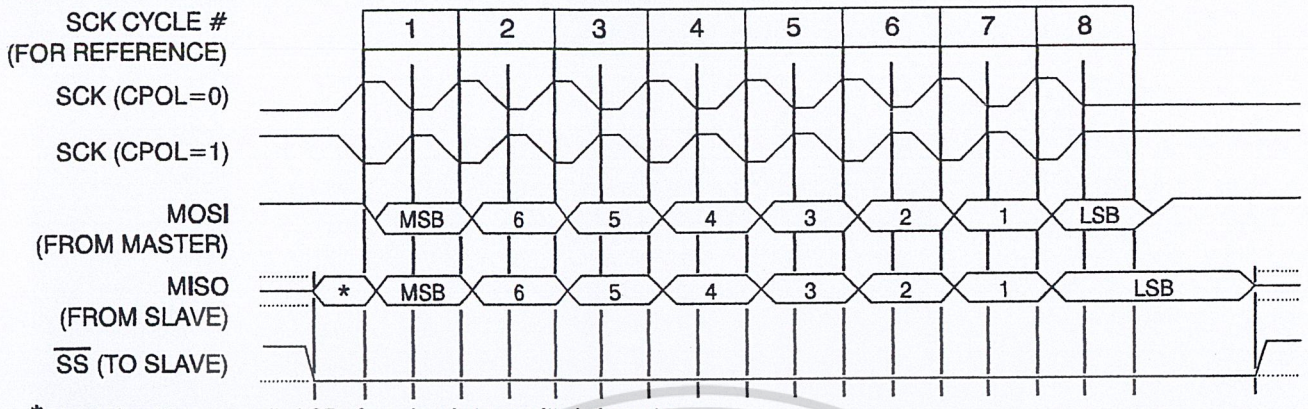
Figure 8. SPI Transfer Format with CPHA = 0



* Not defined but normally MSB of character just received.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9. SPI Transfer Format with CPHA = 1



* Not defined but normally LSB of previously transmitted character.

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, AT89C52 and AT89C55, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Table 10. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

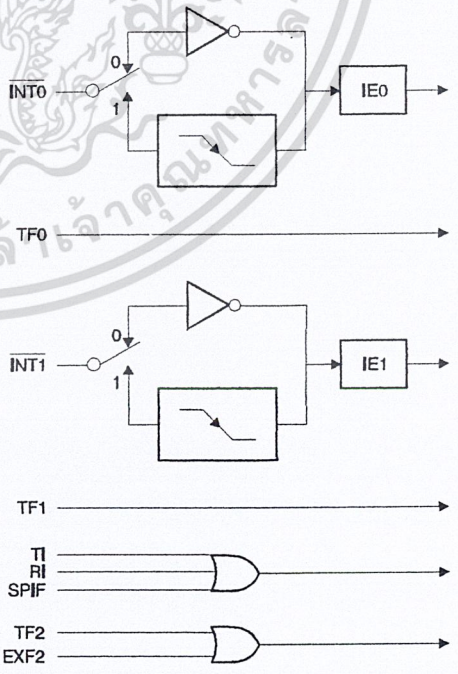
Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TFI, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Figure 10. Interrupt Sources

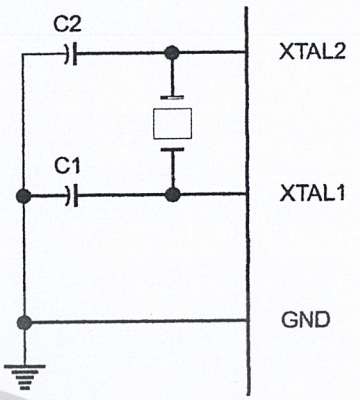


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 11. Oscillator Connections



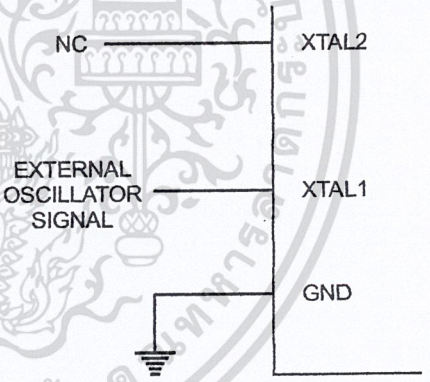
Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits its access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Figure 12. External Clock Drive Configuration



Power Down Mode

In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power

Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Lock Bit Protection Modes (1, 2)

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. EA is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

Notes: 1. U = Unprogrammed.
2. P = Programmed.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and

holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable flash PEROM Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip PEROM Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming modes. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash and EEPROM (Continued)

Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
   Apply power between V_{CC} and GND pins with all other pins floating.
   Set RST pin to 'H'.
   Apply a 4 MHz to 24 MHz clock to XTAL1 pin and wait for
   at least 10 milliseconds.
2. Set $\overline{PSEN}$ pin to 'H'
   ALE pin to 'H'
   $\overline{EA}$ pin to 'H' and all other pins to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
   Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise $\overline{EA}$/VPP to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/$\overline{PROG}$ once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to 'L' and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
   Set XTAL1 to 'L'.
   Set RST and $\overline{EA}$ pins to 'L'.
   Float all other I/O pins.
   Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

$\overline{DATA}$ Polling

The AT89S8252 features $\overline{DATA}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{DATA}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{BSY}$ output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate $\overline{BUSY}$. P3.4 is pulled High again when programming is done to indicate READY.

Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enble instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 KHz.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Downloading (Continued)

Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

   Apply power between V_{CC} and GND pins with all other pins floating.

   Set RST pin to 'H'.

   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 4 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.

Instruction Set

Instruction	Input Format		Operation
	MSB	LSB	
Programming Enable	1010 0101 xxxx	1100 0011 xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 xxxx xxxx	1100 x100 xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa low xxxx	a001 addr xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa low data	a010 addr in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa low xxxx	a101 addr xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa low data	a110 addr in	Write data to Data memory location at selected address.
Write Lock Bits	1010 LLLx BBB 123 xxxx	1100 x111 xxxx	Write lock bits. Set LB1, LB2 or LB3 = '0' to program lock bits.

Notes: 1. $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
2. 'aaaaa' = high order address.
3. 'x' = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash and EEPROM Parallel Programming Modes

Mode	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h(1)	h	x	(See detail timing)					
Chip Erase	H	L	⎍(2)	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L	⎍	12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L	⎍	12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	⎍(2)	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	⎍(2)	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

Notes: 1. 'h' = weakly pulled 'High' internally.
2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
3. P3.4 is pulled Low during programming to indicate RDY/$\overline{BSY}$.
4. 'X' = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 13. Programming the Flash Memory

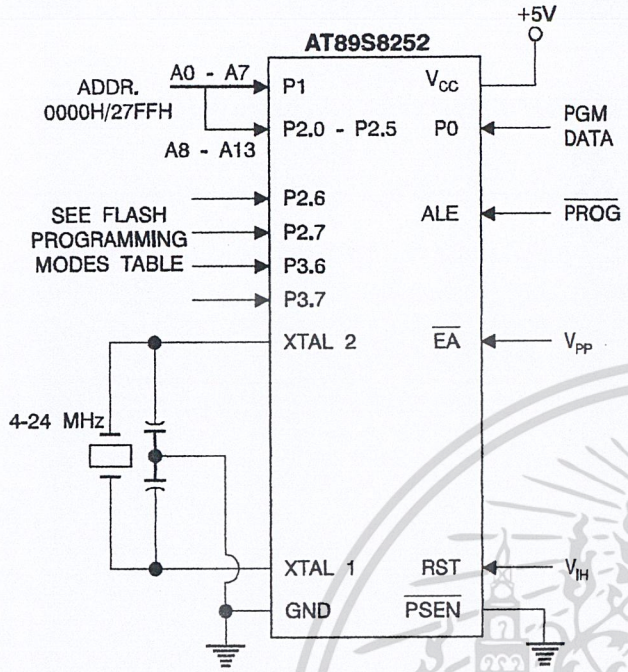


Figure 14. Verifying the Flash Memory

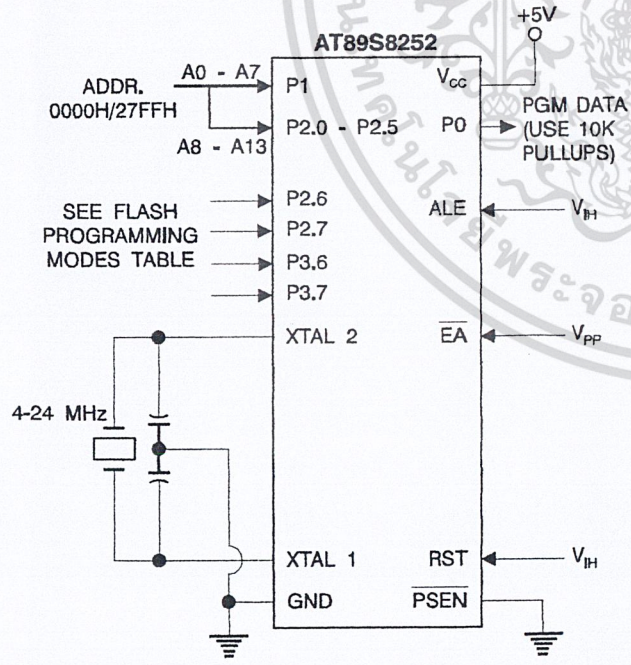
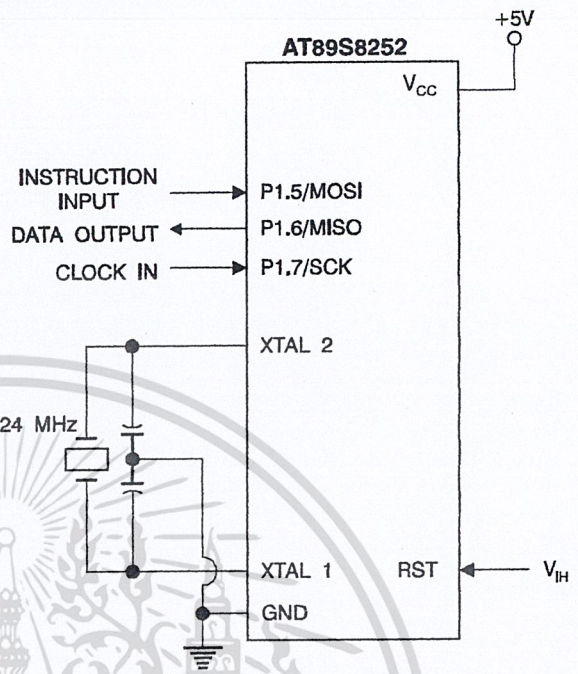


Figure 15. Flash/EEPROM Serial Downloading



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

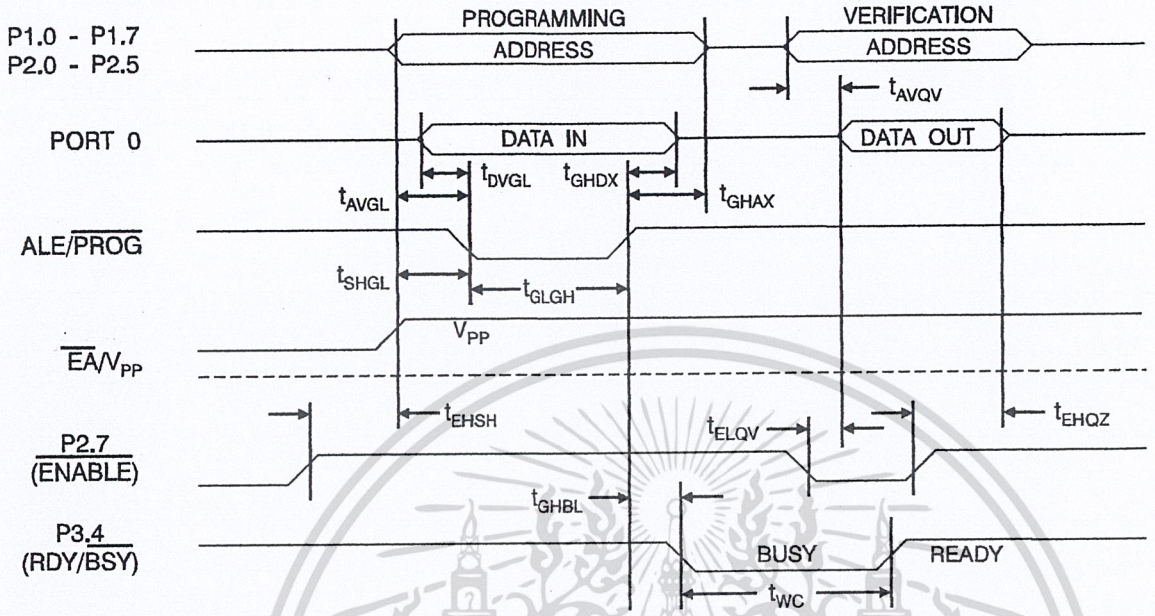
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0V ± 10%

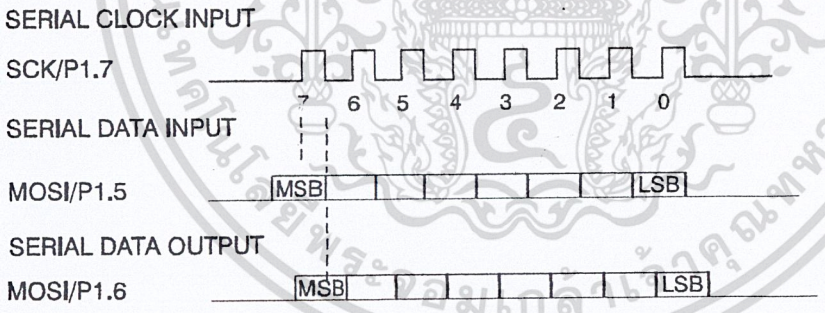
Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



Serial Downloading Waveforms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.7V to 6.0V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage (1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.5	V
V_{OL1}	Output Low Voltage (1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.5	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V		-650	µA
I_{LI}	Input Leakage Current (Port 0, EA)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	kΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode (2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8 bit port:
Port 0: 26 mA
Ports 1,2,3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

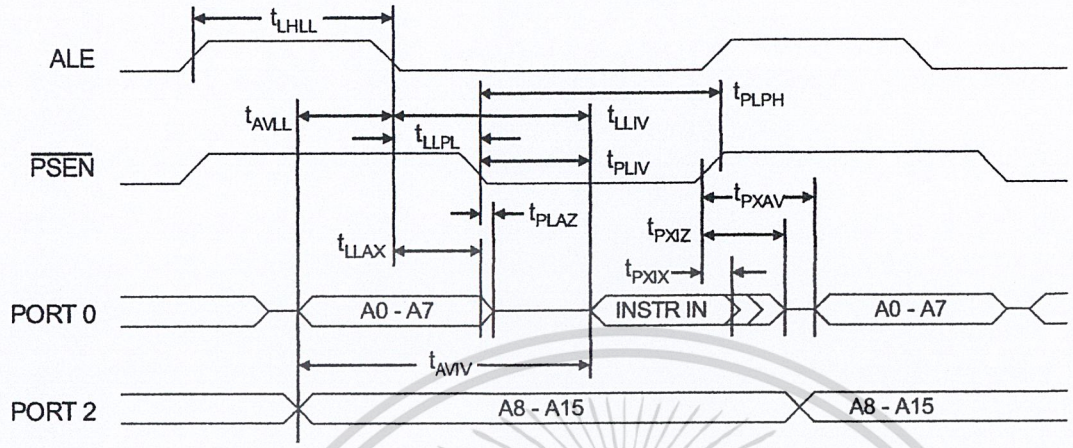
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

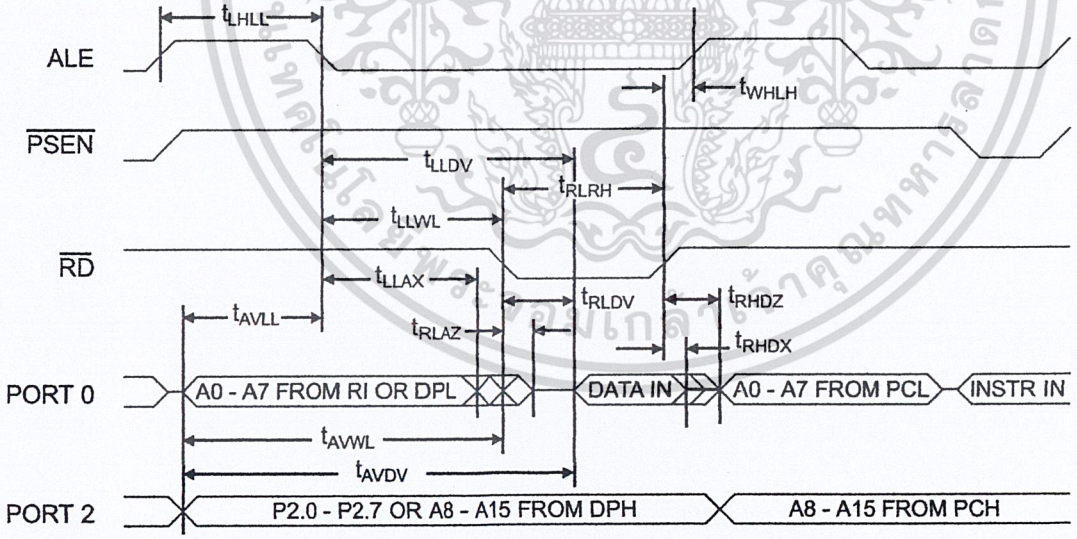
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	28		$t_{CLCL} - 13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL} - 20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL} - 65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL} - 13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL} - 20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL} - 45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL} - 10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL} - 8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL} - 55$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL} - 100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL} - 100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL} - 90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL} - 28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL} - 75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL} - 20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL} - 120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL} - 20$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL} - 20$	$t_{CLCL} + 25$	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

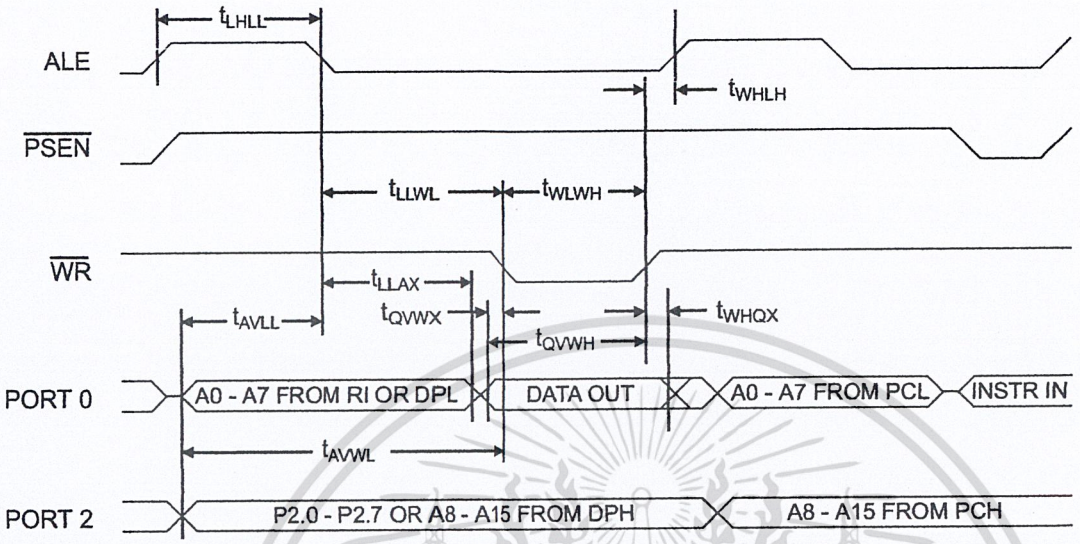


External Data Memory Read Cycle

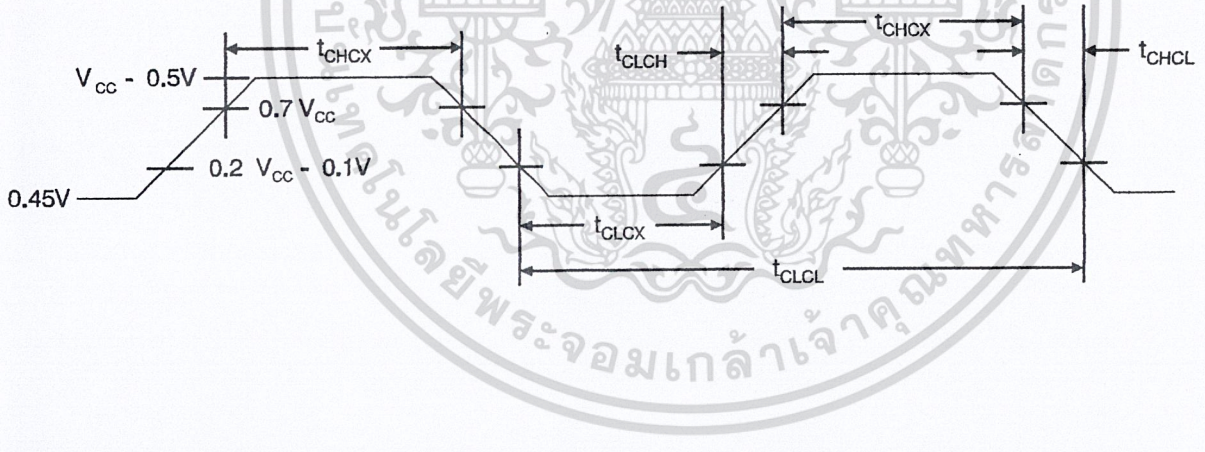


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	VCC = 4.0V to 6.0V		Units	VCC = 2.7V to 6.0V	
		Min	Max		Min	Max
1/tCLCL	Oscillator Frequency	0	24	MHz	0	12
tCLCL	Clock Period	41.6		ns	83.3	
tCHCX	High Time	15		ns	30	
tCLCX	Low Time	15		ns	30	
tCLCH	Rise Time		20	ns		20

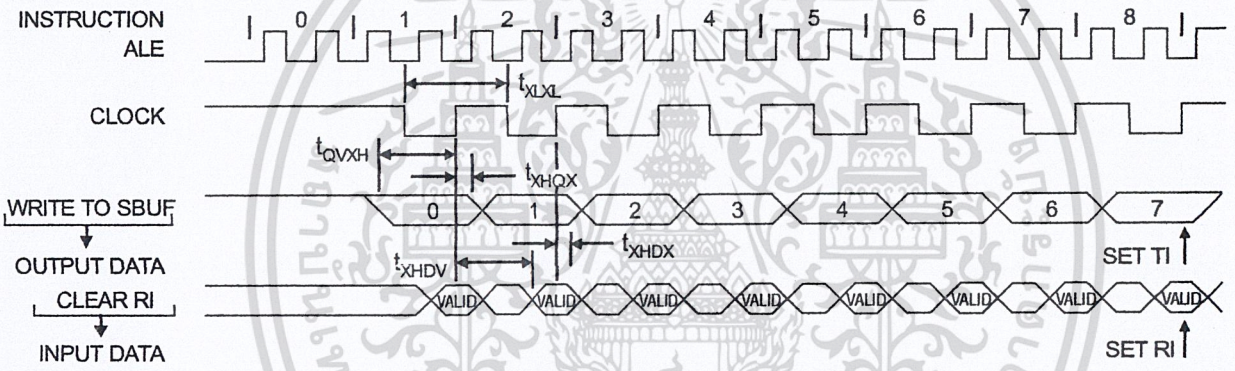
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

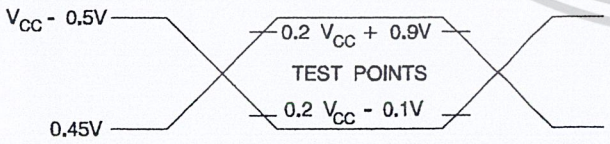
The values in this table are valid for V_{CC} = 2.7V to 6V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$ - 133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$ - 33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$ - 133	ns

Shift Register Mode Timing Waveforms

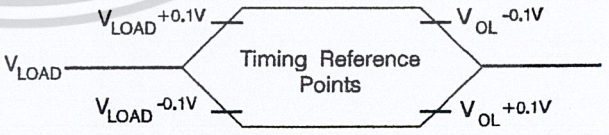


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

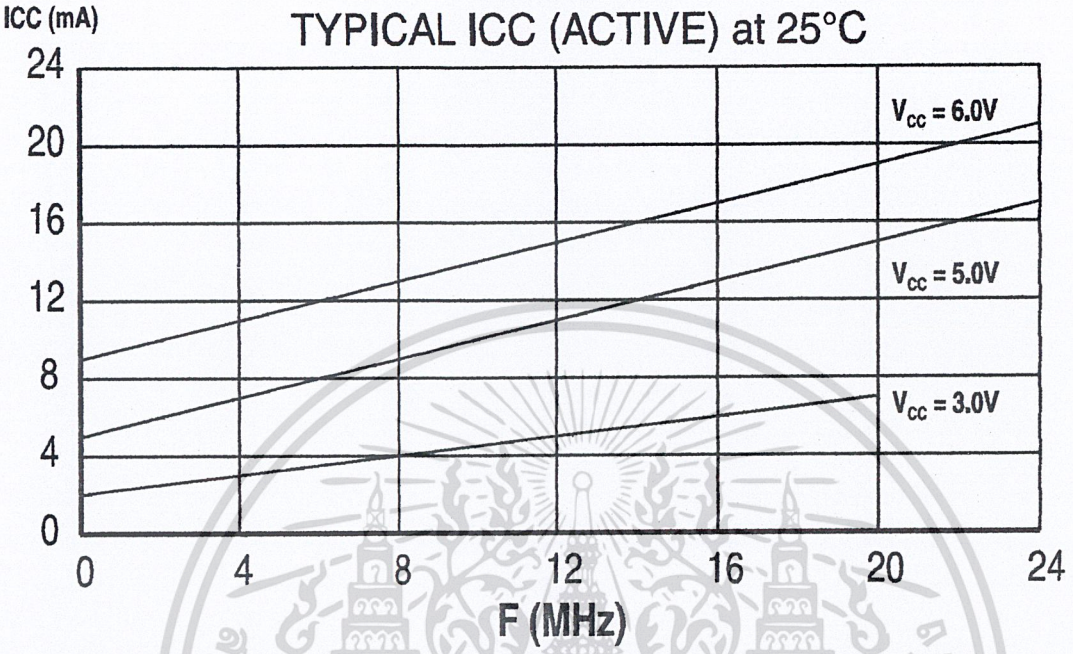
Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

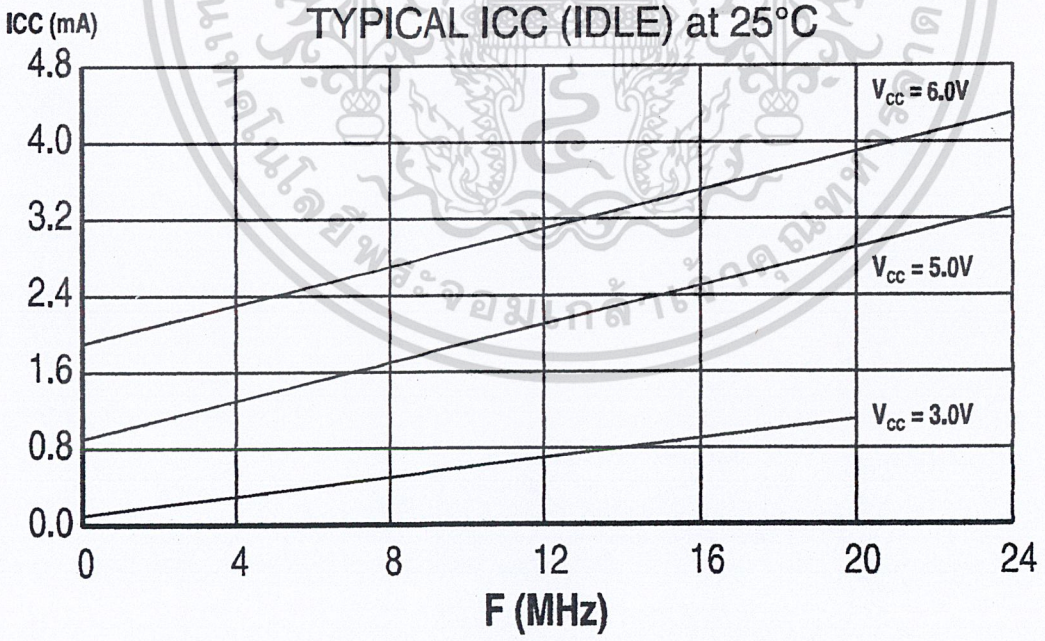
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL ICC (ACTIVE) at 25°C



AT89S8252

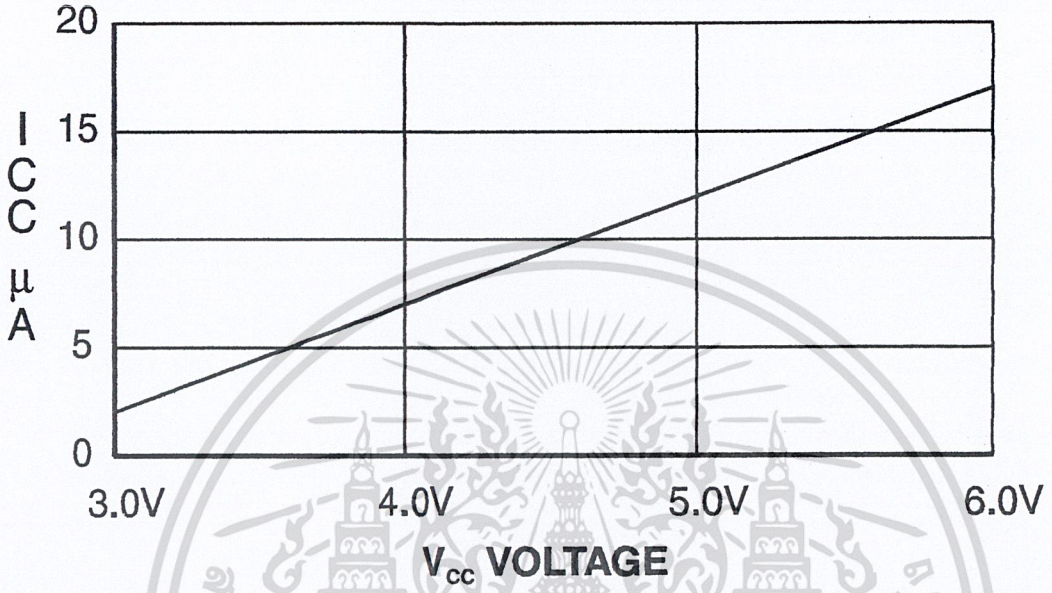
TYPICAL ICC (IDLE) at 25°C



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



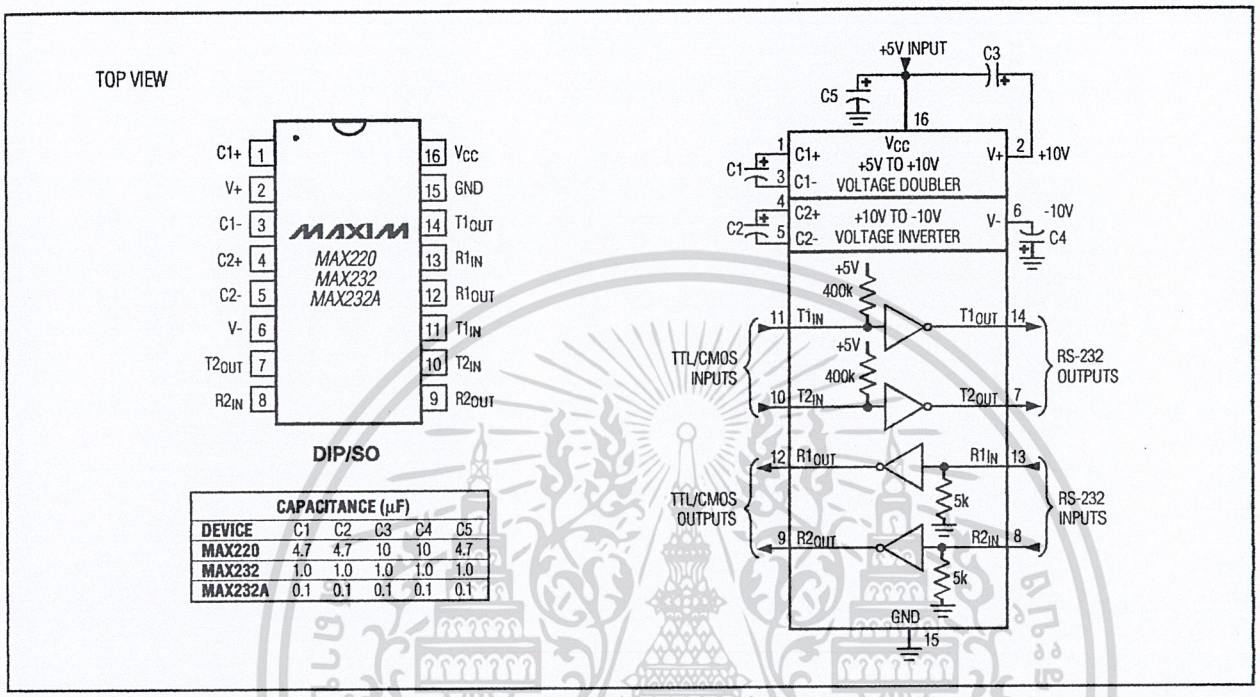
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89S852-12AC AT89S852-12JC AT89S852-12PC AT89S852-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-12AI AT89S852-12JI AT89S852-12PI AT89S852-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
16	4.0V to 6.0V	AT89S852-16AA AT89S852-16JA AT89S852-16PA AT89S852-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	4.0V to 6.0V	AT89S852-20AC AT89S852-20JC AT89S852-20PC AT89S852-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-20AI AT89S852-20JI AT89S852-20PI AT89S852-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	4.0V to 6.0V	AT89S852-24AC AT89S852-24JC AT89S852-24PC AT89S852-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89S852-24AI AT89S852-24JI AT89S852-24PI AT89S852-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

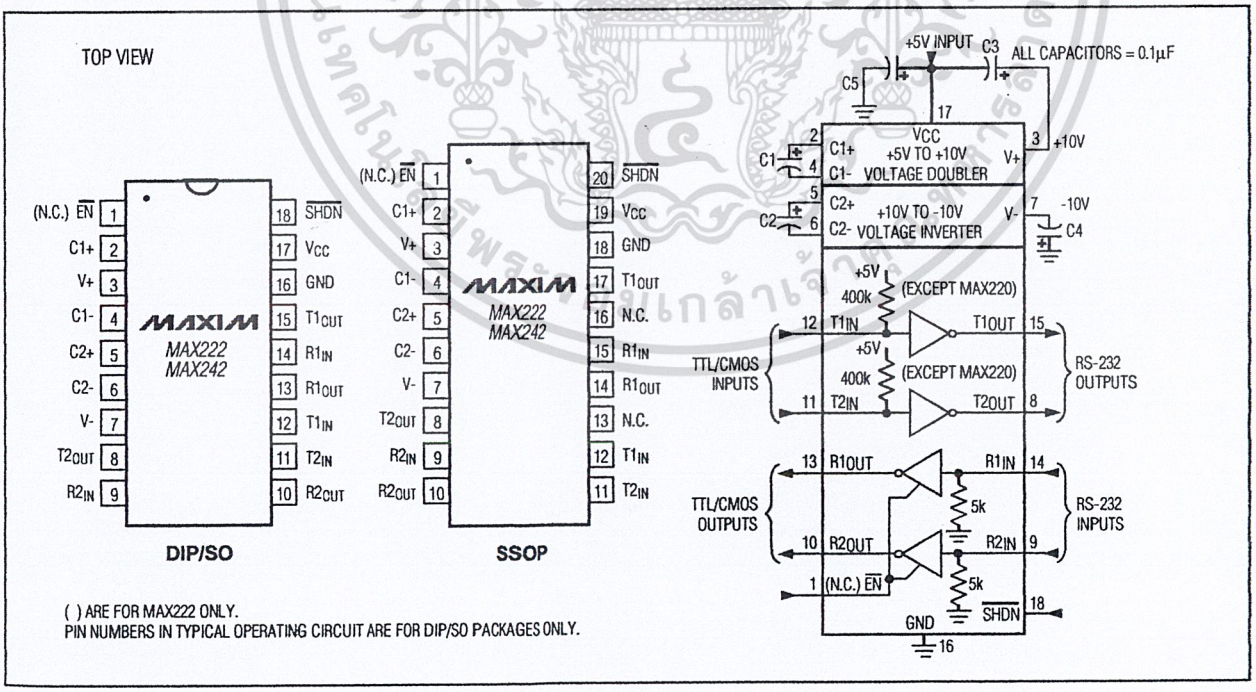


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

สมยศ จุณณะปิยะ,"การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51"
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,กรุงเทพฯ 2543

วรพจน์กรแก้ววัฒนกุล,ชัยวัฒน์ ลิ้มพรจิตรวิไล,Microcontroller Theory&Practical Approach

กฤษดา ใจเย็น,อรรถพล บุญยะโภคา, ชัยวัฒน์ ลิ้มพรจิตรวิไล,เรียนรู้และปฏิบัติการเชื่อมต่อ
คอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม

ฉัททวุฒิพีชผล,พิชิต สันติกุลานนท์,คู่มือเรียน VISUAL BASIC 6,กรุงเทพฯ
โปรวิชั่น,2542,438หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้