สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โมดูลอินเตอร์เฟสกับโปรแกรมVisual Basic

4

นายอารยะ แซ่ตั้ง

ปพ.
อ651ม
2550

เลขหมู่.....
เลขทะเบียน 81615
วัน,เดือน,ปี 19 มิ.ย. 2551

b. 11933628
i.....

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interfacing Module for Visual Basic Programming

Mr.Araya Saetung

A Special Project Submitted in Partial Fulfillment of Requirement for the Degree

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute Technology Ladkrabang

Academic Year 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง โมดูลอินเตอร์เฟสกับโปรแกรมVisual Basic

นักศึกษา นายอารยะ แซ่ตั้ง

ภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สาขาวิชา ฟิสิกส์ประยุกต์ – เครื่องมือวิทยาศาสตร์และอุตสาหกรรม

อาจารย์ที่ปรึกษา รองศาสตราจารย์วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ ดร. ประธาน บุรณะศิริ	
กรรมการ อ. ธรรมรัตน์ แต่งตั้ง	
กรรมการ อ. กาจปัญญา สุวรรณสุโข	
กรรมการที่ปรึกษา รศ. วิชิต ศิริโชติ	

.....

(รองศาสตราจารย์วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ	โมดูลอินเตอร์เฟสกับโปรแกรม Visual Basic
นักศึกษา	นายอารยะ แซ่ตั้ง
ภาควิชา	ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สาขาวิชา	ฟิสิกส์ประยุกต์ เครื่องมือวิทยาศาสตร์ และอุตสาหกรรม
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	รศ.วิชิต ศิริโชติ

บทคัดย่อ

ฮาร์ดแวร์ใช้สำหรับการทดลองโดยการใช้โปรแกรม Visual Basicซึ่งจะเป็นการเชื่อมต่อระหว่าง PC hardware และ GUI Software โดยใช้ API ในการศึกษา และการพัฒนาเครื่องมือโดยใช้ 1) 8-bit dot LED, 2) DC motor, 3) Stepping motor, 4) 7-segment LED และ 5) 3x4 keypad เครื่องมือที่ใช้นี้ติดต่อกับ LPT พอร์ตขนานและพอร์ตอนุกรม โดยมีรายงานการบรรยายและ Source code ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Interfacing Module for Visual Basic Programming
Name	Mr. Araya Saetung
Department	Applied Physics **Faculty of Science**
Program	Applied Physics-Science and industry instrumentation
Academic Year	2007
Special Project Advisors	Assoc. Prof. Wichit Sirichote

ABSTRACT

The hardware modules used for experimenting with Visual Basic Programming have been developed. The linking between PC's hardware and GUI software through the use API has been studied. The developed modules include 1) 8-bit dot LED, 2) DC motor, 3) Stepping motor, 4) 7-segment LED and 5) 3x4 keypad. The module is interfaced through the LPT parallel port and the serial port. The report describes and provides the project file source code of the control program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานพิเศษนี้เสร็จลุล้วงไปได้ด้วยดี ต้องขอขอบคุณ รองศาสตราจารย์ วิชิต ศิริโชติ ที่กรุณาเป็นอาจารย์ปรึกษาโครงงานพิเศษนี้ โดยให้ความรู้ คำปรึกษา และ แบบ อย่างที่ดีใน การแก้ปัญหาต่างที่เกิดขึ้น และขอขอบคุณ พ่อ แม่ ที่ได้อบรมและคอยดูแล และขอขอบคุณ พี่เจี๋ย ที่เป็นที่ปรึกษาทางด้าน Program Visual Basic และ อาจารย์ ทุกท่านที่ให้ความรู้จนสามารถประยุกต์ใช้ความรู้ที่มีอยู่ จนทำโครงงานพิเศษนี้เสร็จลุล้วง

นาย อารยะ แซ่ตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อหา	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	จ
บทที่1 บทนำ	
1.1 ความเป็นมาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการวิจัย	1
1.4 ขั้นตอนการดำเนินงาน	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่2 ทฤษฎีและหลักการ	
2.1 บทนำ	3
2.2 โปรแกรม Visual Basic	11
2.3 หลักการส่งข้อมูลทาง Parallel port	11
2.4 ความรู้เบื้องต้นเกี่ยวกับ Stepping Motor	13
2.5 ความรู้เบื้องต้นเกี่ยวกับ Key Pad	15
2.6 ความรู้เบื้องต้นเกี่ยวกับ LED	17
2.7 ความรู้เบื้องต้นเกี่ยวกับ DC Motor	20
2.8 ความรู้เบื้องต้นเกี่ยวกับ 7-Segment	23
บทที่ 3 วิธีดำเนินการวิจัย	
3.1การออกแบบในส่วนของ ฮาร์ดแวร์	28
3.2การออกแบบในส่วนของซอฟต์แวร์	28
3.3 อุปกรณ์การทดลอง	28
3.4 วิธีการส่งข้อมูลให้ LED ติด	30
3.5การศึกษา 7- Segment และ การส่งข้อมูลให้ 7-Segment	35
3.6ศึกษาการทำงานของ Stepping Motorและการส่งข้อมูลให้ Motorทำงาน	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

3.7การส่งข้อมูลให้ มอเตอร์กระแสตรงทำงาน 45

3.8 การส่งข้อมูลให้ Key Pad ทำงาน 48

3.9 การส่งข้อมูลผ่าน Serial Port โดยใช้ AT89S52 Project Board ใน Visual Basic 53

บทที่4 ผลการทดลอง และ อภิปราย ผลการทดลอง

4.1 ผลการทดลองในบอร์ดเรียนรู้ 8051 SBC 57

4.2 ผลการทดลองในการส่งข้อมูลทาง Parallel Port 57

4.3ผลการทดลองในการส่งข้อมูลทาง Serial Port 58

บทที่5 สรุปผลการทดลองและข้อเสนอแนะ 60

ภาคผนวก / อ้างอิง 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 แสดงการประกาศฟังก์ชันเพื่อเรียกใช้ไฟล์ Inpout32.dll ใน Visual Basic	5
รูปที่ 2.2 แสดงการเข้าสู่โปรแกรม Visual Basic	6
รูปที่ 2.3 แสดงส่วนของ New ในโปรแกรม Visual Basic	6
รูปที่ 2.4 แสดงส่วนของ Existing	7
รูปที่ 2.5 แสดงส่วนของ Recent	7
รูปที่ 2.6 แสดงหน้าต่างที่ให้เขียนข้อมูลต่างๆ	8
รูปที่ 2.7 แสดงการลากเครื่องมือจาก tool มาวางที่ form	9
รูปที่ 2.8 แสดงหน้าต่างที่ให้เขียน Source code	9
รูปที่ 2.9 แสดงผลของการ run	10
รูปที่ 2.10 แสดงหน้าต่างสำหรับการ save	10
รูปที่ 2.11 แสดงการ save ไฟล์ในชื่อ project1	11
รูปที่ 2.12 แสดงลักษณะของหัวต่อแบบ db 25 pin	12
รูปที่ 2.13 แสดงลักษณะของ Stepping motor	14
รูปที่2.14แสดงลักษณะของ KeyPad	15
รูปที่2.15แสดงลักษณะของ KeyPad	15
รูปที่2.16แสดงลักษณะของ KeyPad	16
รูปที่2.17แสดงการส่งข้อมูลที่ Row และ Collumn	16
รูปที่2.18แสดงการทำงานของ LED	17
รูปที่ 2.19 เมื่อ LED ได้รับไบแอสกลับกราฟที่ได้จะมีลักษณะคล้ายกับของซีเนอร์ไดโอด	17
รูปที่ 2.20 แสดงการหาค่าความต้านทานเมื่อทราบค่าแรงดัน Vf และกระแส If	18
รูปที่ 2.21 การใช้ LED เป็นตัวแสดงผลในวงจรไฟสลับที่มีแรงดันต่ำ	19
รูปที่ 2.22 การใช้ LED เป็นตัวแสดงผลในวงจรไฟสลับที่มีแรงดันสูง	19
รูปที่ 2.23 การใช้ LED เป็นตัวแสดงผลการทำงานของฟิวส์	19
รูปที่2.24โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง[11]	20
รูปที่2.25ขั้วแม่เหล็กและขดลวดแม่เหล็กที่ยึดติดกับเฟรม[11]	21
รูปที่2.26โรเตอร์ของมอเตอร์กระแสตรง [11]	21
รูปที่2.27แสดงทิศทางการเคลื่อนที่ของอาร์เมเจอร์ (โรเตอร์) [11]	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่2.28 แบบต่างๆ และ สัญลักษณ์ 23

รูปที่2.29 แสดงตำแหน่งส่วนแสดงผล A- G 23

รูปที่2.30รูป แสดงการนำเอา LED มาต่อกัน แบบคแมมอน K 24

รูปที่2.31 แสดงการนำเอา LED มาต่อกัน แบบคอมมอน A 24

รูปที่2.32แสดง 1 หลัก คอมมอน A ที่ขา 3 กับ 14 ส่วนขา 4,5,6,12 ไม่ได้ใช้ 25

รูปที่2.33แสดง 1 หลัก คอมมอน K ที่ขา 3 กับ 8 25

รูปที่2.34แสดง 2 หลัก คอมมอน K ที่ขา 10(ตัวที่1) กับ 5(ตัวที่ 2) 25

รูปที่2.35แสดง 2 หลัก คอมมอน A ที่ขา 10(ตัวที่1) กับ 5(ตัวที่ 2) 26

รูปที่2.36 แสดง 4 หลัก คอมมอน K ที่ขา 3(ตัวที่1) กับ 5(ตัวที่ 2) กับ 8(ตัวที่3) กับ 10(ตัวที่ 4) 26

รูปที่ 3.1 แสดงลักษณะของ 7-segment 28

รูปที่ 3.2 แสดงลักษณะของ สายParallel Port 28

รูปที่ 3.3 แสดงลักษณะของ Steping Motor 29

รูปที่ 3.4แสดงลักษณะของ Keypad 29

รูปที่ 3.5 แสดงโครงสร้างโปรแกรม ในParallel Project Board 30

รูปที่ 3.6 แสดงโครงสร้างโปรแกรม LED 31

รูปที่ 3.7 แสดงลักษณะของ 7-segment แบบคาโทดร่วมและแบบอาโนดร่วม 35

รูปที่3.8 แสดงโครงสร้างโปรแกรม 7-Segment 36

รูปที่ 3.9 แสดงลักษณะของ Stepping Motor 39

รูปที่ 3.10 แสดงโครงสร้างโปรแกรม Stepping Motor 41

รูปที่ 3.11 แสดงโครงสร้างโปรแกรม มอเตอร์ กระแสตรง 44

รูปที่ 3.12 แสดงโครงสร้างโปรแกรม Key Pad 47

รูปที่ 3.13แสดงลักษณะของ AT89S52 Project Board 51

รูปที่ 3.14 แสดงโครงสร้างโปรแกรม LED 8 หลอด จาก Serial Port 52

รูปที่ 3.15 แสดงรหัส ASCII CODE เทียบกับเลขฐาน 16 54

รูปที่ 4.2 บอร์ด 8952 SBC 56

รูปที่ 4.3แสดง รหัส ASCII 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงานพิเศษ

เนื่องจากโครงงานพิเศษเรื่องนี้ได้ความคิดมาจากการฝึกงาน โดยการได้เห็นระบบการจัดการอุปกรณ์ภายในโรงงานโดยใช้โปรแกรม Visual Basic 6 ในการควบคุมอุปกรณ์และรับค่าจากอุปกรณ์ที่ใช้งานในการผลิตผลิตภัณฑ์ของโรงงาน จากประโยชน์ของโปรแกรม Visual Basic 6 นี้จึงเล็งเห็นว่าโปรแกรมนี้น่าจะนำมาประยุกต์ใช้ได้หลายด้าน เช่น Interface กับเครื่องมือ และทำ database ได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาการใช้โปรแกรม Visual Basic 6
2. เพื่อศึกษาการใช้โปรแกรม Visual Basic 6 ในการ Interface กับอุปกรณ์
3. เพื่อศึกษาการส่งข้อมูลทาง Port Parallel
4. เพื่อศึกษาการส่งข้อมูลทาง Serial Port โดยผ่าน AT89S52 Projectboard

1.3 ขอบเขตการวิจัย

โครงงานพิเศษนี้จะศึกษาการใช้โปรแกรม Visual Basic 6 ในการเชื่อมต่อผ่านทาง Port Parallel และ Interface กับอุปกรณ์และเก็บข้อมูลโดยใช้ Computer PC ในการทำการศึกษา

1.4 ขั้นตอนการดำเนินงาน

1. จัดหา Computer PC ในการใช้เชื่อมต่อกับอุปกรณ์
2. ติดตั้งโปรแกรม Visual Basic 6 ใน Computer PC
3. ทำการศึกษาโปรแกรม Visual Basic 6 โดยศึกษาจากหนังสือคู่มือ
4. ศึกษาการส่งข้อมูลผ่าน Port Parallel โดยใช้ Program Visual Basic 6 Interface กับอุปกรณ์ภายนอก
5. ศึกษาการรับข้อมูลผ่าน Port Parallel โดยใช้ Program Visual Basic 6 Interface กับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้การทำงานของ Program Visual Basic 6 ในการ Interface กับอุปกรณ์
2. ได้เรียนรู้การทำงานของ Program Visual Basic 6 ในการรับข้อมูล
3. ได้เรียนรู้การทำงานของ Program Visual Basic 6 โดยผ่าน Port Parallel
3. ได้เรียนรู้การทำงานของ Program Visual Basic 6 โดยผ่าน Serial Port โดยใช้ AT89S52 Project Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 บทนำ

โปรแกรม Visual Basic (VB) เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่ นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจ็กต์) นั่นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจ็กต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก้ไขคุณสมบัติของออบเจ็กต์นั้นได้โดยตรง ในทุกๆ ออบเจ็กต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจ็กต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจ็กต์

ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่าโพรซีเดอร์ (procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจ็กต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โปรแกรม Visual Basic

เนื่องจาก Visual Basic นั้นไม่ได้เน้นการติดต่อกับ ระบบฮาร์ดแวร์ ทำให้มีคอนโทรลที่เกี่ยวข้องกับ การติดต่อ และควบคุม ฮาร์ดแวร์ มีอยู่ค่อนข้างน้อย สำหรับคอนโทรลมาตรฐาน ตัวเดียวที่มีมาให้สำหรับการ ติดต่อกับ ฮาร์ดแวร์ก็คือ MSComm ซึ่งเน้นการ ติดต่อกับ พอร์ตแบบอนุกรม แต่ พอร์ตขนานนั้นไม่มีคอนโทรลที่ใช้งานได้โดยตรง จึงต้องหามาเพิ่ม นอกเหนือจากการหาคอนโทรล มาใช้งาน เพื่อติดต่อและ ควบคุมฮาร์ดแวร์ แล้วยังมีอีกแนวทางหนึ่งก็คือ การใช้งานฟังก์ชันควบคุม เหมือนกับที่ใช้ฟังก์ชันในภาษาซี เพื่อติดต่อ และควบคุมฮาร์ดแวร์ ซึ่งก็มีโปรแกรมเมอร์เก่งๆ และบริษัทซอฟต์แวร์ดังๆ รวบรวมฟังก์ชันต่างๆ เป็นไลบรารีมีให้เลือกใช้งานกัน ทั้งแบบ ที่ต้องเสียเงินซื้อ และแบบที่ใช้ฟรี ในที่นี้ขอแนะนำตัวที่ใช้งานได้ฟรี นั่นคือ Inpout32.dll

Inpout32.dll เป็นไลบรารีพิเศษที่ใช้ในการเขียนโปรแกรมติดต่อพอร์ตขนาน สำหรับการเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ภายนอกผ่านทางพอร์ตขนาน ด้วย Visual Basic นั้น ตัว VB เองนั้นก็สามารถทำได้ดี แต่การเขียนโปรแกรมควบคุมอุปกรณ์ภายนอกนั้น Visual Basic ไม่มีฟังก์ชันสำหรับติดต่อกับ Parallel Port ได้โดยตรง แต่สามารถเขียนโปรแกรมให้ติดต่อกับ Parallel Port ได้โดยใช้ฟังก์ชันที่เตรียมมาในรูปของฟังก์ชัน API โดยสามารถอ่านข้อมูล และส่งข้อมูล ไปยังพอร์ตที่ต้องการได้ สำหรับไฟล์ Inpout32.dll ที่ได้กล่าวถึงนี้สามารถเข้าไปดาวน์โหลดได้จากhttp://www.logix4u.net/inpout32.htm

2.2.1 การติดตั้งไฟล์ Inpout32.dll

สำหรับการติดตั้งไฟล์ Inpout32.dll นั้นเมื่อเราทำการดาวน์โหลดมาแล้วให้ทำการคลายซิปไฟล์ ออกมาก่อน แล้วนำไฟล์ Inpout32.dll ก็อปปี้ลงในโฟลเดอร์ System ของระบบ Windows

ถ้าเป็น Windows 95, 98 ----------> ให้ก็อปปี้ไว้ที่โฟลเดอร์ C:WindowsSystem

ถ้าเป็น Windows Me, 2000, XP ----------> ให้ก็อปปี้ไว้ที่โฟลเดอร์ C:WindowsSystem32

หรือ C:WinntSystem32

2.2.2 ฟังก์ชันที่ใช้ในการติดต่อกับParallel Port

ในไฟล์ Inpout32.dll จะมีฟังก์ชันที่สามารถสั่งงานพอร์ตขนานได้ โดยมีฟังก์ชันที่จะใช้งานดังนี้

- Private declare sub out lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer) ย่อมาจาก Input from port เป็นคำสั่งที่ใช้รับข้อมูลทีละ 1 ไบต์จากพอร์ตที่กำหนด ปกติจะใช้คู่กับ Out หมายเลขของพอร์ต คือค่าของ n

- OutPort number Data

เป็นคำสั่งที่ใช้สำหรับส่งข้อมูลออกทาง Output Port ค่าของ portnumber จะเป็นเลขจำนวนเต็ม และ data จะมีค่าอยู่ระหว่าง 0 ถึง 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

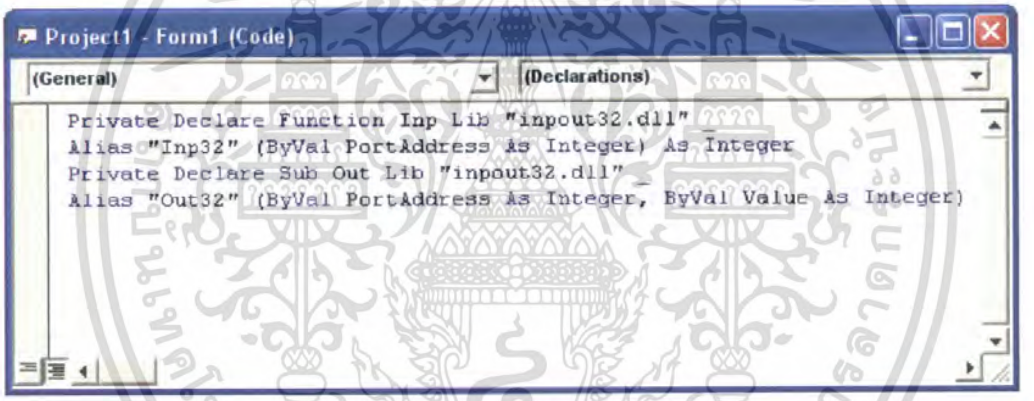
สำหรับการเรียกใช้งานฟังก์ชันใน Visual Basic6 นั้นเราจะเรียกใช้งานเช่นเดียวกับการเรียกใช้ฟังก์ชัน Windows API

2.2.3 การประกาศฟังก์ชันเพื่อเรียกใช้ไฟล์ Inpout32.dll ใน Visual Basic

ในการเรียกใช้งานไฟล์ Inpout32.dll นั้นเราจะต้องทำการประกาศฟังก์ชันที่หน้าต่างของ Code Window เสียก่อนดังนี้

สำหรับการประกาศฟังก์ชันเพื่อเรียกใช้งานไฟล์ Inpout32.dll สำหรับการรับข้อมูล Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress As Integer) As Integer

สำหรับการประกาศฟังก์ชันเพื่อเรียกใช้งานไฟล์ Inpout32.dll สำหรับการส่งข้อมูล Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

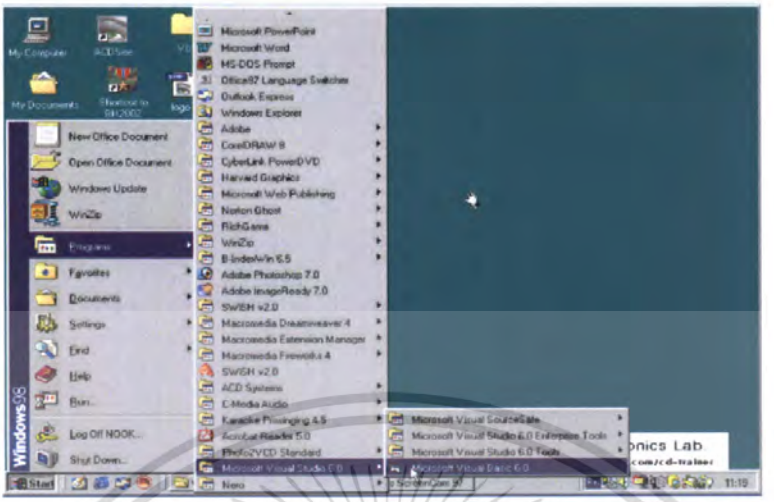


รูปที่ 2.1 แสดงการประกาศฟังก์ชันเพื่อเรียกใช้ไฟล์ Inpout32.dll ใน Visual Basic

2.2.4 การใช้งานโปรแกรม Visual Basic

สำหรับโปรแกรมนี้การทำงานจะมีการทำงานทีละบรรทัดได้ และนอกจากนี้ในการคอมไพล์สามารถที่จะคอมไพล์เป็น .exe หรือเป็นไฟล์ได้ install เช่นกัน จากนี้จะเป็นการแนะนำการเข้าโปรแกรมหลังจากที่ได้ทำการลงโปรแกรมไปเรียบร้อยแล้ว โดยการเริ่มต้นคลิกที่ start แล้วเลือกที่ program เลือกที่ Microsoft Visual Basic และทำการคลิกที่ Microsoft Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการเข้าสู่โปรแกรม Visual Basic

จากรูปที่แสดงขั้นตอนการเข้าสู่โปรแกรมก็จะเข้าสู่หน้าแรกของโปรแกรม Visual Basic ซึ่งจะเป็นหน้าของ New Program และจะประกอบไปด้วยส่วนของ New ส่วนของ Existing และส่วนของ Recent ดังรูปต่อไปนี้

รูปที่ 2.3 แสดงส่วนของ New ในโปรแกรม Visual Basic

New จะเป็นส่วนที่ใช้สร้างโปรแกรมต่าง ๆ ที่ต้องการจะเขียนขึ้นมาและในตอนนี้จะแนะนำตัวแรกที่จะใช้งานคือ standard exe ส่วนต่อมาคือ Existing จะเป็นการดึงเอาข้อมูลที่สร้างไว้แล้วมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

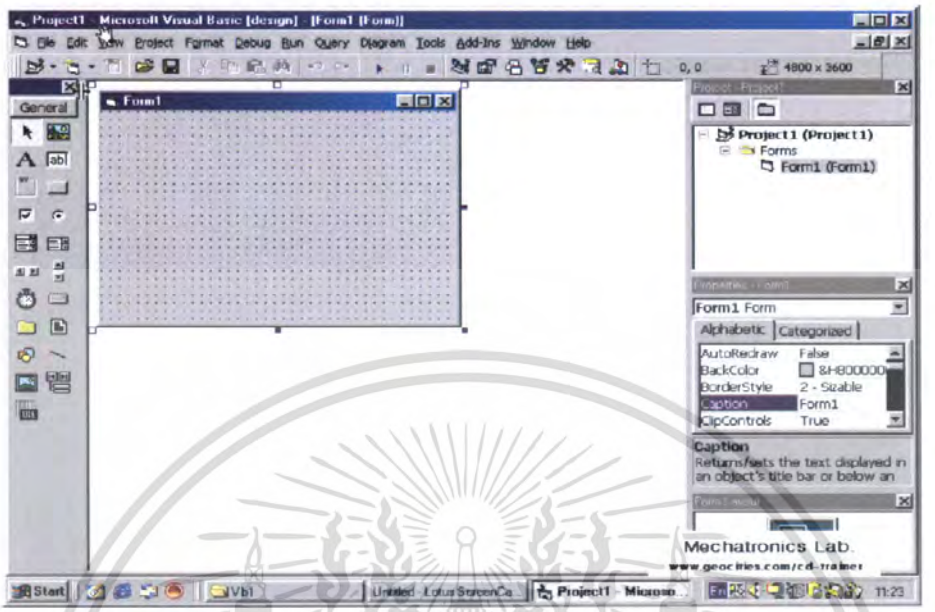
รูปที่ 2.4 แสดงส่วนของ Existing

รูปที่ 2.5 แสดงส่วนของ Recent

และส่วนสุดท้ายคือส่วนของ Recent ซึ่งเป็นส่วนเก็บข้อมูลล่าสุดที่ได้ใช้ข้อมูลตัวนั้นไป

หลังจากที่ได้รู้รายละเอียดของหน้าต่างแรกแล้ว เราจะมาศึกษาในส่วนของ New ซึ่งจะใช้การเขียนโปรแกรม โดยให้ทำการคลิกที่ standard exe ก็จะปรากฏหน้าต่างที่ให้เขียนข้อมูลต่างๆ ซึ่งจะปรากฏเป็นรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

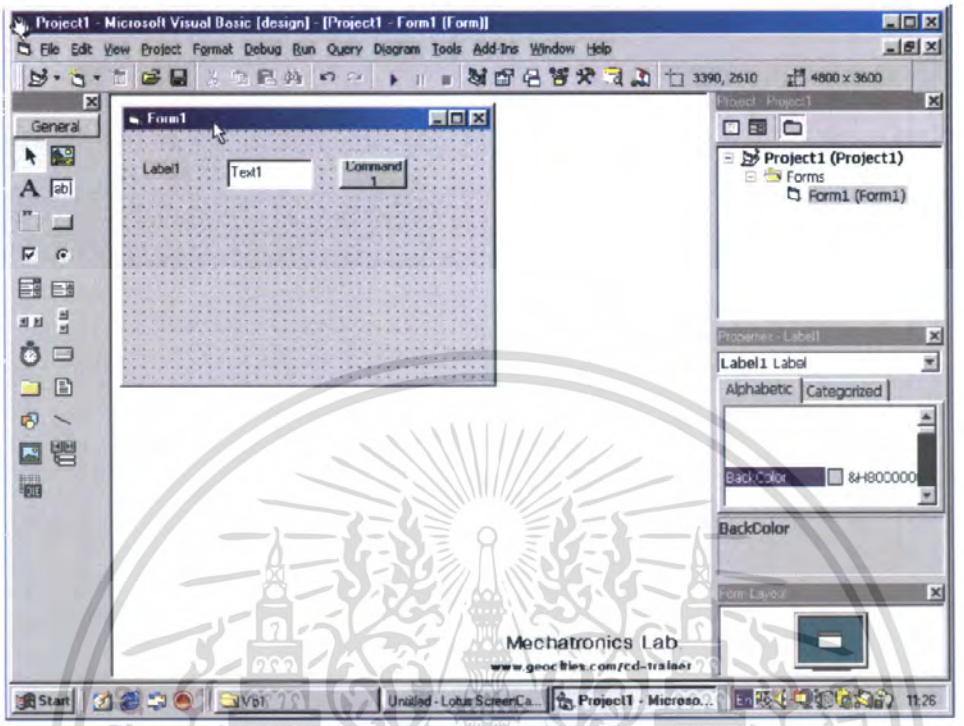


รูปที่ 2.6 แสดงหน้าต่างที่ให้เขียนข้อมูลต่างๆ

ในรูปหน้าต่างนี้จะประกอบด้วย 4 ส่วน ที่ควรจะต้องศึกษาก่อนที่จะทำการเขียนโปรแกรมมีดังนี้

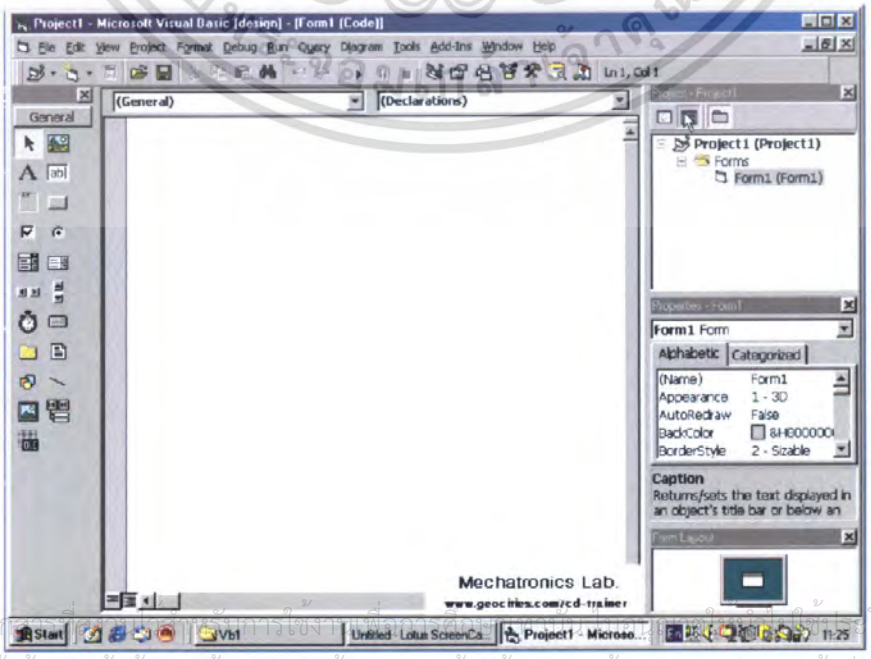
1. ด้านซ้ายเป็นส่วนของ tool bar จะเก็บเครื่องมือต่าง ๆ ที่ใช้สำหรับสร้างงานในส่วนของตัว form
2. ด้านขวาบนเป็นส่วนของ program ใช้เก็บตำแหน่งของ form ต่าง ๆ ที่สร้างขึ้นมา
3. ส่วนถัดมาเป็นส่วน properties ใช้กำหนดคุณสมบัติในส่วนของเครื่องมือใน tool bar ที่ลากมาไว้บน form ว่าต้องการให้เป็นแบบใด
4. ส่วนท้ายสุดเป็นส่วน layout เป็นตัววางตำแหน่งของ form ว่าอยู่ที่ตำแหน่งใดเมื่อทำการ run โปรแกรมแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงการลากเครื่องมือจาก tool มาวางที่ form

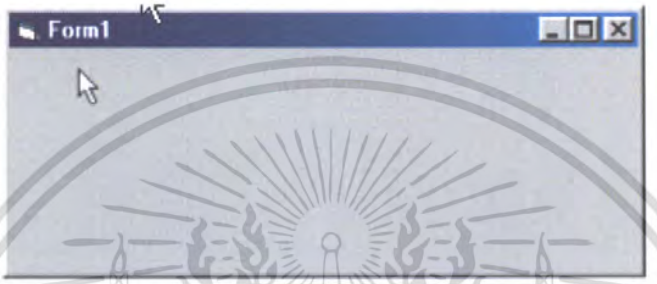
จากรูปที่2.7 เป็นการลากเครื่องมือจาก tool bar มาวางที่ form โดยจะวางที่ตำแหน่งใดก็ได้แต่ต้องกำหนดลำดับเหตุการณ์ของปุ่มก่อนว่าปุ่มใดทำงานก่อนหลัง ในการวางตำแหน่งของปุ่มสามารถทำได้โดยใช้ properties กำหนดคุณสมบัติตามชื่อปุ่มที่ได้ตั้งชื่อเอาไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

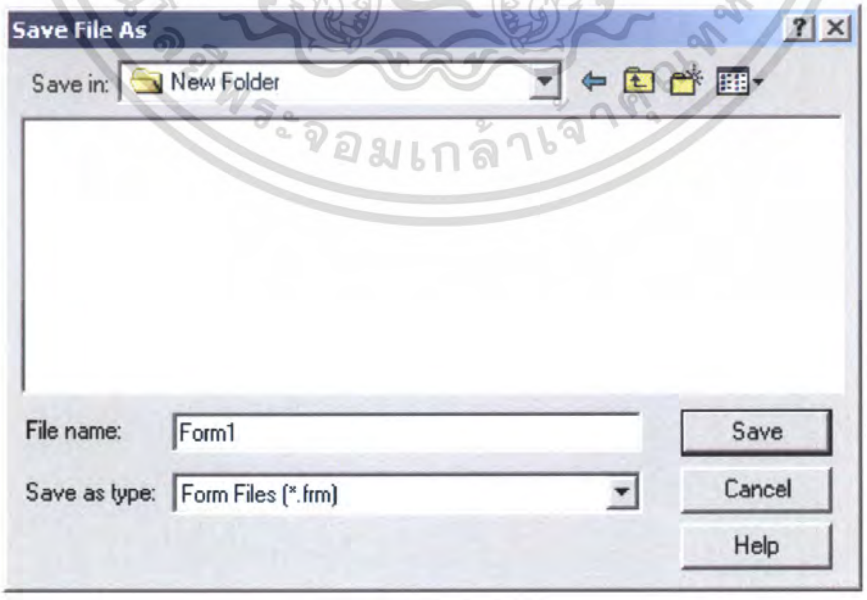
รูปที่ 2.8 แสดงหน้าต่างที่ให้เขียน Source code

จากรูปของส่วนนี้ จะเป็นหน้าต่างที่ให้เขียน source code โดยจะกำหนดให้รูปแบบของปุ่มต่าง ๆ ที่วางอยู่บน form เป็นอย่างไรก็ได้ หน้าจอนี้จะได้มาจากการคลิกที่Start ก็จะปรากฏหน้าจอนี้ขึ้นมาและหากต้องการที่จะกลับสู่หน้าจอ form ก็ให้ทำการคลิกที่End ก็สามารถกลับไปได้เลย



รูปที่ 2.9 แสดงผลของการ run

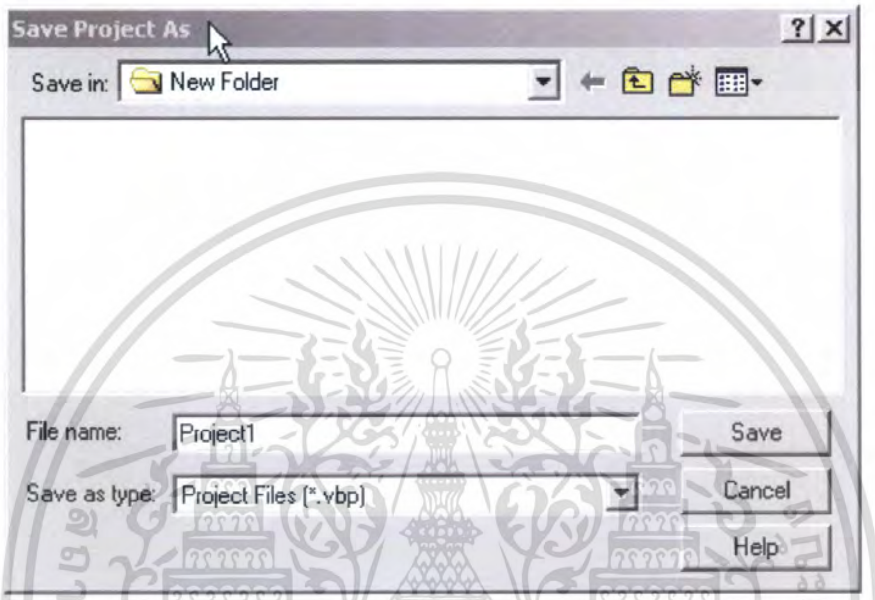
จากรูปที่ 2.9เป็นผลของการ run โดยทำการคลิกที่Starเนื่องจากยังไม่ได้ทำการลากเครื่องมือมาไว้ใน form ผลของการ run จึงไม่มีปุ่มใด ๆ บนผลการ run และคลิก ก็จะกลับสู่หน้าจอของ form และสามารถที่จะเพิ่มเติมข้อมูลเข้าไปได้อีก ต่อมาจะมารู้จักกับวิธีการ save ข้อมูลว่าจะเก็บข้อมูลอย่างไร คลิกเข้าไปที่ file และเลือกที่ save project จากนั้นก็จะปรากฏหน้าต่างขึ้นมา แล้วเลือก folder ที่ต้องการจะเซฟ project (ในกรณีนี้ folder ที่ต้องการเซฟชื่อว่า New Folder)



รูปที่ 2.10 แสดงหน้าต่างสำหรับการ save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ทำการ save ฟอร์มทุกฟอร์มที่เราสร้างขึ้นในโปรเจคนั้น โดยที่ visual basic จะแสดงหน้าต่างให้ save ฟอร์มทุกๆฟอร์มที่เราสร้างขึ้น (สังเกตที่สกุลไฟล์จะเป็น .frm) เมื่อ save ฟอร์มทุกฟอร์มแล้ว ก็ให้ทำการ save ไฟล์ project (.vbp) ดังรูป



รูปที่ 2.11 แสดงการ save ไฟล์ในชื่อ project1

2.3 หลักการส่งข้อมูลทาง parallel port

Parallel port การใช้งานส่วนใหญ่จะใช้กับ printer เป็นหลักและการส่งข้อมูลจะเป็นขนาด 8 bit โดยจะใช้หัวต่อแบบdb25 pin ดังรูปที่ 2.12

รูปที่ 2.12 แสดงลักษณะของหัวต่อแบบ db 25 pin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วในการส่งข้อมูลของ parallel จะเร็วกว่า serial port 8-10 เท่า แต่มีข้อเสียคือส่งข้อมูลได้ไม่ไกลมากนัก parallel port แบ่งหน้าที่ได้ 3 ประเภทใหญ่ๆดังนี้

กลุ่มสัญญาณ	จำนวนเส้นสัญญาณ	ลักษณะสัญญาณ
Data Port	8	8 output pin
Status Port	5	5 output pin(1 invert)
Control Port	4	4 output pin(3 invert)

ลักษณะสัญญาณของพอร์ตขนานทั้งหมด

Pin No	Signal Name	Bit	Direction(in/out)
1	nStrobe	-C0	Output
2	Data 0 (Bit 0)	D0	Output
3	Data 1 (Bit 1)	D1	Output
4	Data 2 (Bit 2)	D2	Output
5	Data 3 (Bit 3)	D3	Output
6	Data 4 (Bit 4)	D4	Output
7	Data 5 (Bit 5)	D5	Output
8	Data 6 (Bit 6)	D6	Output
9	Data 7 (Bit 7)	D7	Output
10	nAck	S6	Input
11	Busy	-S7	Input
12	PaperEnd	S5	Input
13	Select	S4	Input
14	nAutoFeed	-C1	Output
15	nError	S3	Input
16	nInitialize	C2	Output
17	nSelectPrinter	-C3	Output
18-25	Ground	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมติดต่อกับ Port ขนานเราต้องรู้ address ของ Port ขนานที่เราจะติดตั้งก่อนใน Computer โดยติดตั้งดังนี้

1.คลิ๊กปุ่ม Start > Control Panel

2.คลิ๊ก Icon System

3.จะปรากฏหน้าต่าง System Property คลิ๊กที่แถบ Hardware

4.จากนั้นคลิ๊ก Device manager

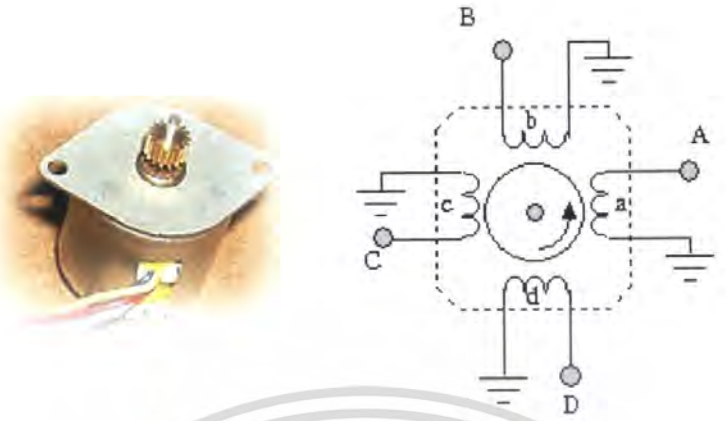
5.จากนั้นคลิ๊ก(COM & LPT)คือ ECP Printer Port (LPT1)

6.จากนั้นคลิ๊ก Resources จากนั้นเลือก I/O Range 0378-037F

2.4 ความรู้เบื้องต้นเกี่ยวกับ Stepping motor

Stepping Motor เป็นมอเตอร์ไฟฟ้าชนิดหนึ่ง ซึ่งมี input เป็นกลุ่ม ของ Binary Voltage และ Output การเคลื่อนที่ในเชิงมุม (หมุน) แกนหมุน (Shaft) เป็น Step โดย Resolution ของ Steping Motor อยู่ในช่วงตั้งแต่ 0.1- 30 องศาซึ่งขึ้นอยู่กับโครงสร้างของ Stepping Motor หรือบอกเป็นจำนวน Step ต่อ 1 รอบ Stepping Motor สามารถควบคุมตำแหน่งการหมุนได้ ซึ่งจะมีความละเอียดของมุมในการหมุนและ Step ที่แตกต่างกันออกไป ทั้งนี้ก็ขึ้นอยู่กับ ชนิดของมอเตอร์และลักษณะการส่งสัญญาณไปควบคุมมอเตอร์ ทำให้ Stepping Motor มีความยืดหยุ่นในการนำมาใช้งาน ทำให้สามารถนำมาประยุกต์ใช้งานได้หลายรูปแบบ ไม่ว่าจะเป็นหุ่นยนต์หรือแขนกลต่างๆ เนื่องจากมีความแม่นยำในการควบคุมตำแหน่งสูง โครงสร้างของ Stepping Motor มีลักษณะดังรูปที่ 2.13 ซึ่งประกอบด้วย ขดลวด stator 4 ขด ล้อมรอบแกนหมุนหลักการทำงาน คือ เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวด Stator Coil a, b, c, d ไม่พร้อมกันนั่นคือ ถ้าเราจ่ายกระแสให้ a ก่อน โดยไม่จ่ายให้ขดอื่น แล้วตามด้วย b, c และ d เรียงตามลำดับ จะทำให้เกิดสนามแม่เหล็ก หมุนวนในลักษณะทวนเข็มนาฬิกา ซึ่งส่วนของ Rotor ที่เป็นแม่เหล็กถาวรก็จะหมุนตามสนามแม่เหล็กไปด้วย คือ ทวนเข็มนาฬิกาในทำนองเดียวกันถ้าเราจ่ายกระแสให้ขด a, d, c, b, a ..... ก็จะให้สนามแม่เหล็กหมุนในทิศทางตามเข็มนาฬิกา ซึ่งส่งผลให้ Rotor หมุนตามเข็มนาฬิกา ด้วย การกำหนดความเร็วของ Stepping Motor ทำได้โดยการเปลี่ยนแปลง ความเร็วของ การเปลี่ยนการจ่ายกระแสจากขดลวดขดหนึ่งไปยังอีกขดหนึ่งให้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงลักษณะของ Stepping motor

การควบคุม Stepping Motor แบบ 4 เฟส

ในการควบคุมการทำงานของ Stepping Motor สามารถแบ่งได้ออกเป็น 3 รูปแบบ คือ

ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)

ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)

ในการควบคุมการหมุนของ Stepping แบบ 4 เฟสนั้น เราจะต้องกระตุ้นให้มอเตอร์หมุนไปแต่ละ Step โดยจ่ายกระแสไฟฟ้าให้กับ Stepping ทีละเฟสตามลำดับ หลักการคือเริ่มจากจ่ายกระแสให้กับขดลวด Stator เฟสที่ 1 จากนั้นกระตุ้นเฟสที่ 2 และ เฟสที่ 3 ไปเรื่อยๆ ตามลำดับ จากนั้นก็วนกลับมาที่ขดลวด Stator เฟสที่ 1 อีกครั้งและวน Loop ไปเรื่อยๆ ก็จะทำให้ Stepping Motor หมุนและในทางกลับกันถ้าต้องการให้ Stepping Motor หมุนกลับทางก็ต้องกระตุ้นขดลวด Stator เฟส 4 เฟส 3 เฟส 2 และ เฟส 1 ตามลำดับ

ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

ในการควบคุม Stepping Motor แบบ 2 เฟสนั้น เราจะต้องจ่ายกระแสไฟฟ้าเพื่อกระตุ้นขดลวดของมอเตอร์ทีละ 2 เฟส ในเวลาเดียวกันและเรียงกันไปตามลำดับซึ่งแสดงดังตารางด้านล่าง โดย Stepping Motor จะหมุนเหมือนกับการควบคุมแบบ เวฟ แต่การควบคุมแบบ 2 เฟสจะให้แรงบิดที่สูงกว่าแบบ เวฟ

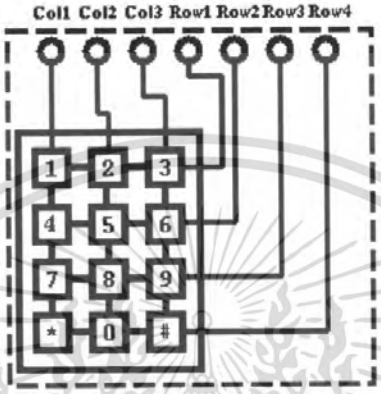
ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

การควบคุม Stepping Motor แบบ ครึ่งสเต็ป จะทำให้เราสามารถเพิ่มความละเอียดในการควบคุมการหมุนของ Stepping Motor ได้แม่นยำมากขึ้นซึ่งเป็นการผสมผสานระหว่างการควบคุมแบบเวฟและแบบ Full Step 2 เฟสเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

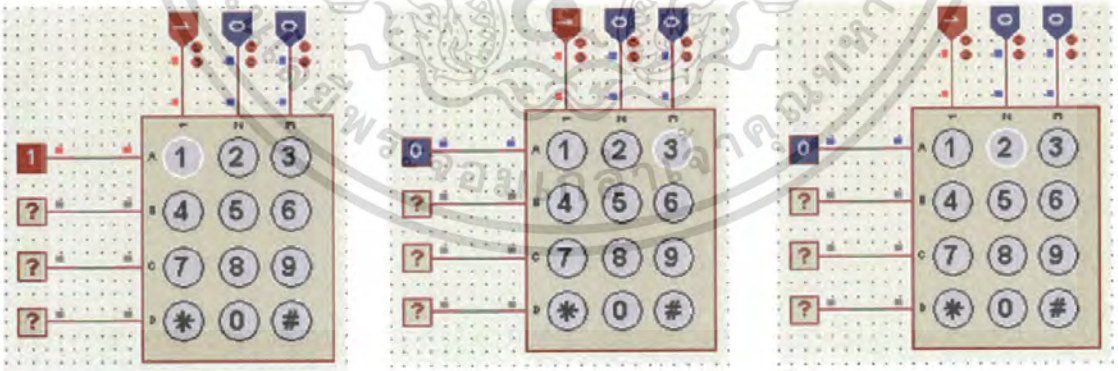
2.5 ความรู้เบื้องต้นของ Keypad

ก่อนอื่นเรามาทำความรู้จักกับ Keypad กันKeypad เป็นสวิตช์ที่มีการเชื่อมต่อในแนวของ Row และ Columnโดยจะแบ่งในลักษณะดังภาพ



รูปที่2.14แสดงลักษณะของ KeyPad

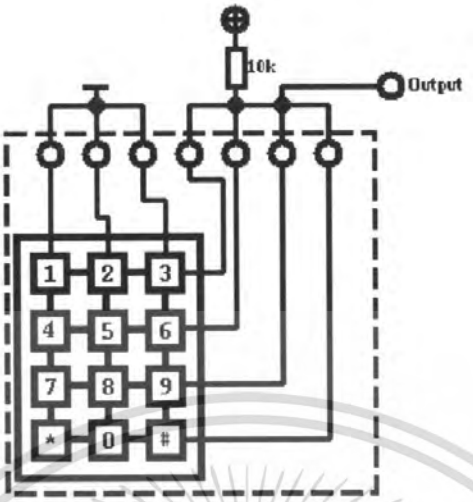
ถ้าอยากรู้ว่าRow หรือ Column ใดที่จะทำงานจะพูดให้เข้าใจง่ายๆก็คือ เวลาที่เรากดสวิตช์ไหน จะเหมือนกับเราลากเส้นในแนวนอนและแนวตั้งให้จุดตัดของสองเส้นผ่านสวิตช์ตัวนั้น



รูปที่2.15แสดงลักษณะของ KeyPad

เช่น ถ้าเรากดสวิตช์ 1 Row1 กับ Column1 คือตัวที่จะทำงานถ้าเรากดสวิตช์ 2 Row1 กับ Column2 คือตัวที่จะทำงานถ้าเรากดสวิตช์ 3 Row1 กับ Column3 คือตัวที่จะทำงานและหากเรานำ Keypad ต่อดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.16แสดงลักษณะของ KeyPad

จะทำให้สวิตช์มีการทำงานเป็นแบบ Active Low คือสวิตช์ใดที่จะทำงานRow และ Columnนั้นจะมีค่าเป็น 0ถ้าไม่เช่นนั้นจะเป็น1

Column			Row				Key	
Col1	Col2	Col3	Row1	Row2	Row3	Row4	Character	Binary code
0	0	0	1	1	1	1	(none)	1111
0	1	1	0	1	1	1	1	0001
1	0	1	0	1	1	1	2	0010
1	1	0	0	1	1	1	3	0011
0	1	1	1	0	1	1	4	0100
1	0	1	1	0	1	1	5	0101
1	1	0	1	0	1	1	6	0110
0	1	1	1	1	0	1	7	0111
1	0	1	1	1	0	1	8	1000
1	1	0	1	1	0	1	9	1001
0	1	1	1	1	1	0	*	1010
1	0	1	1	1	1	0	0	0000
1	1	0	1	1	1	0	#	1011

รูปที่2.17แสดงการส่งข้อมูลที่ Row และ Collumn

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความรู้เบื้องต้นของ LED

LED ไดโอดเปล่งแสงเป็นไดโอดชนิดพิเศษ ที่สามารถเปล่งแสงออกมาได้ ไม่ว่าจะเป็นแสง ในช่วงที่สายตามองเห็น

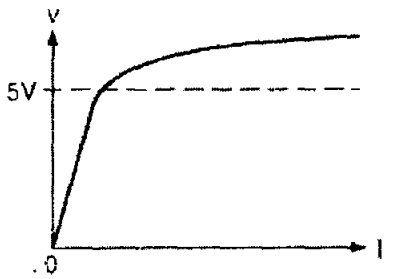
เมื่อเปรียบเทียบกับหลอดไฟชนิดทังสเตนแล้ว LED จะมีประสิทธิภาพสูงกว่าประมาณ 10-15 เท่า และยังมีการตอบสนองต่อแสงที่เร็วกว่าด้วย คือประมาณ 0.1 ไมโครวินาที เมื่อเทียบกับหลอดไฟชนิดทังสเตน ซึ่งใช้เวลาในระดับมิลลิวินาที ดังนั้นโดยทั่ว ๆ ไป LED จึงเหมาะที่จะใช้เป็นตัวแสดงผลหรือใช้เป็นไฟกะพริบ

LED สีต่าง ๆ ที่กระแส I_f 20 มิลลิแอมป์

LED	สีแดง	สีส้ม	สีเหลือง	สีเขียว
ค่าแรงดันฟอร์เวิร์ด V_f (โวลต์)	1.8	2.0	2.1	2.2

รูปที่2.18แสดงการทำงานของ LED

เมื่อ LED ได้รับกระแสไบแอสตรงจะมีแรงดันตกคร่อม LED ประมาณ 2 โวลต์ ดังแสดงในตารางที่ 1 ซึ่งเป็นค่าแรงดันไบแอสตรงของ LED สีต่าง ๆ ที่กระแสไบแอสตรง 20 มิลลิแอมป์ และถ้า LED ได้รับการไบแอสกลับ ก็จะเกิดปรากฏการณ์อะวาลานซ์หรือซีเนอร์เช่นเดียวกับไดโอดชนิดอื่น ๆ แต่ค่าแรงดันนั้นจะมีค่าน้อยกว่า ดังแสดงในรูปที่ 1 โดยทั่ว ๆ ไปแล้ว LED จะทนแรงดันย้อนกลับได้สูงสุดประมาณ 3-5 โวลต์

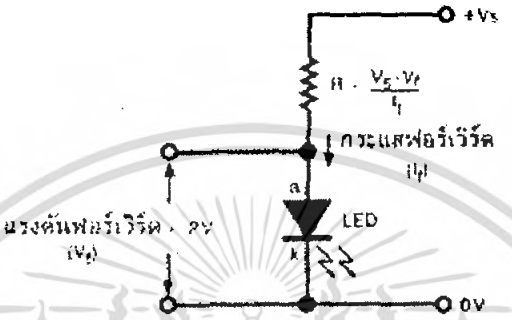


รูปที่ 2.19 เมื่อ LED ได้รับไบแอสกลับกราฟที่ได้จะมีลักษณะคล้ายกับของซีเนอร์ไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

81615

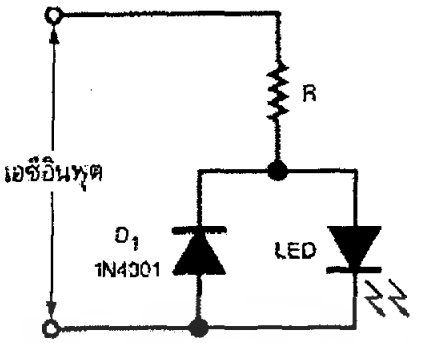
ในการนำ LED ไปใช้งาน จะต้องต่อตัวต้านทานจำกัดกระแสอนุกรมเข้ากับ LED ด้วย เพื่อป้องกันไม่ให้ LED เสียหาย รูปที่ 2 แสดงการหาค่าตัวต้านทาน R เพื่อให้ได้กระแสที่พอเหมาะแก่ LED โดย R จะต่อที่ขั้วแคโทด หรือแอโนดของ LED ก็ได้ความสว่างของ LED จะเป็นสัดส่วนโดยตรงกับปริมาณกระแสที่ไหลผ่านตัวมัน โดยทั่ว ๆ ไป LED จะสามารถทนกระแสได้สูงสุดประมาณไม่เกิน 30-40 มิลลิแอมป์



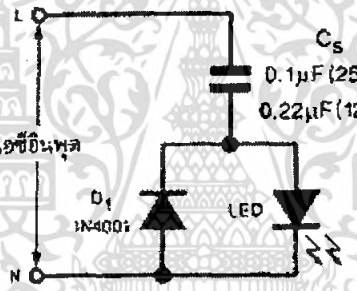
รูปที่ 2.20 แสดงการหาค่าความต้านทานเมื่อทราบค่าแรงดัน Vf และกระแส If

LED ยังสามารถนำไปใช้กับไฟสลับได้ด้วยดังแสดงในรูปที่ 3 โดยการนำไดโอดธรรมดามาต่อกลับขั้วขนานกับ LED เพื่อป้องกันไม่ให้ LED ได้รับไบแอสกลับ สำหรับค่า R ควรจะมีค่าเป็นครึ่งหนึ่ง ของเมื่อคำนวณได้จากวงจรไฟตรง เพื่อให้ความสว่างเท่าเดิม และถ้าวงจรนี้ใช้กับแหล่งจ่ายไฟสลับที่มีค่าแรงดันสูง ๆ ตัวต้านทาน R ก็จะต้องมีอัตราทนกำลังสูง ๆ ตามไปด้วย เช่นถ้าใช้กับแหล่งจ่ายไฟ 250 โวลต์ ตัวต้านทานควรอัตราทนกำลังอย่างน้อยที่สุด 2.5 โวลต์ เมื่อคิดที่กระแสที่ผ่าน LED เฉลี่ยประมาณ 10 มิลลิแอมป์หรือถ้าไม่ต้องการใช้ตัวต้านทานที่มีอัตราทนกำลังสูง ๆ ก็อาจใช้ตัวเก็บประจุจำกัดกระแสที่ผ่าน LED ตามต้องการได้เช่นเดียวกัน โดยตัวเก็บประจุ Cs จะมีอัตรากำลังสูญเสียน้อยมาก เนื่องจากกระแสและแรงดันที่ Cs มีเฟสต่างกัน 90 องศา สำหรับค่าของ Cs ประมาณ 0.1 และ 0.22 ไมโครฟารัดสำหรับไฟสลับ 50-60 เฮิรตซ์ 250 โวลต์ และ 125 โวลต์ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

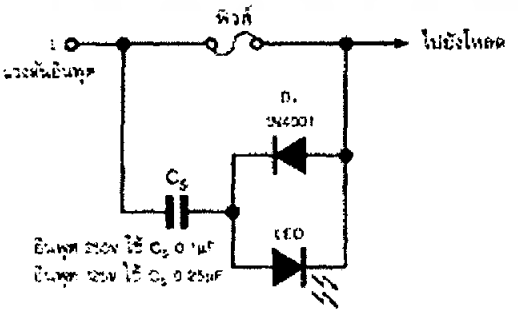


รูปที่ 2.21 การใช้ LED เป็นตัวแสดงผลในวงจรไฟสลับที่มีแรงดันต่ำ



รูปที่ 2.22 การใช้ LED เป็นตัวแสดงผลในวงจรไฟสลับที่มีแรงดันสูง

วงจรในรูปที่ 5 LED ยังสามารถใช้เป็นตัวแสดงผลการทำงานของฟิวส์ได้ด้วย ในสภาวะปกติวงจรจะถูกลัดโดยฟิวส์แต่เมื่อฟิวส์ขาดจะทำให้วงจรนี้ทำงานและกระแสโหลด นั้นจะถูกจำกัดโดย Cs

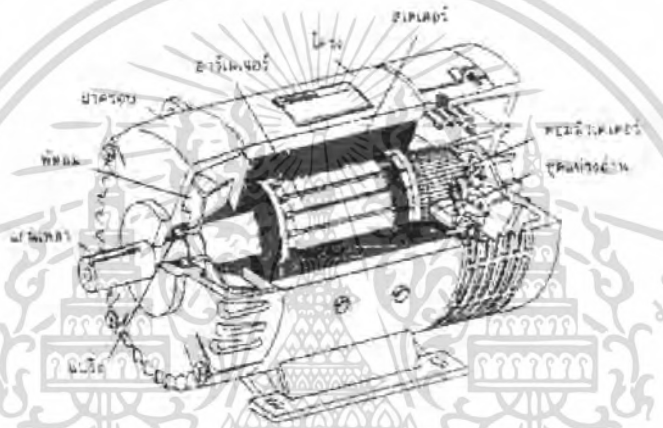


รูปที่ 2.23 การใช้ LED เป็นตัวแสดงผลการทำงานของฟิวส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ความรู้เบื้องต้นของ DC Motor

มอเตอร์ คือ เครื่องกลไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล มอเตอร์แบ่งออกเป็น 2 ประเภท ใหญ่ ๆ คือ มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) และมอเตอร์ ไฟฟ้ากระแสสลับ (AC Motor) ในที่นี่จะกล่าวถึงเฉพาะ มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เท่านั้น มอเตอร์ไฟฟ้ากระแสตรงนั้นจะใช้ในงานในด้านการขับเคลื่อนในแบบต่าง ๆ ที่มีอัตราเร็วไม่สูงมากนัก เนื่องจากมอเตอร์ไฟฟ้ากระแสตรงนั้นมีแรงบิดเริ่มต้นที่สูง (starting torque) สามารถควบคุมควบคุมอัตราเร็วได้ค่อนข้างง่าย แต่มีข้อเสียคือมีโครงสร้างที่ค่อนข้างซับซ้อนมากจึงจำไม่เหมาะที่จะใช้ในงานที่มีอัตราเร็วค่อนสูงมาก ๆ



รูปที่2.24 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง[11]

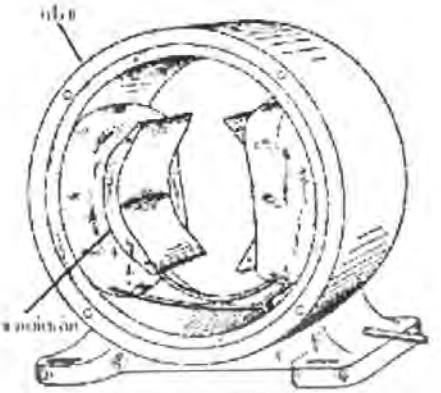
โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง (D.C. Motor) ประกอบด้วย 2 ส่วนหลัก ๆ คือ ส่วนที่อยู่กับที่ และส่วนที่เคลื่อนที่

ส่วนที่อยู่กับที่

เฟรม คือ เป็นโครงสร้างภายนอก ที่เรามองเห็นเป็นตัวมอเตอร์ จะทำหน้าที่เป็นเส้นทางเดินของสนามแรงแม่เหล็ก และเป็นที่ยึดส่วนต่าง ๆ ให้แข็งแรง

ขั้วแม่เหล็ก จะประกอบด้วย แกนขั้วแม่เหล็ก ส่วนนี้จะติดอยู่กับเฟรมและขดลวดสนามแม่เหล็ก (Field coil) ที่พันรอบ ๆ แกนขั้วแม่เหล็ก จะทำหน้าที่รับกระแสจากภายนอก และสร้างสนามแม่เหล็ก ซึ่งจะทำให้เกิดแรงบิดขึ้น (Torque)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.25ขั้วแม่เหล็กและขดลวดแม่เหล็กที่ยึดติดกับเฟรม[11]

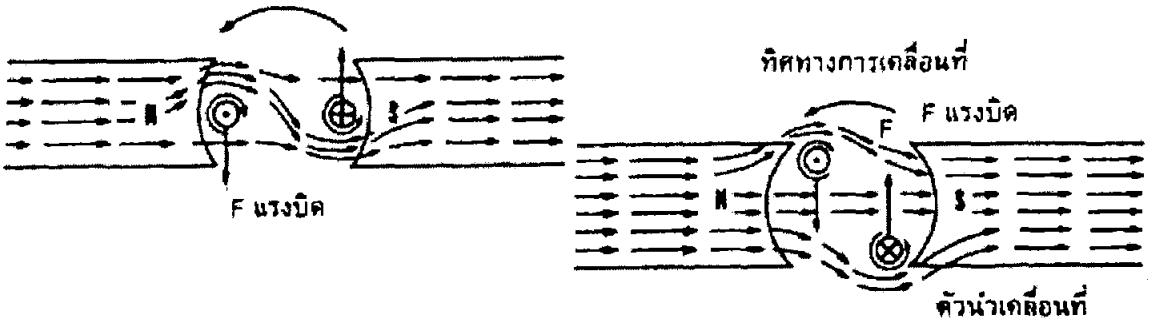
ส่วนที่เคลื่อนที่หรือโรเตอร์ (rotor) จะมีขดลวดอาร์เมเจอร์(Armature Winding)ที่พันอยู่บนแกนเหล็กอาร์เมเจอร์ (Armature core) และมีคอมมิวเตเตอร์ยึดติดอยู่ที่ปลายของขดลวดอาร์เจอร์ ดังรูป

รูปที่2.26 โรเตอร์ของมอเตอร์กระแสตรง [11]

ซึ่งในส่วนนี้ คอมมิวเตเตอร์จะทำหน้าที่ในการสัมผัสกับแปรงถ่านคาร์บอน (Carbon Brushes)ที่อยู่ในมอเตอร์เพื่อที่จะให้มีกระแสไหลผ่านไปยังขดลวดอาร์เมเจอร์ ทำให้เกิดการสร้างสนามแม่เหล็กขึ้นเพื่อให้เกิดการหักล้างและเสริมกันกันกับสนามแม่เหล็กที่เกิดจากขดลวดแม่เหล็ก ซึ่งจะทำให้มอเตอร์หมุนได้

หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง (D.C. Motor) เมื่อมีกระแสไหลผ่านเข้าไปในมอเตอร์ กระแสจะแบ่งออกไป 2 ทาง คือ ส่วนที่หนึ่งจะผ่านเข้าไปที่ขดลวดสนามแม่เหล็ก (Field coil) ทำให้เกิดสนามแม่เหล็กขึ้นและอีกส่วนหนึ่งจะผ่านแปลงถ่านคาร์บอนและผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์เมเจอร์ทำให้เกิดสนามแม่เหล็กขึ้นเช่นกัน ซึ่งทั้งสองสนามจะเกิดขึ้นขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็กแล้วจะไม่มีการตัดกัน จะมีแต่การหักล้างและการเสริมกัน ซึ่งทำให้เกิดแรงบิดในอาร์เมเจอร์ ทำให้อาร์เมเจอร์หมุนซึ่งในการหมุนนั้นจะเป็นไปตามกฎมือซ้ายของเฟลมมิ่ง (fleming's left hand rule)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.27แสดงทิศทางการเคลื่อนที่ของอาร์เมเจอร์ (โรเตอร์) [11]

รายละเอียดพื้นฐานของมอเตอร์ รายละเอียดพื้นฐานของมอเตอร์ที่จะนำมาพิจารณาเลือกใช้กับงานต่าง ๆ ที่จะกล่าวถึงมีอยู่ 4 อย่าง คือ แรงดันไฟฟ้า (voltage) การไหลของกระแส (current dawn) ความเร็ว (speed) แรงบิด (torque)

แรงดันไฟฟ้า (voltage) มอเตอร์ทุกตัวจะมีแรงดันไฟฟ้าใช้งานที่แตกต่างกันตามคุณสมบัติของมอเตอร์แต่ละตัวที่ผู้ผลิตกำหนดมาเช่น มอเตอร์ไฟฟ้ากระแสตรง 12 โวลต์ เป็นต้น สำหรับมอเตอร์ไฟฟ้ากระแสตรงนั้นสามารถใช้ไฟกระแสตรงหรือกระแสสลับก็ได้ แต่ถ้าเป็นมอเตอร์ไฟฟ้ากระแสสลับจะใช้ไฟกระแสสลับเท่านั้น และแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์จะมีผลต่ออัตราเร็วและแรงบิดของมอเตอร์คือ ถ้าหากแรงดันไฟฟ้ามากอัตราเร็วและแรงบิดของมอเตอร์ก็จะมากด้วย

การไหลของกระแส การไหลของกระแสนั้นจะกล่าวถึงในกรณีที่มอเตอร์ได้รับกระแสจากแหล่งจ่าย ในกรณีที่มอเตอร์ไม่ได้ต่อกับโหลดใด ๆ นั้นจะมีกระแสไหลผ่านน้อย แต่ในกรณีที่มีการใช้งานต่อกับโหลดจะมีปริมาณกระแสที่เพิ่มมากขึ้น การไหลของกระแสนั้นมีความจำเป็นเพราะถ้าหากกระแสไม่พอแล้วมอเตอร์ก็จะไม่มีกำลังเพียงพอสำหรับการขับโหลด และกระแสไฟฟ้าที่จ่ายให้กับมอเตอร์จะมีผลต่ออัตราเร็วและแรงบิดของมอเตอร์ด้วย คือ ถ้าหากจ่ายกระแสไฟฟ้าให้กับมอเตอร์มากอัตราเร็วและแรงบิดของมอเตอร์ก็จะมากด้วย

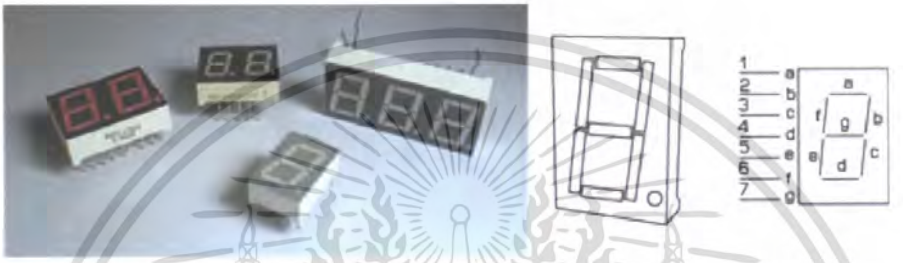
อัตราเร็ว ส่วนใหญ่มอเตอร์กระแสตรงจะมีอัตราเร็วปกติที่ 4000-7000 รอบต่อนาที ซึ่งอัตราเร็วของมอเตอร์สามารถลดลงหรือเพิ่มขึ้นได้ตามความต้องการของผู้ใช้ ถ้าหากต้องการใช้งานที่ต้องการความเร็วมากก็ต้องเลือกมอเตอร์ที่มีอัตราเร็วสูง เป็นต้น

แรงบิด เป็นแรงที่มอเตอร์กระทำกับโหลดในการพิจารณาเลือกมอเตอร์นั้นถ้าหากมีแรงบิดน้อยจะใช้งานได้กับโหลดที่ไม่หนักมากแต่ถ้ามีแรงบิดมากสามารถใช้งานกับโหลดที่มีน้ำหนักมากได้ ในการพิจารณาเลือกใช้งานมอเตอร์จึงจำเป็นต้องรู้ข้อมูลพื้นฐานของมอเตอร์เพื่อที่จะเป็นข้อพิจารณาในการเลือกใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

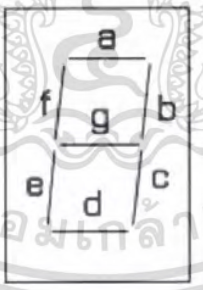
2.8 ความรู้เบื้องต้นของ 7-Segment

ตัวแสดงผล 7 ส่วน หรือที่เราเรียกว่า 7 Segment เป็นอุปกรณ์อิเล็กทรอนิกส์ประเภท Display เช่นเดียวกับไดโอดแปลงแสง หรือ LED ตัว 7 Segment เองนั้นภายในก็คือ LED 7ตัว(หรือมากกว่า) มาต่อกันเป็นรูปตัวเลข 8 นั้นเองครับ ดังนั้นการใช้งาน 7 Segment จะเหมือนกับการใช้งาน LED นั้นเอง แต่จะแตกต่างกันอย่างไร รองดูนะครับ



รูปที่2.28 แบบต่างๆ และ สัญลักษณ์

ที่ตัว ส่วนแสดงผล 7 Segment จะมีชื่อกำกับอยู่ (อันนี้ต้องจำให้ได้นะครับ) โดยจะไล่จาก A,B, C, D, E, F, G และจุด เป็นต้น

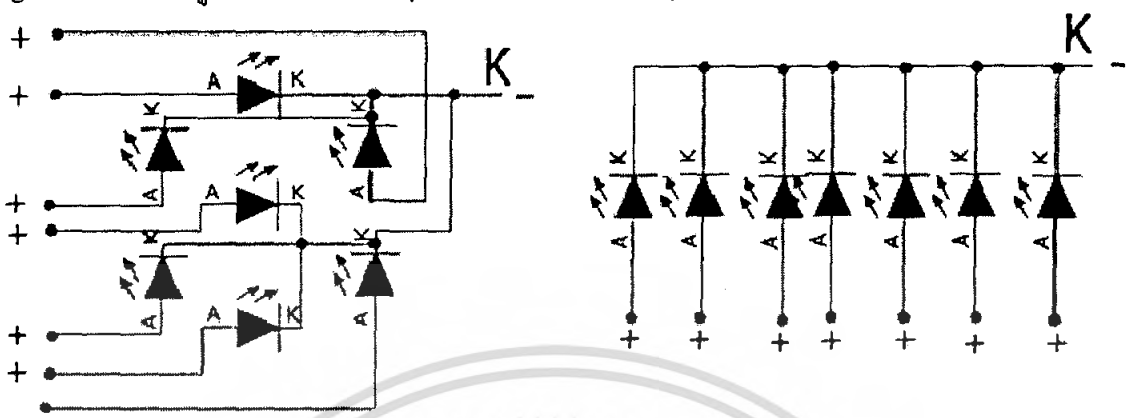


รูปที่2.29 แสดงตำแหน่งส่วนแสดงผล A- G

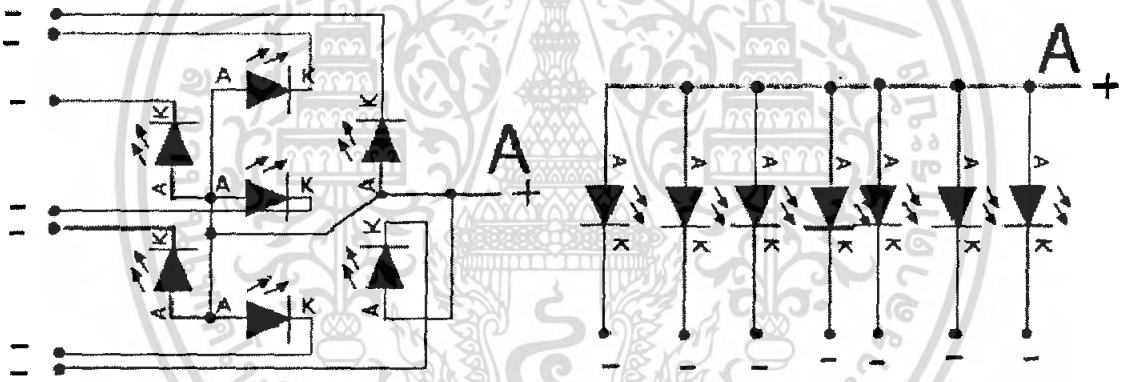
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อ LED ภายในตัว7 Segment

7 Segment นั้นจะมีอยู่ 2 คอมมอนหลักๆ คือ แบบคอมมอน A (อาโนท) และแบบคอมมอนK (คาโทด)



รูปที่2.30รูป แสดงการนำเอา LED มาต่อกัน แบบคแมมอน K



รูปที่2.31 แสดงการนำเอา LED มาต่อกัน แบบคอมมอน A

การต่อแบบคอมมอน A เราจะใช้ขั้วลบ (-) ป้อนให้ที่ขา A - G ส่วนไฟบวก (+) จะมาป้อนที่จุดรวมของขา A

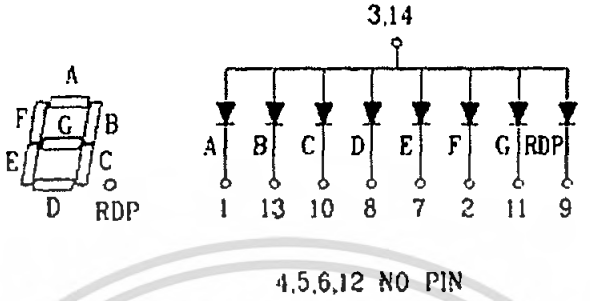
การต่อแบบคอมมอน K เราจะใช้ขั้วบวก (+) ป้อนให้ที่ขา A - G ส่วนไฟลบ (-) จะมาป้อนที่จุดรวมของขา K

จากรูปจะเห็นว่าเป็นการจำลองโดยการใช้ LED มาต่อกัน 8 ตัว จะได้เป็นเลข 8 แทนการใช้ 7 Segment ได้ครับ

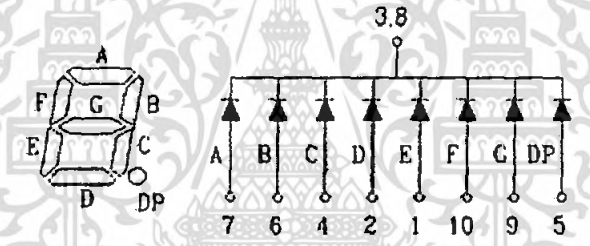
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดูสัญลักษณ์การต่อภายใน 7 Segment

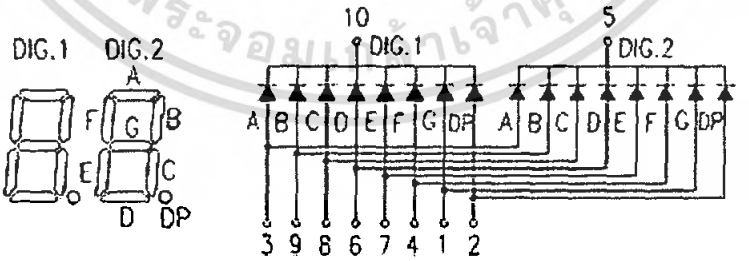
รูปดังกล่าวต่อไปนี้จะแสดงการต่อ LED ไว้ภายใน ซึ่งจะมีทั้งคอมมอน A และ K และแบบรวม โดยที่สัญลักษณ์ จะแสดงตำแหน่งของขา LED ไว้ให้ด้วย



รูปที่2.32แสดง 1 หลัก คอมมอน A ที่ขา 3 กับ 14 ส่วนขา 4,5,6,12 ไม่ได้ใช้

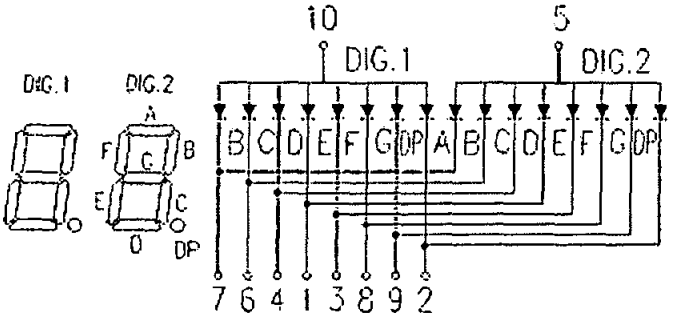


รูปที่2.33แสดง 1 หลัก คอมมอน K ที่ขา 3 กับ 8

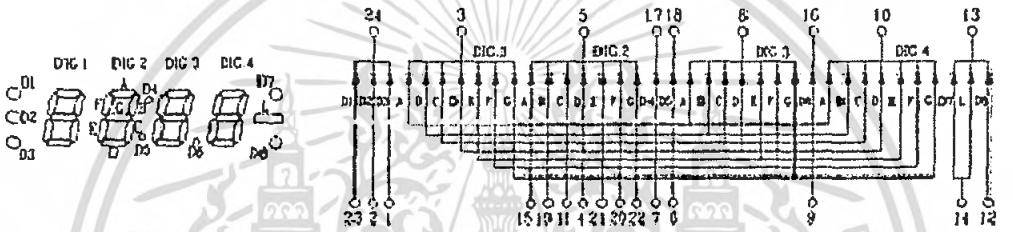


รูปที่2.34แสดง 2 หลัก คอมมอน K ที่ขา 10(ตัวที่1) กับ 5(ตัวที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.35แสดง 2 หลัก คอมมอน A ที่ขา 10(ตัวที่1) กับ 5(ตัวที่ 2)



รูปที่2.36 แสดง 4 หลัก คอมมอน K ที่ขา 3(ตัวที่1) กับ 5(ตัวที่ 2) กับ 8(ตัวที่3) กับ 10(ตัวที่ 4)

การ ตรวจสอบขาของ 7-Segment (ถ้าไม่มี datasheet)

โดยปกติถ้าเราไปซื้อ 7 Segment ตามร้านทั่วๆไปนั้นเขาจะไม่มี datasheet ครับ ซึ่งจะต้องเป็นหน้าที่ของเราเองครับ ว่าจะต้องตรวจสอบตำแหน่งขา ตรวจสอบคอมมอนให้แน่ใจเสียก่อนครับ ซึ่งการตรวจสอบก็ไม่ได้ยากเย็นอะไรนักหรอกครับแต่ต้องใช้เวลาและความอดทนนิดหน่อยเท่านั้นเอง (7Segment บางขาก็ไม่ได้ใช้งานครับ) ซึ่งจะแนะนำวิธีที่ผมเคยใช้อยู่บ่อยๆดังนี้

1.<u>การตรวจสอบโดยใช้มัลติมิเตอร์แบบเข็ม</u>

จะใช้ย่านวัดโอห์มในการวัด โดยปรับย่านวัดไปที่ X1 ก่อน จากนั้นใช้ที่วัด วัดไปที่ขาของ 7 segment เรื่อยๆ จนกว่าจะเจอว่าขาอะไรเป็นขารวม หรือขาคอมมอน หากแน่ใจแล้วว่าขาที่ได้เป็นขาคอมมอน ให้ดูที่มิเตอร์ว่าขาคอมมอนของเรานั้นต่อกับสายสีอะไรของมัลติมิเตอร์

ถ้าต่ออยู่กับสายสีดำ แสดงว่าเป็นคอมมอน A

ถ้าต่ออยู่กับสายสีแดง แสดงว่าเป็นคอมมอน K

** การจ่ายไฟของย่านวัดค่าโอห์มจะจ่ายสลับขั้วกัน

จากนั้นเมื่อเราหาได้แล้ว เราก็ทำการหาขาที่เหลือคือ ขา A - G และ จุด ต่อไปได้ไม่ยากเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. <u>การตรวจสอบโดยใช้ถ่านไฟฉายธรรมดา</u>

เราจะใช้ไฟประมาณ 3V ในการตรวจสอบ โดยทำแบบเดียวกับการใช้มิเตอร์ คือต้องหาขาร่วมให้ได้ก่อน และเมื่อแน่ใจแล้วว่าได้ขาร่วมหรือขาคอมมอนแล้วดูที่สายไฟว่าต่ออยู่กับขั้วไปอะไร

ถ้าต่ออยู่กับขั้วบวก(+) แสดงว่าเป็นคอมมอน A

ถ้าต่ออยู่กับขั้วลบ(-) แสดงว่าเป็นคอมมอน K

ซึ่งจะกลับกับขั้วของมัลติมิเตอร์

จากนั้นเมื่อเราหาได้แล้ว เราก็ทำการหาขาที่เหลือคือ ขา A - G และ จุด ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินการวิจัย

3.1 การออกแบบในส่วนของ ฮาร์ดแวร์

ใช้ Project Board ในการทำการทดลองในส่วนต่างๆ โดย Project Board นี้จะมีอุปกรณ์ที่ใช้ในการทดลอง 5 การทดลอง คือ 1 .LED 8 ดวง 2. DC Motor 3 . Stepping Motor 4 . 7-Segment และ 5 . Key Pad ในการทดลอง ส่ง ข้อมูลทาง Parallel Port และในการทดลองการส่งข้อมูล ผ่านทาง Serial Port

3.2 การออกแบบในส่วนของซอฟต์แวร์

การออกแบบในส่วนของ ซอฟต์แวร์ จะแบ่งเป็น 2ส่วน คือ ซอฟต์แวร์ของ 89S52ที่ใช้ในการแปลงข้อมูลจากโปรแกรม Visual Basic เพื่อให้ข้อมูลผ่านทาง Serial Port ออกทาง P1.0-P1.7 ของ Board 8051 SBC ส่วนที่ 2 จะเป็นการส่งข้อมูลผ่านทาง Parallel Port โดยตรง กับ Board ที่ได้ทำขึ้นมา

3.3 อุปกรณ์การทดลอง

3.3.1 LED

3.3.2 7-Segment ดังรูปที่ 3.2

รูปที่ 3.1 แสดงลักษณะของ 7-segment

3.3.3 สาย Parallel Port ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 แสดงลักษณะของ สายParallel Port

3.3.4 Stepping Motor ดังรูป

รูปที่ 3.3 แสดงลักษณะของ Steping Motor

3.3.5 KEY PAD

รูปที่3.4 แสดงลักษณะของ Keypad

3.3.6 IC 74 LS 48

3.3.7 Connector (Parallel Port ตัวผู้)

3.3.8 ตัวต้านทาน 270 ohm

3.3.9 Connector

3.4.0 Capacitor 10uF,470uF

3.4.1 Socket 16 ขา

3.4.2 RC jack

3.4.3 สายไฟ

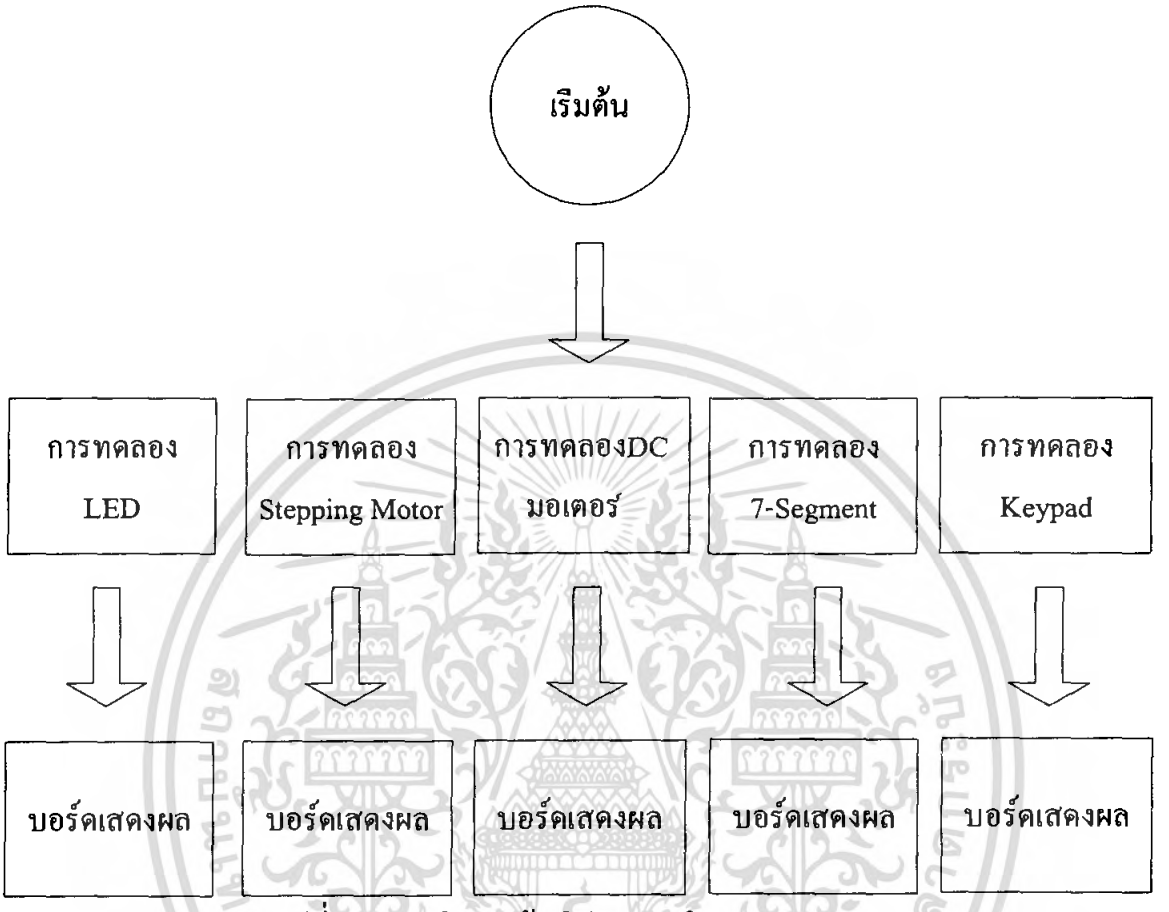
3.4.4 IC เบอร์ ULN2003

3.4.5 DC Motor

3.4.6 L 293D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart ของการทดลองใน Parallel Project Board



รูปที่3.5 แสดงโครงสร้างโปรแกรม ใน Parallel Project Board

3.4 วิธีการส่งข้อมูลให้ LED ติด

Output ของport คือ &378 และ&37A

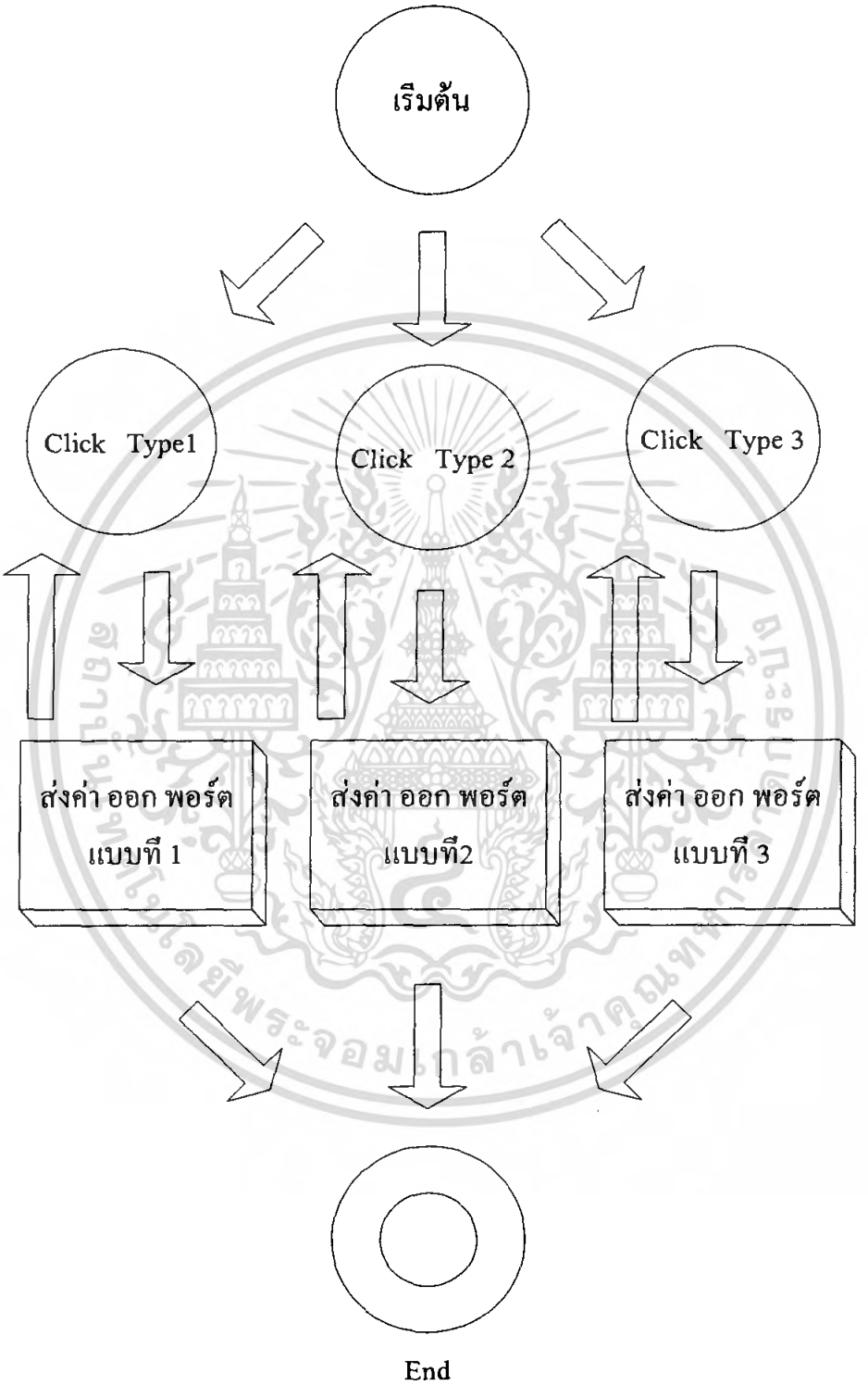
Input ของ Port คือ &379

Out pwrite1, &H1 -> outputจะออกจากขา D0 =1 D1=0 D2=0 D3=0 D4=0 D5=0 D6=0 D7=0

Out pwrite1, &H10 -> outputจะออกจากขา D0 =0 D1=0 D2=0 D3=0 D4=1 D5=0 D6=0 D7=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart แสดง การทำงานของ LED



รูปที่3.6 แสดงโครงสร้างโปรแกรม LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโค้ดของ Program เป็นดังนี้

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress As Integer) As
Integer
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal Value
As Integer)
Public pwrite1 As Integer
Public pwrite2 As Integer
Public pread As Integer
Private Sub Form_Load()
   Command1.Caption = "D0"
   Command2.Caption = "D1"
   Command3.Caption = "D2"
   Command4.Caption = "D3"
   Command5.Caption = "D4"
   Command6.Caption = "D5"
   Command7.Caption = "D6"
   Command8.Caption = "D7"
   Command9.Caption = "/C0"
   Command10.Caption = "/C1"
   Command11.Caption = "C2"
   Command12.Caption = "/C3"
   Label1.Caption = "input from port="
   Label1.Alignment = 2

   Timer1.Interval = 1
   pwrite1 = &H378
   pwrite2 = &H37A
   pread = &H379
End Sub
Private Sub command1_Click()
Out pwrite1, &H1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command2_Click()
Out pwrite1, &H2
End Sub

Private Sub Command3_Click()
Out pwrite1, &H4
End Sub

Private Sub Command4_Click()
Out pwrite1, &H8
End Sub

Private Sub Command5_Click()
Out pwrite1, &H10
End Sub

Private Sub Command6_Click()
Out pwrite1, &H20
End Sub

Private Sub Command7_Click()
Out pwrite1, &H40
End Sub

Private Sub Command8_Click()
Out pwrite1, &H80
End Sub

Private Sub Command9_Click()
Out pwrite2, &HA
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

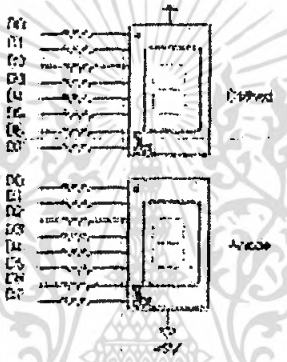
```
Private Sub command10_click()
Out pwrite2, &H9
End Sub

Private Sub command11_click()
Out pwrite2, &HF
End Sub

Private Sub command12_click()
Out pwrite2, &H3
End Sub
Private Sub timer1_timer()
   n = Inp(pread)
   Label1.Caption = "input from port=" & n
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การศึกษา 7- segment และ การส่งข้อมูลให้ 7-segment

ในทางด้านฮาร์ดแวร์ 7-segment LED นั้นเป็นอุปกรณ์ ทาง อิเล็กทรอนิกส์ที่มีหลอดแสดงผล LED จำนวน 7 ตัวบรรจุอยู่ในตัวถังเดียวกัน ใช้สำหรับการแสดงผลแบบตัวเลข ส่วนมากเราจะพบเห็นในเครื่องคิดเลข นาฬิการะบบดิจิตอล หัวจ่ายในปั๊มน้ำมัน และการแสดงตัวเลขบนสกอร์บอร์ดในสนามกีฬา เป็นต้น

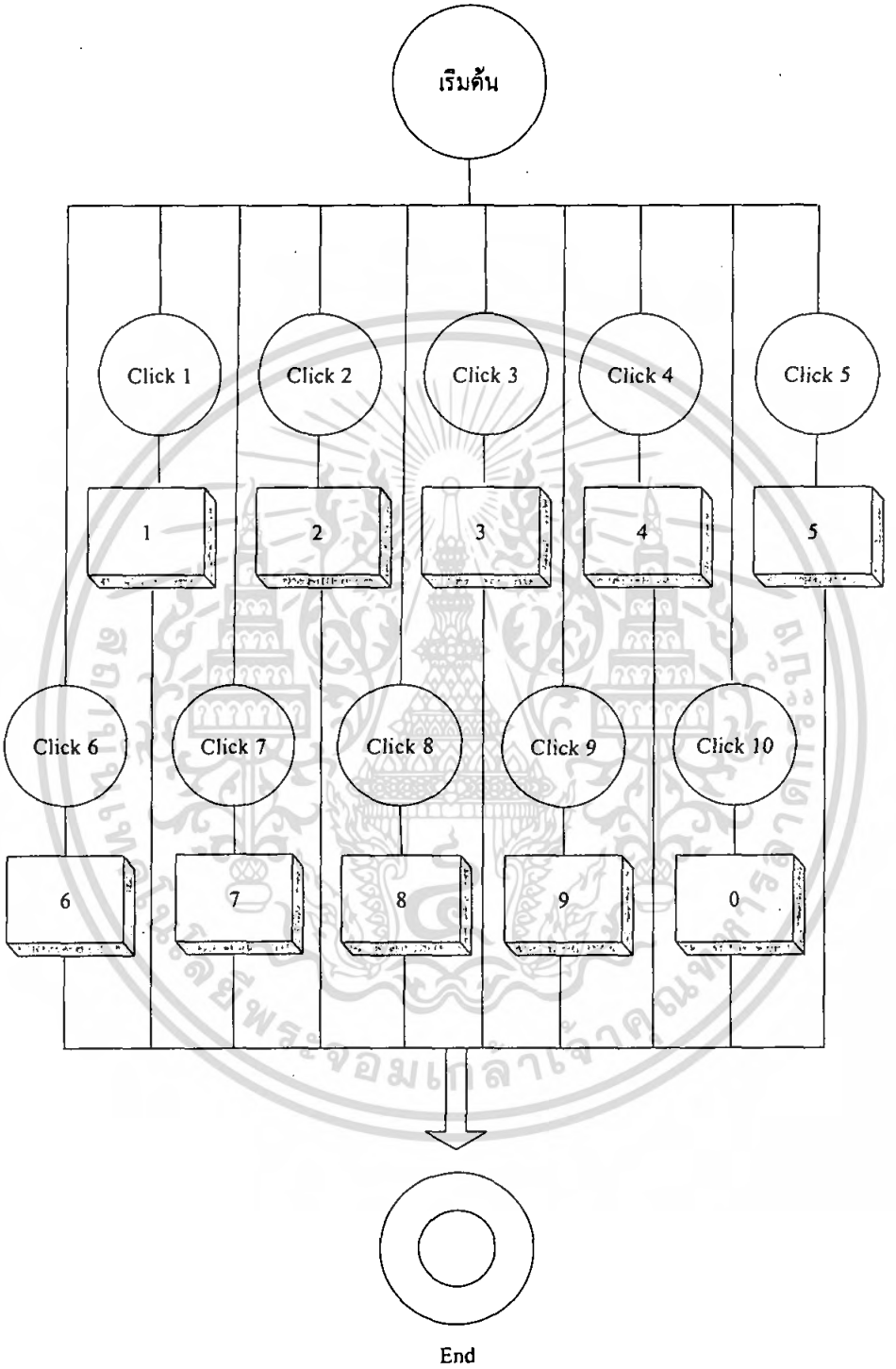
7-segment นั้นมีให้ใช้งาน 2 ชนิด คือ แบบคาโทดร่วม (Common Cathode) และแบบอาโนดร่วม (Common Anode) ซึ่งแต่ละเซกเมนต์จะมีชื่อขา ดังรูป



รูปที่ 3.7 แสดงลักษณะของ 7-segment แบบคาโทดร่วมและแบบอาโนดร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart แสดงการทำงาน ของ 7- Segment



รูปที่3.8 แสดงโครงสร้างโปรแกรม 7-Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเชื่อมต่อคอมพิวเตอร์เพื่อขับหลอดแสดงผลแบบ 7-segment นั้น สามารถกระทำได้หลายแบบหลายวิธี โดยโค้ดของโปรแกรมมีดังนี้

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal portaddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal portaddress As Integer, ByVal value
As Integer)
Public pwrite As Integer
Private Sub Command1_Click()
Out pwrite, &H1
End Sub

Private Sub Command2_Click()
Out pwrite, &H2
End Sub

Private Sub Command3_Click()
Out pwrite, &H3
End Sub

Private Sub Command4_Click()
Out pwrite, &H4
End Sub

Private Sub Command5_Click()
Out pwrite, &H5
End Sub

Private Sub Command6_Click()
Out pwrite, &H6
End Sub

Private Sub Command7_Click()
```

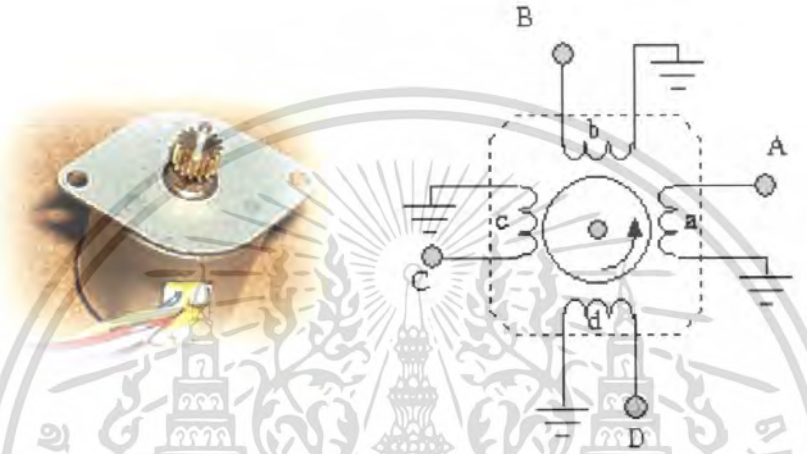
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Out pwrite, &H7
End Sub

Private Sub Command8_Click()
Out pwrite, &H8
End Sub

Private Sub Command9_Click()
Out pwrite, &H9
End Sub

Private Sub Command10_Click()
Out pwrite, &H0
End Sub
Private Sub Form_Load()
Command1.Caption = "1"
Command2.Caption = "2"
Command3.Caption = "3"
Command4.Caption = "4"
Command5.Caption = "5"
Command6.Caption = "6"
Command7.Caption = "7"
Command8.Caption = "8"
Command9.Caption = "9"
Command10.Caption = "0"
pwrite = &H378
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ศึกษาการทำงานของ stepping Motorและการส่งข้อมูลให้ Motorทำงาน

3.4.1 ศึกษาโครงสร้างของ Stepping Motor ซึ่งมีลักษณะดังรูป ซึ่งประกอบด้วย ขดลวด stator 4 ขด ล้อมรอบแกนหมุน



รูปที่ 3.9 แสดงลักษณะของ Stepping Motor

3.4.2 ศึกษาการควบคุม Stepping motor แบบ 4 เฟส ซึ่งมี 3 รูปแบบ คือ

3.4.2.1 ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)

ขั้นตอนการทำงาน

ตารางที่ 1 แสดงการทำงานของสเต็บมอเตอร์

step	เฟส4	เฟส3	เฟส2	เฟส1
Step1	ON	OFF	OFF	OFF
Step2	OFF	ON	OFF	OFF
Step3	OFF	OFF	ON	OFF
Step4	OFF	OFF	OFF	ON
Step5	ย้อนกลับ Step ที่ 1			
.....	.....			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

ขั้นตอนการทำงาน

ตารางที่ 1 แสดงการทำงานของสเต็บมอเตอร์

Step	เฟส4	เฟส3	เฟส2	เฟส1
Step1	ON	ON	OFF	OFF
Step2	OFF	ON	ON	OFF
Step3	OFF	OFF	ON	ON
Step4	ON	OFF	OFF	ON
Step5	ย้อนกลับ Step ที่ 1			

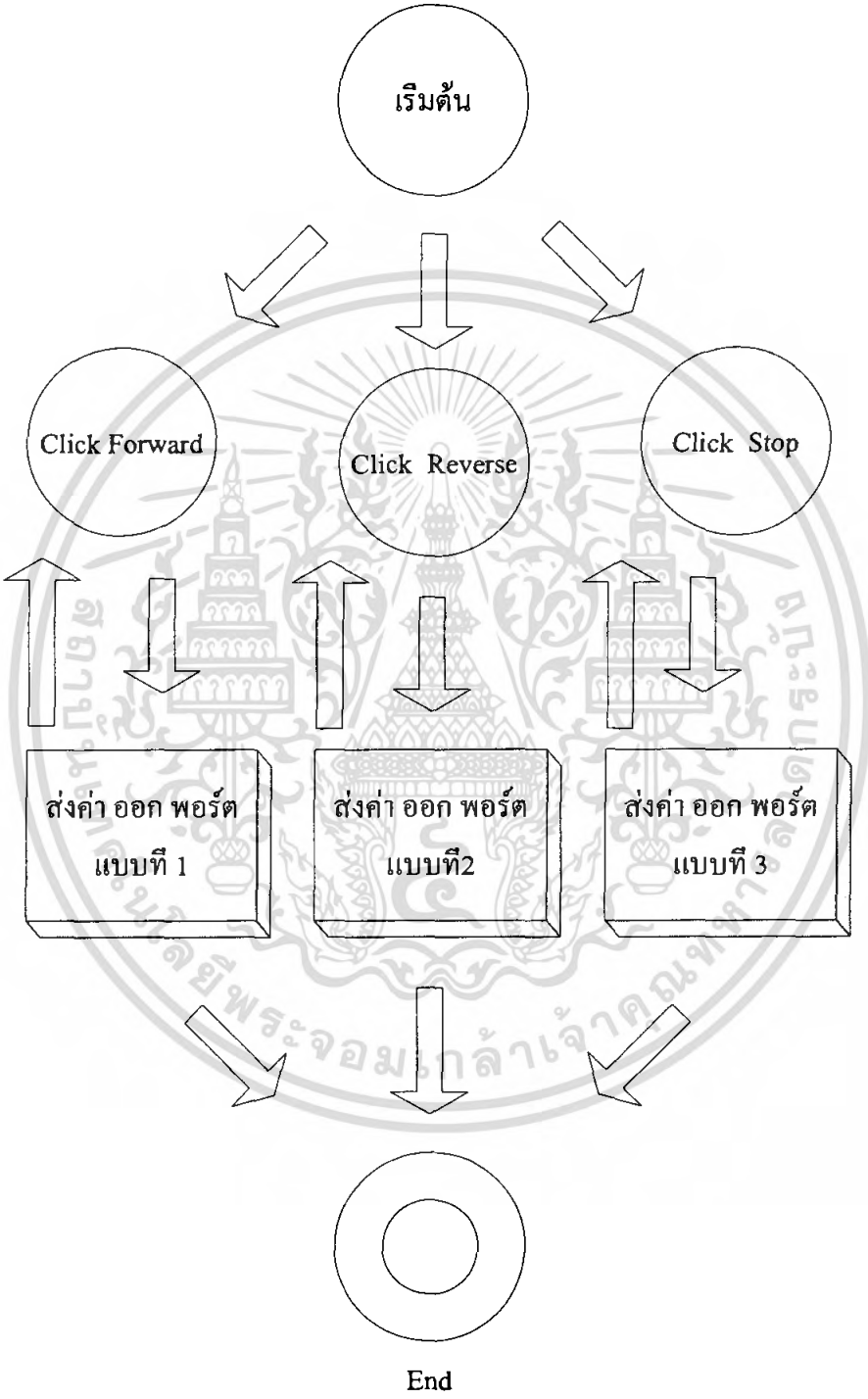
ขั้นตอนการทำงาน

ตารางที่ 1 แสดงการทำงานของสเต็บมอเตอร์

Step	เฟส4	เฟส3	เฟส2	เฟส1
Step1	ON	OFF	OFF	OFF
Step2	ON	ON	OFF	OFF
Step3	OFF	ON	OFF	OFF
Step 4	OFF	ON	ON	OFF
Step5	OFF	OFF	ON	ON
Step6	OFF	OFF	ON	ON
Step7	OFF	OFF	ON	ON
Step8	ON	OFF	OFF	OFF
Step9	ย้อนกลับ Step ที่ 1			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart แสดง การทำงานของ Stepping Motor



รูปที่3.10 แสดงโครงสร้างโปรแกรม Stepping Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ด ของโปรแกรมเป็นดังนี้

```
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal portaddress As Integer, ByVal value
As Integer)
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Public pwrite As Integer
Public t As Integer
Private Sub Command1_Click()
   Timer1.Enabled = True
   Timer2.Enabled = False
   Timer3.Enabled = False
End Sub
Private Sub Command2_Click()
   Timer1.Enabled = False
   Timer2.Enabled = True
   Timer3.Enabled = False
End Sub
Private Sub command3_click()
   Timer1.Enabled = False
   Timer2.Enabled = False
   Timer3.Enabled = True
End Sub
Private Sub Form_Load()
Command1.Caption = "type1"
Command2.Caption = "type2"
Command3.Caption = "type3"
Timer1.Interval = 10000
Timer2.Interval = 10000
Timer3.Interval = 10000
'Timer1.Enabled = False
'Timer2.Enabled = False
'Timer3.Enabled = False
HScroll1.Min = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HScroll1.Max = 700
HScroll1.value = 350
pwrite = &H378
End Sub
Private Sub Timer1_Timer()
Do
  DoEvents
    Out pwrite, &H1
    Sleep (t)
    Out pwrite, &H2
    Sleep (t)
    Out pwrite, &H4
    Sleep (t)
    Out pwrite, &H8
    Sleep (t)
    Out pwrite, &H10
    Sleep (t)
    Out pwrite, &H20
    Sleep (t)
    Out pwrite, &H40
    Sleep (t)
    Out pwrite, &H80
    Sleep (t)
  Loop
End Sub

Private Sub Timer2_Timer()
  Do
  DoEvents
    Out pwrite, &H30
    Sleep (t)
    Out pwrite, &H60
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Sleep (t)
    Out pwrite, &HC0
    Sleep (t)
    Out pwrite, &H81
    Sleep (t)
    Out pwrite, &H3
    Sleep (t)
    Out pwrite, &H6
    Sleep (t)
    Out pwrite, &HC
    Sleep (t)
    Out pwrite, &H18
    Sleep (t)
  Loop
End Sub

Private Sub Timer3_Timer()
  Do
  DoEvents
    Out pwrite, &H50
    Sleep (t)
    Out pwrite, &HA0
    Sleep (t)
    Out pwrite, &H41
    Sleep (t)
    Out pwrite, &H82
    Sleep (t)
    Out pwrite, &H5
    Sleep (t)
    Out pwrite, &HA
    Sleep (t)
    Out pwrite, &H14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

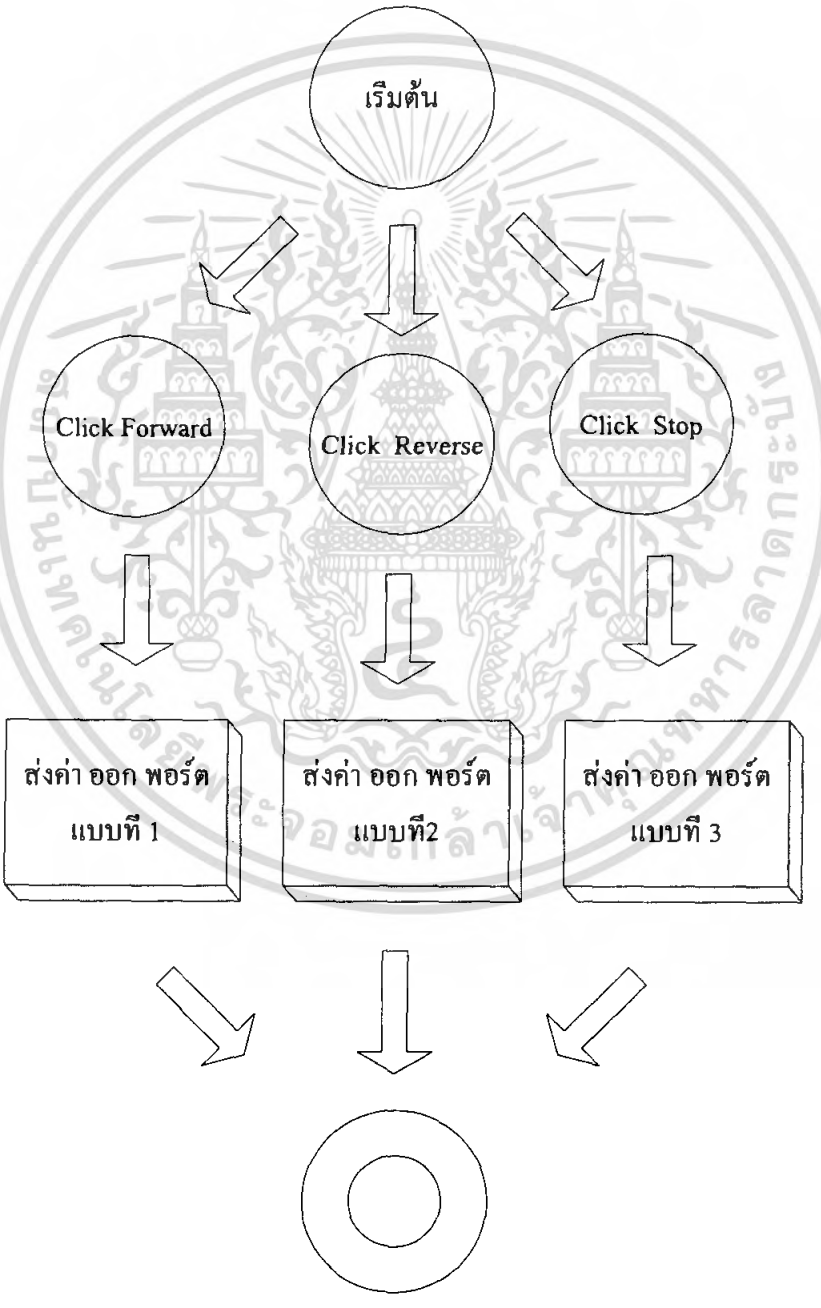
```
        Sleep (t)
        Out pwrite, &H82
        Sleep (t)
    Loop
End Sub
Private Sub HScroll1_Change()
        t = HScroll1.value
End Sub
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การส่งข้อมูลให้ มอเตอร์กระแสตรงทำงาน

การส่งข้อมูลให้มอเตอร์ทำงานนั้น จะใช้ IC เบอร์ L 293D โดยที่ Code ของ Program จะสั่งให้ข้อมูลออกทางขา 2 และขา 3 ของ Parallel Port โดยใช้ L293D ในการเพิ่มกระแส

Flow Chart แสดงการทำงานของมอเตอร์ กระแสตรง



รูปที่3.11 แสดงโครงสร้างโปรแกรม มอเตอร์ กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal portaddress As Integer, ByVal value
As Integer)
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Public pwrite As Integer
Public t As Integer

Private Sub Command5_Click()
  Do
    DoEvents
    Out pwrite, &H1
    Sleep (t)
    Out pwrite, &H2
    Sleep (t)
    Out pwrite, &H4
    Sleep (t)
    Out pwrite, &H8
    Sleep (t)
  Loop
End Sub

Private Sub Command6_Click()
  Do
    DoEvents
    Out pwrite, &H8
    Sleep (t)
    Out pwrite, &H4
    Sleep (t)
    Out pwrite, &H2
    Sleep (t)
    Out pwrite, &H1
    Sleep (t)
  Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

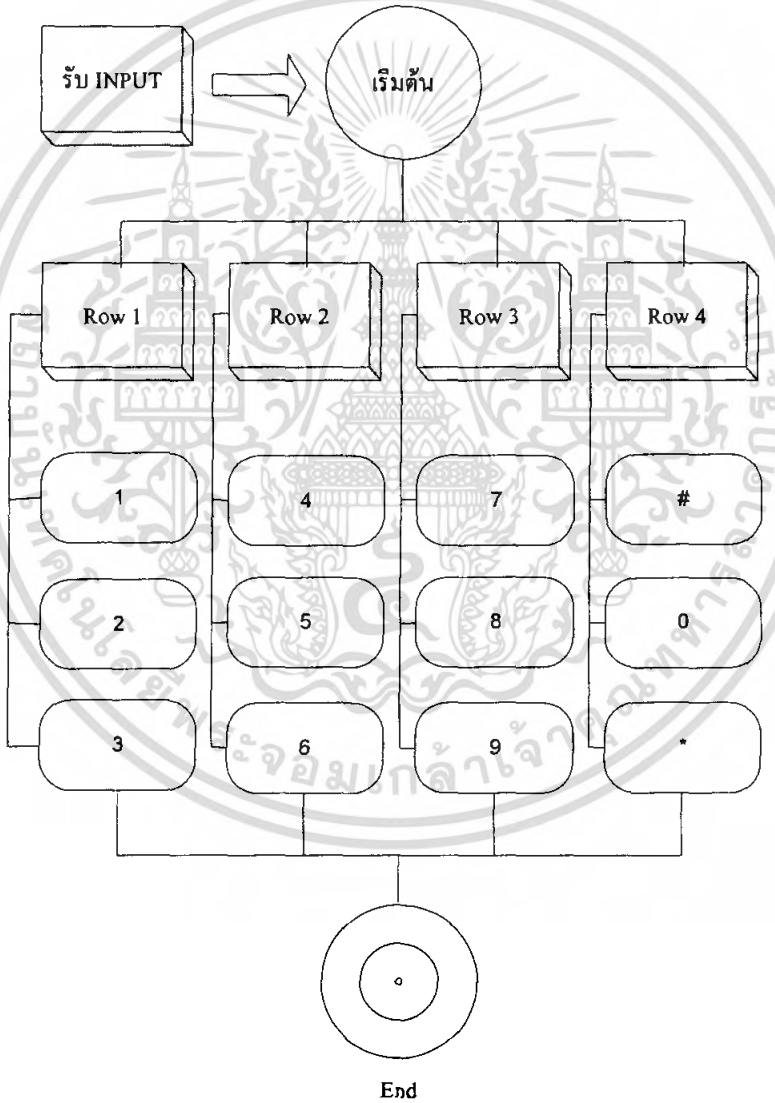
```
End Sub

Private Sub Command7_Click()
   Out pwrite, &H0
End Sub

Private Sub Form_Load()
Command5.Caption = "<--reverse"
Command6.Caption = "forward-->"
Command7.Caption = "Stop"
HScroll1.Min = 1
HScroll1.Max = 100
HScroll1.value = 50
Timer1.Interval = 10000
pwrite = &H378
End Sub

Private Sub Timer1_Timer()
   t = HScroll1.value
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การส่งข้อมูลให้ Key Pad ทำงาน

เป็นการส่งข้อมูลโดยให้เป็นค่า (0) ที่เริ่มต้นขา2 ของ แล้วไป ขา3 ขา4 และ ขา5โดย ทำการวนลูปสแกน Parallel Port โดยที่ขารับข้อมูลคือ 12 , 13 , 15 ข้อมูลที่เข้าไปในData Port คือ 1,2,3,4,5,6,7,8,9,0,*,#

Flow Chart แสดงการทำงานของ Key Pad



รูปที่3.12 แสดงโครงสร้างโปรแกรม Key Pad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ทดลองคือ

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" _
(ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal value As Integer)
Public pwrite As Integer
Public pread As Integer
Public i As Integer
Public n As Integer
Private Sub Form_Activate()
    Do
      i = &HE
      Out pwrite, &HE
      Call delay
      Call row1
      i = &HD
      Out pwrite, &HD
      Call delay
      Call row2
      i = &HB
      Out pwrite, &HB
      Call delay
      Call row3
      i = &H7
      Out pwrite, &H7
      Call delay
      Call row4
    Loop
End Sub

Private Sub Form_Load()
    Label1.Caption = "input from switch:"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

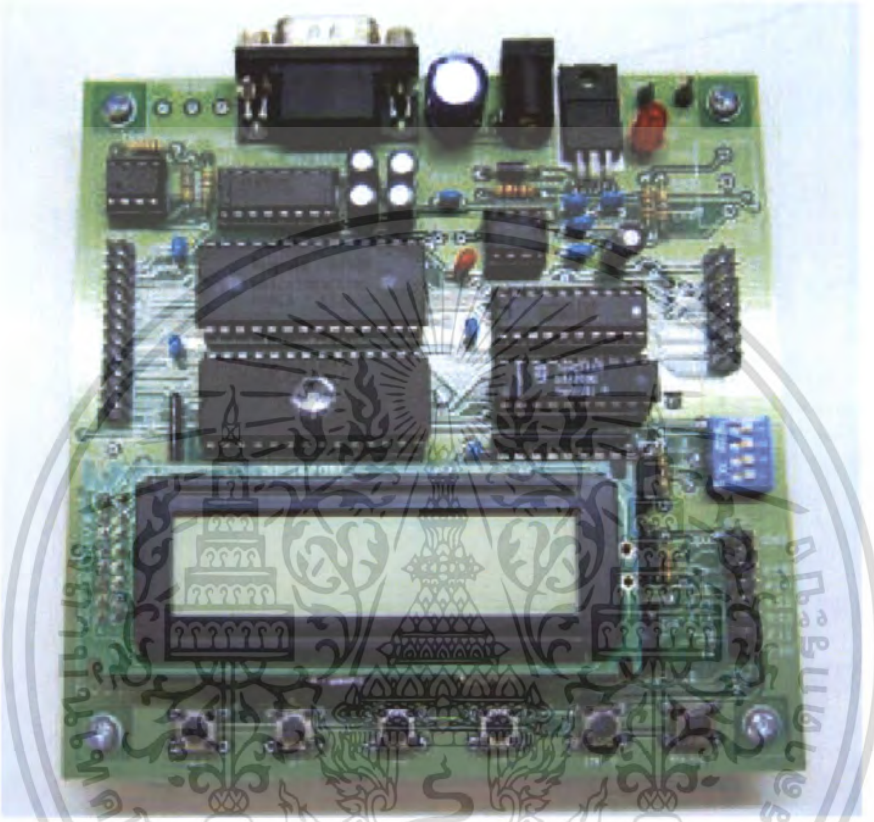
```
    Label2.Caption = "DaTa"
    pwrite = &H378
    pread = &H379
    Timer1.Interval = 1
End Sub

Private Sub Timer1_Timer()
  n = Inp(pread)
End Sub

Sub delay()
  For i = 1 To 32000
  DoEvents
  Next i
End Sub
Sub row1()
    Select Case n
      Case 119
        Label2.Caption = "1"
      Case 111
        Label2.Caption = "2"
      Case 95
        Label2.Caption = "3"
    End Select
End Sub
Sub row2()
    Select Case n
      Case 119
        Label2.Caption = "4"
      Case 111
        Label2.Caption = "5"
      Case 95
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Label2.Caption = "6"
        End Select
    End Sub
    Sub row3()
        Select Case n
            Case 119
                Label2.Caption = "7"
            Case 111
                Label2.Caption = "8"
            Case 95
                Label2.Caption = "9"
        End Select
    End Sub
    Sub row4()
        Select Case n
            Case 119
                Label2.Caption = "*"
            Case 111
                Label2.Caption = "0"
            Case 95
                Label2.Caption = "#"
        End Select
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

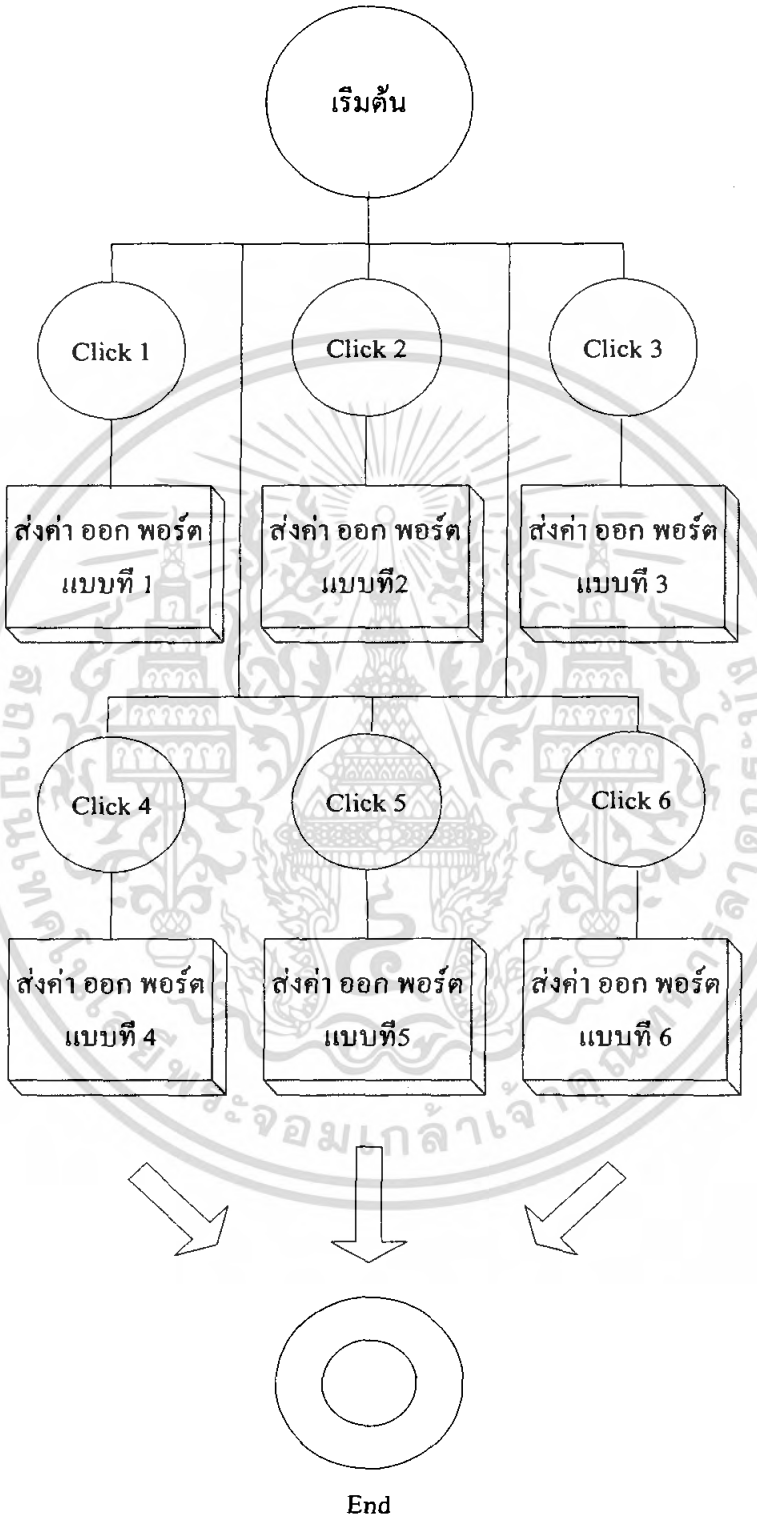
3.9 การส่งข้อมูลผ่าน Serial Port โดยใช้ AT89S52 Project Board ใน Visual Basic



รูปที่3.13 แสดงลักษณะของ AT89S52 Project Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart แสดงการทำงานของโปรแกรม LED 8 หลอด ที่มี Output 6 แบบ จาก Serial Port



รูปที่3.14 แสดงโครงสร้างโปรแกรม LED 8 หลอด จาก Serial Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code Visual Basic เขียนผ่าน AT89S52 Project Board

โดยที่ MSComm1.Output = "5" หมายความว่า ส่งรหัส ASCII ออก Serial Port

เช่น 5 คือ 35 ในเลขฐาน 16 ----------> 3 5

0011 0101

โดยที่ program visual Basic จะส่ง ASCII Code ไปที่ AT89S52 Project Board แล้วทำการส่ง Output ออกทาง P1.0-P1.7 ของ AT89S52 ซึ่งถ้าต้องการ output แบบใด ก็จะต้องเทียบกับรหัส ASCII

```
Private Sub Command1_Click()
MSComm1.Output = "0"
End Sub
Private Sub Command2_Click()
MSComm1.Output = "1"
End Sub
Private Sub Command3_Click()
MSComm1.Output = "2"
End Sub
Private Sub Command4_Click()
MSComm1.Output = "3"
End Sub
Private Sub Command5_Click()
MSComm1.Output = "4"
End Sub
Private Sub Command6_Click()
MSComm1.Output = "5"
End Sub
Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.Settings = "9600,n,8,1"
MSComm1.PortOpen = True
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

รูปที่3.15 แสดงรหัส ASCII CODE เทียบกับเลขฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและอภิปรายผลการทดลอง

4.1ผลการทดลองของ Parallel Project board

จากการทดลองใน Parallel project board ได้ผลตามโปรแกรมที่ได้ทำการเขียนเป็นตัวอย่างได้โดยที่การทดลองที่ 1 เป็น LED 8 ดวง จะมีการกระพริบได้ทั้ง 3 แบบ ตามการเขียน Program
การทดลองที่ 2 เป็นการบังคับทิศทางของ Motor กระแสตรง ให้เคลื่อนที่ไปทางซ้ายเมื่อกด reverseและไปทางขวาเมื่อกด Forward การทดลองที่ 3 เป็นทดลองให้ Stepping Motor เคลื่อนที่ในทิศทาง Forward และ Reverse ตาม Program เช่นกัน การทดลองที่ 4 เป็นการทดลอง กด 7 Segment ติดตามตัวเลขในProgram
การทดลองที่ 5 เป็นการทดลองการรับค่า จาก Key Pad โดยให้ค่าแสดง เข้ามาในตัว Program Visual Basic
ทุกการทดลอง มีตัวอย่างการเขียน Program และCode ใน CD
ท้ายเล่ม

4.2ผลการทดลองในบอร์ดเรียนรู้ 8051 SBC

รูปที่ 4.1 บอร์ด 8952 SBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการใช้ Microcontroller 89S52 ในการอัดโปรแกรมเพื่อแปลง สัญญาณจากProgram Visual Basic ซึ่งส่งเข้ามาทาง RS232 ผ่าน Max232 เพื่อแปลงระดับแรงดันจาก +/- 12 Volt เป็น 5 Volt

4.3ผลการทดลองในการส่งข้อมูลทาง Serial Port

ผลการทดลองในการส่งข้อมูลทาง Serial Port พบว่า การส่งข้อมูล จะส่งเป็น รหัส ASCII จากหลักการทำงานของ AT89S52 Project Board ที่รับค่าจาก Serial Port จาก Program Visual Basic แล้ว ทำการเรียงสัญญาณ ขนาด 8 bit ที่ รับมาจาก Program Visual Basic แล้วส่งค่าออก Port P1.0-P1.7 ของ AT89S52 Project Board สามารถให้สัญญาณตาม Program ที่เขียนขึ้นได้
โดยสามารถแสดงหลักการทำงานโดย Flow Chart ข้างล่าง
โดยที่คำสั่งจะเป็นดังเช่นตัวอย่างนี้

33 = 0011 0011

3 3

0 0 1 1 0 0 1 1

```
Private Sub Command4_Click()
MSComm1.Output = "3"
End Sub
```

34 = 0011 0100

3 4

0 0 1 1 0 1 0 0

```
Private Sub Command5_Click()
MSComm1.Output = "4"
End Sub
```

35 = 0011 0101

3 5

0 0 1 1 0 1 0 1

```
Private Sub Command6_Click()
MSComm1.Output = "5"
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

จากคำสั่ง MScomm1.Output จะเป็นตัวกำหนด Port การส่งข้อมูลให้ส่งที่ Com Port หรือ RS 232 นั่นเอง โดยที่ Output ที่ออกจะถูกกำหนดอยู่ใน " " ซึ่งเมื่อใส่ตัวเลขหรือตัวอักษรลงในช่องจะส่งข้อมูลเป็นรหัส ASCII ซึ่งต้องเทียบกับตารางเพื่อแปลงเป็นเลขฐาน 16 หรือเลขฐาน 2 (8 bit)

โดยที่ รหัส ASCII จะต้องดูจากดรูปที่4.2

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

รูปที่4.2แสดง รหัส ASCII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

สรุปผลการทดลองและข้อเสนอแนะ

โครงงานพิเศษนี้สามารถส่งข้อมูลจาก Visual Basic มา คอนโทรล Hardware โดยผ่าน Serial Port และ Parallel Port ได้ โดยการคอนโทรล Serial Port นั้นจะต้องเขียน Program ภาษา C ใช้ Keil คอมไพล์ และอัดลง AT89S52 โดยให้ AT89S52 มีหน้าที่ในการเรียง สัญญาณ ที่ออกจาก Visual Basic เข้า 89S52 SBC โปรเจคบอร์ดแล้วแสดงผลออก LED ที่ ขา P1.0-P1.7

ในการคอนโทรล ผ่าน Parallel Port นั้นจะส่งข้อมูลเป็น Hight(1) และ Low(0) โดยการทดลองนี้ได้สร้างบอร์ดที่ใช้เฉพาะกับการส่งข้อมูลทาง Parallel Port ในโปรแกรม Visual Basic ซึ่งมีแรงดันไฟที่ +5 volt

ข้อเสนอแนะและแนวทางพัฒนา

1. สามารถที่จะเขียน program อื่นๆ ในการแปลง รหัส ASCII เป็นเลขฐาน 16 เพื่อใช้ในการคอนโทรล Hardware ได้
2. การเขียน Program Visual Basic สามารถ เขียนคอนโทรลโดยผ่านสายส่งข้อมูลอื่นๆได้ อีกเช่น *RJ* 45, RS 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ธาริน สิทธิธรรมชารี. **สร้างโปรแกรมบน Windows ด้วย Visual Basic Version 6** กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2548.

www.logix4u.net/inpout32.html

www.thaieasyelec.com

www.vbthailand.com

www.wara.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

January 1992

DM74LS48
BCD to 7-Segment Decoder

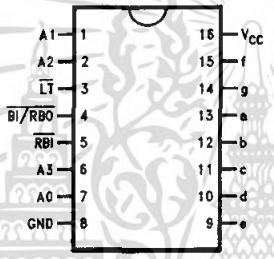
General Description

The 'LS48 translates four lines of BCD (8421) input data into the 7-segment numeral code and provides seven corresponding outputs having pull-up resistors, as opposed to totem pole pull-ups. These outputs can serve as logic signals, with a HIGH output corresponding to a lighted lamp segment, or can provide a 1.3 mA base current to npn lamp driver transistors. Auxiliary inputs provide lamp test, blanking and cascadable zero-suppression functions.

The 'LS48 decodes the input data in the pattern indicated in the Truth Table and the segment identification illustration.

Connection Diagram

Dual-In-Line Package



TL/F/10172–1

Order Number DM74LS48M or DM74LS48N
See NS Package Number M16A or N16E

©1995 National Semiconductor Corporation

RRD-B30M105/Printed in U. S. A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM74LS48			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Output Current			−50	μA
I_{OL}	Low Level Output Current			6.0	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} Min, I_{OH} = Max, V_{IL} = Max	2.4			V
I_{OFF}	Output High Current Segment Outputs	V_{CC} = Min, V_O = 0.85V	−1.3			mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min			0.5	V
		I_{OL} = 2.0 mA, V_{CC} = Min			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max, V_O = 0V at $\overline{BI}/\overline{RBO}$ (Note 2)	−0.3		−2	mA
I_{CCH}	Supply Current	V_{CC} = Max, V_{IN} = 4.5V			38	mA

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	C_L = 15 pF		Units
		Min	Max	
t_{PLH} t_{PHL}	Propagation Delay Time A_n to a–g		100 100	ns
t_{PLH} t_{PHL}	Propagation Delay Time $\overline{RBI}$ to a–f		100 100	ns

Note: $\overline{LT}$ = HIGH, A_0–A_3 = HIGH.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Numerical Designations—Resultant Displays

TL/F/10172–4

Truth Table

Decimal Or Function	Inputs						Outputs							
	$\overline{LT}$	$\overline{RBI}$	A_3	A_2	A_1	A_0	$\overline{BI/RBO}$	a	b	c	d	e	f	g
0 (Note 1)	H	H	L	L	L	L	H	H	H	H	H	H	H	L
1 (Note 1)	H	X	L	L	L	H	H	L	H	H	L	L	L	L
2	H	X	L	L	H	L	H	H	H	L	H	H	L	H
3	H	X	L	L	H	H	H	H	H	H	H	L	L	H
4	H	X	L	H	L	L	H	L	H	H	L	L	H	H
5	H	X	L	H	L	H	H	H	L	H	H	L	H	H
6	H	X	L	H	H	L	H	L	L	H	H	H	H	H
7	H	X	L	H	H	H	H	H	H	H	L	L	L	L
8	H	X	H	L	L	L	H	H	H	H	H	H	H	H
9	H	X	H	L	L	H	H	H	H	H	L	L	H	H
10	H	X	H	L	H	L	H	L	L	L	H	H	L	H
11	H	X	H	L	H	H	H	L	L	H	H	L	L	H
12	H	X	H	H	L	L	H	L	H	L	L	L	H	H
13	H	X	H	H	L	H	H	H	L	L	H	L	H	H
14	H	X	H	H	H	L	H	L	L	L	H	H	H	H
15	H	X	H	H	H	H	H	L	L	L	L	L	L	L
$\overline{BI}$ (Note 2)	X	X	X	X	X	X	L	L	L	L	L	L	L	L
$\overline{RBI}$ (Note 3)	H	L	L	L	L	L	L	L	L	L	L	L	L	L
$\overline{LT}$ (Note 4)	L	X	X	X	X	X	H	H	H	H	H	H	H	H

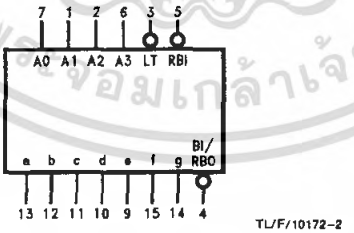
Note 1: $\overline{BI/RBO}$ is wired-AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$). The blanking out ($\overline{BI}$) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{RBI}$) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW.

Note 2: When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level, regardless of the state of any other input condition.

Note 3: When ripple-blanking input ($\overline{RBI}$) and inputs A_0, A_1, A_2, and A_3 are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a LOW level and the ripple-blanking output ($\overline{RBO}$) goes to a LOW level (response condition).

Note 4: When the blanking input/ripple-blanking output ($\overline{BI/RBO}$) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a HIGH level.

Logic Symbol

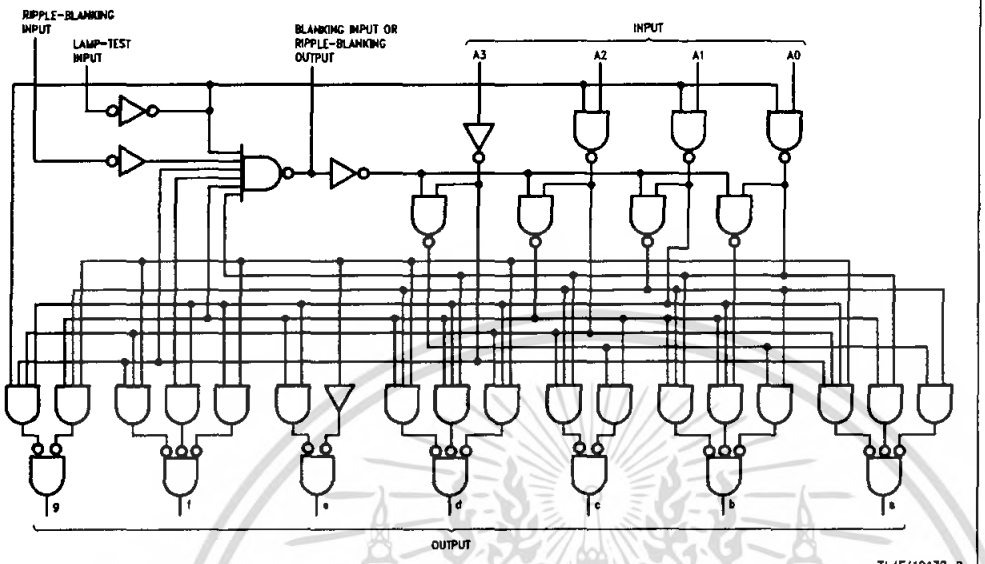


TL/F/10172–2

V_{CC} = Pin 16
GND = Pin 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

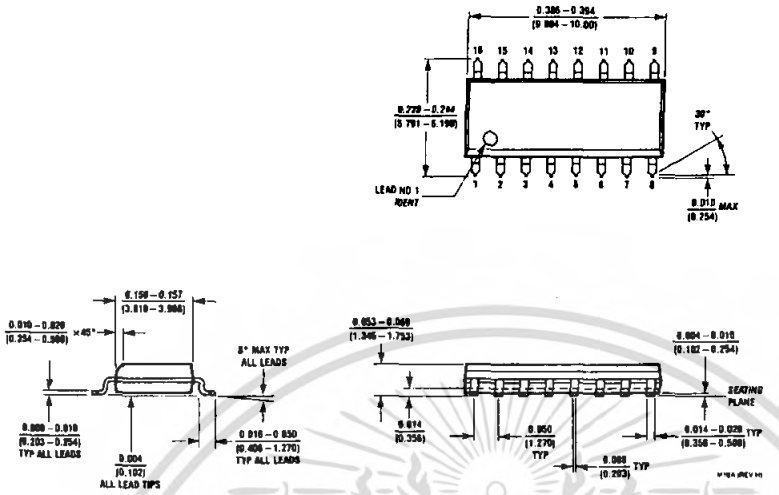
Logic Diagram



TL/F/10172–3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

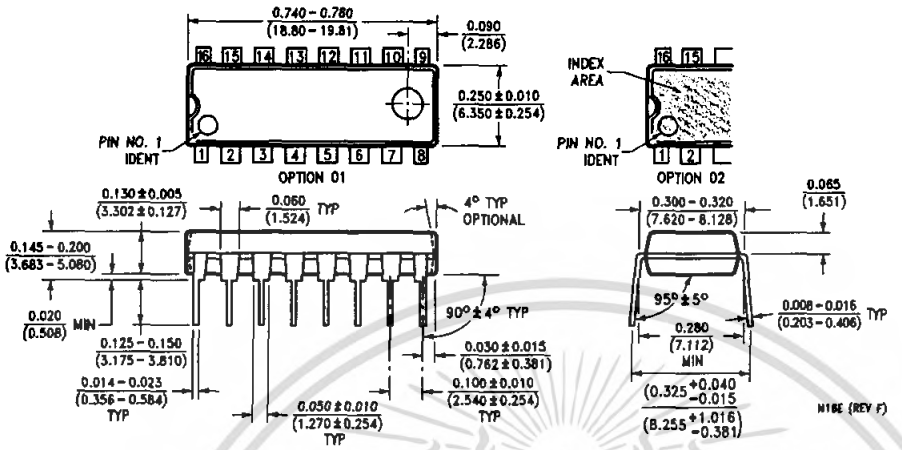
Physical Dimensions inches (millimeters)



16-Lead Small Outline Molded Package (M)
Order Number DM74LS48M
NS Package Number M16A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) (Continued)



16-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS48N
NS Package Number N16E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This datasheet has been downloaded from:

www.DatasheetCatalog.com

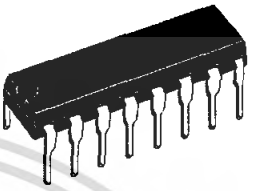
Datasheets for electronic components.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ULN2001A-ULN2002A
ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A

SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

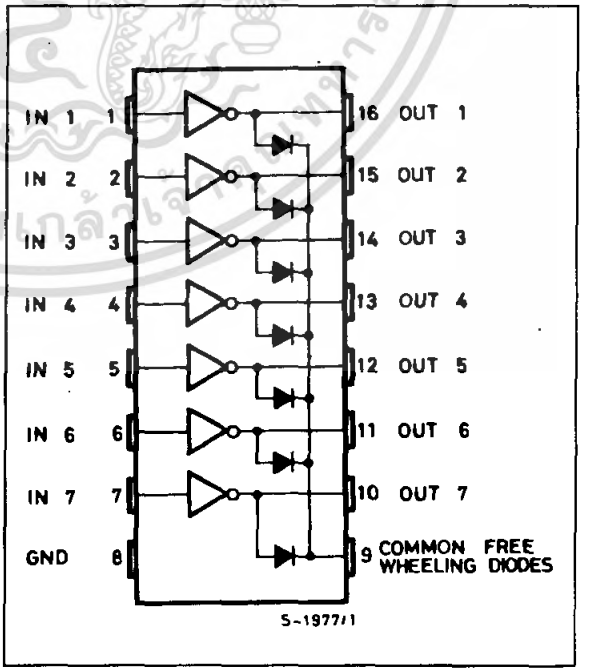
The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6–15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal printheads and high power buffers.

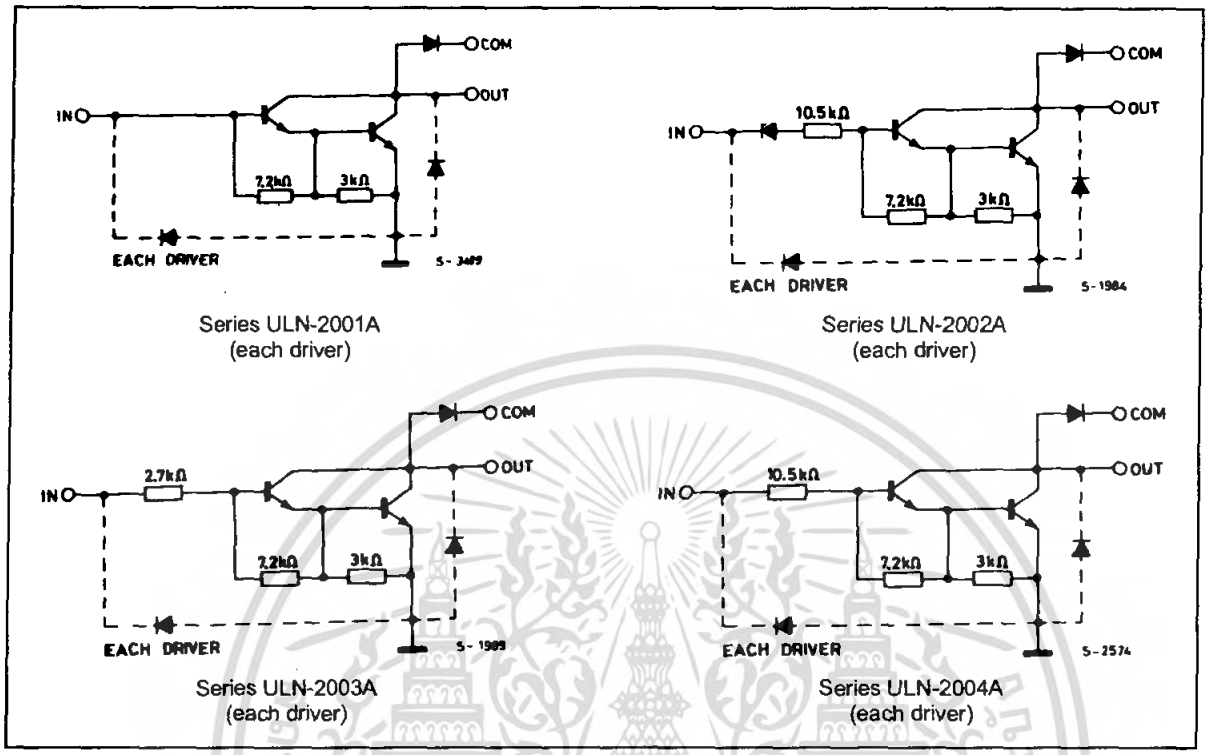
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM



Series ULN-2001A (each driver)

Series ULN-2002A (each driver)

Series ULN-2003A (each driver)

Series ULN-2004A (each driver)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter		DIP16	SO16	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	70	120	°C/W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V			50	µA	1a
		T_{amb} = 70°C, V_{CE} = 50V			100	µA	1a
		T_{amb} = 70°C for ULN2002A V_{CE} = 50V, V_i = 6V			500	µA	1b
		for ULN2004A V_{CE} = 50V, V_i = 1V			500	µA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA		0.9	1.1	V	2
		I_C = 200 mA, I_B = 350µA		1.1	1.3	V	2
		I_C = 350mA, I_B = 500µA		1.3	1.6	V	2
$I_{i(on)}$	Input Current	for ULN2002A, V_i = 17V		0.82	1.25	mA	3
		for ULN2003A, V_i = 3.85V		0.93	1.35	mA	3
		for ULN2004A, V_i = 5V		0.35	0.5	mA	3
		V_i = 12V		1	1.45	mA	3
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2V for ULN2002A I_C = 300mA			13	V	5
		for ULN2003A I_C = 200mA			2.4		
		I_C = 250mA			2.7		
		I_C = 300mA			3		
		for ULN2004A I_C = 125mA			5		
		I_C = 200mA			6		
		I_C = 275mA			7		
		I_C = 350mA			8		
h_{FE}	DC Forward Current Gain	for ULN2001A V_{CE} = 2V, I_C = 350mA	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
I_R	Clamp Diode Leakage Current	V_R = 50V			50	µA	6
		T_{amb} = 70°C, V_R = 50V			100	µA	6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEST CIRCUITS

Figure 1a.

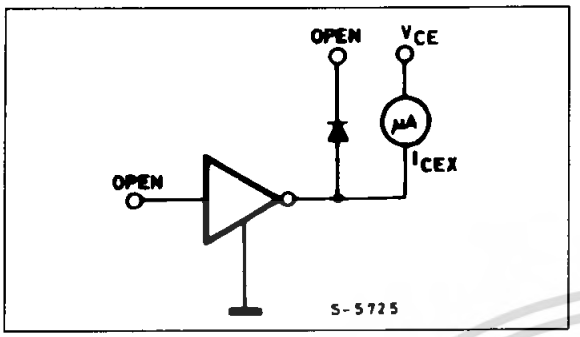


Figure 1b.

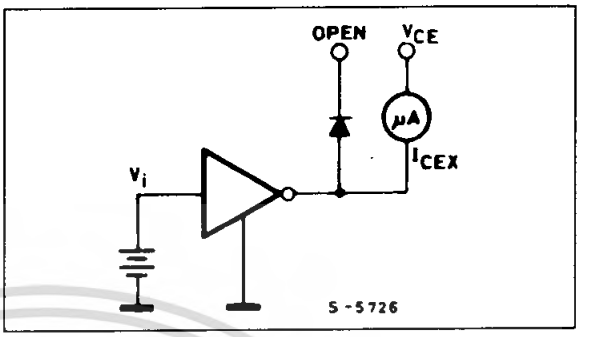


Figure 2.

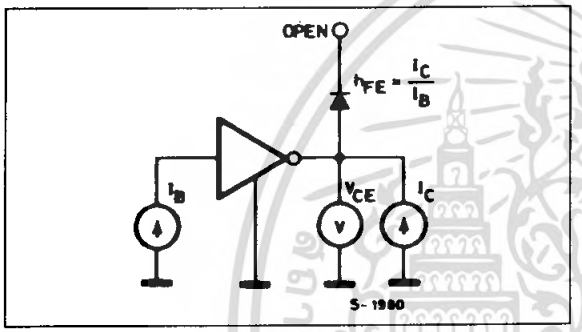


Figure 3.

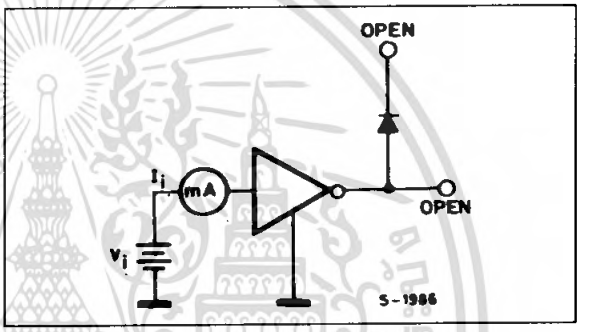


Figure 4.

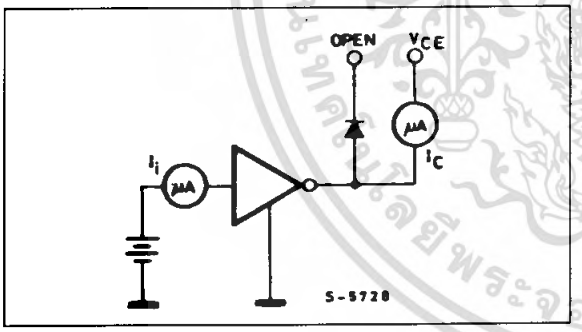


Figure 5.

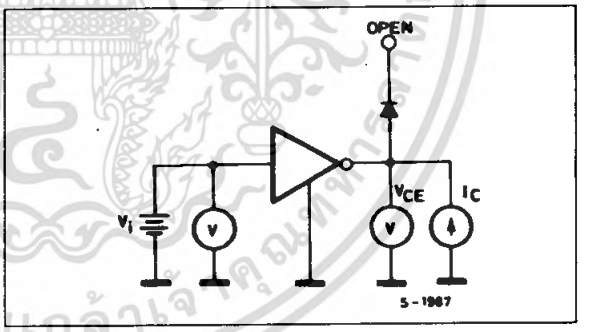


Figure 6.

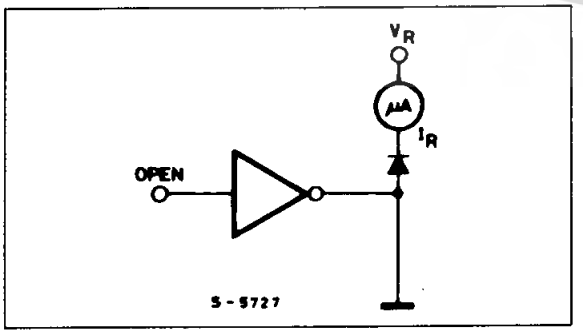
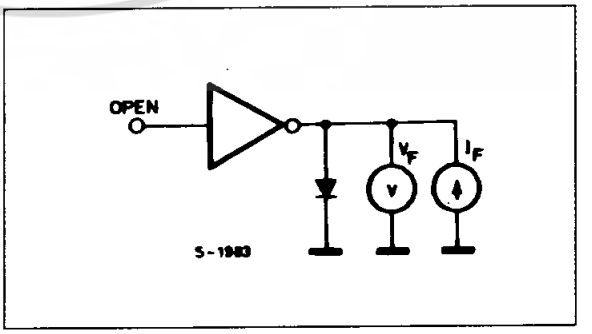


Figure 7.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 8: Collector Current versus Input Current

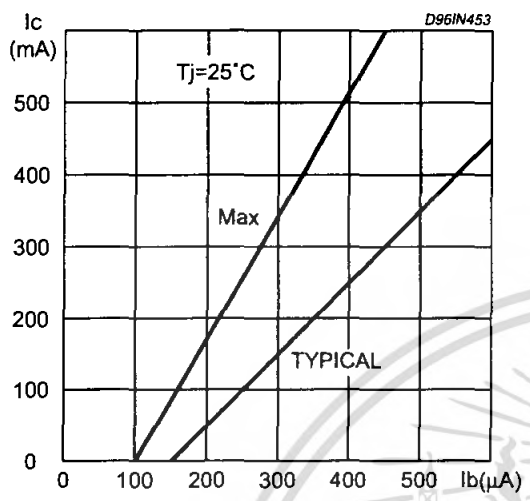


Figure 9: Collector Current versus Saturation Voltage

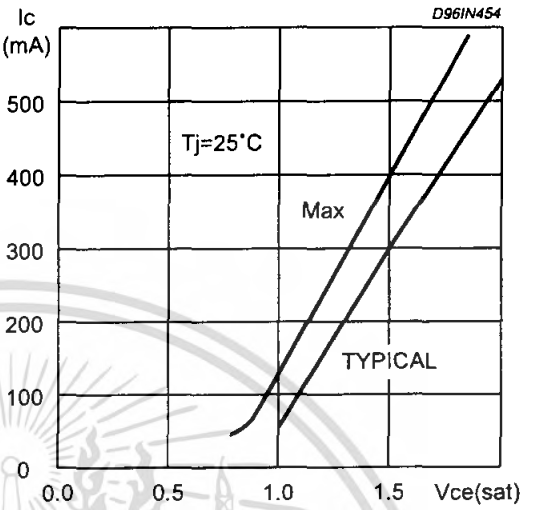


Figure 10: Peak Collector Current versus Duty Cycle

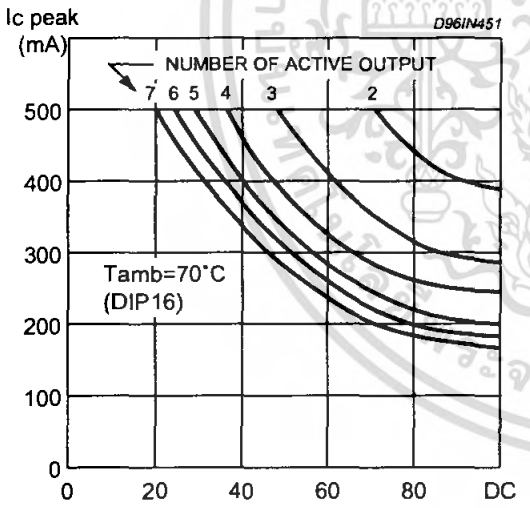
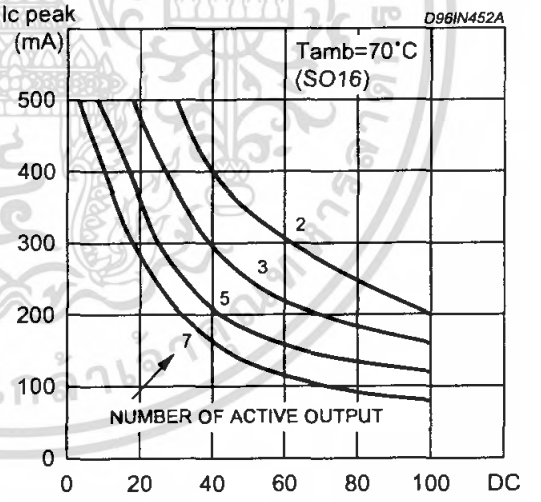


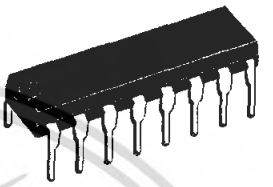
Figure 11: Peak Collector Current versus Duty Cycle



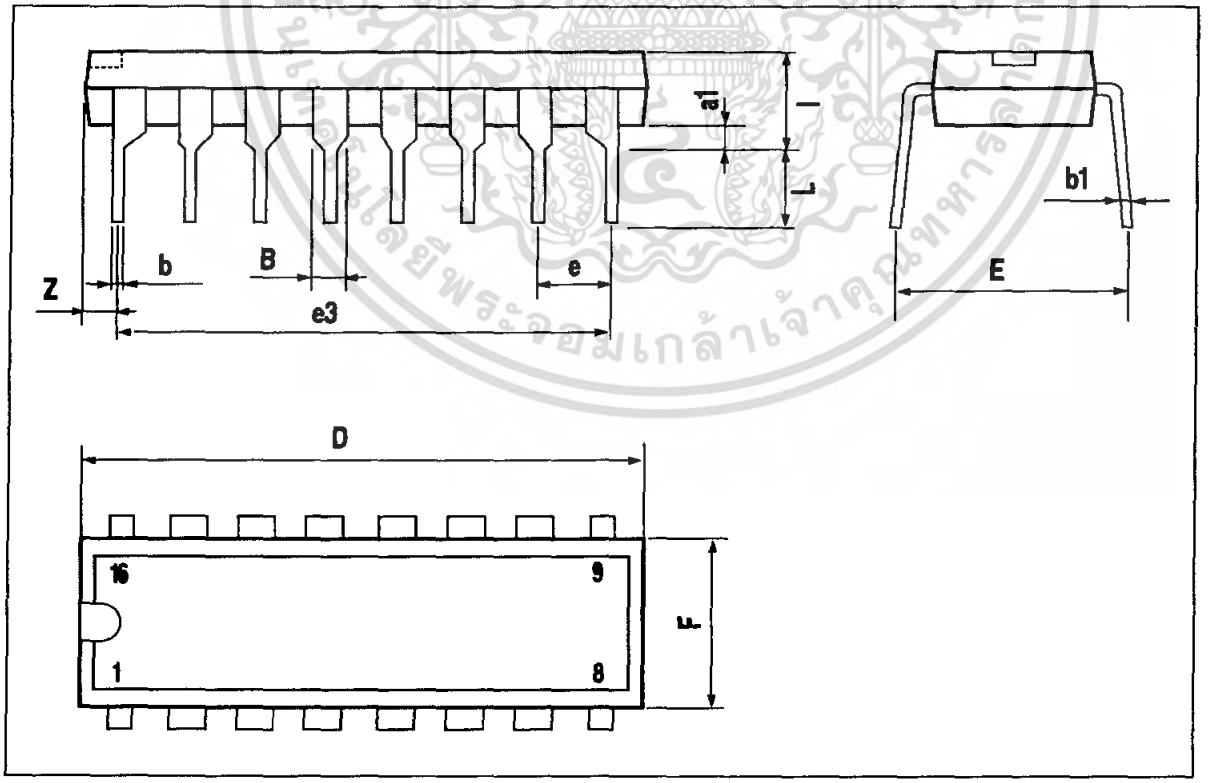
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

OUTLINE AND MECHANICAL DATA



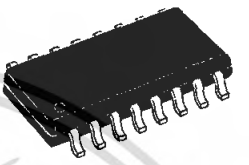
DIP16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

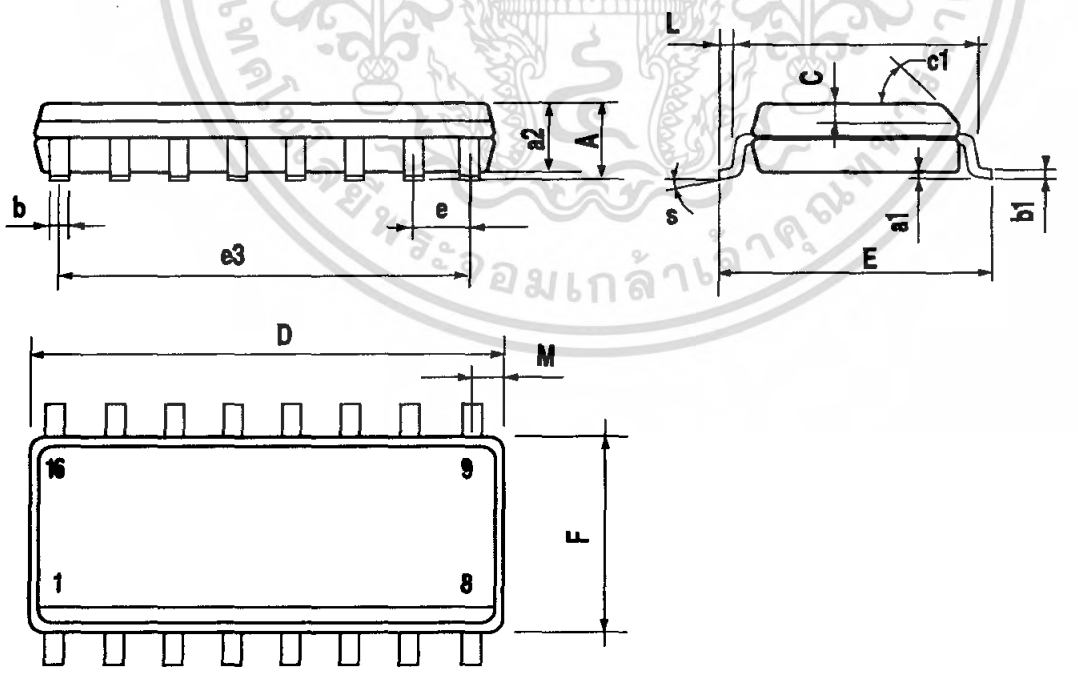
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.75			0.069
a1	0.1		0.25	0.004		0.009
a2			1.6			0.063
b	0.35		0.46	0.014		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.020	
c1	45° (typ.)					
D (1)	9.8		10	0.386		0.394
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F (1)	3.8		4	0.150		0.157
G	4.6		5.3	0.181		0.209
L	0.4		1.27	0.016		0.050
M			0.62			0.024
S	8°(max.)					

OUTLINE AND MECHANICAL DATA



SO16 Narrow

(1) D and F do not include mold flash or protrusions. Mold flash or potrusions shall not exceed 0.15mm (.006inch).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

© 2002 STMicroelectronics – Printed in Italy – All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

http://www.st.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

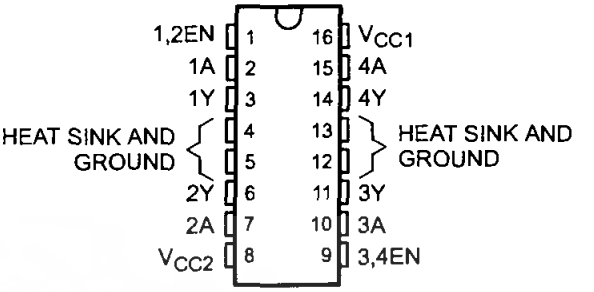
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L293, L293D
QUADRUPLE HALF-H DRIVERS

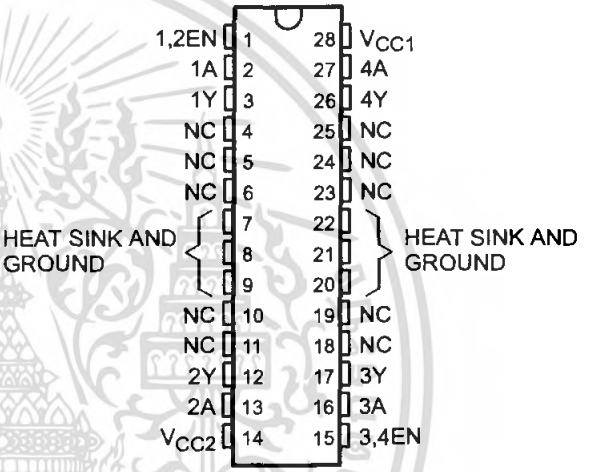
SLRS008B – SEPTEMBER 1986 – REVISED JUNE 2002

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- ***Internal ESD Protection***
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functional Replacements for SGS L293 and SGS L293D**
- ***Output Current 1 A Per Channel* (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

N, NE PACKAGE
(TOP VIEW)



DWP PACKAGE
(TOP VIEW)



description

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A V_{CC1} terminal, separate from V_{CC2}, is provided for the logic inputs to minimize device power dissipation.

The L293and L293D are characterized for operation from 0°C to 70°C.

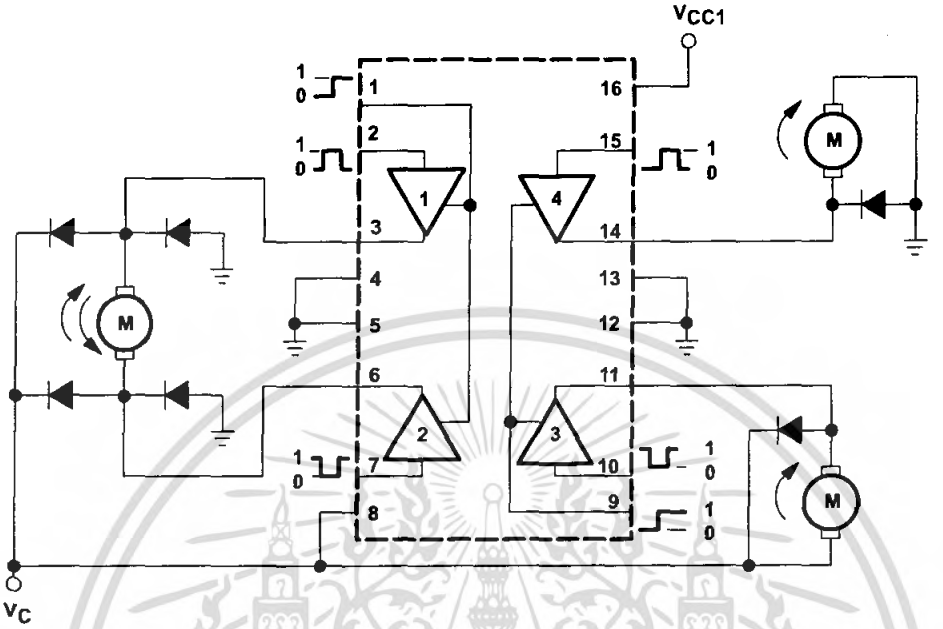
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

block diagram



NOTE: Output diodes are internal in L293D.

TEXAS INSTRUMENTS AVAILABLE OPTIONS

T_A	PACKAGE
	PLASTIC DIP (NE)
0°C to 70°C	L293NE L293DNE

Unitrode Products from Texas Instruments

AVAILABLE OPTIONS

T_A	PACKAGED DEVICES	
	SMALL OUTLINE (DWP)	PLASTIC DIP (N)
0°C to 70°C	L293DWP L293DDWP	L293N L293DN

The DWP package is available taped and reeled. Add the suffix TR to device type (e.g., L293DWPTR).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

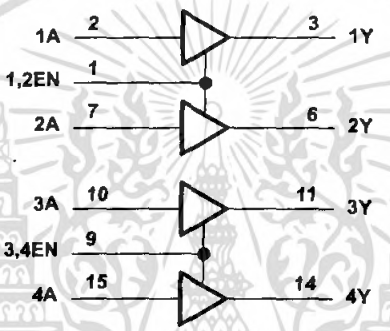
FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

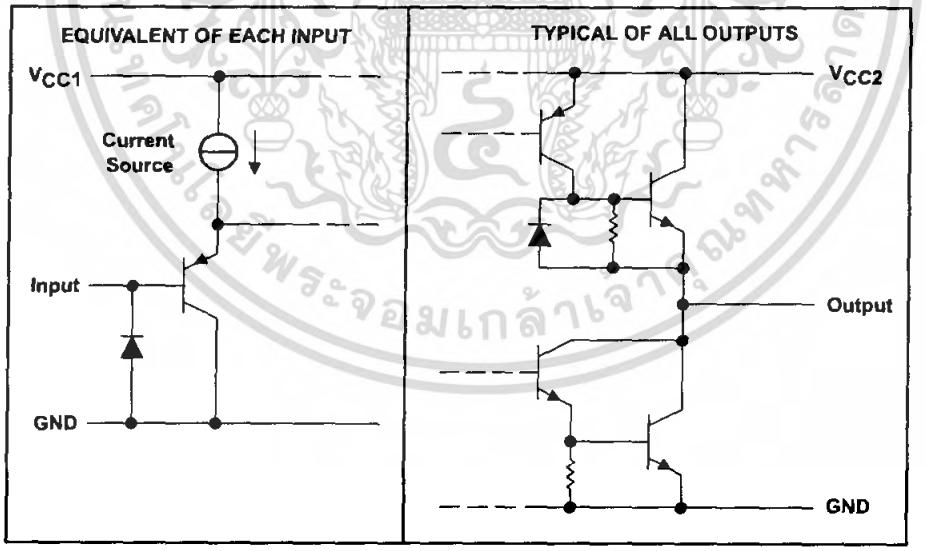
H = high level, L = low level, X = irrelevant,
Z = high impedance (off)

† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

logic diagram

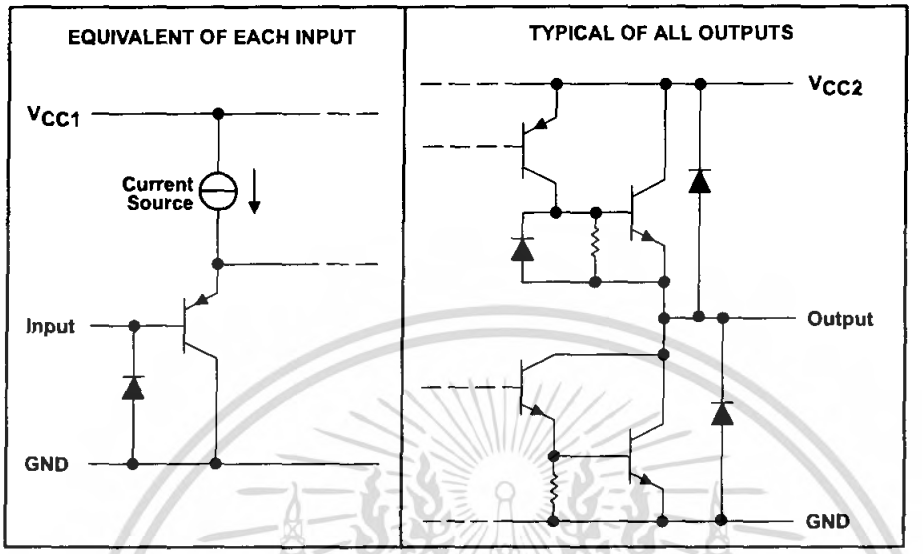


schematics of inputs and outputs (L293)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematics of inputs and outputs (L293D)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC1} (see Note 1)	36 V
Output supply voltage, V_{CC2}	36 V
Input voltage, V_I	7 V
Output voltage range, V_O	–3 V to V_{CC2} + 3 V
Peak output current, I_O (nonrepetitive, t ≤ 5 ms): L293	±2 A
Peak output current, I_O (nonrepetitive, t ≤ 100 µs): L293D	±1.2 A
Continuous output current, I_O: L293	±1 A
Continuous output current, I_O: L293D	±600 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Notes 2 and 3)	2075 mW
Continuous total dissipation at 80°C case temperature (see Note 3)	5000 mW
Maximum junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to the network ground terminal.
2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C.
3. For operation above 25°C case temperature, derate linearly at the rate of 71.4 mW/°C. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection may be activated at power levels slightly above or below the rated dissipation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recommended operating conditions

			MIN	MAX	UNIT
	Supply voltage	V_{CC1}	4.5	7	V
		V_{CC2}	V_{CC1}	36	
V_{IH}	High-level input voltage	$V_{CC1} \le 7$ V	2.3	V_{CC1}	V
		$V_{CC1} \ge 7$ V	2.3	7	V
V_{IL}	Low-level output voltage		−0.3†	1.5	V
T_A	Operating free-air temperature		0	70	°C

† The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

electrical characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		L293: I_{OH} = −1 A L293D: I_{OH} = −0.6 A		V_{CC2}−1.8	V_{CC2}−1.4		V
V_{OL}	Low-level output voltage		L293: I_{OL} = 1 A L293D: I_{OL} = 0.6 A			1.2	1.8	V
V_{OKH}	High-level output clamp voltage		L293D: I_{OK} = −0.6 A			V_{CC2} + 1.3		V
V_{OKL}	Low-level output clamp voltage		L293D: I_{OK} = 0.6 A			1.3		V
I_{IH}	High-level input current	A	V_I = 7 V			0.2	100	µA
		EN				0.2	10	
I_{IL}	Low-level input current	A	V_I = 0			−3	−10	µA
		EN				−2	−100	
I_{CC1}	Logic supply current		I_O = 0	All outputs at high level		13	22	mA
				All outputs at low level		35	60	
				All outputs at high impedance		8	24	
I_{CC2}	Output supply current		I_O = 0	All outputs at high level		14	24	mA
				All outputs at low level		2	6	
				All outputs at high impedance		2	4	

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

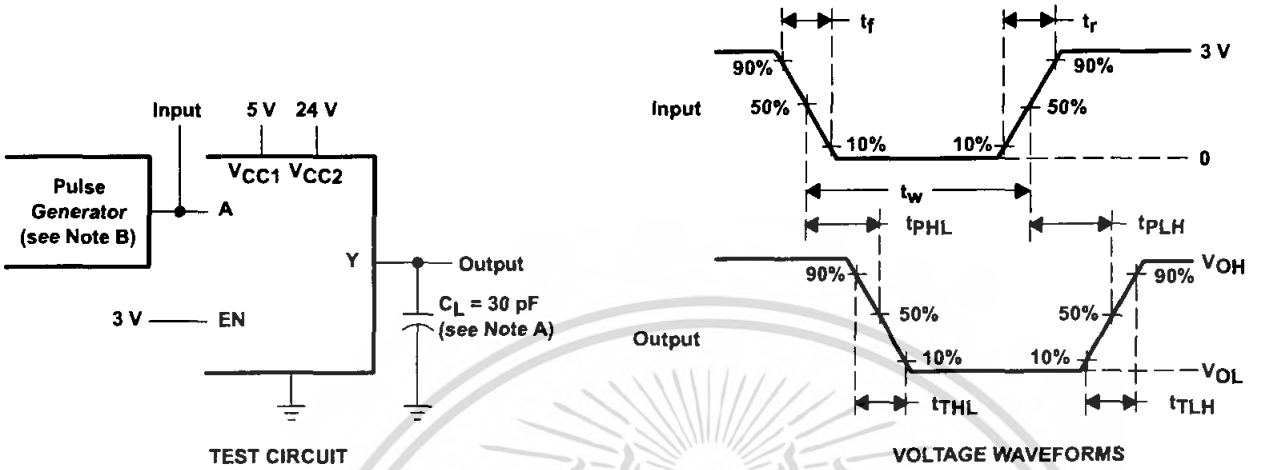
PARAMETER		TEST CONDITIONS	L293NE, L293DNE			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		800		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			400		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	L293DWP, L293N L293DDWP, L293DN			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		750		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			200		ns
t_{TLH}	Transition time, low-to-high-level output			100		ns
t_{THL}	Transition time, high-to-low-level output			350		ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



NOTES: A. C_L includes probe and jig capacitance.
B. The pulse generator has the following characteristics: $t_r \le 10$ ns, $t_f \le 10$ ns, $t_w = 10$ µs, PRR = 5 kHz, $Z_O = 50\ \Omega$.

Figure 1. Test Circuit and Voltage Waveforms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

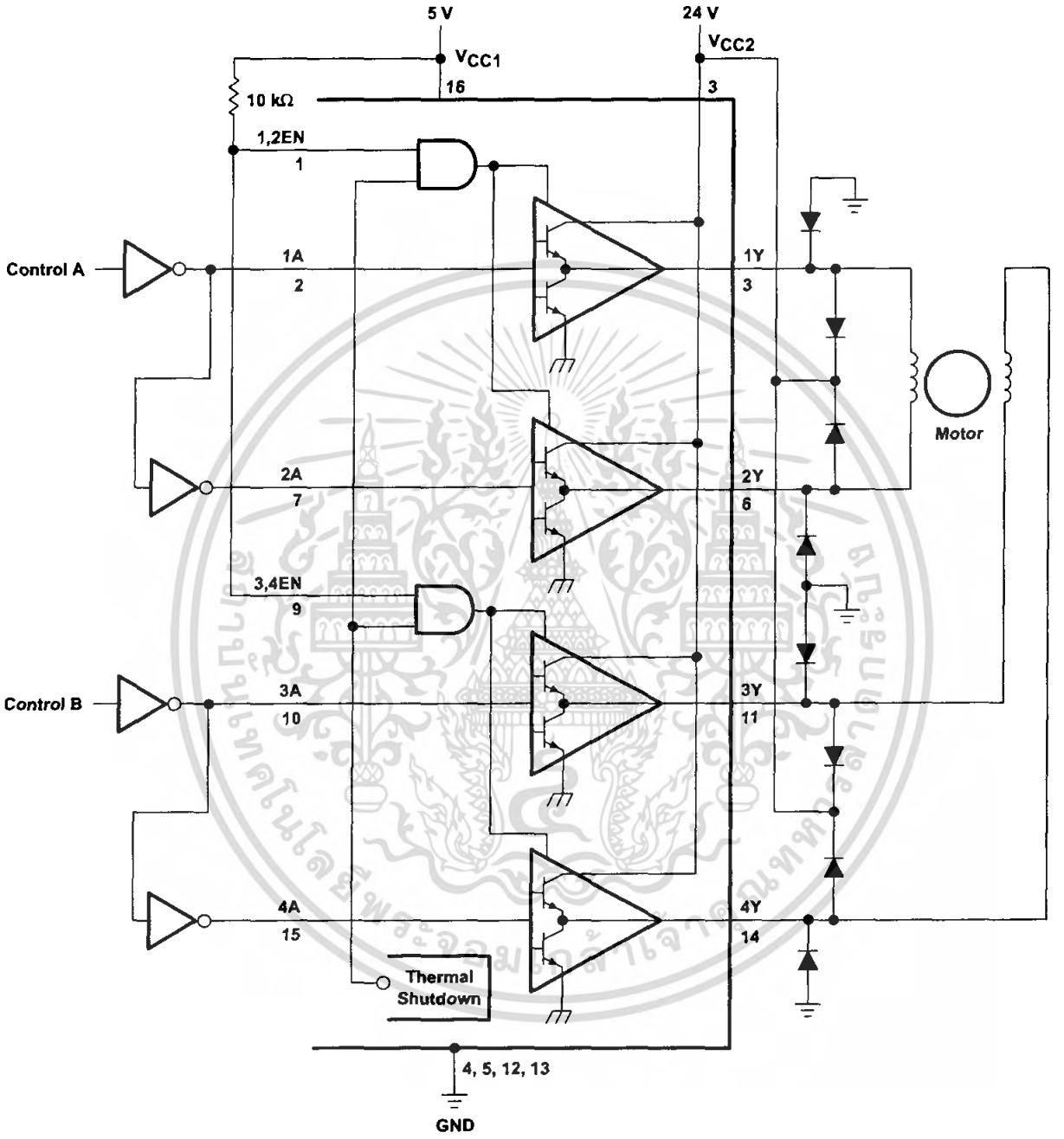


Figure 2. Two-Phase Motor Driver (L293)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

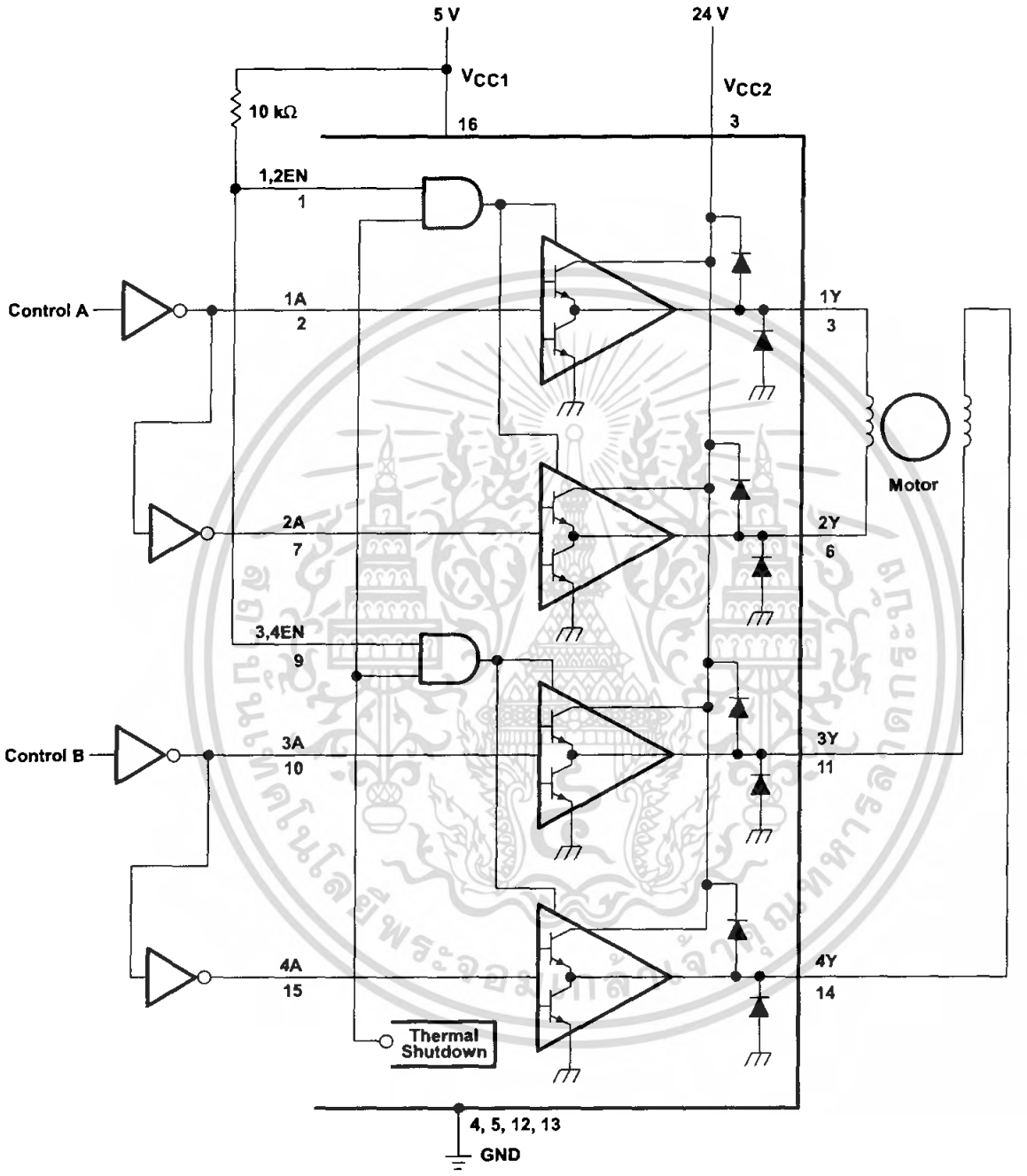
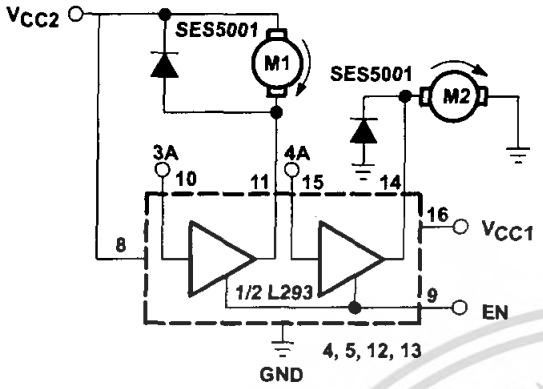


Figure 3. Two-Phase Motor Driver (L293D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

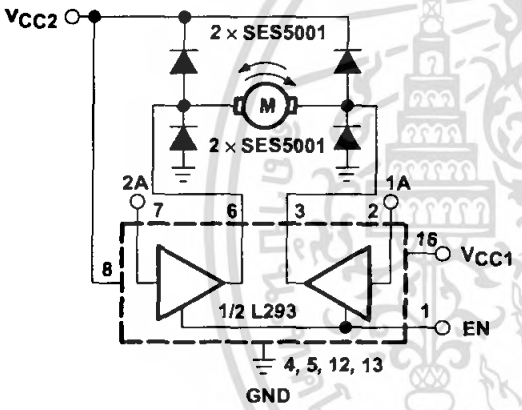
APPLICATION INFORMATION



EN	3A	M1	4A	M2
H	H	Fast motor stop	H	Run
H	L	Run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

L = low, H = high, X = don't care

Figure 4. DC Motor Controls (connections to ground and to supply voltage)



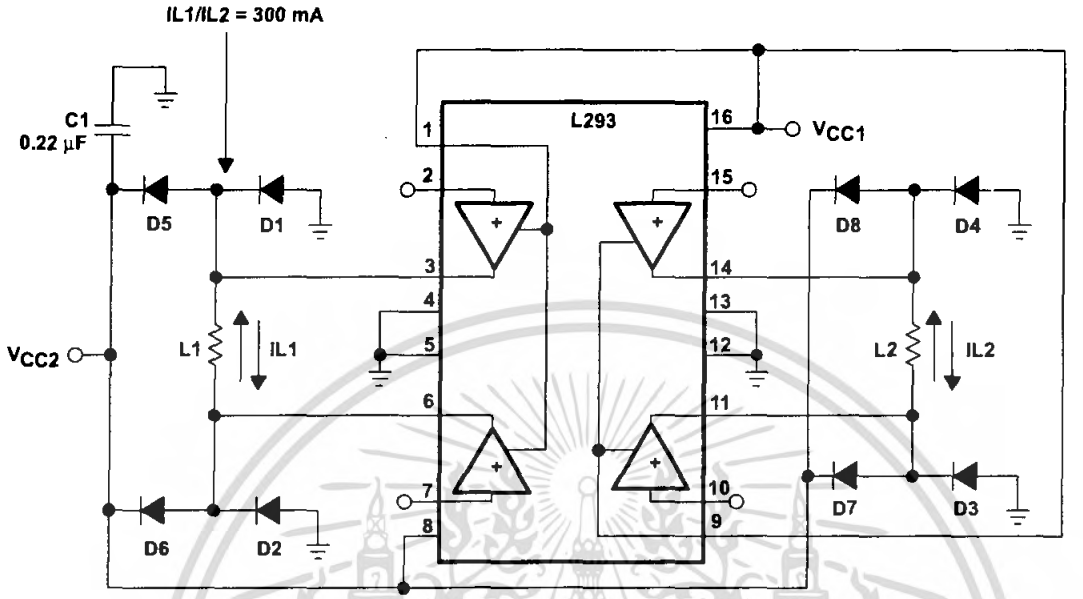
EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 5. Bidirectional DC Motor Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION



D1–D8 = SES5001

Figure 6. Bipolar Stepping-Motor Control

mounting instructions

The Rthj-amp of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heatsink.

Figure 9 shows the maximum package power P_{TOT} and the θ_{JA} as a function of the side l of two equal square copper areas having a thickness of 35 µm (see Figure 7). In addition, an external heat sink can be used (see Figure 8).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not be longer than 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำ[illegible]ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

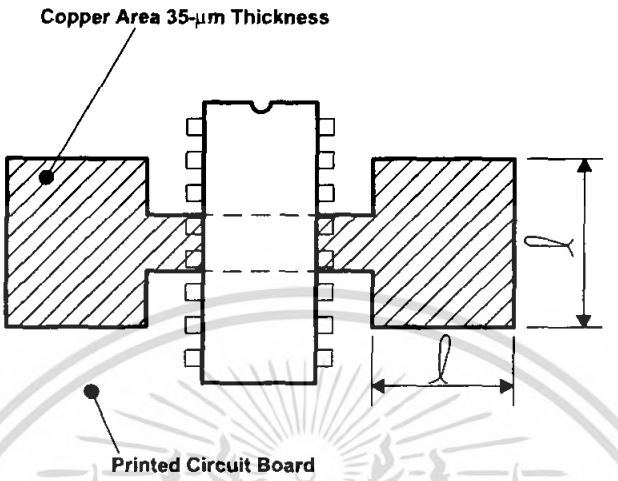


Figure 7. Example of Printed Circuit Board Copper Area (used as heat sink)

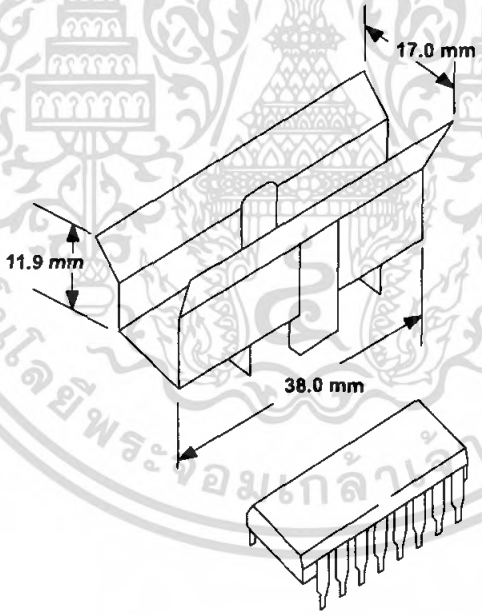


Figure 8. External Heat Sink Mounting Example (θ_{JA} = 25°C/W)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION

MAXIMUM POWER AND JUNCTION
vs
THERMAL RESISTANCE

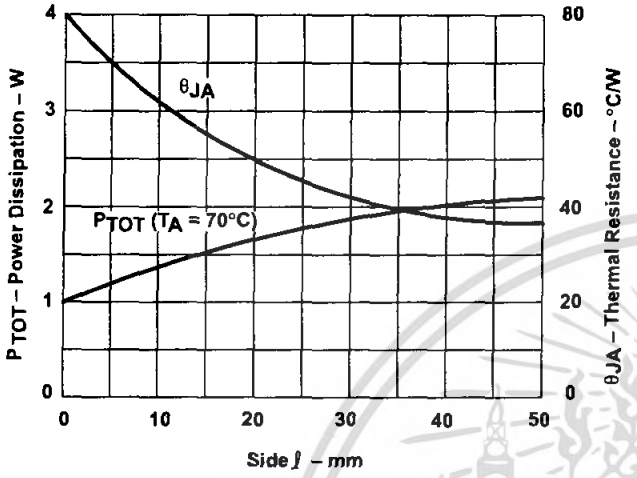


Figure 9

MAXIMUM POWER DISSIPATION
vs
AMBIENT TEMPERATURE

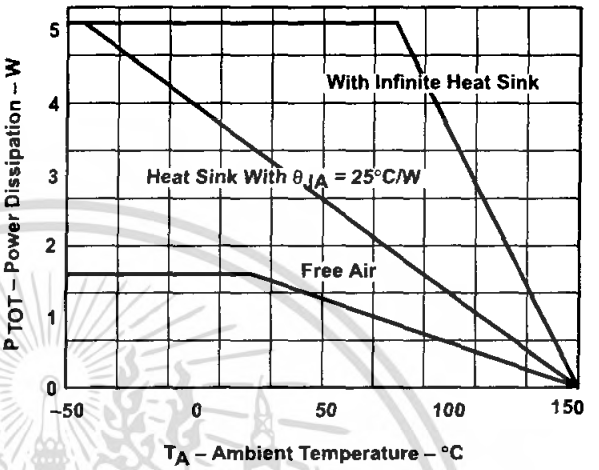


Figure 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Copyright © 2002, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8051SBC V1.0

Wichit Sirichote, kswichit@kmitl.ac.th

New Microprocessor learning board demonstrates exemplary design of 8-bit single board computer. More peripheral and easy to adapt to be a dedicated controller.

I decided to rename C52EVB to be 8051SBC. Since the cpu can be any 8051 compatible chips with 40-pin DIP package. The old day chips, say 8031, 8751, 8032 or the new chips, 89C51, 89C52, 89C51RD2 can be used without any problem. Some may think that your old chips were useless, now you can use it for many projects. The monitor program of new 8051SBC was placed at external 32kB EPROM. With this space, you can customize your own monitor code or even add the application program and have the DIP switch to boot it when power up. The 32kB SRAM still be mapped as external code and data memory. We can test the program in RAM, when it firms then put it in the space in EPROM. The memory and i/o decoder is now using GAL (Generic Array Logic).

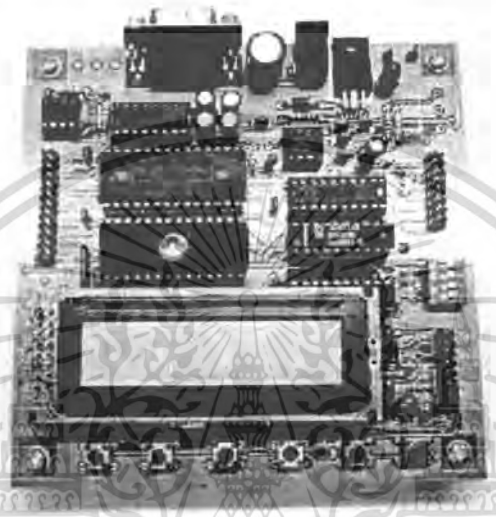


Figure 1: The prototype of 8051SBC (click for big picture).

The hardware schematic was provided to be an exemplary design for student to learn basic design of 8-bit microprocessor board. The board is a generic tool for every microprocessor lab. I provided all materials in public domain, so you may build it for your lab. Student can learn assembly programming, HLL programming and hardware interfacing.

The new 8051SBC features;

- **CPU:** Any 8051 compatible with 40-pin DIP package @11.0592MHz
- **MEMORY:** 27C256, 32kB EPROM for monitor program
- 62256, 32KB SRAM for both code and data space
- **I/O:** direct cpu bus interface 2x16 line LCD
- 8-bit input port, 74LS244
- 8-bit output port, 74HC573
- **MEMORY and I/O Decoder:** GAL16V8D
- **EEPROM:** 24LC256, 32KB serial eeprom
- **RTC:** Real-time clock, DS1307 with +3V Lithium backup
- **ADC:** LTC1298, SPI interface 2-channel 12-bit Analog-to-Digital Converter
- **I/O pins:** P1,P3 of 8051 cpu, 16-bit I/O port
- **Debug LED:** single dot LED connected to P1.7
- **Keypad and DIPSW:** 4-bit keypad and 4-bit DIP switch
- **RS232 Level Converter:** MAX232
- **RS485:** 75176 differential transceiver
- **Serial Interface:** 9600 8n1
- **Monitor Program:** Modified PAULMON2 including new commands

Hardware Schematic

Complete schematic for cpu and memory is shown in Figure 2. U7 is 40-pin 8051 compatible chip. The xtal is 11.0592MHz. JP1 selects internal or external code memory. With the external monitor mode, we make EA# to GND. The data bus D0-D7 are tied to memory chip, U1, U2 and to the GPIO, U3, U4 directly. Both memory chips are 32kB, so each chip needs A0-A14. A15 is used for memory decoder that decodes between first 32kB and second 32kB of external code memory. When A15 = 0, U1 the monitor rom was selected. The OE# of U1 is tied to PSEN#, so the code memory will enable only when address range 0x0000-0x7FFF of code memory are accessed. The upper memory space from address 0x8000-0xFFFF makes A15 = 1. When A15 = 1, U2 or

SRAM will be selected. This memory space was decoded to be overlapped between external data and code memory. So while running under monitor mode, we can then load the hex code into the SRAM(data memory) and when JUMP from monitor program to address says 0x8000, the PSEN# that AND with RD# and ties to OE# of SRAM will enable the RAM to be code memory.

Since we have decoded the lower space, 0x0000-0x1FFF for code memory, this left a big space for external data memory from 0x0000-0x1FFF. The 8051SBC then borrowed this area to make the I/O that can access as the memory space with RD# and WR# combination with address decoder.

We can say that this area is memory mapped I/O. U3 is an 8-bit D-FF for output port. U4 is 8-bit tristate for input port. The LCD also connects cpu bus directly. To make the I/O decoder for each chip, we use A8, A9, A10 and A15 to select each block. The I/O map shows the address range for each device. U6 is PLD made from GAL16V8. It provides the memory and I/O decoder, later you will see details description on how to implement the decoder.

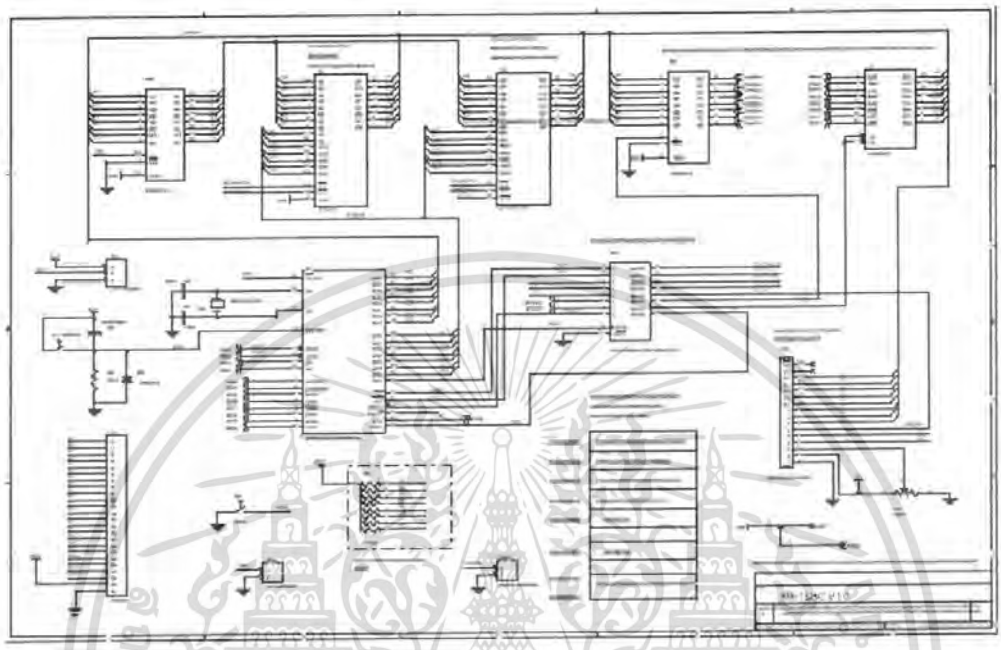


Figure 2: Complete hardware schematic, CPU, memory, PLD

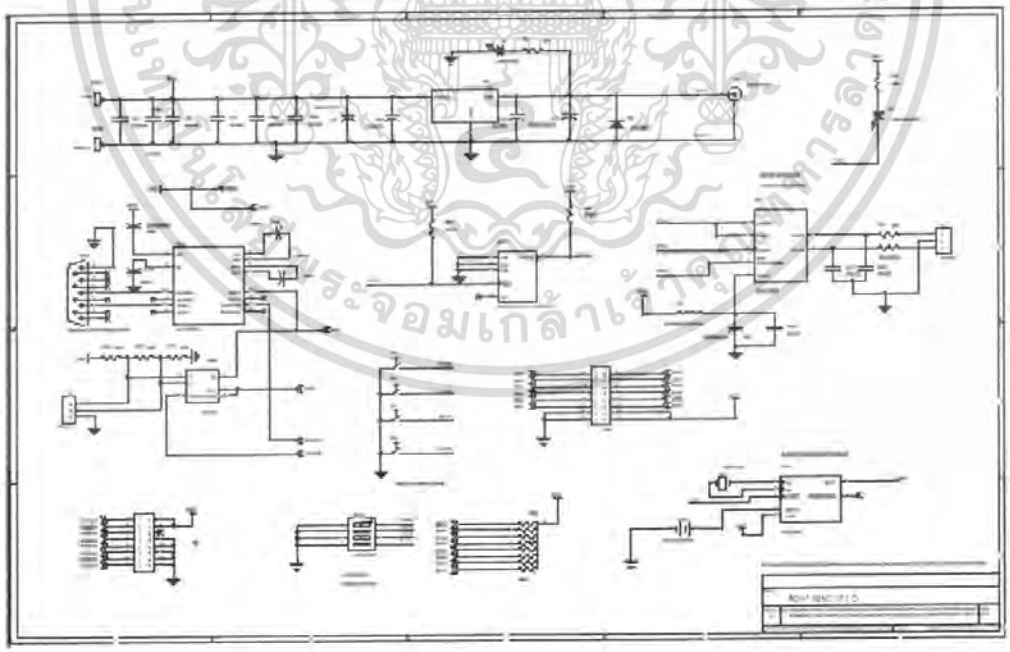


Figure 3: RS232, RS485, DC supply, EEPROM, ADC, RTC, keypad, debug LED.

U8 is +5V voltage regulator. D4 is protection diode. The RS232 level converter is MAX232. Notice that we have the RS485 transceiver as well. The TxD pin from CPU can tie to both input, T1IN (U10) and D(U12) directly. Howvere the RxD pin is the output of AND logic between R1OUT(U10) and R(U12). U11 is 32kB I2C EEPROM. It uses for program storage or the aplliaction program can use it. U9 is a popular 12-bit ADC having SPI interface. The reference voltage uses +5V. The two channels analog input has RC filter. U13 is RTC with I2C interface. It shares the I2C signal with EEPROM, U11. SW1 is 4-bit DIP swicth for system settings. User can use S3, S4, S5 and S6 for application program.

ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU list

I have tested some of 40-pin DIP package 8051 compatible CPU with the 8051SBC, table 1 shows the reason for each chip and

you may find the right for your applications.

8751	old chip with ceramic package and quartz window for UV, new generation will see how the old chip look likes.
8031	romless 8051, 128 bytes RAM
8032	romless 8051, 256 bytes RAM
8052	basic interpreter with EA = +VDD
AT89S8252	SPI, WDT, 2kB EEPROM, 8kB FLASH, 256 bytes RAM
AT89C51	4kB FLASH 128 bytes RAM
AT89C52	8kB FLASH 256 bytes RAM
DALLAS DS500T	8kB loadable RAM RTC

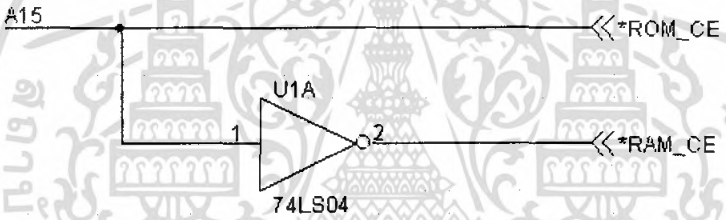
Table 1: CPU that can be used with 8051SBC.

GAL Decoder

The reason that took me to use GAL is to reduce the number of discrete gate for memory and i/o decoder. You may noticed that, with a C52EVB, only one chip, the 74HC00 was used for memory and 8255 decoder. However with new 8051SBC, I need to decode the LCD and simple port. This needs more chip, so better try with 20-pin popular GAL from Lattice, the GAL16V8D. The GAL can emulate logic function and can reprogram many times.

Before getting into GAL's matter, let begin with how to replace the logic gate using logic equation. Now let see the actual decoder circuit. First take a look for memory decoder.

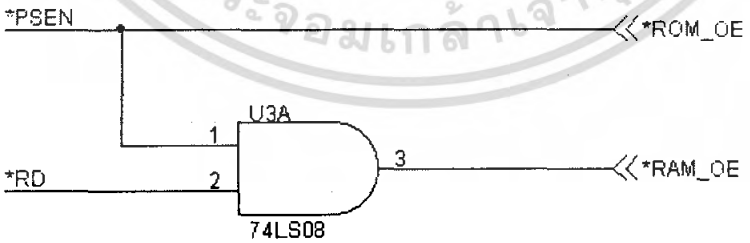
The EPROM will place at 0x0000 to 0x7FFF. And the SRAM will place at 0x8000 to 0xFFFF. We see that we can use A15 to be a chip select control signal for EPROM directly. Whereas for the SRAM it needs to invert A15 in order to select SRAM when A15 becomes '1' for address 0x8000-0xFFFF.



Note * is active low signal.

To read or fetch the content of EPROM, the place we store monitor program, we must provide *PSEN signal tied to output enable pin.

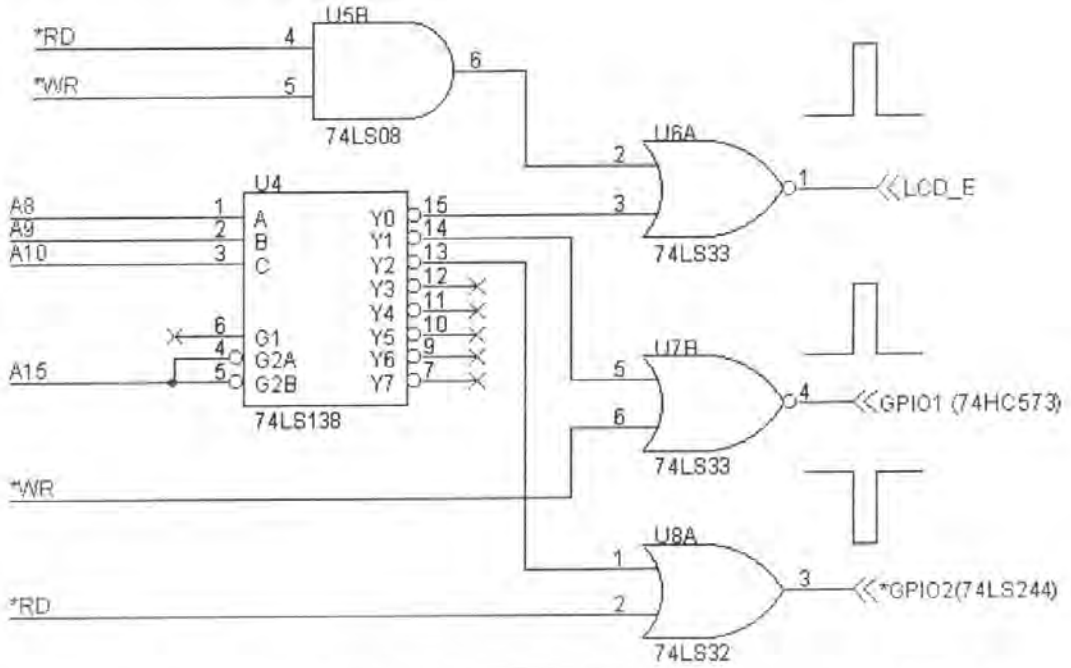
However for SRAM, we will use it for user program running as well as the place where we can edit the machine code, so we must make it to be code and data memory space. This can be made easily with *RD and *PSEN logical AND together. The circuit would be,



A bit complicate for i/o decoder, as shown below we see the popular decoder chip, the 74LS138, 3-to-8 line decoder and some gates were used to provide control signal for the i/o devices. The i/o space uses the external data memory with address 0x0000 to 0x7FFF. A15 must be low to enable the output of 74LS138. A8-A10 selects the active low output for each i/o.

Let look at the first device, the LCD. Y0 will active low when all inputs are low, or A8-A10 and A15 = '0'. The datasheet of LCD says, the signal E of LCD will active for both read or write cycle, so we AND them together with U5B. U6B provides the positive pulse when Y0 and pin 6 of U5B low. This signal, LCD_E can then directly tie to signal E of the LCD.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Similarly for GPIO1 or 74HC573, latch signal. Now GPIO1 uses Y1, that means A8 must be '1', A9,A10 and A15 must be '0'. The WR signal is logical OR with Y1. The NOR gate U7B provides the positive pulse similar to the LCD_E signal. This signal will control the D FF latch. Dada is loaded into FF and the is latched at high to low transition.

For the input port made by tri-state buffer, 74LS244, the enable signal was made by Y2 and RD signal. However now A9 must be 1' and A8, A10 and A15 must be '0'.

With above decoder, we then need more chips to make it. Now let see how the GAL chip looks like.

GAL16V8D

The GAL chip was Lattice GAL16V8D. The GAL is a programmable logic device (PLD). We can write the logic equation, convert it to JED file and program the GAL with JED file. Then we can replace above circuit with single GAL chip. The GAL16V8 has 10 inputs and 8 outputs. The chip can reprogram many times. In this design I used only one chip to replace such memory and i/o decoder circuit.

Using PLD Assembler

To implement the logic with GAL, we must first edit the logic equation and have the PLD assembler to translate the equation file to JEDEC file. I found OPALjr 2.0 from google search. It's very easy to use. Now let see the equation of above decoder.

```
; 8051SBC PLD Equation
; Memory and I/O decoder for 8051 Single Board Computer
; Wichit Sirichote, kswichit@kmitl.ac.th
; May 1, 2003
; PLD: Lattice GAL16V8D

CHIP 8051SBC GAL16V8

 a15=1 rd=2 wr=3 a8=4 a9=5 a10=6 rs232=7 rs485=8 psen=9 rom_ce=12
 ram_ce=13 ram_oe=14 rom_oe=15 lcd_e=16 gpio1=17 gpio2=18 rxd=19

EQUATIONS

ram_oe = rd * psen

rxd = rs232 * rs485

rom_ce = a15

ram_ce = /a15

rom_oe = psen
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/lcd_e = rd * wr + a8 + a9 + a10 + a15

/gpio1 = /a8 + a9 + a10 + a15 + wr

gpio2 = a8 + /a9 + a10 + a15 + rd
```

Note. * is logical AND, + is logical OR and / is logical NOT.

There is some i/o pin available on GAL, I also used them for sharing between rs232 and rs485. The output is tie to RxD pin of the cpu. See schematic for detail.

When finish editing the equation file, click MODULE -> EQU2JED, it will convert the equation file to JEDEC file. Let look at the output JEDEC file.

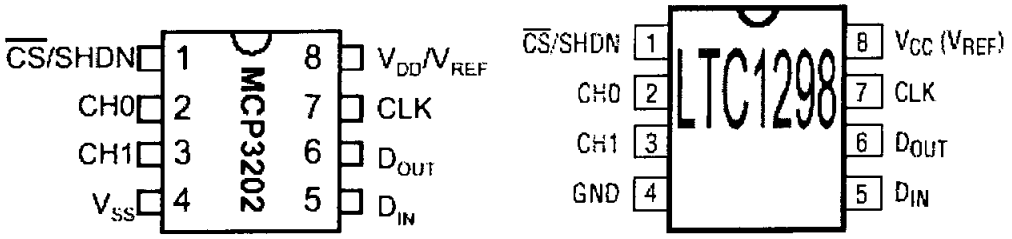
```
GAL16V8
EQN2JED - Boolean Equations to JEDEC file assembler (Version V101)
Copyright (c) National Semiconductor Corporation 1990-1993
Assembled from "D:\C52EVBV3\O\8051SBC.EQN". Date: 5-1-103
*
NOTE PINS a15:1 rd:2 wr:3 a8:4 a9:5 a10:6 rs232:7 rs485:8 psen:9*
NOTE PINS rom_ce:12 ram_ce:13 ram_oe:14 rom_oe:15 lcd_e:16 gpio1:17*
NOTE PINS gpio2:18 rxd:19*
NOTE GALMODE SMALL*
QF2194*QP20*F0*
L0000
11111111111111111111011101111111*
L0256
11111111011111111111111111111111
11111111111110111111111111111111
11111111111111110111111111111111
11011111111111111111111111111111
01111111111111111111111111111111*
L0512
11111111101111111111111111111111
11111111111101111111111111111111
11111111111111110111111111111111
11011111111111111111111111111111
11110111111111111111111111111111*
L0768
01110111111111111111111111111111
11111111011111111111111111111111
11111111111101111111111111111111
11111111111111110111111111111111
11011111111111111111111111111111*
L1024
11111111111111111111111111110111*
L1280
01111111111111111111111111110111*
L1536
11101111111111111111111111111111*
L1792
11011111111111111111111111111111*
L2048
11001111*
L2128
10000000111110001111100011111000100000001000000010000000100000000*
L2192
10*
C5053*
0000
```

The JEDEC file is standard file for any GAL programmer. I used GALBLAST to program it.

Before you can use GAL or any PLD efficiently, you must have knowledge on the fundamental of digital logic. The PLD is just to help hardware design more easy.

12-bit ADC converter

I found Microchip 12-bit ADC converter, MCP3202 that can replace Linear Technology LTC1298 directly. Below is pin diagram of 8-pin DIP package. MCP3202 is cheaper than LTC1298. Both chips are pins compatible. The 8051SBC can use either one without any problem.



Software

With the external 32kB EPROM, the 27C256, we then need the EPROM programmer. I used **willem EPROM programmer**. Also here in Thailand **http://se-ed.net/mpu51/** You may build it easily. Later I'll add the EPROM programmer attach to the 8051SBC.

The main monitor code space is approx. 8kB. This fits exactly in 89C52 chip. So you can add many things into 32kB EPROM. During firmware developing, I used parallel eeprom, 28C256 from ATMEL. I made the adapter from two sockets. The socket swaps pin 1 and pin 14. If you have the 28C256, you can use my idea.

Some may got many ceramic EPROM, 27C256. To erase it, you need Ultraviolet light. When your code firms, you can write the program into a cheap 27C256 OTP (One Time Program).

The monitor program composes of three files.

1. **8051SBC.ASM**
2. **myextra1.ASM**
3. **new.c**

The 8051SBC.ASM and myextra1.asm are modified version of PAULMON2 free monitor program written by Paul Stoffregen. Both files fit in 8kB space. The new commands is now under developing with Micro-C compiler. The available for download was tested on RAM and then recompile with MEDIUM memory model. In addition to such monitor code, I also experimented with Camel FORTH. EEPROM booting still working with this version excepts the boot switch is now move to DIP switch bit 0. To start Cemel FORTH, when set DIP switch bit 1 to '0' after power up, it will enter Camel FORTH.

The Intel hex file for the monitor program including new commands and Camel FORTH is here, **8051SBC.HEX**.

Below is the sample of new commands.

```
8051SBC Microprocessor Learning Board

ADDR:8000> Help

Standard Commands
 ?- This help list
 M- List programs
 R- Run program
 L- Download
 U- Upload
 N- New location
 J- Jump to memory location
 H- Hex dump external memory
 I- Hex dump internal memory
 E- Editing external ram
 C- Clear memory
User Installed Commands
 D- Disassemble
 S- Single-Step
 Q- Quick home
 Z- ZAP EEPROM
 W- NEW COMMANDS

ADDR:8000>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Notice that letter D is now changed to Disassemble, and L changed Load Intel hex file. Command Q will get back to address

0x8000. Z command still exists. Now let see new command with letter W.

```
ADDR:8000>  NEW COMMANDS


8051SBC new commands (press ? for help or q to quit)

>>

Help Menu
A ASCII Table
D Dec2Hex
R Read GPIO2
Q Quit
I i/o address
B Binary image
? help

>>Decimal to Hex converter

Enter decimal number = -32768
Dec -> Hex -> Bin
-32768 -> 8000 -> 1000000000000000

>>Read GPIO2 (8-bit input port)

FD 11111101

>>i/o address decoded by PLD

LCD   command write    0x0000
LCD   data write       0x0001
LCD   command read     0x0002
LCD   data read        0x0003
GPIO1 8-bit output port 0x0100
GPIO2 8-bit input  port 0x0200
expansion i/o space    0x0300-0x07FF

>>
```

Now let make DIP switch bit 1 to '0' and then power up the board.

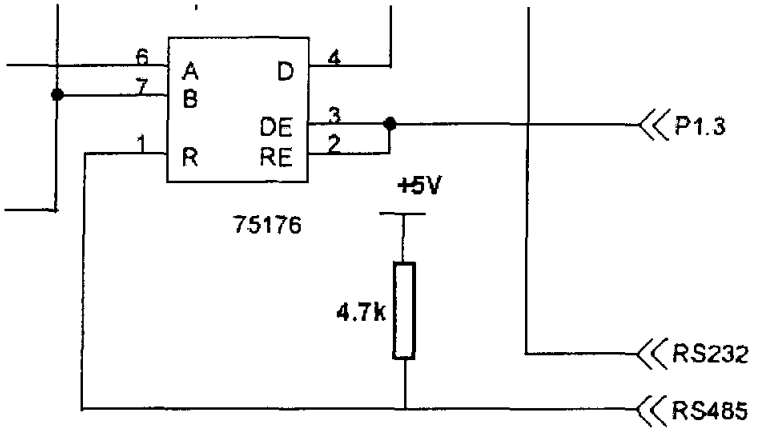
```
8051 CamelForth v1.6 27 Mar 2000
 ok
2a emit *ok
 ok
: star 2a emit ; ok
 ok
 ok
star *ok
 ok
 ok
```

The sample above is a common first word for FORTH beginner. We define word 'star' with 2a put to data stack and emit it to screen. I used to try FORTH with my design 80C88, since I was a student in 1984. However the Harvard architecture of 8051 may not suitable to make use of FORTH more efficient. The reason why I added FORTH to the 8051 board is just to experiment the startup booting.

RS485

I found some board has a problem on serial port. When power up the board, the prompt was printed on screen repeatedly. I fixed it by adding a 4.7k pull-up resistor at pin 1 of 75176 RS485 driver. See the picture below.

ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Download

Schematic	8051SBC.pdf
Monitor program	8051SBC.asm myextra1.asm new.c
Intel Hex files	mypaulm2.hex myextra1.hex new.hex
Intel Hex file (three files)	8051SBC.hex
8051SBC Reference Manual	8051SBCreference.pdf
PCB gerber files	8051gerber.zip
OrCAD 9.1 MAX file	8051sbc.max
OrCAD 9.1 Schematic and Netlist files	orcad8051sbc.zip
8051SBC BOM	bom8051sbc.txt
Equation and JEDEC file for GAL16V8D	8051sbc.eqn 8051sbc.jed

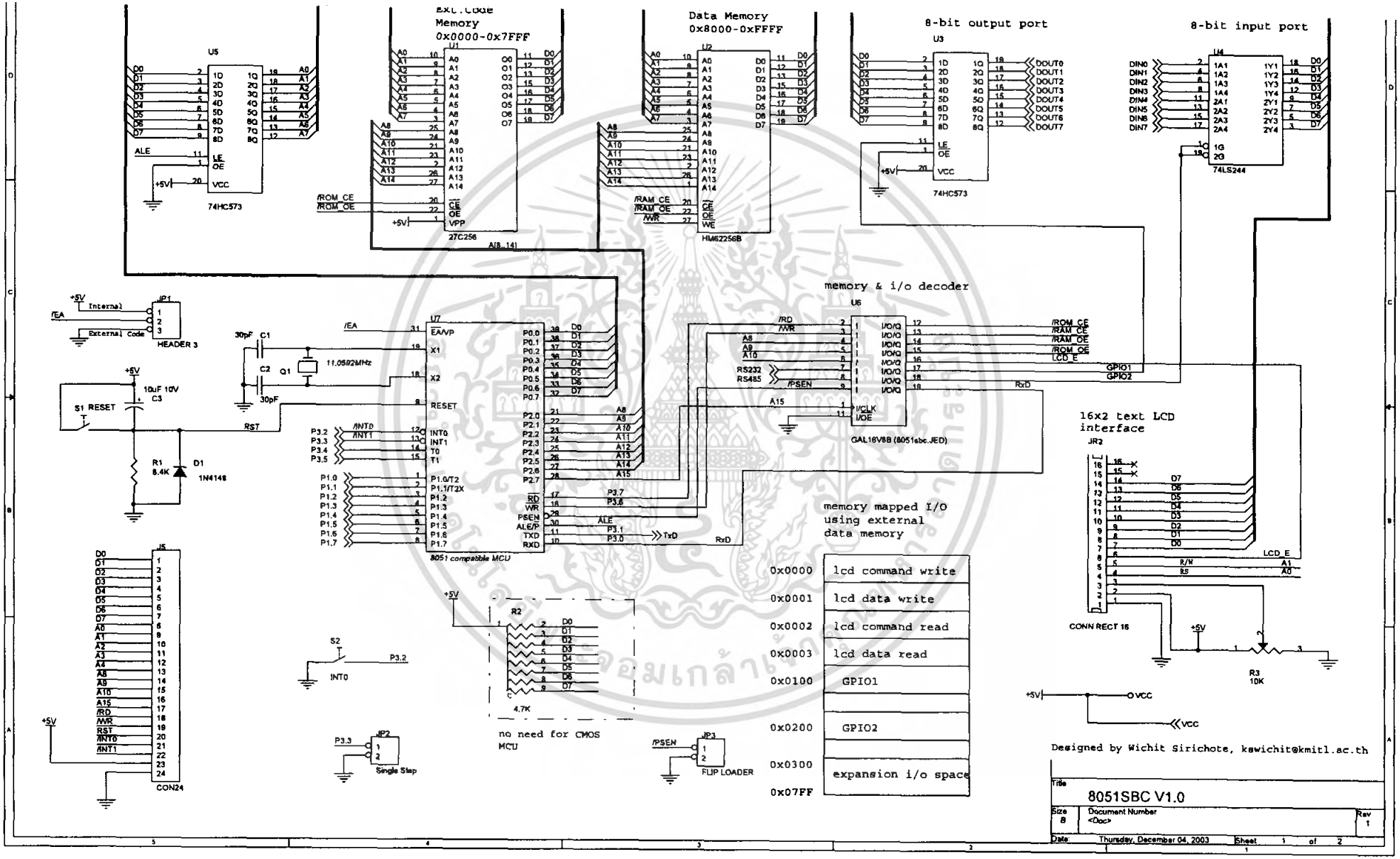
Contribution to 8051SBC

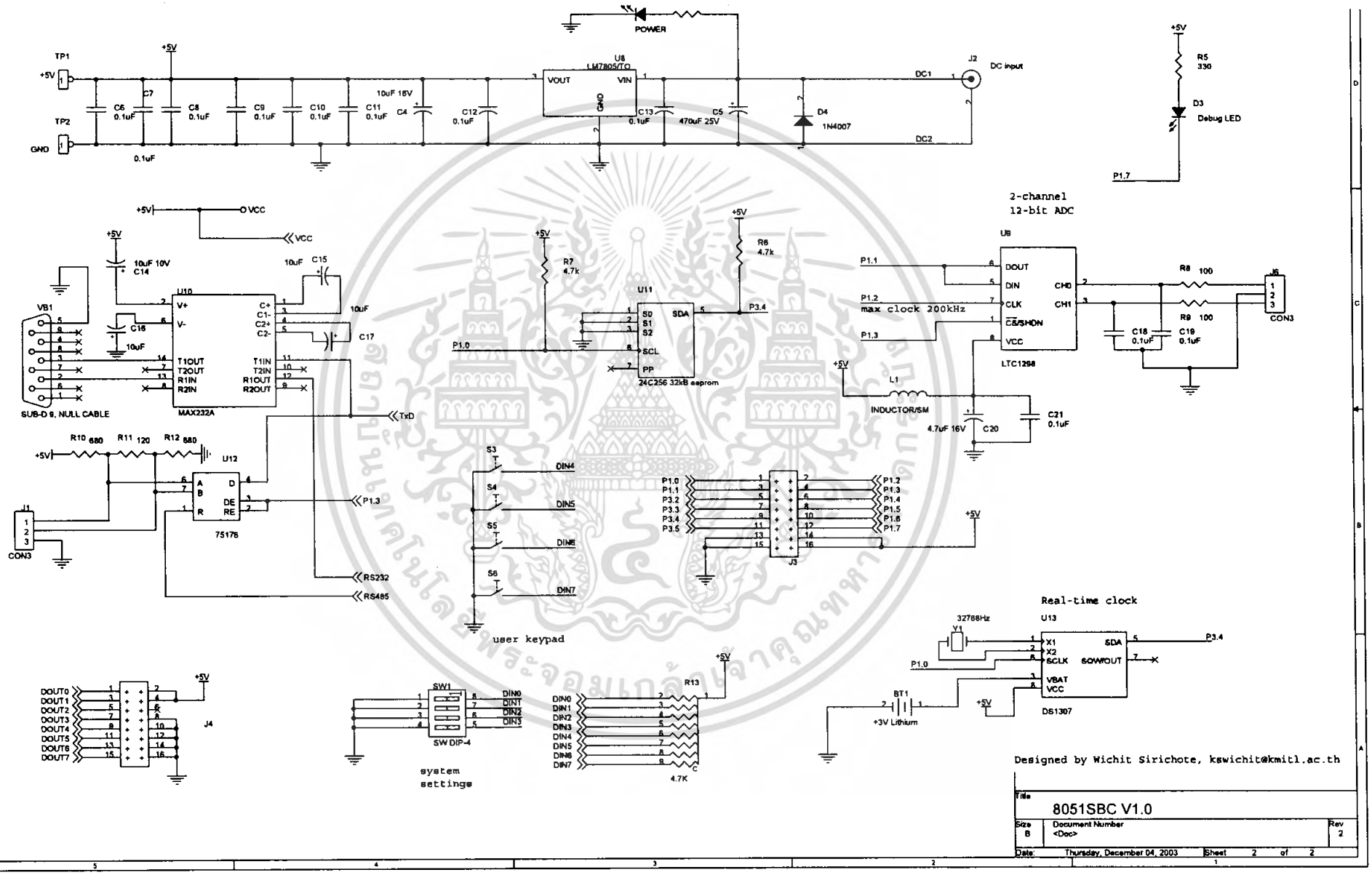
- SBC51Lib.zip A library file for LCD module on SBC51 written by Muhammad Kamran, Department of Electronics, University of Peshawar, NWFP, Pakistan.

Build Your Own
µControllers Projects

30 Nov 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Ext. Code Memory 0x0000-0x7FFF
Data Memory 0x8000-0xFFFF
8-bit output port
8-bit input port
74HC573
74LS244
27C256
HM62256B
memory & i/o decoder
GAL16V8B (8051sbc.JED)
8051 compatible MCU
11.0592MHz
HEADER 3
16x2 text LCD interface
CONN RECT 16
memory mapped I/O using external data memory
0x0000 lcd command write
0x0001 lcd data write
0x0002 lcd command read
0x0003 lcd data read
0x0100 GPIO1
0x0200 GPIO2
0x0300 expansion i/o space
0x07FF
no need for CMOS MCU
Single Step
FLIP LOADER
CON24
Designed by Wichit Sirichote, kswichit@kmitl.ac.th
8051SBC V1.0
Thursday, December 04, 2003


POWER
TP1
TP2
GND
+5V
C6 0.1uF
C7 0.1uF
C8 0.1uF
C9 0.1uF
C10 0.1uF
C11 0.1uF
C4 10uF 16V
C12 0.1uF
U8 LM7805/TO
VOUT
VIN
GND
C13 0.1uF
C5 470uF 25V
D4 1N4007
J2 DC input
DC1
DC2
R5 330
D3 Debug LED
P1.7
2-channel 12-bit ADC
U8
DOUT
DIN
CLK
CS/SHDN
VCC
CH0
CH1
LTC1298
P1.1
P1.2
P1.3
max clock 200kHz
R8 100
R9 100
C18 0.1uF
C19 0.1uF
J6
CON3
L1
INDUCTOR/SM
4.7uF 16V C20
C21 0.1uF
U10
MAX232A
V+
V-
C+
C1-
C2+
C2-
T1OUT
T2OUT
R1IN
R2IN
T1IN
T2IN
R1OUT
R2OUT
C14 10uF 10V
C15 10uF
C16 10uF
C17 10uF
VB1
SUB-D 9, NULL CABLE
TxD
R10 680
R11 120
R12 680
U12
75176
J1
CON3
P1.3
RS232
RS485
U11
24C256 32kB eeprom
SDA
SCL
SD
S1
S2
PP
R7 4.7k
R6 4.7k
P1.0
P3.4
S3
S4
S5
S6
DIN4
DIN5
DIN6
DIN7
user keypad
P1.0
P1.1
P3.2
P3.3
P3.4
P3.5
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
J3
Real-time clock
32768Hz
Y1
U13
X1
X2
SCLK
SQW/OUT
VBAT
VCC
DS1307
BT1
+3V Lithium
P3.4
P1.0
DOUT0
DOUT1
DOUT2
DOUT3
DOUT4
DOUT5
DOUT6
DOUT7
J4
SW1
SW DIP-4
DIN0
DIN1
DIN2
DIN3
system settings
R13
4.7K
Designed by Wichit Sirichote, kswichit@kmitl.ac.th
Title
8051SBC V1.0
Size B
Document Number <Doc>
Rev 2
Date: Thursday, December 04, 2003
Sheet 2 of 2