ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

ปริญญานิพนธ์ ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

8051 MICROCONTROLLER INTERFACE BOARD

นักศึกษา

1. นายนพพร วัฒนสิทธิ์
2. นางสาวรจนา อัชชาศัย
3. นางสาวศุภรัตน์ โพธิ์สิทธิ์
4. นายเอกรินทร์ บุญยเนตร

T 032819

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1. อาจารย์ไพบูลย์ พวงวงศ์ตระกูล
2. อาจารย์ปิยะ จิตธรรมมาภิรมย์

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. อาจารย์ไพบูลย์ พวงวงศ์ตระกูล	
2. อาจารย์กิติพงค์ มะโน	
3. อาจารย์อำพล ทองระอา	
4. อาจารย์สุรพงษ์ สิริพงศ์ดี	
5. อาจารย์สุระชัย พิมพ์สาลี	

วันเดือนปีที่สอบ วันที่ 22 พฤศจิกายน พ.ศ. 2541 เวลา 12:00 น. ถึง 13:00 น.

สถานที่สอบ ห้อง ค 310 คณะครุศาสตร์อุตสาหกรรม

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.ดร.ธีรพล เทพหัสดิน ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 12 เดือน ก.ย. พ.ศ. 42

เลขหมู่.....
เลขทะเบียน 32819
วัน, เดือน, ปี 10 มิ.ย. 2542

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

8051 MICROCONTROLLER INTERFACE BOARD

นายนพพร วัฒนสิทธิ์

นางสาวรจนา อัชชาศัย

นางสาวศุภรัตน์ โพธิ์สิทธิ์

นายเอกรินทร์ บุญยเนตร

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

8051 MICROCONTROLLER INTERFACE BOARD

ผู้จัดทำ

1. นายนพพร วัฒนสิทธิ์
2. นางสาวรจนา อัชชาศัย
3. นางสาวศุภรัตน์ โพธิ์สิทธิ์
4. นายเอกรินทร์ บุญยเนตร

อาจารย์ที่ปรึกษา

ลงนาม.....

(อาจารย์ไพบูลย์ พวงวงศ์ตระกูล)

ลงนาม.....

(อาจารย์ปิยะ จิตธรรมมาภิรมย์)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....

(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

นายนพพร วัฒนสิทธิ์

นางสาวรจนา อัชชาศัย

นางสาวศุภรัตน์ โพสิทธิ์

นายเอกรินทร์ บุญยเนตร

อาจารย์ที่ปรึกษา

อาจารย์ไพบูลย์ พวงวงศ์ตระกูล

อาจารย์ปิยะ จิตธรรมมาภิรมย์

ปีการศึกษา 2541

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 โดยออกแบบชุดทดลองให้สามารถทำงานได้โดยใช้พอร์ต 8255 ของซิงเกิ้ลบอร์ดรุ่น ET-8032 V2.0 และไมโครคอนโทรลเลอร์ 8051 ภายในชุดทดลองประกอบด้วย ชุดแสดงผลข้อมูล, ชุดแสดงผลแบบจุด, วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก, ชุดแสดงระดับสัญญาณ, ชุดแสดงผลตัวเลข 7 ส่วน, บัซเซอร์, มอเตอร์กระแสตรง, วงจรสร้างสภาวะลอจิก, วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล, ชุดตรวจจับความเร็ว, สเต็ปปิ้งมอเตอร์, คีย์แพดและวงจรสร้างแรงดัน 0-2.55 โวลต์ นอกจากนี้ได้ออกแบบโปรแกรมทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ซึ่งชุดทดลองนี้สามารถนำไปใช้เป็นสื่อการเรียนการสอนภายในสถานศึกษาต่างๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

II

8051 MICROCONTROLLER INTERFACE BOARD

MR.NOPPORN WATTANASIT

MISS.ROTJANA ACHASAI

MISS.SUPPARAT PHOSITE

MR.EAKRIN BOONYANET

ADVISORS

MR.PAIBOON PONGWONGTRAGULL

MR.PIYA JITTHAMMAPIROM

1998

ABSTRACT

This thesis presents an experimental set of 8051 microcontroller . It was designed to work by using 8255 and 8051 microcontroller . 8051 Microcontroller Interface Board composes of port monitor, dot matrix display, D/A converter, bargraph display, 7-segment display, buzzer, DC motor, switch input, A/D converter, stepping motor, keypad and potentiometer. We also design and implement the program for this experimental set, which it able to use at institute of education.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

III

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงได้ด้วยดี เนื่องจากการให้ความช่วยเหลือและการให้คำแนะนำปรึกษาจากอาจารย์ที่ปรึกษาปริญญานิพนธ์และอาจารย์ประจำภาควิชาทุกท่านตลอดจนสมาชิกในกลุ่มที่ร่วมมือกันทำงานจนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้ด้วยดี

อนึ่ง คณะผู้จัดทำปริญญานิพนธ์ฉบับนี้ ต้องขอขอบคุณ คุณพ่อ คุณแม่ที่ได้ให้การอบรมเลี้ยงดู สนับสนุนส่งเสริมทางด้านการศึกษามาโดยตลอด อีกทั้งคณะครู-อาจารย์ ทุกท่านที่ได้ให้วิชาความรู้ การอบรมรวมทั้งขอขอบคุณ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุนด้านงบประมาณ ในการจัดทำปริญญานิพนธ์ฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	3
1.3 เนื้อหาโดยสังเขป	4
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 กล่าวนำ	5
2.2 ไมโครคอนโทรลเลอร์ เบอร์ 8051	5
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ เบอร์ 8051	6
2.2.2 โครงสร้างของไมโครคอนโทรลเลอร์ เบอร์ 8051	6
2.2.3 สถาปัตยกรรมของ 8051	8
2.2.4 การจัดการหน่วยความจำของ 8051	11
2.2.5 ฐานเวลาในการทำงานของ 8051	12
2.2.6 การทำงานของ 8051	13
2.3 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	14
2.3.1 รายละเอียดทางไฟฟ้าของการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	15
2.4 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	16
2.4.1 การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแบบประมาณค่าหลายครั้ง	19
2.4.2 ADC0808	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.5 สเต็ปปิ้งมอเตอร์	24
2.5.1 การกระตุ้นและควบคุมการหมุนของสเต็ปปิ้งมอเตอร์	25
2.5.2 การเชื่อมต่อ MCS-51 กับสเต็ปปิ้งมอเตอร์	27
บทที่ 3 การออกแบบและการสร้าง	29
3.1 กล่าวนำ	29
3.2 การออกแบบและการสร้าง	30
3.3 การออกแบบด้านฮาร์ดแวร์	31
3.3.1 ชุดแสดงผลข้อมูลขนาด 8 บิต	31
3.3.2 ชุดแสดงผลแบบจุดขนาด 5x7	31
3.3.3 ชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	34
3.3.4 ชุดแสดงระดับสัญญาณ	36
3.3.5 ชุดแสดงผลตัวเลข 7 ส่วน	37
3.3.6 ชุดสร้างสัญญาณเสียงด้วยบัซเซอร์	40
3.3.7 ชุดควบคุมมอเตอร์กระแสตรง	40
3.3.8 ชุดสร้างสภาวะลอจิกขนาด 8 บิต	42
3.3.9 ชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	43
3.3.10 ชุดตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด	44
3.3.11 ชุดควบคุมสเต็ปปิ้งมอเตอร์	45
3.3.12 ชุดคีย์แพดขนาด 4x3	47
3.3.13 ชุดปรับแรงดัน 0-2.55 โวลต์	47
3.4 การออกแบบด้านซอฟต์แวร์	48
บทที่ 4 ผลการทดลองและทดสอบ	60
4.1 กล่าวนำ	60
4.2 การทดลองชุดแสดงผลข้อมูลขนาด 8 บิต	63
4.3 การทดลองชุดสร้างสภาวะลอจิกขนาด 8 บิต	63
4.4 การทดลองชุดแสดงผลตัวเลข 7 ส่วนขนาด 2 หลัก	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.5 การทดลองชุดสร้างสัญญาณเสียงบัซเซอร์	64
4.6 การทดลองชุดสร้างสัญญาณแสดงผลด้วยไดโอดเปล่งแสง	65
4.7 การทดลองชุดควบคุมมอเตอร์กระแสตรง	65
4.8 การทดลองชุดควบคุมสเต็ปปิ้งมอเตอร์	66
4.9 การทดลองชุดแสดงผลแบบจุดขนาด 5X7	66
4.10 การทดลองชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	66
4.11 การทดลองชุดแสดงระดับสัญญาณและชุดปรับแรงดัน 0-2.55 โวลต์	67
4.12 การทดลองชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	67
4.13 การทดลองชุดวัดความสว่างแสง	68
4.14 การทดลองชุดตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด	68
4.15 การทดลองคีย์แพด	69
บทที่ 5 สรุป อภิปรายและข้อเสนอแนะ	70
5.1 บทสรุป	70
5.1.1 วงจรของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	70
5.1.2 ใบงานการทดลอง	71
5.2 ปัญหาและแนวทางแก้ไข	71
5.2.1 ปัญหาในส่วนของฮาร์ดแวร์	71
5.2.2 ปัญหาในส่วนของซอฟต์แวร์	72
5.3 ข้อเสนอแนะและแนวทางในการพัฒนา	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ก คู่มือการใช้งาน	74
ภาคผนวก ข ใบงานการทดลอง	88
ภาคผนวก ค วงจรและลายวงจรพิมพ์	235
ภาคผนวก ง รายการอุปกรณ์	244
ภาคผนวก จ รูปต้นแบบของชุดทดลอง	250
บรรณานุกรม	252
ประวัติผู้แต่ง	253

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051	5
ตารางที่ 2.2 การจ่ายกระแสแบบเวฟให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์	26
ตารางที่ 2.3 การจ่ายกระแสแบบ 2 เฟสให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์	26
ตารางที่ 2.4 การจ่ายกระแสแบบครึ่งสเต็ปให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์	27
ตารางที่ 3.1 การกำหนดคอลัมน์ในการแสดงผลของชุดแสดงผลแบบจุด	33
ตารางที่ 3.2 การทำงานของ 4511	38
ตารางที่ 3.3 อินพุตของ D0 และ D1	39
ตารางที่ 3.4 บิตควบคุมการทำงานและทิศทางการหมุนของมอเตอร์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 1.1 โครงสร้างภายในของ Z80	2
รูปที่ 1.2 โครงสร้างภายในของ 8051	3
รูปที่ 2.1 โครงสร้างของ 8051	6
รูปที่ 2.2 สถาปัตยกรรมภายในของ 8051	9
รูปที่ 2.3 ลักษณะขาภายนอกของ 8051	9
รูปที่ 2.4 การจัดพื้นที่หน่วยความจำสำหรับโปรแกรมของ 8051	11
รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูล	12
รูปที่ 2.6 เวลาพื้นฐานของ 8051 และลำดับของช่วงเวลาใน 1 แมชชีนไซเคิล	12
รูปที่ 2.7 ลำดับสถานะการทำงานใน MCS-51	14
รูปที่ 2.8 วงจร R-2R Ladder	15
รูปที่ 2.9 ADC ขนาด 3 บิต	17
รูปที่ 2.10 ความผิดพลาดของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	18
รูปที่ 2.11 การแปลงสัญญาณแบบประมาณค่าหลายครั้ง	19
รูปที่ 2.12 แผนผังการทำงานของวงจรแปลงสัญญาณแบบประมาณค่าหลายครั้ง	19
รูปที่ 2.13 ผังงานของโปรแกรมแปลงสัญญาณแบบประมาณค่าหลายครั้ง	20
รูปที่ 2.14 แผนผังการทำงานของ ADC0808	22
รูปที่ 2.15 การจัดขาของ ADC0808	23
รูปที่ 2.16 แผนผังเวลาของ ADC0808	24
รูปที่ 2.17 สเต็ปปิ้งมอเตอร์ 4 เฟส แบบยูนิโพลาร์เพอร์มาเนนแม็กเนต	25
รูปที่ 2.18 แผนผังการทำงานของการเชื่อมต่อ MCS-51 กับสเต็ปปิ้งมอเตอร์	28
รูปที่ 2.19 ตัวอย่างการขับมอเตอร์แบบไบโพลาร์	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 2.20 ตัวอย่างการขับมอเตอร์แบบยูนิโพลาร์	28
รูปที่ 3.1 ส่วนประกอบของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	30
รูปที่ 3.2 วงจรแสดงผลข้อมูลขนาด 8 บิต	31
รูปที่ 3.3 ชุดแสดงผลแบบจุดขนาด 5×7	32
รูปที่ 3.4 วงจรชุดแสดงผลแบบจุดขนาด 5×7	33
รูปที่ 3.5 การต่อพอร์ตกับชุดแสดงผลแบบจุดขนาด 5×7	34
รูปที่ 3.6 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	35
รูปที่ 3.7 โครงสร้างของชุดแสดงระดับสัญญาณ	36
รูปที่ 3.8 วงจรแสดงระดับแรงดัน	36
รูปที่ 3.9 การต่อขาวงจรแสดงผลตัวเลข 7 ส่วน	37
รูปที่ 3.10 วงจรแสดงผลตัวเลข 7 ส่วน	37
รูปที่ 3.11 บิตที่ใช้กับวงจรแสดงผลตัวเลข 7 ส่วน	38
รูปที่ 3.12 วงจรสร้างสัญญาณเสียงด้วยบัซเซอร์	40
รูปที่ 3.13 การจ่ายแรงดันให้กับมอเตอร์เป็นช่วงๆ	41
รูปที่ 3.14 วงจรควบคุมมอเตอร์กระแสตรง	41
รูปที่ 3.15 วงจรสร้างสภาวะลอจิกขนาด 8 บิต	43
รูปที่ 3.16 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	44
รูปที่ 3.17 เอาต์พุตของวงจรตรวจจับความเร็วเมื่อมอเตอร์หมุน 1 รอบ	45
รูปที่ 3.18 วงจรตรวจจับความเร็ว	45
รูปที่ 3.19 วงจรควบคุมสเต็ปปิ้งมอเตอร์	46
รูปที่ 3.20 สเต็ปปิ้งมอเตอร์แบบ 2 เฟส 6 สาย	46
รูปที่ 3.21 วงจรคีย์แพด	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปภาพ	หน้า
รูปที่ 3.22 วงจรปรับแรงดัน 0-2.55 โวลต์	48
รูปที่ 3.23 ผังงานของโปรแกรมแสดงผลข้อมูลขนาด 8 บิต	48
รูปที่ 3.24 ผังงานของโปรแกรมสร้างสภาวะลอจิกขนาด 8 บิต	49
รูปที่ 3.25 ผังงานของโปรแกรมสร้างสัญญาณเสียงด้วยบัซเซอร์	49
รูปที่ 3.26 ผังงานของโปรแกรมแสดงตัวเลข 7 ส่วนขนาด 2 หลัก	50
รูปที่ 3.27 ผังงานของโปรแกรมสร้างสัญญาณแสดงผลด้วยไดโอดเปล่งแสง	51
รูปที่ 3.28 ผังงานของโปรแกรมควบคุมมอเตอร์กระแสตรง	52
รูปที่ 3.29 ผังงานของโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์	53
รูปที่ 3.30 ผังงานของโปรแกรมแสดงผลแบบจุดขนาด 5×7	54
รูปที่ 3.31 ผังงานของโปรแกรมแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก	54
รูปที่ 3.32 ผังงานของโปรแกรมแสดงระดับสัญญาณ	55
รูปที่ 3.33 ผังงานของโปรแกรมปรับแรงดัน 0-2.55 โวลต์	55
รูปที่ 3.34 ผังงานของโปรแกรมแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล	55
รูปที่ 3.35 ผังงานของโปรแกรมตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด	56
รูปที่ 3.36 ผังงานของโปรแกรมรับข้อมูลจากคีย์แพด	57
รูปที่ 4.1 ภาพด้านบนของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	60
รูปที่ 4.2 ภาพด้านล่างของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	61
รูปที่ 4.3 ภาพด้านบนของชุดแปลงคอนเน็กเตอร์	61
รูปที่ 4.4 การวางตำแหน่งของวงจรบนชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	62
รูปที่ 4.5 การวางตำแหน่งของคอนเน็กเตอร์บนชุดแปลงคอนเน็กเตอร์	62
รูปที่ 4.6 การเชื่อมต่อชุดทดลองกับซิงเกิ้ลบอร์ดรุ่น ET-8032 V2.0	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

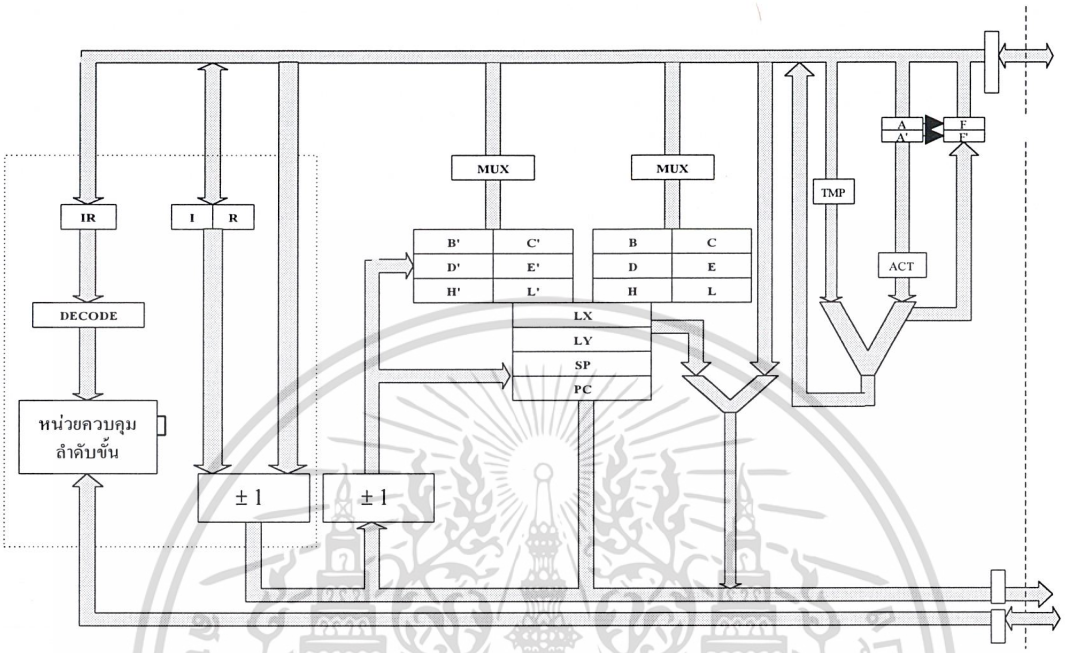
ปัจจุบันได้มีการนำไมโครโพรเซสเซอร์เข้ามาใช้ในชีวิตประจำวันมากขึ้น โดยจะสังเกตเห็นได้จากการนำไปใช้ในด้านอุตสาหกรรม, ด้านการผลิต, ด้านการโทรคมนาคมสื่อสาร รวมถึงสิ่งอำนวยความสะดวกด้านต่างๆ ไมโครโพรเซสเซอร์ที่นิยมใช้กันมากตัวหนึ่งก็คือ Z80 เพราะชุดคำสั่งของ Z80 มีข้อดีคือ ใช้กระทำกับข้อมูลเป็นบล็อก (Block) ซึ่งสามารถนำไปใช้ได้ดีกับระบบที่ต้องการฐานข้อมูล (Database) แต่การที่ Z80 มีชุดคำสั่งและมีรีจิสเตอร์ให้ใช้งานมากมายนั้น ทำให้เกิดความสับสนในการใช้งานได้ เพราะไม่รู้ว่าจะใช้คำสั่งใดกับรีจิสเตอร์ตัวใดดี นอกจากนี้การต่อใช้งาน Z80 โดยพื้นฐานจะต้องประกอบด้วยไอซีอย่างน้อยอีก 5 ตัวคือ รอม, แรม, พอร์ตอินพุตและเอาต์พุต, วงจรถอดรหัส, วงจรกำเนิดความถี่ (Oscillator) ทำให้ไม่เหมาะที่จะใช้กับระบบควบคุมที่มีขนาดเล็ก เพราะสิ้นเปลืองเนื้อที่และค่าใช้จ่าย ทั้งยังมีความยุ่งยากและซับซ้อนในการติดต่อหรือการเขียนโปรแกรมควบคุมการทำงานด้วยภาษาแอสแซมบลี้ดังนั้นจึงมีการผลิตไมโครคอนโทรลเลอร์ เบอร์ 8051 ขึ้นมาใช้งานเพราะเหมาะสำหรับงานควบคุมโดยเฉพาะ การใช้งานก็ไม่จำเป็นต้องต่อไอซีอื่นเพิ่ม แต่ละคำสั่งจะใช้เวลาในการทำงานไม่เกิน 2 ไมโครวินาที (ที่สัญญาณนาฬิกาขนาด 12 เมกะเฮิรตซ์) ทำให้โปรแกรมที่เขียนขึ้นมาทำงานได้รวดเร็ว นอกจากนี้ยังสามารถเขียนโปรแกรมควบคุมการทำงานได้ทั้งภาษาเบสิกและภาษาแอสแซมบลี้ ดังนั้นคณะผู้จัดทำจึงได้นำไมโครคอนโทรลเลอร์ เบอร์ 8051 มาใช้เพื่อเป็นการพัฒนาความสามารถในการนำไมโครคอนโทรลเลอร์ เบอร์ 8051 มาประยุกต์ใช้งาน

จากรูปที่ 1.1 แสดงให้เห็นถึงโครงสร้างภายในของ Z80 ซึ่งสามารถอ้างแอดเดรสของรอมและแรมได้ 64 กิโลไบต์ ติดต่อพอร์ตอินพุตและเอาต์พุตได้ 256 พอร์ต และสามารถรับอินเตอร์รัพต์ได้ 2 แหล่งเท่านั้น คือ NMI และ INT

และสำหรับรูปที่ 1.2 แสดงให้เห็นถึงโครงสร้างภายในของไมโครคอนโทรลเลอร์ เบอร์ 8051 ซึ่งภายในซีพียูนอกจากจะมี หน่วยควบคุมและหน่วยคำนวณทางคณิตศาสตร์และตรรกเหมือนที่มีใน Z80 แล้วยังประกอบไปด้วยหน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ หน่วยความจำข้อมูลภายในขนาด 128 ไบต์, วงจรนับ/จับเวลาขนาด 16 บิต 2 แชนเนล, อินพุต/เอาต์พุต พอร์ต 32 บิต, พอร์ตอนุกรมและสามารถอ้างอินเตอร์รัพต์ได้ 6 แหล่ง อ้างแอดเดรสหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมและหน่วยความจำข้อมูลได้อย่างละ 64 กิโลไบต์ ส่วนพอร์ตอินพุตและเอาต์พุตนั้นใช้เทคนิค Memory Map I/O

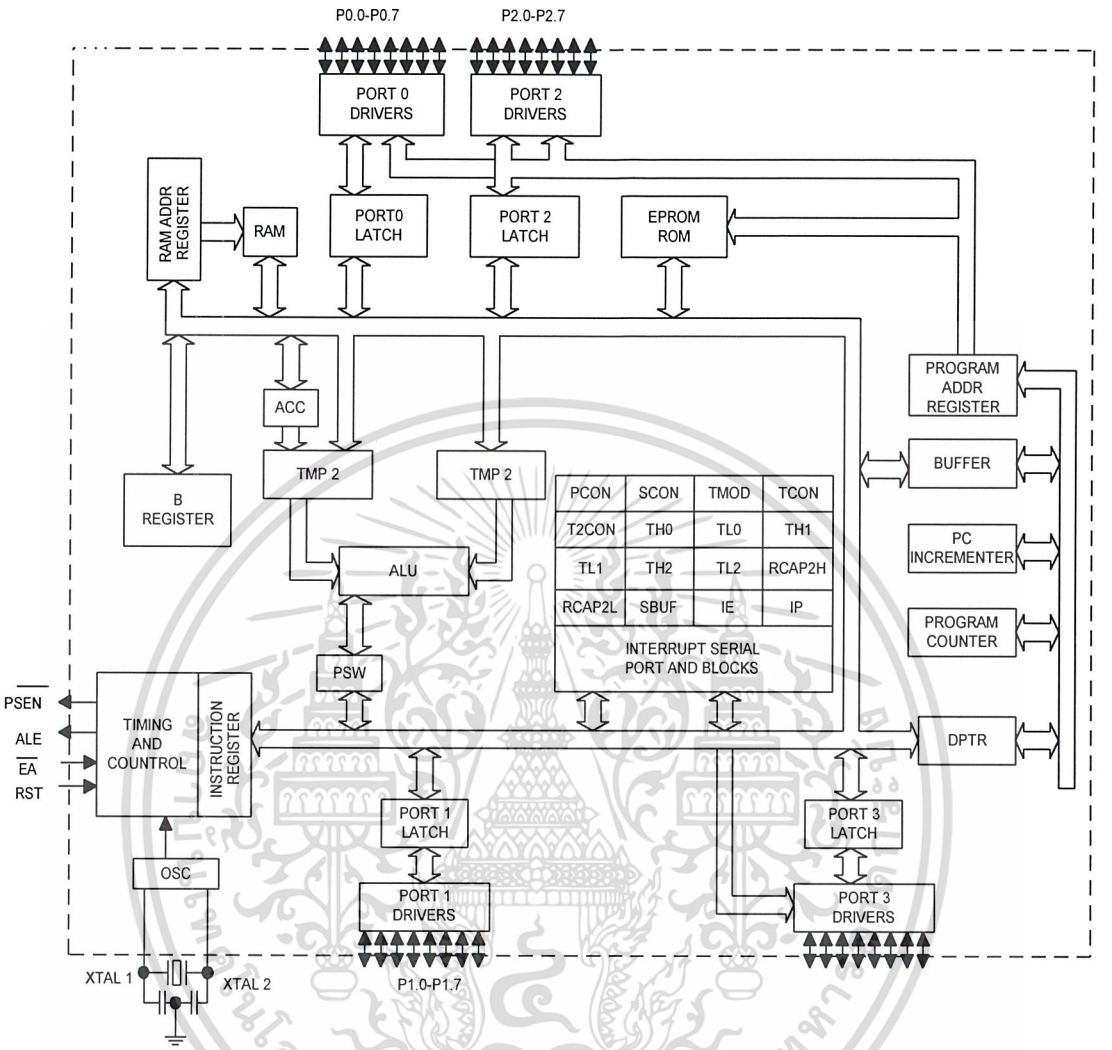


รูปที่ 1.1 โครงสร้างภายในของ Z80

จะเห็นได้ว่าระบบควบคุมที่ใช้ไมโครคอนโทรลเลอร์ เบอร์ 8051 เป็นซีพียูนั้นสามารถทำงานได้คล่องตัวกว่า, มีเสถียรภาพดีกว่า, มีขนาดเล็กกะทัดรัดและประหยัดค่าใช้จ่าย ด้วยเหตุนี้ทางผู้จัดทำจึงได้เลือกใช้ ไมโครคอนโทรลเลอร์ 8051 เป็นตัวประมวลผล

วัตถุประสงค์ของการทำปริญญานิพนธ์ เพื่อศึกษาการทำงาน, การเชื่อมต่อของอุปกรณ์อิเล็กทรอนิกส์ร่วมกับไมโครคอนโทรลเลอร์ เบอร์ 8051 และเพื่อทำการออกแบบและจัดสร้างชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ เบอร์ 8051 รวมทั้งการเขียนโปรแกรมควบคุมการทำงานเพื่อให้ ไมโครคอนโทรลเลอร์ เบอร์ 8051 ทำงานได้ตามแนวทางที่กำหนดไว้ นอกจากนี้ยังเพื่อการออกแบบและสร้างใบงานการทดลอง, คู่มือในการใช้ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ เบอร์ 8051 อีกด้วย ทั้งนี้เพื่อเป็นแนวทางการศึกษาและพัฒนาความรู้ด้านไมโครคอนโทรลเลอร์ เบอร์ 8051 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 โครงสร้างภายในของ 8051

1.2 ขีดความสามารถของโครงงาน

ขอบเขตของการทำปริญญานิพนธ์ เพื่อนำไมโครคอนโทรลเลอร์ เบอร์ 8051 มาใช้ในศึกษาคำสั่งในการควบคุมการทำงานกับอุปกรณ์ต่อร่วม โดยใช้ไอซี 8255 เป็นตัวกลาง เช่น ชุดแสดงผลข้อมูลเป็นตัวแสดงผลแบบตัวเลข 0-9 จำนวน 2 หลัก ซึ่งประโยชน์ที่คาดว่าจะได้รับจากการทำโครงงานนี้ก็คือ ความรู้และความเข้าใจในหลักการทำงานและวิธีการเชื่อมต่อของอุปกรณ์อิเล็กทรอนิกส์ร่วมกับไมโครคอนโทรลเลอร์ เบอร์ 8051 ทั้งยังสามารถออกแบบและสร้างชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 พร้อมทั้งเขียนโปรแกรมควบคุมการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งเป็นบทต่างๆ เพื่อความสะดวกและความง่ายต่อการทำความเข้าใจในการศึกษา ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาทางสถาปัตยกรรมของไมโครคอนโทรลเลอร์ เบอร์ 8051 หลักการทำงานของไมโครคอนโทรลเลอร์ 8051 และทฤษฎีการเชื่อมต่อ

บทที่ 3 การออกแบบและการสร้างชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ซึ่งประกอบด้วย วงจรชุดแสดงผลข้อมูลขนาด 8 บิต, วงจรควบคุมมอเตอร์กระแสตรง, วงจรชุดแสดงผลแบบจุดขนาด 5×7, วงจรชุดสร้างสัญญาณเสียงด้วยบัซเซอร์, ชุดสร้างสัญญาณแสดงผลด้วยไดโอดเปล่งแสง, วงจรชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก, วงจรชุดแสดงระดับสัญญาณ, วงจรชุดปรับแรงดัน 0-2.55 โวลต์, วงจรชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล, วงจรตรวจวัดความสว่างแสง, วงจรชุดสร้างสภาวะลอจิกขนาด 8 บิต, วงจรชุดตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด, วงจรชุดคีย์แพด 4×3, วงจรชุดสเต็ปปิ้งมอเตอร์ รวมทั้งลักษณะการทำงานของซอฟต์แวร์ ที่ใช้ควบคุมการทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

บทที่ 4 การทดลองและผลการทดลองการทำงานของชุดทดลองการเชื่อมต่อไมโครโทรลเลอร์ 8051 อาทิเช่น ชุดแสดงผลข้อมูลขนาด 8 บิต, ชุดแสดงผลแบบจุดขนาด 5×7, ชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก, ชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล, ชุดแสดงระดับสัญญาณและชุดแสดงผลตัวเลข 7 ส่วน, ชุดสเต็ปปิ้งมอเตอร์

บทที่ 5 สรุปปัญหา แนวทางแก้ไขปัญหาที่เกิดขึ้นตลอดจนข้อบกพร่องที่พบในการทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 พร้อมทั้งแนวคิดพัฒนาชุดทดลองเพื่อให้มีประสิทธิภาพในการใช้งานที่สูงกว่าเดิม

ในภาคผนวกแสดงรายละเอียดของโปรแกรมและรายการอุปกรณ์ต่างๆ ที่ใช้ในการจัดทำโครงงาน ดังนี้

ภาคผนวก ก คู่มือการใช้งาน

ภาคผนวก ข ใบงานการทดลอง

ภาคผนวก ค วงจรและลายวงจรพิมพ์

ภาคผนวก ง รายการอุปกรณ์

ภาคผนวก จ รูปต้นแบบของชุดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 เป็นชุดทดลองสำหรับใช้ทำการทดลองควบคุมอุปกรณ์ต่างๆ ให้สามารถทำงานได้ โดยใช้ไอซี 8255 เป็นตัวกลางระหว่างไมโครคอนโทรลเลอร์ 8051 กับชุดทดลอง เพื่อให้สามารถทำการออกแบบและจัดสร้างชุดทดลองได้ ต้องรู้หลักการทำงานของไมโครคอนโทรลเลอร์ 8051 และการเชื่อมต่อกับ 8255 นอกจากให้สามารถสร้างชุดทดลองได้แล้วยังเป็นแนวทางในการพัฒนาโครงงานชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 และเนื้อหาที่ได้กล่าวไว้ข้างต้นได้แสดงรายละเอียดตามหัวข้อต่อไปนี้

2.2 ไมโครคอนโทรลเลอร์ เบอร์ 8051

ไมโครคอนโทรลเลอร์ในตระกูล MCS–51 ที่มีขนาด 8 บิต ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ต่างๆ ดังตารางที่ 2.1 ทุกๆ เบอร์จะมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดของหน่วยความจำภายในและภายนอกที่แตกต่างกัน เพื่อความเหมาะสมในการนำไปใช้งานตามความต้องการต่างๆ แต่เดิม 8051 ถูกสร้างด้วยวิธี HMOS I แต่ในปัจจุบันได้สร้างด้วยวิธี HMOS II จึงมีชื่อเป็น 8051AH ไมโครคอนโทรลเลอร์ในตระกูล 51 นั้น ถึงแม้ว่าจะมีหลายเบอร์แต่เราก็จะเรียกว่าเป็น “8051” ซึ่งหมายถึงไมโครคอนโทรลเลอร์ตระกูล 51 ส่วนเบอร์ 8032 และ 8052 มีหน่วยความจำภายในเพิ่มขึ้นและมีวงจรนับ/จับเวลา ขนาด 16 บิต เพิ่มขึ้นดังตารางที่ 2.1

ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051

เบอร์	หน่วยความจำภายใน		จำนวนไทเมอร์ / เคาน์เตอร์	อินเตอร์รัพต์ หมายเลข
	เก็บโปรแกรม	เก็บข้อมูล		
8052H	8K x 8 ROM	256 x 8 ROM	3 x 16-Bit	6
8051H	4K x 8 ROM	128 x 8 ROM	2 x 16 Bit	5
8051	4K x 8 ROM	128 x 8 ROM	2 x 16 Bit	5
8032AH	ไม่มี	256 x 8 ROM	3 x 16-Bit	6
8031AH	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8031	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8751H	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5
80751H-12	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5

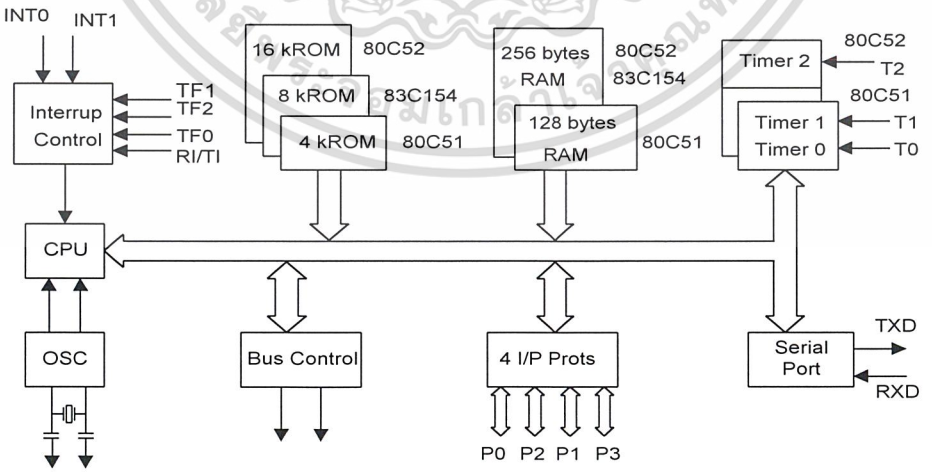
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ เบอร์ 8051

1. หน่วยประมวลผลกลางขนาด 8 บิต
2. หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์
3. หน่วยความจำข้อมูลภายในขนาด 128 ไบต์
4. อ้างตำแหน่งของหน่วยความจำโปรแกรมได้ถึง 64 กิโลไบต์
5. อ้างตำแหน่งของหน่วยความจำข้อมูลได้ถึง 64 กิโลไบต์
6. หน่วยความจำโปรแกรมและข้อมูลที่อยู่ภายนอกไอซีแยกจากกันอย่างละ 64 กิโลไบต์
7. มีพอร์ตอินพุต/เอาต์พุต แบบขนานจำนวน 4 พอร์ต แยกกันอย่างอิสระ
8. มีวงจรนับ/จับเวลา ขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
9. มีพอร์ตการสื่อสารอนุกรม รับส่งข้อมูลได้ในเวลาเดียวกัน สามารถเลือกรูปแบบการส่งได้ 4 รูปแบบ
10. รับสัญญาณอินเตอร์รัพต์ได้ 6 แหล่ง กระโดดไปทำงานตอบสนองได้ 5 ตำแหน่ง
11. มีวงจรออสซิลเลเตอร์ภายใน
12. นำข้อมูลมา AND, OR หรือทำการ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

2.2.2 โครงสร้างของไมโครคอนโทรลเลอร์ เบอร์ 8051

ภายใน 8051 จะประกอบขึ้นด้วยเกตชนิดต่างๆ เช่น AND, OR, NOT ซึ่งแต่ละเกตเหล่านี้จะนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง, วงจรสร้างสัญญาณนาฬิกา เป็นต้น โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ 8051 จะประกอบด้วย 3 ส่วนหลักๆ ดังนี้

1. ซีพียู (Central Processing Unit) ส่วนนี้จะทำหน้าที่สร้างสัญญาณควบคุมการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออก ซึ่งส่วนควบคุมการขัดจังหวะ และส่วนควบคุมบัสก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณวงจรควบคุมจากซีพียูนี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่งที่มีการกำหนดไว้และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุกๆ ส่วนทำงานประสานกันอย่างถูกต้อง

ในซีพียูยังประกอบด้วยส่วนประมวลผล (Arithmetic Logic Unit) ทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในหน่วยความจำที่ต้องการ

2. หน่วยความจำ (Memory) มีไว้สำหรับจดจำข้อมูล ซึ่งในการนำข้อมูลเข้าและออกจากหน่วยความจำ เราจำเป็นต้องรู้ตำแหน่งของหน่วยความจำ (Address) ในการนำข้อมูลเข้าไปเก็บไว้ในหน่วยความจำ เรียกว่า การเขียนข้อมูลและการนำข้อมูลออกจากหน่วยความจำ เรียกว่า การอ่านข้อมูล ในไมโครคอนโทรลเลอร์ 8051 ข้อมูลในแต่ละตำแหน่งจะมีขนาด 8 บิต ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะสามารถเก็บข้อมูลมีค่าได้ระหว่าง 00000000_2 ถึง 11111111_2 หรือ 00H ถึง 0FFH ในการติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่ม คือ

2.1 ตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ซึ่งในไมโครคอนโทรลเลอร์ 8051 จะมีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลที่มีขนาดสูงสุดชนิดละ 65,536 ตำแหน่ง (64 กิโลไบต์) ดังนั้นการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ 65,536)

2.2 ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำในตำแหน่งที่เราต้องการ

2.3สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำเพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล โดยวงจรถอดรหัสคำสั่งจะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

3. อุปกรณ์อินพุต/เอาต์พุต (Input/Output Device) เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือนำข้อมูลออกจากไมโครคอนโทรลเลอร์ 8051 ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกได้ อุปกรณ์อินพุต/เอาต์พุต ได้แก่ อินพุต/เอาต์พุตพอร์ตแบบขนาน, วงจรนับ/จับเวลา 0, วงจรนับ/จับเวลา 1, พอร์ตสื่อสารอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 พอร์ตแบบขนาน เป็นที่สำหรับใช้รับส่งข้อมูลซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัวไมโครคอนโทรลเลอร์ เบอร์ 8051 มีทั้งหมด 4 พอร์ตโดยแต่ละพอร์ตจะรับส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตจะใช้งานมากกว่า 1 อย่างก็ได้

3.2 วงจรนับ/จับเวลา 0 และวงจรนับ/จับเวลา 1 เป็นวงจรนับที่สามารถทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอกไมโครคอนโทรลเลอร์ เบอร์ 8051 หรือจำนวนของสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์ เบอร์ 8051 ก็ได้ สามารถตั้งค่าเริ่มต้นของการนับและอ่านค่าการนับได้โดยซีพียู

3.3 พอร์ตอนุกรม ซีพียูจะอ่านและเขียนข้อมูลกับพอร์ตอนุกรมเป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกจากไมโครคอนโทรลเลอร์ เบอร์ 8051 เรียงไปทีละบิตออกจากขา TXD และในการรับข้อมูลก็จะรับเข้ามาทีละบิตทางขา RXD และจัดเรียงใหม่เป็น 8 บิต เพื่อให้ซีพียูอ่านไปใช้งานต่อไป

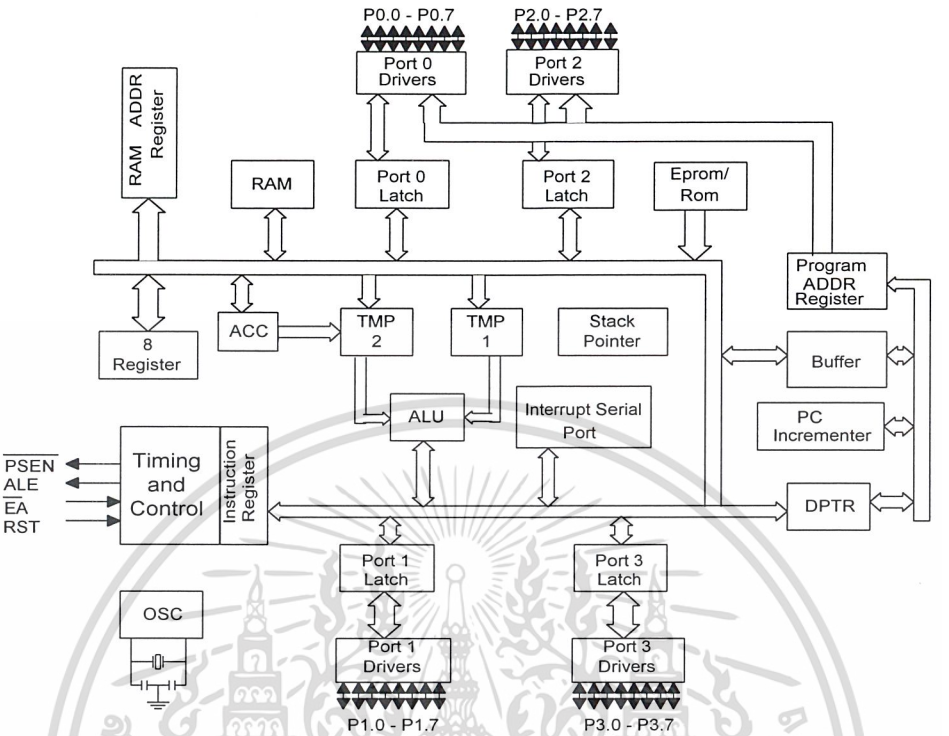
ในไมโครคอนโทรลเลอร์ เบอร์ 8051 มีพอร์ตให้ใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่างๆ ได้มากมาย การจะนำพอร์ตไปใช้งานจะต้องเขียนโปรแกรมขึ้นมาควบคุม

2.2.3 สถาปัตยกรรมของ 8051

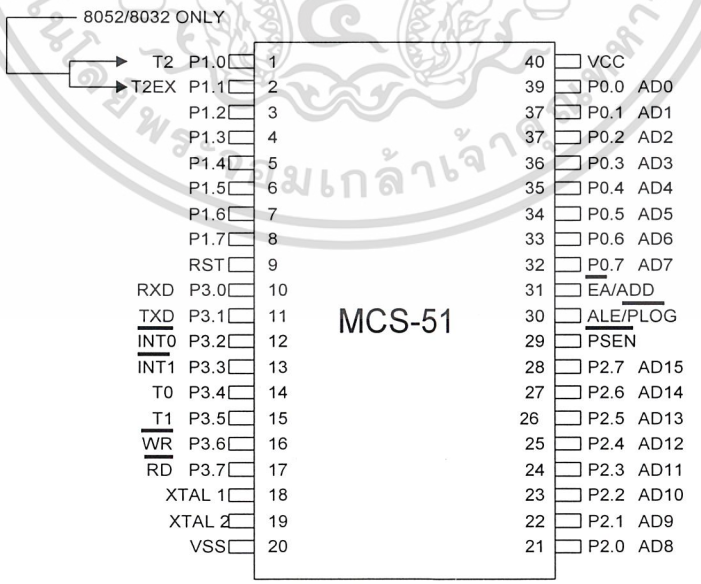
ในรูปที่ 2.2 เป็นรูปแสดงสถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ เบอร์ 8051 ซึ่งอธิบายถึงส่วนประกอบย่อยๆ ภายในตัวของไมโครคอนโทรลเลอร์ เบอร์ 8051 และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขาของไมโครคอนโทรลเลอร์ เบอร์ 8051 ที่มีอยู่ 40 ขาดังรูปที่ 2.3 ไมโครคอนโทรลเลอร์ เบอร์ 8051 บรรจุอยู่ในวงจรรวมแบบ DIP (Dual Inline Package) แบบ 40 ขา ดังนี้

VCC	(ขา 40) ต่อไฟเลี้ยง +5 โวลต์
VSS	(ขา 20) ต่อลงกราวด์
PORT 0	(ขา 32-29) มีทั้งหมด 8 บิต คือ P0.0-P0.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไปใช้เป็นตัวส่งแอดเดรสไบต์ต่ำ (A0-A7) และรับส่งข้อมูล (D0-D7) จากหน่วยความจำภายนอก
PORT 1	(ขา 1-8) มีทั้งหมด 8 บิต คือ P1.0-P1.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไป
PORT 2	(ขา 21-28) มีทั้งหมด 8 บิต คือ P2.0-P2.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไปและใช้เป็นตัวส่งแอดเดรสไบต์สูง (A8-A15) เพื่อใช้ติดต่อกับหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 สถาปัตยกรรมภายในของ 8051



รูปที่ 2.3 ลักษณะขาภายนอกของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT 3 (ขา 10-17) มี ทั้งหมด 8 บิต คือ P3.0-P3.7 ใช้งานเป็นอินพุต/เอาต์พุต พอร์ตทั่วไปและใช้งานในหน้าที่พิเศษดังนี้

P3.0 /RXD (Serial Input Port) ใช้รับข้อมูลแบบอนุกรม

P3.1 /TXD (Serial Output Port) ใช้ส่งข้อมูลแบบอนุกรม

P3.2 / $\overline{\text{INT 0}}$ (External Interrupt) ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.3 / $\overline{\text{INT 1}}$ (External Interrupt) ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.4 /T0 (Timer/Counter 0 External Input) ใช้เป็นอินพุตให้วงจรนับ/จับเวลาชุดที่ 0

P3.5 /T1 (Timer/Counter 1 External Input) ใช้เป็นอินพุตให้วงจรนับ/จับเวลาชุดที่ 1

P3.3 / $\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาควบคุมการเขียนข้อมูลจากหน่วยความจำภายนอก

P3.3 / $\overline{\text{RD}}$ (External Data Memory Read Strobe) ขาควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก

RST (ขา 9) Reset ใช้สำหรับรีเซ็ตวงจรทุกวงจรภายในชิพ เพื่อเริ่มต้นการทำงานใหม่ ในการรีเซ็ตต้องป้อนลอจิก “1” นานอย่างน้อย 2 แมชชีนไซเคิล

ALE (ขา 30) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอกเพื่อควบคุมการแลตซ์ค่าแอดเดรสไบต์ต่ำจากพอร์ต 0

$\overline{\text{PSEN}}$ (ขา 29) Program Strobe Enable เป็นขาสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขานี้ Active มีลอจิกเป็น “0” จะอ่านโปรแกรมจากหน่วยความจำภายนอก และถ้าเป็นการอ่านหน่วยความจำโปรแกรมภายในขานี้จะไม่ Active

$\overline{\text{EA}}$ (ขา 31) External Access เป็นขาที่ใช้สำหรับเลือกว่าให้ทำงานจากหน่วยความจำโปรแกรมภายในหรือหน่วยความจำโปรแกรมภายนอกชิพ เมื่อขานี้ Active มีลอจิกเป็น “0” จะเป็นการทำงานตามคำสั่งในหน่วยความจำโปรแกรมภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

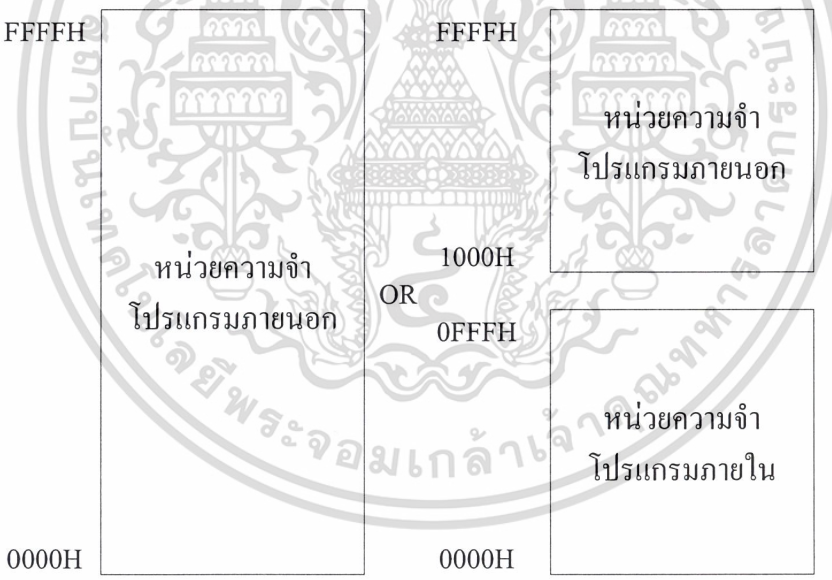
XTAL1 (ขา 19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์

XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิล-เลเตอร์

2.2.4 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกเป็น 2 แบบตามลักษณะการใช้งาน ดังนี้

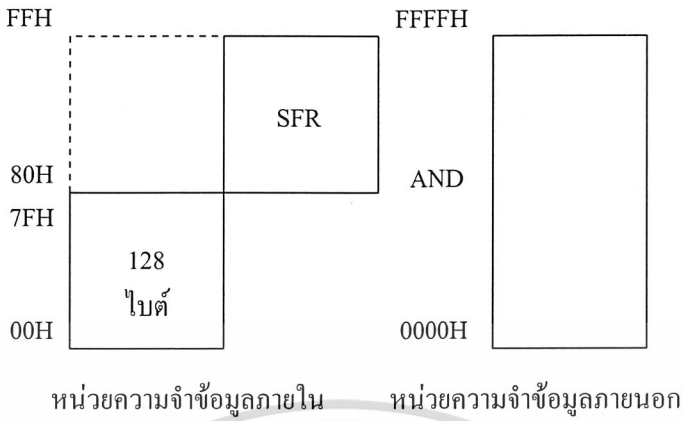
1. หน่วยความจำโปรแกรม (Program Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปของภาษาเครื่องซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำโปรแกรมไปทำการถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้เป็นแบบรอมและผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ ส่วนที่เป็นหน่วยความจำโปรแกรมก็คือแรมขนาด 4 กิโลไบต์ ดังรูปที่ 2.4



รูปที่ 2.4 การจัดพื้นที่หน่วยความจำสำหรับโปรแกรมของ 8051

2. หน่วยความจำข้อมูล (Data Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูล สามารถอ่านและเขียนข้อมูลได้ ซึ่งหน่วยความจำภายในมีขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำภายนอกไอซีมีขนาด 64 กิโลไบต์ ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

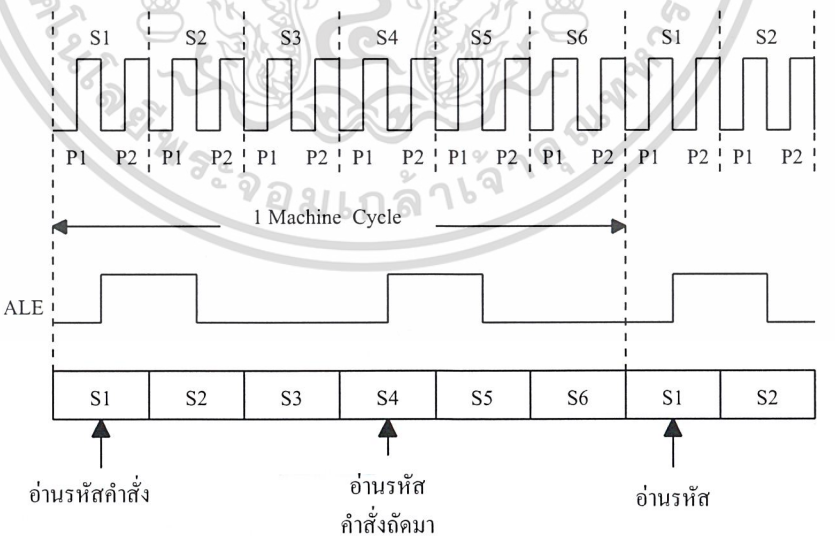


รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูล

2.2.5 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์

แมชชีนไซเคิล (Machine Cycle) คือรอบการทำงานของคำสั่ง เป็นค่าที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ถ้าเป็นคำสั่งที่ซับซ้อนมากก็จะต้องใช้เวลานาน 2 -3 แมชชีนไซเคิล

1 แมชชีนไซเคิล จะประกอบด้วยสัญญาณนาฬิกาจำนวน 12 ลูก โดยสัญญาณนาฬิกาแต่ละลูกเรียกว่าเฟส (Phase) สัญญาณนาฬิกา 2 เฟส รวมกันเป็น 1 สเตท (State) เพราะฉะนั้นใน 1 แมชชีนไซเคิลจึงมีทั้งหมด 6 สเตท



รูปที่ 2.6 เวลาพื้นฐานของ 8051 และลำดับของช่วงเวลาใน 1 แมชชีนไซเคิล

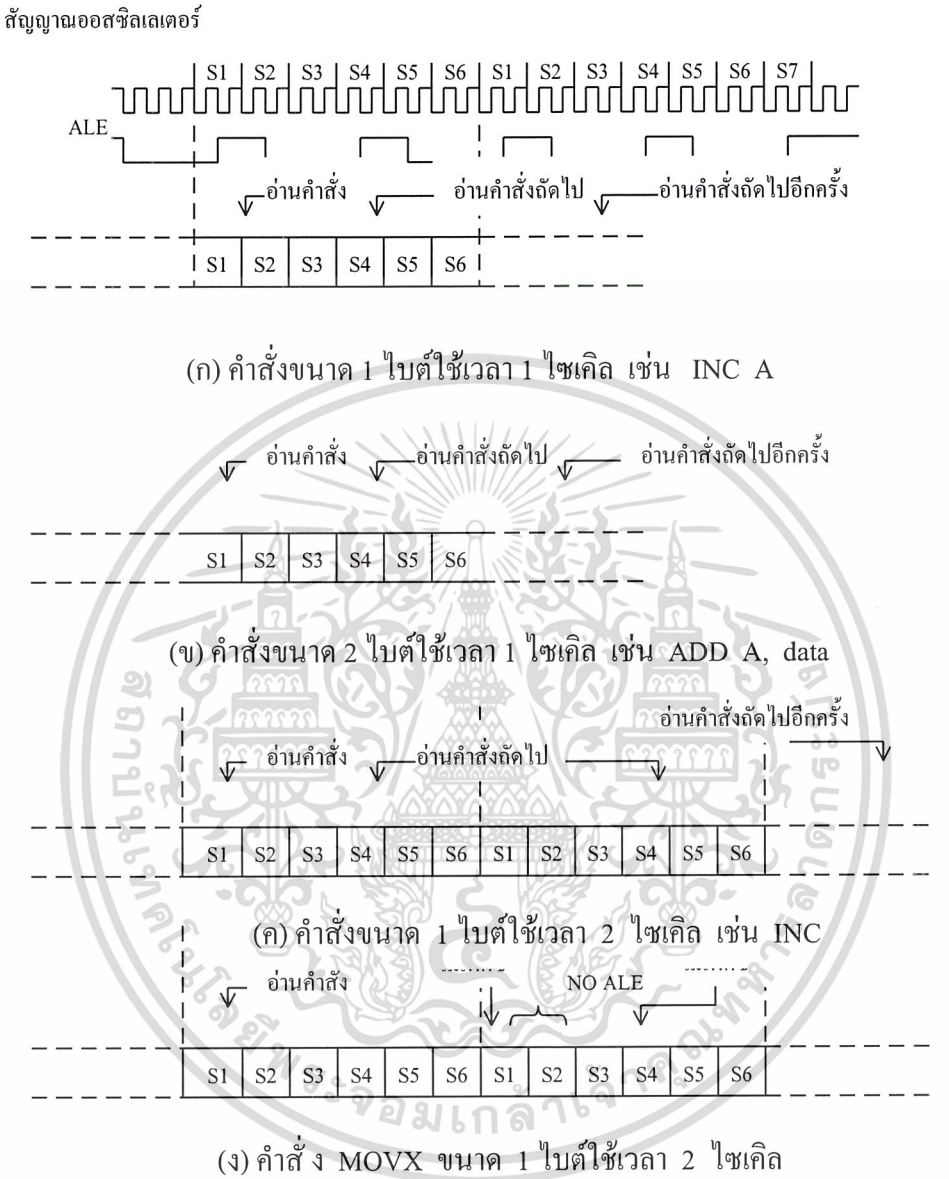
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การทำงานของ 8051

จากรูปที่ 2.7 เมื่อป้อนไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์ เบอร์ 8051 ซึ่งมีวงจรรีเซ็ต เมื่อเปิดเครื่องจะเกิดการรีเซ็ตการทำงานภายในไมโครคอนโทรลเลอร์ เบอร์ 8051 เริ่มจากภาคโปรแกรมเคาน์เตอร์ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนเส้นทางหมายเลข 1 เส้นทางนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR register ค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็นรอมภายใน หรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา $\overline{EA}$ ซึ่งต่ออยู่กับส่วนวงจรเวลาและควบคุม ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{EA}$ เป็นการเลือกใช้รอมภายใน 8051 โดยที่วงจรเวลาและควบคุมจะสร้างสัญญาณไปยังรอมภายใน ให้ส่งข้อ-มูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาทางเส้นทางหมายเลข 2 ข้อมูลจากรอม ถูกส่งไปยังเส้นทางหมายเลข 3 ที่เรียกว่า เส้นทางข้อมูลภายใน แล้วนำไปเก็บไว้ที่รีจิสเตอร์ IR (Instruction Register) เพื่อส่งไปให้กับวงจรเวลาและควบคุมทำการถอดรหัสแล้วควบคุมการทำ-งานส่วนอื่นๆ ต่อไป ในกรณีที่เลือกรอมภายนอก โดยการป้อนสัญญาณลอจิก 1 เข้าไปที่ขา $\overline{EA}$ จะทำให้วงจรเวลาและควบคุมส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนเส้นทางหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยังเส้นทางข้อมูลภายในแล้วไปเก็บที่รีจิสเตอร์ IR เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจากรอมภายใน การทำงานในช่วงส่งค่าตำแหน่งในหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ในรีจิสเตอร์ IR เรียกว่าช่วงของการเฟตช์ (Fetch) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle

คำสั่งแต่ละคำสั่งของไมโครคอนโทรลเลอร์ เบอร์ 8051 จะใช้เวลาทำงานเพียง 1, 2 หรือ 3 แมชชีนไซเคิล แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 แมชชีนไซเคิล จะใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของไมโครคอนโทรลเลอร์ เบอร์ 8051 จะใช้เวลาการทำงาน 12, 24 หรือ 36 ไซเคิลของสัญญาณนาฬิกา ลำดับการทำงานของไมโครคอนโทรลเลอร์ 8051 ดังแสดงในรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

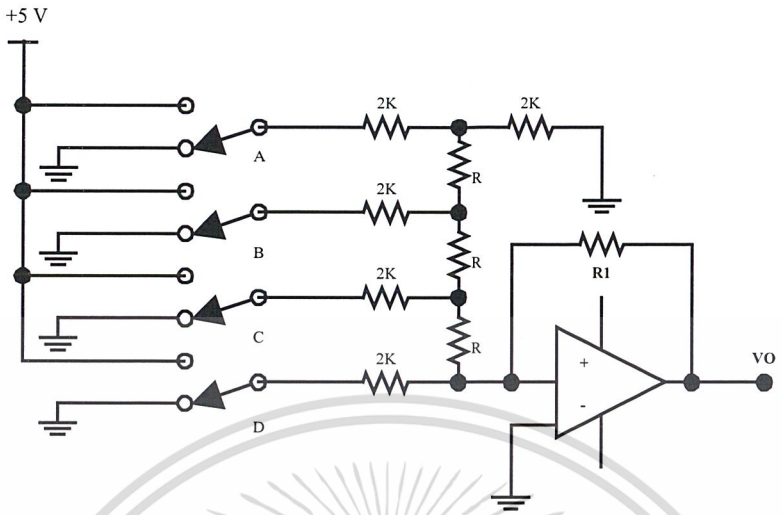


รูปที่ 2.7 ลำดับสถานะการทำงานในไมโครคอนโทรลเลอร์ 8051

2.3 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

การแปลงกลุ่มของสัญญาณดิจิตอล ให้กลายเป็นปริมาณของแรงดันหรือกระแส เราเรียกวิธีการแบบนี้ว่า การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก วงจรพื้นฐานของการแปลงสัญญาณดิจิตอลให้เป็นสัญญาณแอนะล็อกนั้น เราจะใช้ออปแอมป์ เป็นตัวรวบรวมสัญญาณและอาศัยเทคนิคในการต่อวงจรที่อินพุตของออปแอมป์ ซึ่งวงจรพื้นฐานที่จะกล่าวถึงนี้คือ วงจร R-2R Ladder ดังแสดงในรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 วงจร R-2R Ladder

วงจร R-2R Ladder มีสมการของแรงดันเอาต์พุต

Vo = (Vref/16)xD โวลต์

เมื่อ Vref = แรงดันอ้างอิง 2.55 โวลต์

D = เลขฐานสิบที่แปลงมาจากสัญญาณอินพุต

<u>ตัวอย่าง</u> มีอินพุตป้อนเข้าวงจร R-2R เป็น 0101 โดยที่ Vref = 2.55 โวลต์ สามารถคำนวณหาค่า Vo ได้โดย

จากสมการ Vo = (Vref/16)xD

Vref = 2.55 โวลต์

D = 0101 = 5 ในฐานสิบ

Vo = (2.55/16)x 5

= 0.7968 โวลต์

2.3.1 รายละเอียดทางไฟฟ้าของการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

1. รีโซลูชัน

รีโซลูชั่นจะกำหนดโดยจำนวนบิตของสัญญาณดิจิตอล เช่นสัญญาณดิจิตอล 8 บิต จะมีระดับแรงดันเอาต์พุต 256 ระดับที่ไม่ซ้ำกันเลย ($2^8 = 256$) ดังนั้นรีโซลูชั้นจะมีค่าเท่ากับจำนวนที่ต่ำสุดหารด้วยจำนวนสูงสุดก็คือ 1/256 นั้นเอง นิยมบอกในรูปของเปอร์เซ็นต์ เช่น สัญญาณดิจิตอล 12 บิต จะมีเปอร์เซ็นต์รีโซลูชั่นเท่ากับ $1/2^{12} = \frac{1}{4096} \times 100 = 0.024$ เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ความผิดพลาด

เป็นตัวเลขแสดงว่าค่าที่ได้นั้นผิดพลาดไปจากค่าที่ถูกต้องเพียงใดปกติจะระบุเป็นเปอร์เซ็นต์ของค่าสูงสุด ตัวอย่างเช่น วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกมีแรงดันสูงสุด 10 โวลต์ และมีความผิดพลาด ±0.2 เปอร์เซ็นต์ ดังนั้นคิดในรูปแรงดันจะได้ (0.2/100)×10 โวลต์ เท่ากับ 20 มิลลิโวลต์ โดยทั่วไปวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกควรมีความผิดพลาดไม่เกิน ±1/2 ของหลัก LSB ตัวอย่างเช่น วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก 8 บิต หลัก LSB เท่ากับ (1/256)×100 เท่ากับ 0.39 ดังนั้นควรมีค่าความผิดพลาดเท่ากับ ±0.19 เปอร์เซ็นต์ หรือเท่ากับ ±1/2 ของหลัก LSB นั่นเอง ค่าความผิดพลาดที่กล่าวมานี้เรียกว่า Relative Accuracy นอกจากนี้ค่าความผิดพลาดยังมีอีกค่าหนึ่งได้แก่ค่า Differential Linearity ซึ่งเป็นค่าความผิดพลาดที่คิดอยู่ในรูปของสเต็ปไซซ์ หมายความถึงสเต็ปไซซ์ที่ได้นั้นผิดพลาดจากค่าที่เป็นจริงไปเท่าไร เช่น วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกมีขนาดของสเต็ป-ไซซ์ 0.625 โวลต์ มี Differential Linearity ±0.01 เปอร์เซ็นต์ FS (Full Scale 9.375 V) ดังนั้นขนาดของสเต็ปไซซ์จะมีค่าความผิดพลาด 0.9375 มิลลิโวลต์ ในวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกราคาแพงๆ จะมีค่าความผิดพลาดต่ำประมาณ 0.001 เปอร์เซ็นต์ ส่วนพวกที่ใช้งานทั่วๆ ไปจะมีค่าประมาณ 0.01-0.1 เปอร์เซ็นต์

3. ความเร็วในการทำงาน

ความเร็วของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก เรียกว่า เซ็ตติ้งไทม์ (Setting Time) หมายถึงเวลาสูงสุดที่ต้องการในการเปลี่ยนเอาต์พุตจาก “0” ถึงฟูลสเกล เมื่ออินพุตเปลี่ยนจากค่าต่ำสุดมาสูงสุด ปกติจะมีค่าอยู่ระหว่าง 1.20 ไมโครวินาที ถ้าเป็นวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกแบบกระแสเอาต์พุตจะมีเซ็ตติ้งไทม์สั้นกว่าวงจรแปลงสัญญาณดิจิตอลเป็นแอนะล็อกแบบแรงดันเอาต์พุต

2.4 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล (A/D) คือการที่เราเปลี่ยนสัญญาณที่เป็นสัญญาณแอนะล็อกได้แก่ สัญญาณที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง เช่น ระดับของแรงดันไฟฟ้า หรือปริมาณของกระแสไฟฟ้า ให้กลายเป็นสัญญาณดิจิตอลที่อยู่ในรูปของเลขฐานสอง คือ “0” กับ “1” ซึ่งเป็นสัญญาณที่ไม่ขึ้นอยู่กับฟังก์ชันของเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเปลี่ยนสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิตอลนั้นจะต้องคำนึงถึง

1. ความละเอียดของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

ความละเอียดของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล ขึ้นอยู่กับจำนวนบิตของเอาต์พุตยิ่งมีจำนวนบิตของเอาต์พุตมากเท่าไร ความละเอียดของการแปลงสัญญาณของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลก็จะยิ่งมากขึ้นทั้งนี้เพราะว่าแรงดันอินพุตที่ป้อนเข้ามาถ้าหากมีการเปลี่ยนแปลงอยู่ในช่วง +1/2LSB ถึง -1/2LSB จะไม่มีผลอะไรต่อเอาต์พุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลเลย ตัวอย่างเช่น การแปลงแรงดันไฟฟ้าให้เป็นสัญญาณดิจิตอลขนาด 3 บิต เอาต์พุตจะประกอบด้วย D0, D1 และ D2 โดยที่ D0 เป็นบิตที่มีความสำคัญต่ำสุด (Least Significant Bit (LSB)) และบิต D2 เป็นบิตที่มีความสำคัญสูงสุด (Most Significant Bit (MSB)) ดังแสดงในรูปที่ 2.9

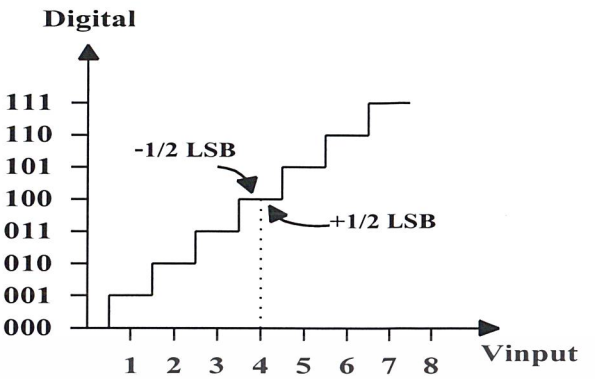
รูปที่ 2.9 ADC ขนาด 3 บิต

คุณลักษณะในทางอุดมคติของการแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิตอล 3 บิต จะให้ 1 LSB เทียบเท่ากับแรงดัน 1 โวลต์ ดังนั้นเมื่อป้อนอินพุตแรงดันเข้ามาเป็น 0, 1 และ 2 โวลต์ เอาต์พุตที่ได้จะเป็น 000, 001 และ 010 ตามลำดับ ถ้าหากแรงดันอินพุตที่ป้อนเข้ามามีค่าเป็นเศษส่วน เช่น มีแรงดันอินพุตเท่ากับ 1.25 โวลต์ เอาต์พุตที่ได้ก็ยังคงเป็น 001 อยู่ ที่เป็นอย่างนี้ก็เพราะว่าแรงดันอินพุตที่ป้อนเข้ามามีค่าอยู่ในช่วง +1/2 LSB ถึง -1/2 LSB จะไม่มีผลต่อเอาต์พุต

ดังนั้นจะเห็นได้ว่าแรงดันที่ป้อนเข้ามาอยู่ในช่วง 0.5-1.5 โวลต์ เอาต์พุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล ก็ยังคงมีค่าเป็น 001 อยู่ตลอด แต่ถ้าหากแรงดันอินพุตมีค่าเท่ากับ 1.6 โวลต์ เอาต์พุตที่ได้จะเป็น 010 ซึ่งความผิดพลาดของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลนี้เราจะต้องคำนึงถึงในการที่จะออกแบบวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32819



รูปที่ 2.10 ความผิดพลาดของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

จากรูปที่ 2.10 แสดงให้เห็นว่าวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลไม่สามารถแยกความแตกต่างได้ถ้าแรงดันอินพุตมีค่าความแตกต่างของแรงดันน้อยกว่า 1 LSB แต่เพื่อที่จะทำให้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแยกความแตกต่างของแรงดันอินพุตได้ดีขึ้น จะต้องทำให้ 1 LSB มีช่วงแรงดันที่น้อยที่สุด โดยการเพิ่มจำนวนบิตของเอาต์พุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลให้มากขึ้น

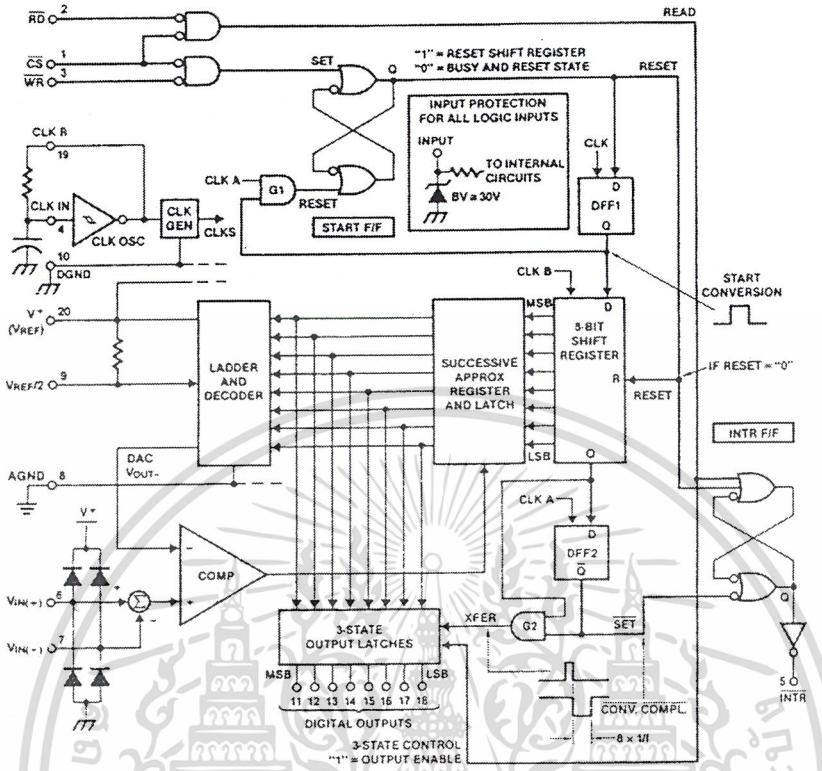
2. เวลาการเปลี่ยนแปลง

การแปลงสัญญาณทางดิจิตอลให้เป็นสัญญาณแอนะล็อกนั้นไม่ได้เกิดขึ้นในทันทีทันใด ต้องมีการผ่านกระบวนการต่างๆ ดังนั้นจึงต้องใช้ระยะเวลาหนึ่งจึงจะได้สัญญาณแอนะล็อกออกมา ซึ่งระยะเวลาที่ถูกใช้ไปนี้เรียกว่า “เวลาการแปลง” (Conversion Time) โดยปกติจะมีค่าเป็นไมโครวินาที ซึ่งค่านี้ดูได้จากข้อมูลของไอซีที่เราใช้ เช่น เบอร์ ADC0804 ที่ใช้ในชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีค่าเวลาการแปลงประมาณ 100 ไมโครวินาที

3. วิธีการแปลงสัญญาณ

ในการแปลงสัญญาณจะมีอยู่ด้วยกันหลายวิธี แต่วิธีที่เป็นที่นิยม มีความเร็วในการแปลงสัญญาณสูงและราคาไม่สูงมากนัก มีความละเอียดพอสมควร เราเรียกวิธีการแปลงสัญญาณแบบนี้ว่า “การแปลงสัญญาณแบบประมาณค่า” ดังแสดงในรูปที่ 2.11

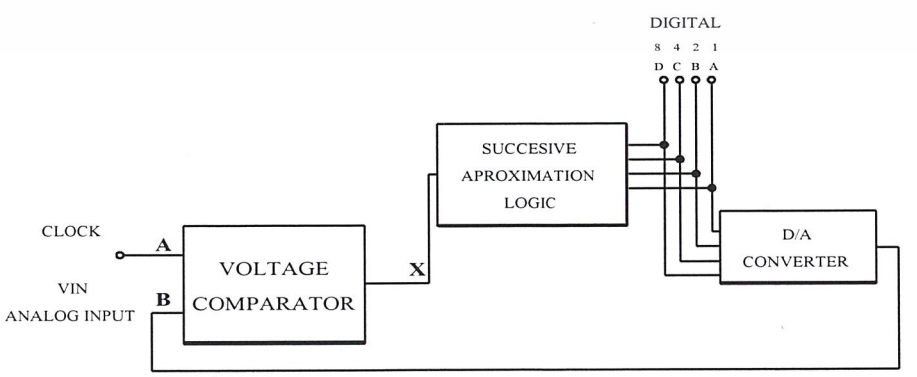
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การแปลงสัญญาณแบบประมาณค่าหลายครั้ง

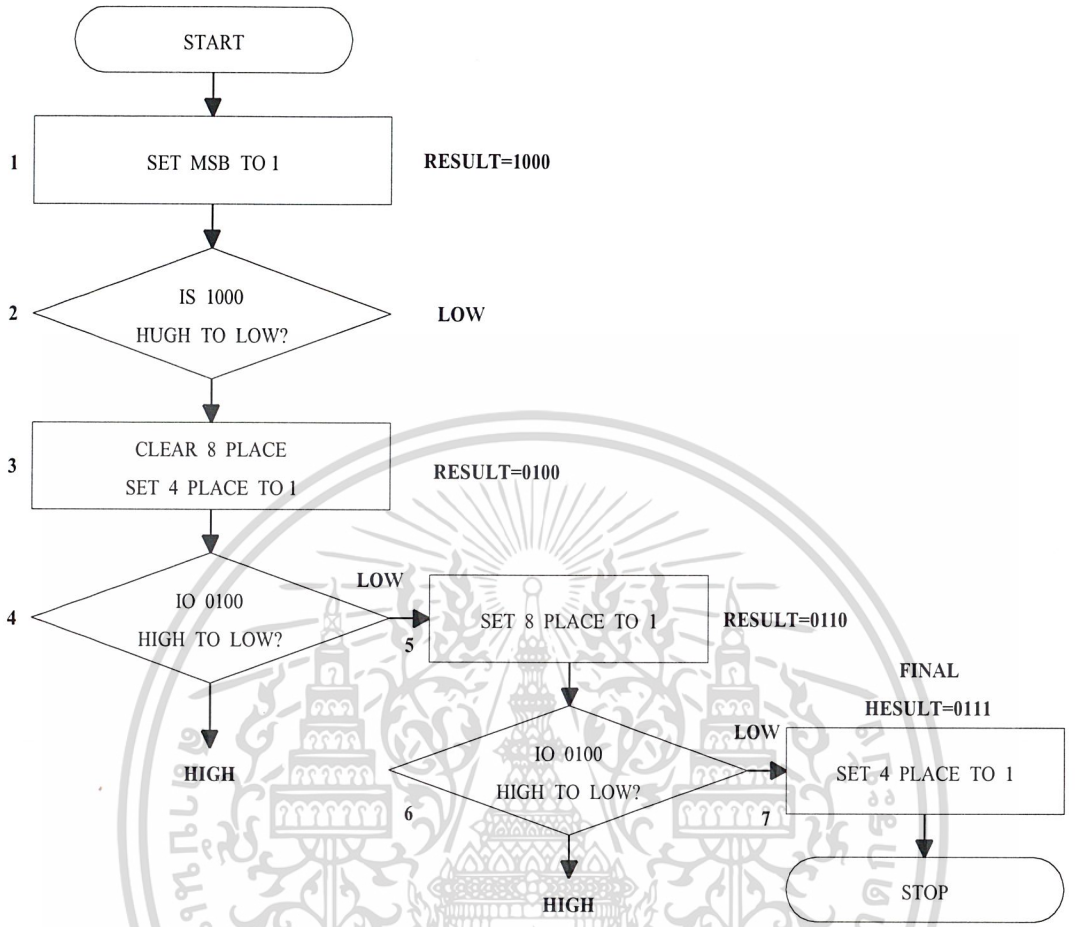
2.4.1 การแปลงสัญญาณแอนะล็อกเป็นดิจิตอลแบบประมาณค่าหลายครั้ง

ในชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะใช้การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแบบประมาณค่าหลายครั้ง โดยจะประกอบด้วยวงจรเปรียบแรงดันดิจิตอลเป็นแอนะล็อกและวงจรการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแบบประมาณค่าหลายครั้ง ดังแสดงในรูปที่ 2.12 และผังงานของโปรแกรม ดังแสดงในรูปที่ 2.13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 แผนผังการทำงานของวงจรการแปลงสัญญาณแบบประมาณค่าหลายครั้ง



รูปที่ 2.13 ผังงานของโปรแกรมแปลงสัญญาณแบบประมาณค่าหลายครั้ง

การทำงานของผังงานโปรแกรมแปลงสัญญาณแบบประมาณค่าหลายครั้ง

จากรูปที่ 2.12 สมมุติว่าแรงดันแอนะล็อกที่อินพุตเท่ากับ 7 โวลต์ วงจร Successive Approximate Logic จะต้องให้บิต MSB เป็นลอจิก “1” ทำให้ได้รหัสไบนารี่ “1000” ป้อนผ่านวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก ไปยังอินพุตของวงจรเปรียบเทียบแรงดัน ซึ่งจะเปรียบเทียบว่าแรงดันเปรียบเทียบที่ป้อนกลับสูงหรือต่ำกว่าแรงดันที่อินพุตถ้าแรงดันเปรียบเทียบสูงกว่าวงจร Successive Approximate Logic ก็จะเคลียร์บิต MSB หรือบิต 2^3 ให้เป็นลอจิก “0” แล้วเซ็ตบิตถัดไปคือบิตที่ 2^2 ให้เป็นลอจิก “1” ผลที่ได้คือรหัส “0100” ซึ่งจะเปลี่ยนเป็นแรงดับป้อนกลับไปเปรียบเทียบกับแรงดันอินพุตอีกคราวนี้ถ้าแรงดันเปรียบเทียบต่ำกว่าแรงดันที่อินพุต วงจรลอจิกก็จะเซ็ตบิตถัดไปคือบิตที่ 2^1 ให้เป็นลอจิก “1” ผลที่ได้คือรหัส “0110” แล้วนำไปเปรียบเทียบใหม่ ถ้าแรงดันเปรียบเทียบยังต่ำกว่าแรงดันอินพุตอีก วงจรลอจิกก็จะเซ็ตบิตถัดไปคือบิต 2^0 ให้เป็นลอจิก “1” ผลที่ได้คือรหัส “0111” เมื่อนำไปเปรียบเทียบที่แรงดันอินพุตปรากฏว่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

วงจรเปรียบเทียบก็จะควบคุมให้วงจร Successive Approximate Logic จะหยุดการทำงานข้อมูลรหัสไบนารี่ที่อยู่ในวงจรลอจิกควบคุมก็คือ “0111” ซึ่งจะเป็นรหัสไบนารี่แทนค่าแรงดันแอนะล็อกที่อินพุต 7 โวลต์นั่นเอง

ข้อดีของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแบบประมาณค่าหลายครั้งนี้ก็คือ จะใช้เวลาในการเปลี่ยนสัญญาณเร็วมาก เพราะไม่ต้องใช้วิธีนับเรียงลำดับไปเรื่อยๆ เหมือนกับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลแบบอื่นๆ

2.4.2 ADC0808

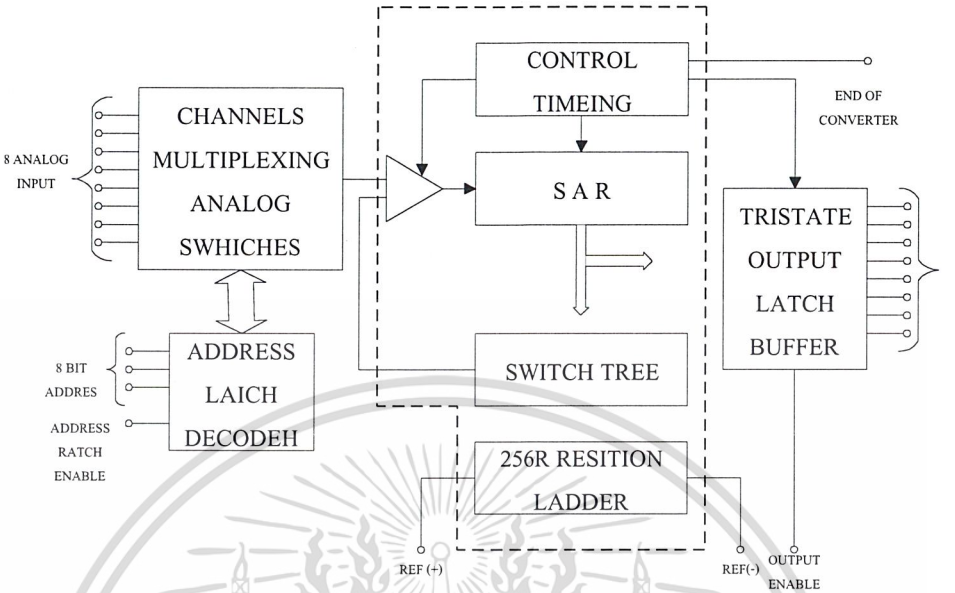
คุณสมบัติของ ADC0808

1. รีโซลูชันขนาด 8 บิต
2. ค่าความผิดพลาดเท่ากับ $\pm \frac{1}{2}$ LSB
3. ไม่มีรหัสความผิดพลาด
4. ความเร็วในการแปลงสัญญาณ 100 ไมโครวินาที
5. แรงดันไฟที่ใช้ 5 VDC
6. รับสัญญาณอินพุตได้ 8 แชนเนล
7. มีเอาต์พุตแบบ Tri-State with Latched Control Logic
8. รับแรงดันแอนะล็อกอินพุต 0-5 VDC
9. กำลังไฟที่ใช้ 15 มิลลิวัตต์
10. ใช้วิธีการแปลงสัญญาณแบบประมาณค่าหลายครั้ง
11. ย่านการเปลี่ยนแปลงอุณหภูมิ -40°C ถึง +85°C

จากรูปที่ 2.14 วงจรรับสัญญาณแอนะล็อกขนาด 8 ช่องแบบมัลติเพลกซ์ (8 Channel Multiplexing Switch) ทำหน้าที่เลือกรับสัญญาณแอนะล็อกอินพุตว่าจะรับเอาสัญญาณแอนะล็อกจากช่องไหนเข้าสู่วงจรส่วนของวงจร 8 ช่องแบบมัลติเพลกซ์นี้ถูกควบคุมการเลือกสัญญาณอินพุตด้วยวงจร Address Latch and Decoder

วงจร Address Latch and Decoder ทำหน้าที่ในการรับข้อมูลแอดเดรสจากภายนอกมาทำการถอดรหัสและสัญญาณที่ถอดรหัสได้นี้จะนำไปควบคุมการเลือกรับสัญญาณอินพุตของวงจร 8 ช่องแบบมัลติเพลกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แผนผังการทำงานของ ADC0808

วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลขนาด 8 บิตประกอบด้วย วงจร 256R Register Addr และวงจร Switch Tree ทำหน้าที่เป็นวงจรสร้างแรงดันอ้างอิง เพื่อนำไปเปรียบเทียบกับสัญญาณแอนะล็อกอินพุต

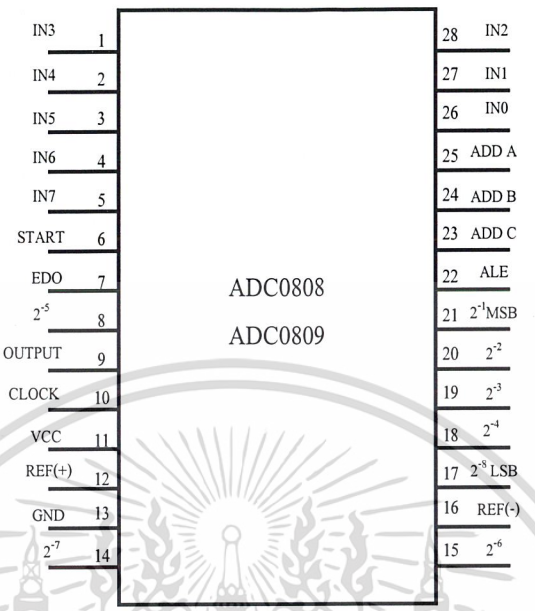
วงจร Comparator ทำหน้าที่เปรียบเทียบแรงดันแอนะล็อกอินพุตกับสัญญาณที่ได้จากวงจร Switch Tree

วงจร Successive Approximate Register (SAR) จะทำหน้าที่แปลงสัญญาณที่ได้จากการเปรียบเทียบให้เป็นสัญญาณดิจิตอลโดยวิธีประมาณค่าหลายครั้ง

วงจร Control & Timing ทำหน้าที่ในการรับสัญญาณควบคุมจากอุปกรณ์ภายนอก และส่งสัญญาณไปควบคุมวงจรภายใน เช่นรับสัญญาณเริ่มต้นที่ขา Start หรือส่งสัญญาณสิ้นสุดการแปลงสัญญาณ (End of Conversion) ไปยังอุปกรณ์ภายนอก

วงจร Tri-State Output Latch/Buffer ทำหน้าที่ค้างข้อมูลดิจิตอลขนาด 8 บิต เพื่อให้วงจรภายนอกรับข้อมูลนี้ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



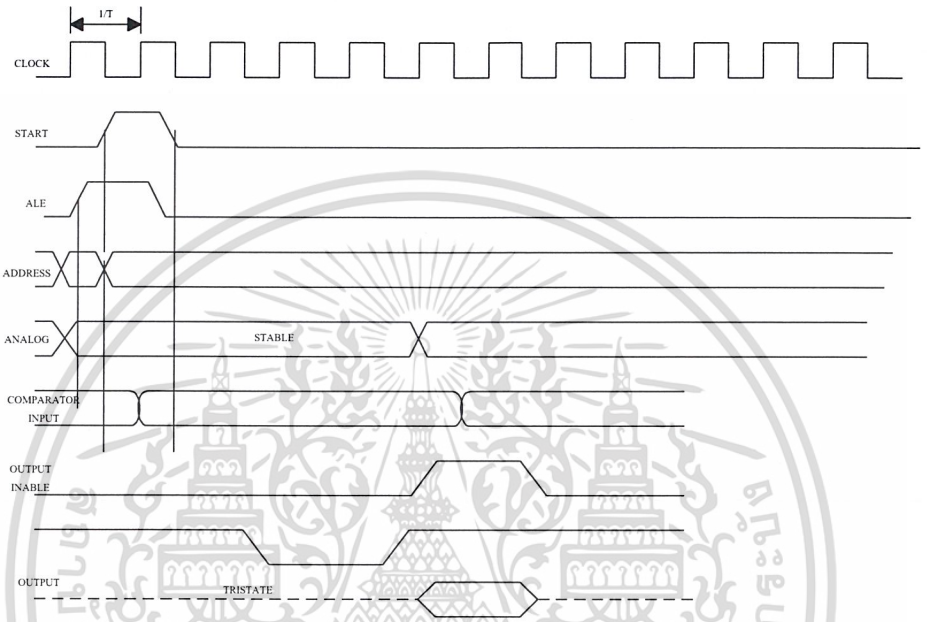
รูปที่ 2.15 การจัดขาของ ADC 0808

จากรูปที่ 2.15 แสดงขาสัญญาณของ ADC0808 ซึ่งแต่ละขาจะมีหน้าที่ดังนี้คือ

IN0-IN7	เป็นขาที่รับสัญญาณแอนะล็อกที่ต้องการแปลงเป็นสัญญาณดิจิตอล
REF (+)	เป็นขาที่ต่อกับแรงดันอ้างอิงบวก
REF (-)	เป็นขาที่ต่อกับแรงดันอ้างอิงลบ
CLK	รับสัญญาณจากสัญญาณนาฬิกาภายนอกเพื่อควบคุมจังหวะการทำงานภายในของ ADC0808
DB0-DB7	เป็นขาข้อมูลดิจิตอลเอาต์พุตที่ได้จากการแปลงสัญญาณแอนะล็อกอินพุต
ADD A-ADD C	เป็นขารับสัญญาณแอดเดรสเพื่อเป็นตัวกำหนดตำแหน่งของการรับสัญญาณอินพุตว่าจะรับสัญญาณอินพุตจากแชนเนลไหน
ALE	แอคทีฟที่โลจิก “1” เป็นขาที่รับสัญญาณเพื่อทำการค้างข้อมูลแอดเดรสที่เข้ามาแทนที่ ADD A-ADD C ให้ค้างไว้ที่ (Latch) วงจร Address Latch Decoder ภายใน ADC0808
START	แอคทีฟที่โลจิก “1” เป็นสัญญาณกำหนดการเริ่มต้นการแปลง สัญญาณ
EOC	แปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล นั่นคือถ้าที่ขา EOC เป็นลอจิก “0” แสดง ว่าสัญญาณแอนะล็อกถูกแปลงเป็นสัญญาณดิจิตอลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

OE แอคทีฟที่โลจิก “1” เป็นขาอินพุตเพื่อรับสัญญาณการอ่านข้อมูลจากอุปกรณ์ภายนอก



รูปที่ 2.16 แผนผังเวลาของ ADC 0808

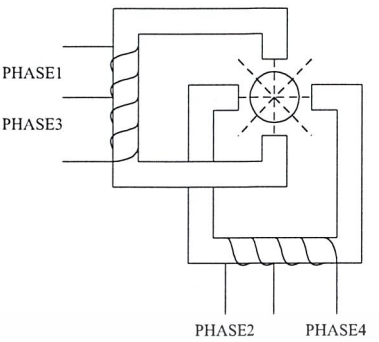
2.5 สเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่งซึ่งสามารถควบคุมได้ด้วยไมโครคอนโทรลเลอร์ ลักษณะการทำงานของสเต็ปปิ้งมอเตอร์จะเคลื่อนที่เป็นสเต็ปซึ่งอาจเป็นสเต็ปละ 1, 8, 5, 7.5 องศา แล้วแต่ชนิดของมอเตอร์ ส่วนใหญ่สเต็ปปิ้งมอเตอร์จะใช้ในงานควบคุมระบบดิจิตอล อุปกรณ์ในงานอิเล็กทรอนิกส์อุตสาหกรรม หรือเครื่องมือวัดและระบบควบคุมอื่นๆ

สเต็ปปิ้งมอเตอร์จะประกอบด้วยส่วนที่สำคัญ 2 ส่วน คือ

1. โรเตอร์ (Rotor) เป็นส่วนที่หมุนได้ จะเป็นแม่เหล็กถาวรและอื่นๆ
2. สเตเตอร์ (Stator) เป็นส่วนที่อยู่กับที่ จะเป็นขดลวดหลายๆ ขด ดังแสดงในรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 สเต็ปปิ้งมอเตอร์ 4 เฟส แบบยูนิโพลาร์เพอร์มาเนนแม็กเนต

ข้อดีของสเต็ปปิ้งมอเตอร์เมื่อเปรียบเทียบกับดีซีมอเตอร์

1. การควบคุมสามารถที่จะควบคุมแบบลูปเปิด ได้โดยไม่ต้องอาศัยตัวตรวจจับการหมุน
2. ไม่ต้องใช้แปรงถ่าน ทำให้ไม่มีส่วนที่สึกหรอ และปัญหาเรื่องการสปาร์คที่ทำให้เกิดสัญญาณรบกวน
3. การควบคุมโดยไมโครคอนโทรลเลอร์ทำได้ง่าย

สเต็ปปิ้งมอเตอร์แบ่งออกเป็น 3 ชนิด คือ

1. วาริเอเบิลรีลักแตนซ์ (VR) เป็นการหมุนได้อิสระในขณะที่ไม่ได้จ่ายไฟให้โรเตอร์จะทำด้วยเหล็กอ่อน มีลักษณะเป็นฟันเลื่อยรูปทรงกระบอก จะมีมุมสเต็ปขนาด 7.5 องศา หรือ 15 องศา
2. เพอร์มาเนนแม็กเนต (PM) ตัวโรเตอร์จะทำด้วยแม่เหล็กถาวรเป็นรูปทรงกระบอกฟันเลื่อยที่ตัวสเตเตอร์จะมีขดลวดพันอยู่ตัว โรเตอร์จะหมุนไม่เป็นอิสระ ในขณะที่ไม่ได้ป้อนไฟเพราะแรงดูดจากแม่เหล็กถาวร
3. ไฮบริดก์ เป็นสเต็ปปิ้งมอเตอร์ที่นำเอาข้อดีของแบบ VR และ PM เข้ามารวมกันไว้จะมีตัวสเตเตอร์คล้ายกับแบบ VR ที่โรเตอร์จะมีแม่เหล็กกำลังสูงติดอยู่ ทำให้แรงบิดสูงมีความแม่นยำในการหมุนสูงและมีขนาดกระทัดรัด ขนาดของมุมสเต็ปจะเป็น 0.9 องศา และ 1.8 องศา

2.5.1 การกระตุ้นและการควบคุมการหมุนของสเต็ปปิ้งมอเตอร์

การทำให้สเต็ปปิ้งมอเตอร์เคลื่อนที่ไปทีละสเต็ป ทำได้โดยการจ่ายไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ซึ่งจะต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้องการป้อนพัลส์กระตุ้นสเต็ปปิ้งมอเตอร์สามารถทำได้ 3 รูปแบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แบบเวฟ (Wave) เป็นการป้อนกระแสให้กับขดลวดแต่ละขดของ สเต็ปปิ้งมอเตอร์ที่ละขดเรียงลำดับกันได้ ลักษณะการขับแบบนี้จะทำให้แรงบิดน้อย ดังแสดงในตารางที่ 2.2
2. แบบ 2 เฟส (Two phase) มีลักษณะคล้ายกับแบบเวฟ แต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดย จ่ายกำลังไฟไปที่ขดลวดโดย 2 ขดลวดที่อยู่ใกล้กันในเวลาเดียวกัน เรียงถัดกันไป เช่นเดียวกับแบบเวฟขึ้นอยู่กับทิศทางของการหมุน การเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นจะทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน ข้อเสียของการกระตุ้นแบบนี้ คือ การกระตุ้นแบบนี้จะต้องจ่ายกำลังไฟฟ้ามากขึ้น การทำงานต่างๆ จะแสดงให้เห็นในตารางที่ 2.3

ตารางที่ 2.2 การจ่ายกระแสแบบเวฟให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์

STEP	PHASE			
	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

ตารางที่ 2.3 การจ่ายกระแสแบบ 2 เฟส ให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์

STEP	PHASE			
	1	2	3	4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แบบครึ่งสเต็ป (Half step) เป็นแบบที่ได้จากการผสมระหว่างการกระตุ้นแบบเวฟและแบบ 2 เฟส ดังแสดงในตารางที่ 2.4 เพื่อเพิ่มจำนวนสเต็ปต่อรอบอีกหนึ่งเท่าตัว แรงบิดได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง และแต่ละสเต็ปเกิดจากแรงดึงของขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกันความถูกต้องของตำแหน่งจึงเพิ่มมากขึ้น ที่สำคัญการกระตุ้นแบบนี้จะต้องทำการหมุน 2 สเต็ปจึงจะเท่ากับ 1 สเต็ปของ 2 แบบแรกส่วนการจ่ายกำลังไฟฟ้าต้องใช้เหมือนกับแบบ 2 เฟส

ตารางที่ 2.4 การจ่ายกระแสแบบครึ่งสเต็ป ให้กับขดลวดแต่ละขดของสเต็ปปิ้งมอเตอร์

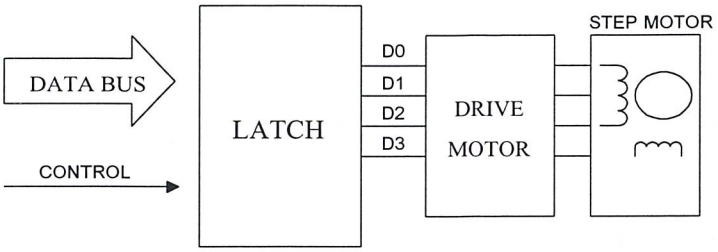
STEP	PHASE			
	1	2	3	4
1	1	1	0	0
2	0	1	0	0
3	0	1	1	0
4	0	0	1	0
5	0	0	1	1
6	0	0	0	1
7	1	0	0	1
8	1	0	0	0

2.5.2 การเชื่อมต่อ ไมโครคอนโทรลเลอร์ 8051 กับสเต็ปปิ้งมอเตอร์

โดยทั่วไปรูปแบบของการเชื่อมต่อกับสเต็ปปิ้งมอเตอร์จะมีรูปแบบ ดังผังการทำงานในรูปที่ 2.18 ซึ่งสามารถที่จะควบคุมได้ทั้งทิศทางและตำแหน่ง

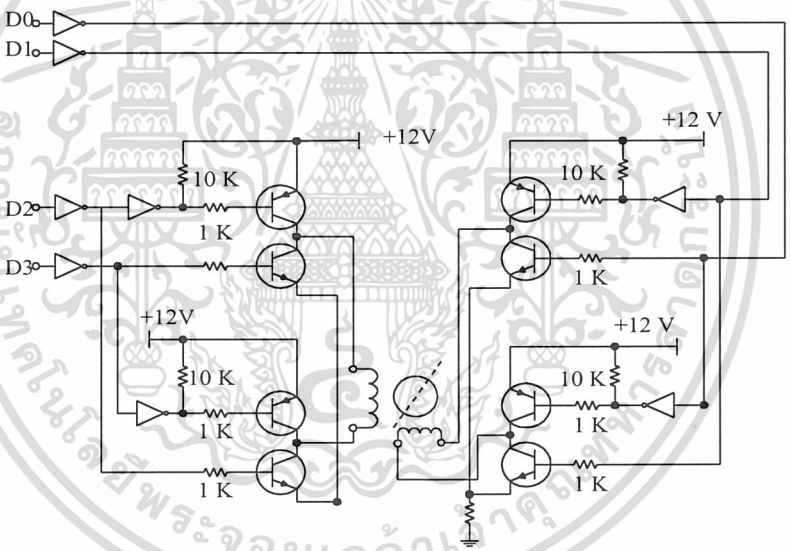
จากรูปที่ 2.18 จะเห็นว่าการเชื่อมต่อกับสเต็ปปิ้งมอเตอร์จะเหมือนกับการเชื่อมต่อกับอุปกรณ์เอาต์พุตของไมโครคอนโทรลเลอร์ 8051 ทั่วไป โดยสัญญาณที่ใช้ในการเลือกเอาต์พุตนั้นเองส่วนชุดขับมอเตอร์จะใช้สำหรับจ่ายกระแสให้กับตัวสเต็ปปิ้งมอเตอร์ ซึ่งชุดขับมอเตอร์นี้จะขึ้นอยู่กับชนิดของการพันขดลวด (แบบไบโพลาร์หรือยูนิโพลาร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

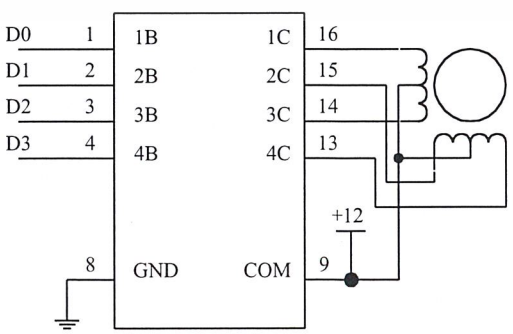


รูปที่ 2.18 แผนผังการทำงานของการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 กับสเต็ปปิ้งมอเตอร์

ชุดสำหรับขับมอเตอร์สามารถต่อวงจรได้หลายแบบขึ้นอยู่กับความสามารถของผู้ออกแบบ สำหรับในรูปที่ 2.19 จะเป็นตัวอย่างของการขับมอเตอร์แบบไบโพลาร์ ส่วนในรูปที่ 2.20 จะเป็นตัวอย่างของการขับมอเตอร์แบบยูนิโลาร์



รูปที่ 2.19 ตัวอย่างการขับมอเตอร์แบบไบโพลาร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.20 ตัวอย่างการขับมอเตอร์แบบยูนิโพลาร์โดยใช้ไอซีเบอร์ ULN2003A เป็นตัวขับกระแส

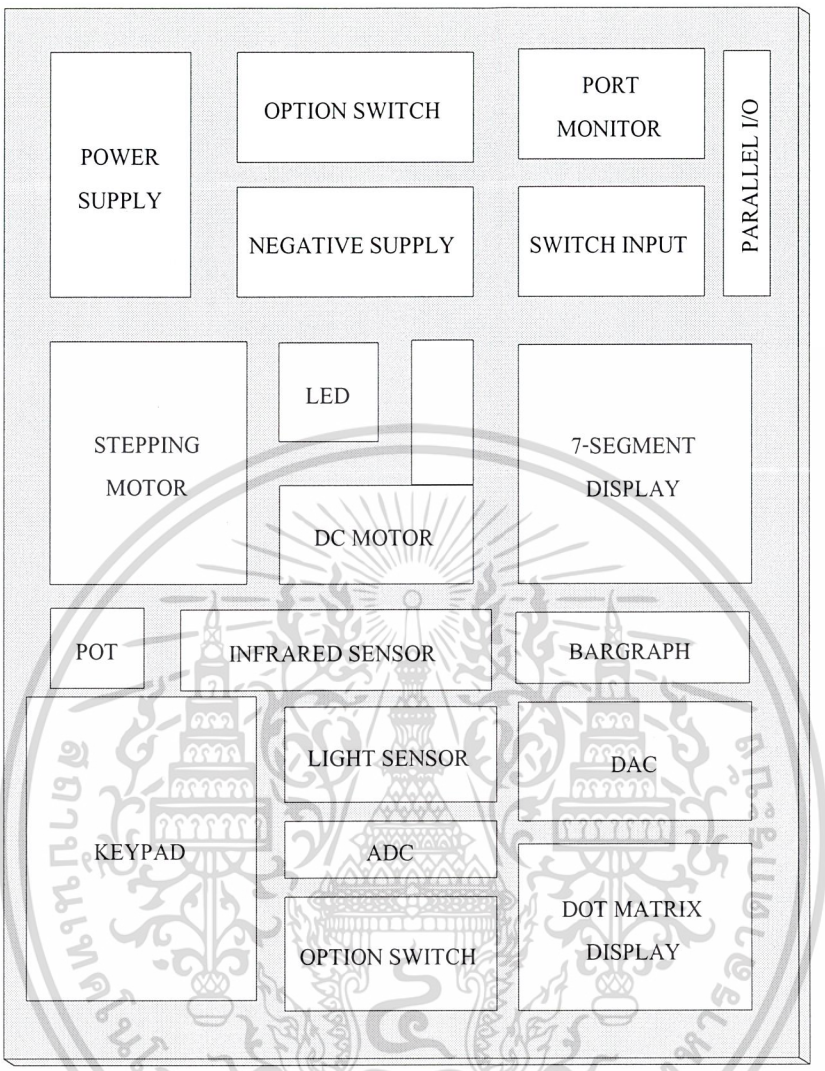
บทที่ 3
การออกแบบและการสร้าง

3.1 กล่าวนำ

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะประกอบหลายส่วนด้วยกันดังแสดงในรูปที่ 3.1 ซึ่งในการออกแบบจะต้องเริ่มทำการออกแบบทีละส่วน โดยส่วนประกอบหลักในชุดทดลองนั้นจะประกอบด้วย

1. ชุดแสดงผลข้อมูลขนาด 8 บิต
2. ชุดแสดงผลแบบจุดขนาด 5×7
3. ชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก
4. ชุดแสดงระดับสัญญาณ
5. ชุดปรับแรงดัน 0-2.55 โวลต์
6. ชุดแสดงผลตัวเลข 7 ส่วน
7. ชุดสร้างสัญญาณแสดงผลด้วยไดโอดเปล่งแสง
8. ชุดสร้างสัญญาณเสียงด้วยบัซเซอร์
9. ชุดควบคุมมอเตอร์กระแสตรง
10. ชุดสร้างสภาวะลอจิกขนาด 8 บิต
11. ชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล
12. ชุดตรวจวัดความสว่างของแสง
13. ชุดตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด
14. ชุดควบคุมสเต็ปปิ้งมอเตอร์
15. ชุดคีย์แพดขนาด 4 x 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ส่วนประกอบของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

3.2 การออกแบบและการสร้าง

ในการออกแบบและสร้างชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 สามารถแบ่งออกได้เป็น 2 ส่วน ดังนี้

1. ด้านฮาร์ดแวร์
2. ด้านซอฟต์แวร์

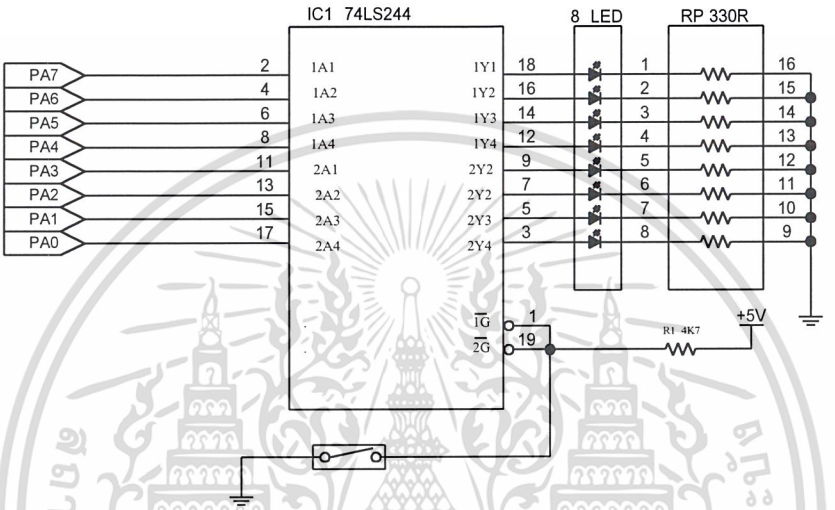
รายละเอียดในการสร้างวงจรของแต่ละส่วนนั้นจะกล่าวในส่วนของเนื้อหาในลำดับต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบด้านฮาร์ดแวร์

3.3.1 ชุดแสดงผลข้อมูลขนาด 8 บิต

หลอดไดโอดเปล่งแสง (Light Emiting Diode) จะทำหน้าที่เป็นตัวแสดงผลข้อมูลในแต่ละบิตของพอร์ตเอาต์พุตจะประกอบด้วยหลอดไดโอดเปล่งแสง จำนวน 8 หลอด



รูปที่ 3.2 วงจรแสดงผลข้อมูลขนาด 8 บิต

จากรูปที่ 3.2 ไอซี 74LS240 จะต่อเป็นบัฟเฟอร์ระหว่างพอร์ตของ 8255 กับหลอดไดโอดเปล่งแสง เมื่อทำให้ 74LS240 อยู่ในสภาวะ Enable หลอดไดโอดเปล่งแสง จะติดสว่างเมื่อมีอินพุตลอจิกเป็น “1” เอาต์พุตของ 74LS240 จะเป็น “0” ทำให้เกิดความต่างศักย์ของแรงดันระหว่างแหล่งจ่าย +5 โวลต์ กับเอาต์พุตของ 74LS240 เป็นผลให้มีกระแสจำนวนหนึ่งไหลผ่านตัวต้านทาน 330 โอห์ม ทำให้หลอดไดโอดเปล่งแสงติดสว่าง

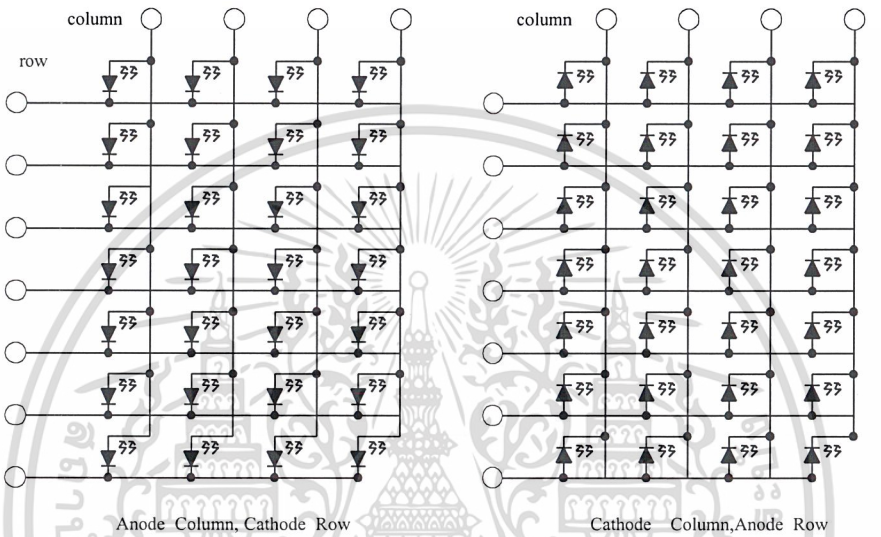
ถ้าอินพุตมีลอจิกเป็น “0” เอาต์พุตของ 74LS04 ก็จะมีลอจิกเป็น “1” ทำให้ไม่เกิดความต่างศักย์ขึ้นระหว่างแหล่งจ่าย +5 โวลต์ กับเอาต์พุตของ 74LS04 ทำให้ไม่เกิดกระแสไหลผ่านตัวต้านทานมายังหลอดไดโอดเปล่งแสง หลอดไดโอดเปล่งแสงจึงไม่ติดสว่าง

3.3.2 ชุดแสดงผลแบบจุดขนาด 5×7

หลอดไดโอดเปล่งแสง ที่เห็นอยู่ทั่วไปจะมีลักษณะเป็นหลอดเดี่ยวๆ จะใช้เป็นตัวแสดงผลแบบติด-ดับ (ON-OFF) แต่ถ้านำเอาหลอดไดโอดเปล่งแสงเหล่านี้มารวมกันให้เป็นกลุ่มแถวในแนวนอนและแนวตั้ง รูปแบบของการแสดงผลก็จะเปลี่ยนไป การติด-ดับของหลอดไดโอดเปล่งแสงสามารถที่จะสร้างให้เกิดเป็นข้อความหรือรูปลักษณ์ต่างๆ ตามที่เราต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

กลุ่มของหลอดไดโอดเปล่งแสง ที่ว่านี้มีชื่อเรียกว่า ชุดแสดงผลแบบจุดขนาด 5×7 (Dot Matrix Display) ซึ่งจะมีอยู่ด้วยกันหลายแบบ หลายขนาดเช่น 4×4, 5×7, 5×8, 7×5, 8×8, 16×16 มีทั้งแบบสีเดียวหรือแบบหลายสี และยังแบ่งตามการใช้งานเป็นแบบแอโนดที่หลัก, แคโทดที่แถว และแคโทดที่หลัก, แอโนดที่แถว

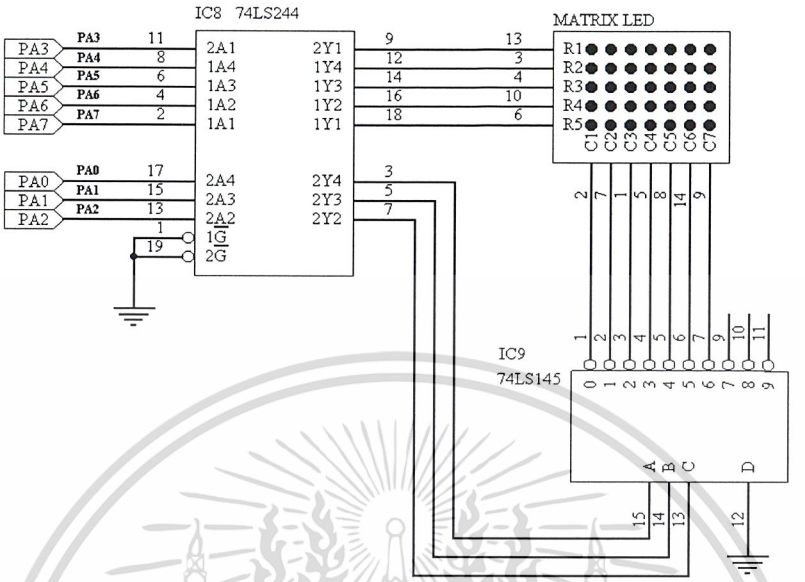


รูปที่ 3.3 ชุดแสดงผลแบบจุดขนาด 5×7 (Dot Matrix Display)

ในชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะใช้ชุดแสดงผลแบบจุดขนาด 5×7 (Dot Matrix Display) แบบแคโทดที่หลัก, แอโนดที่แถว โดยที่ชุดแสดงผลแบบจุดนี้จะต่อเข้ากับพอร์ต PA โดยบิต 0-2 เป็นบิตกำหนดคอลัมน์ บิต 3-7 เป็นบิตกำหนดแถว

จากวงจรในรูปที่ 3.4 บิต 0-7 จะต่อผ่านบัฟเฟอร์ (74LS244) แยกบิต 0-2 เข้า 74LS145 ซึ่งเป็นไอซี BCD to Decimal Decoder (4 to 10) แต่จะป้อนอินพุตให้ที่ขา A, B และ C ส่วนขา D จะต่อลงกราวด์เอาไว้ทำให้ 74LS145 นับแค่ 0-6 เท่านั้น 74LS145 จะทำหน้าที่เหมือนสวิตช์คอยต่อให้คอลัมน์ของชุดแสดงผลแบบจุดลงกราวด์ทีละคอลัมน์ตามอินพุตที่ป้อนเข้ามา ดังในตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



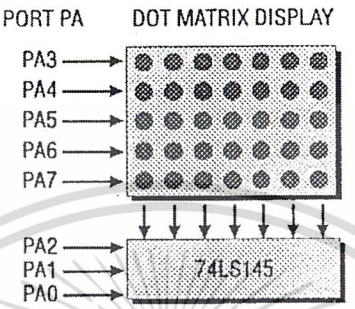
รูปที่ 3.4 วงจรชุดแสดงผลแบบจุดขนาด 5×7 (Dot Matrix Display)

ตารางที่ 3.1 การกำหนดคอลัมน์ในการแสดงผลของชุดแสดงผลแบบจุด

อินพุตของ 74LS145			Column ที่ถูกต่อลงกราวด์								CODE
C	B	A									
PA2	PA1	PA0	C1	C2	C3	C4	C5	C6	C7	C8	
0	0	0	⊗								00
0	0	1		⊗							01
0	1	0			⊗						02
0	1	1				⊗					03
1	0	0					⊗				04
1	0	1						⊗			05
1	1	0							⊗		06
1	1	1								⊗	07

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนบิต 3-7 เป็นบิตข้อมูลของแถวเมื่อต้องการให้แถวใดติดก็กำหนดให้บิตของแถวนั้นเป็น “1” นำรหัสที่ได้มารวมกัน เพื่อใช้เป็นข้อมูลที่จะส่งออกมาทางพอร์ต PA การแสดงผลแสดงในรูปที่ 3.5



รูปที่ 3.5 การต่อพอร์ตกับชุดแสดงผลแบบจุดขนาด 5×7 (Dot Matrix Display)

เช่นต้องการให้คอลัมน์ 5, แถว 1-5 ติดสว่าง

คอลัมน์ 5 ที่จะต้องถูกต่อลงกราวด์ รหัส = 04H

แถว 1-5 ติดหมด รหัส = F8H (1 1 1 1 1 X X X)

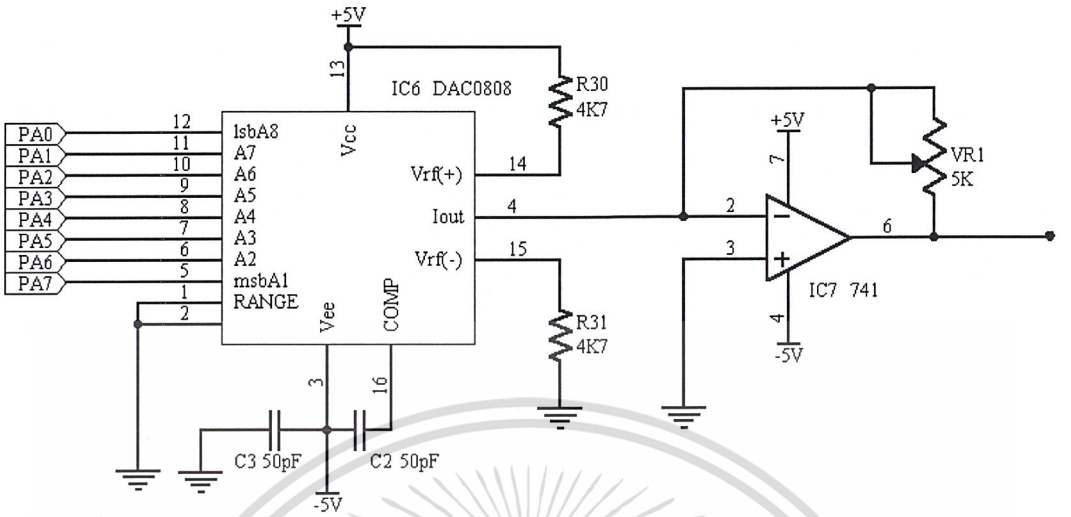
ข้อมูลที่จะถูกส่งออกมาที่พอร์ต PA คือ 04H + F8H = FCH

วิธีนี้จะเห็นได้ว่าเราสามารถที่จะสั่งให้ไดโอดเปล่งแสง หลอดใดหลอดหนึ่งในจำนวน 35 หลอด ที่มีอยู่ติดสว่างได้ตามต้องการ แต่ไม่สามารถที่จะสั่งให้ไดโอดเปล่งแสง ทุกหลอดติดสว่างพร้อมๆ กันได้จะต้องอาศัยเทคนิคในการสั่งให้ติดทีละคอลัมน์ต่อเนื่องกันไปด้วยความเร็วที่มากกว่า 24 ครั้งต่อวินาที จะทำให้ดูเหมือนว่าหลอดไดโอดเปล่งแสง ทุกหลอดติดสว่างพร้อมกันหมด

3.3.3 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีไอซี DAC0808 เป็นตัวแปลงสัญญาณดิจิตอล 8 บิต ให้เป็นกระแส แล้วใช้ออปแอมป์ 741 แปลงกระแสที่ได้ให้เป็นแรงดันให้อยู่ในช่วง 0-2.55 VDC ซึ่งวงจรแสดงดังในรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

จากวงจรสามารถที่จะคำนวณหาค่าแรงดัน Vo ได้โดยอาศัยสมการ

เมื่อ Vref = แรงดันอ้างอิง (กำหนดให้เท่ากับ 5 โวลต์)

Rf = ตัวต้านทานที่ต่อระหว่างขา 13 และ 14 ของ DAC0808

Ro = ตัวต้านทานป้อนกลับของออปแอมป์

Do = ลอจิกของสัญญาณดิจิตอล

<u>ตัวอย่าง</u> ป้อนสัญญาณดิจิตอลให้กับวงจรเป็น 10101010 จงคำนวณหาค่าแรงดัน Vo

จากสมการ $V = Vref \times Ro(D7/2+D6/4+D5/8+D4/16+D3/32+D2/64+D1/128+D0/256)$

Vref = 5 โวลต์

Rf = 4.7 กิโลโอห์ม

Ro = 2.35 กิโลโอห์ม

$Vo = (5/4.7)\times 2.35\times(1/2+0/4+1/8+0/16+1/32+0/64+1/128+0/256)$

$= 2.55\times(0.5+0.125+0.03125+0.0078125)$

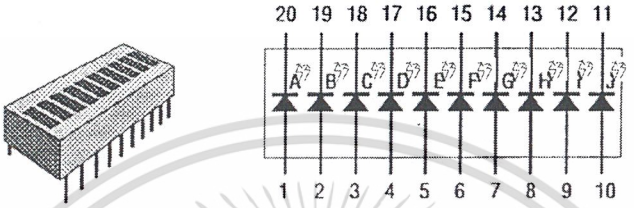
$= 2.55\times 0.664$

$\therefore \quad Vo = 1.6932$ โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 ชุดแสดงระดับสัญญาณ

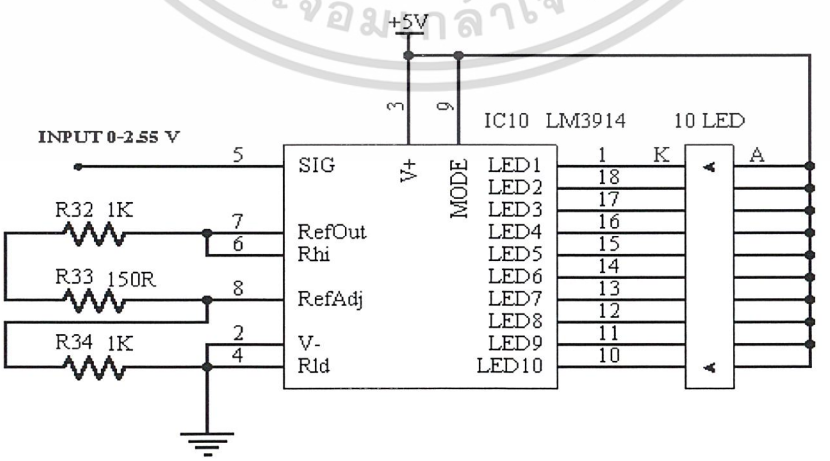
ชุดแสดงระดับสัญญาณจัดเป็นตัวแสดงผลอีกแบบหนึ่ง ที่เกิดจากการนำเอาหลอดไดโอดเปล่งแสงสี่เหลี่ยมมาเรียงติดกันเป็นแถว มีตั้งแต่ 2 หลอด ไปจนถึง 10 หลอด แล้วถูกยึดเป็นชิ้นเดียวกัน แต่ละหลอดจะต่อขาออกมีทั้งแบบแถวเดี่ยวและแถวคู่ ดังรูปที่ 3.7



รูปที่ 3.7 โครงสร้างของชุดแสดงระดับสัญญาณ

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 นำเอาชุดแสดงระดับดันมาเป็นหน่วยแสดงผลโดยต่อเข้ากับไอซี LM3914 ซึ่งเป็นตัวขับการแสดงผลแบบจุดและแบบแท่ง ภายในจะประกอบไปด้วย ออปแอมป์จำนวน 10 ตัว ทำหน้าที่เป็นตัวเปรียบเทียบระหว่างแรงดันอินพุตกับแรงดันอ้างอิงที่กำหนด ถ้าแรงดันอินพุตมีค่ามากกว่าแรงดันอ้างอิงของออปแอมป์แต่ละตัว เอาต์พุตของออปแอมป์แต่ละตัวจะเป็นศูนย์ ทำให้หลอดไดโอดเปล่งแสงที่ต่ออยู่ติดสว่าง อินพุตของ LM3914 จะรับมาจาก POT หรือวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก ปรับที่ SW 1/1

แรงดันอินพุตจะอยู่ในช่วง 0-2.55 โวลต์ LM3914 สามารถแสดงผลได้ 10 ช่อง ฉะนั้นแรงดันอินพุตของวงจรชุดแสดงระดับดันจะมีค่าเท่ากับ 0.255 V/Bar ดังในรูปที่ 3.8



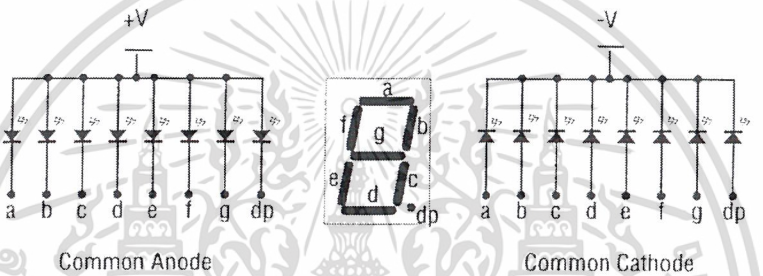
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 วงจรแสดงระดับแรงดัน

3.3.5 ชุดแสดงผลตัวเลข 7 ส่วน

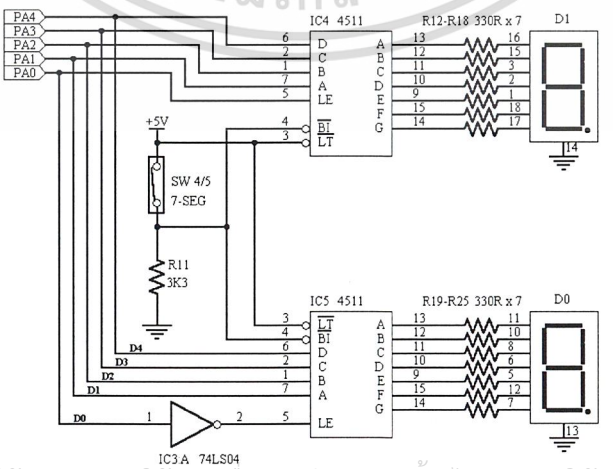
วงจรแสดงผลตัวเลข 7 ส่วน เป็นตัวแสดงผลอีกแบบหนึ่งที่แสดงผลออกมาเป็นตัวเลข ภายในจะประกอบไปด้วยหลอดไดโอดเปล่งแสง จำนวน 8 ดวงเป็นเซกเมนต์ a, b, c, d, e, f, g และเป็นจุดทศนิยม (dp) ไดโอดเปล่งแสงจะต่อเข้ากันอยู่หนึ่งขา ซึ่งอาจจะเป็นขาแคโทด (Cathode) หรือขาแอโนด(Anode) ดังแสดงในรูปที่ 3.9 แบ่งวงจรแสดงผลตัวเลข 7 ส่วน ออกเป็นประเภทตามลักษณะการต่อขาภายในร่วมกันเป็น

1. แบบแคโทดร่วม (Common Cathode)
2. แบบแอโนดร่วม (Common Anode)



รูปที่ 3.9 การต่อขาวงจรแสดงผลตัวเลข 7 ส่วน

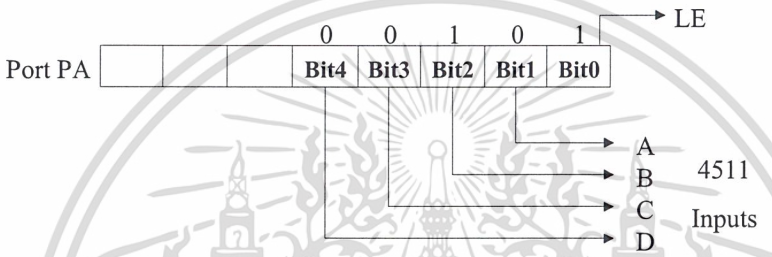
การแสดงผลของวงจรแสดงผลตัวเลข 7 ส่วน จำเป็นต้องอาศัยวงจรถอดรหัส (Decode) และวงจรขับเพื่อรับอินพุตเป็น BCD มาแสดงผลเป็นตัวเลขฐานสิบ ในปัจจุบันจะทำเป็นไอซีสำเร็จรูปอย่าง เช่น 7447, 4511 ซึ่งจะเป็น BCD to 7-Segment ชนิด 4 อินพุต จะเห็นได้ว่าการควบคุมวงจรแสดงผลตัวเลข 7 ส่วน แต่ละหลักจะต้องใช้สัญญาณอินพุตจำนวน 4 บิต ดังนั้นถ้าหากต้องการแสดงผลหลายๆ หลักก็จะต้องใช้จำนวนบิตของอินพุตมากขึ้นตามไปด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 วงจรแสดงผลตัวเลข 7 ส่วน

รูปที่ 3.10 แสดงวงจรแสดงผลตัวเลข 7 ส่วน ของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีอยู่ 2 หลักคือ D0 และ D1 โดยทั้ง 2 หลักจะใช้สัญญาณอินพุตรวมกันแค่ 5 บิต ต่อเข้ากับ 4511 จำนวน 2 ตัว ตารางที่ 3.2 แสดงการทำงานของ 4511 วงจรแสดงผลตัวเลข 7 ส่วน ที่ใช้จะเป็นแบบแคโถดร่วมจากวงจรของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะเห็นว่าบิต 0 จะต่อเข้ากับขา LE ทำหน้าที่ควบคุมว่าจะให้หลัก D0 หรือ D1 ทำงาน ขา LE ของทั้งสองหลักจะมีสภาวะที่ตรงข้ามกันตลอด ถ้าหลักใดที่ขา LE เป็น “0” หลักนั้นก็จะทำงานดังรูปที่ 3.11



รูปที่ 3.11 บิตที่ใช้กับวงจรแสดงผลตัวเลข 7 ส่วน

ตารางที่ 3.2 การทำงานของ 4511

INPUTS							OUTPUTS							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	DISPLAY
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (ต่อ) การทำงานของ 4511

INPUTS							OUTPUTS							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X								

เช่น อินพุตของ 4511 เป็น 00011 บิต 0 เป็น “1” แสดงว่าหลัก D0 จะทำงานแสดงผลออกมาเป็นตัวเลข 1 ที่ วงจรแสดงผลตัวเลข 7 ส่วน

ตารางที่ 3.3 อินพุตของ D0 และ D1

Bit4	Bit3	Bit2	Bit1	Bit0	CODE	7-Segment D1	7-Segment D0
0	0	0	0	0	00H	-	0
0	0	0	0	1	01H	0	-
0	0	0	1	0	02H	-	1
0	0	0	1	1	03H	1	-
0	0	1	0	0	04H	-	2
0	0	1	0	1	05H	2	-
0	0	1	1	0	06H	-	3
0	0	1	1	1	07H	3	-
0	1	0	0	0	08H	-	4
0	1	0	0	1	09H	4	-
0	1	0	1	0	0AH	-	5
0	1	0	1	1	0BH	5	-
0	1	1	0	0	0CH	-	6
0	1	1	0	1	0DH	6	-
0	1	1	1	0	0EH	-	7
0	1	1	1	1	0FH	7	-
1	0	0	0	0	10H	-	8
1	0	0	0	1	11H	8	-
1	0	0	1	0	12H	-	9
1	0	0	1	1	13H	9	-

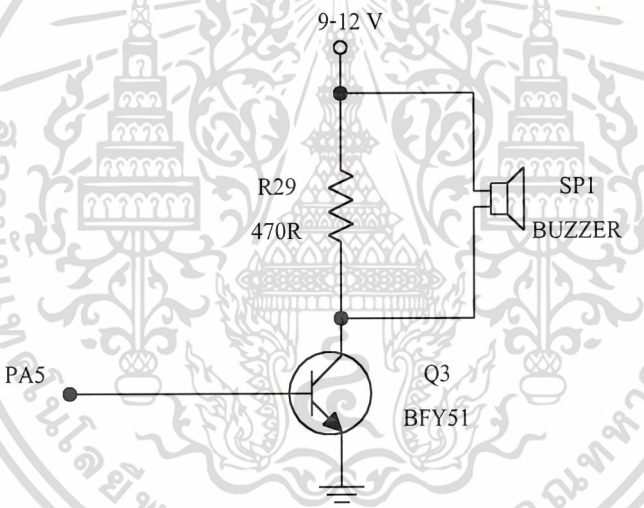
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6 วงจรสร้างสัญญาณเสียงด้วยบัซเซอร์

บัซเซอร์เป็นอุปกรณ์ที่สามารถสร้างสัญญาณเสียงออกมา บัซเซอร์ที่ใช้อยู่ทั่วไปแบ่งออกเป็น 2 ชนิดคือ

1. ชนิด A เป็นแบบที่ทำงานอิสระ เมื่อป้อนแรงดันไฟฟ้าให้กับตัวมัน ก็จะสามารถสร้างสัญญาณเสียงออกมาได้ สัญญาณที่ออกมาไม่สามารถเปลี่ยนแปลงได้

2. ชนิด B เป็นแบบที่ต้องสร้างสัญญาณพัลส์ ที่มีความถี่แตกต่างกันป้อนให้กับตัวบัซเซอร์จึงจะสามารถสร้างสัญญาณเสียงออกมาได้ตามความถี่ที่ป้อนให้ บัซเซอร์ชนิดนี้โดยทั่วไปจะเรียกว่า “เปียซโซ” ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะใช้บัซเซอร์แบบชนิด B เพราะฉะนั้นการที่จะสั่งให้ตัวบัซเซอร์สร้างสัญญาณเสียงออกมาได้จะต้องสร้างสัญญาณพัลซ์ที่มีความถี่ที่แตกต่างกันป้อนให้กับบัซเซอร์นี้ ดังรูปที่ 3.12

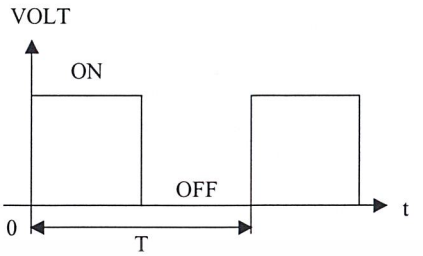


รูปที่ 3.12 วงจรสร้างสัญญาณเสียงด้วยบัซเซอร์

3.3.7 ชุดควบคุมมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์ให้ความเร็วคงที่ตามที่เราต้องการเป็นเรื่องที่ทำได้ค่อนข้างยาก ถ้าใช้การควบคุมแบบธรรมดาทั่วไปคือ การควบคุมแรงดันที่จ่ายให้กับมอเตอร์ การจะควบคุมความเร็วของมอเตอร์ให้ได้ความเร็วที่คงที่ ต้องควบคุมกระแสที่จ่ายให้กับมอเตอร์ ซึ่งวิธีนี้เรียกว่าการควบคุมแบบ Pulse Width Modulation (PWM) เป็นการควบคุมกระแสที่จ่ายให้กับมอเตอร์ โดยการจ่ายแรงดันให้กับมอเตอร์เป็นช่วงๆ ดังรูปที่ 3.13 กระแสไฟฟ้าที่มอเตอร์ได้รับจะเป็นค่าเฉลี่ยที่เกิดขึ้นในช่วง “ON” โดยความถี่ของการ ON และ OFF คงที่

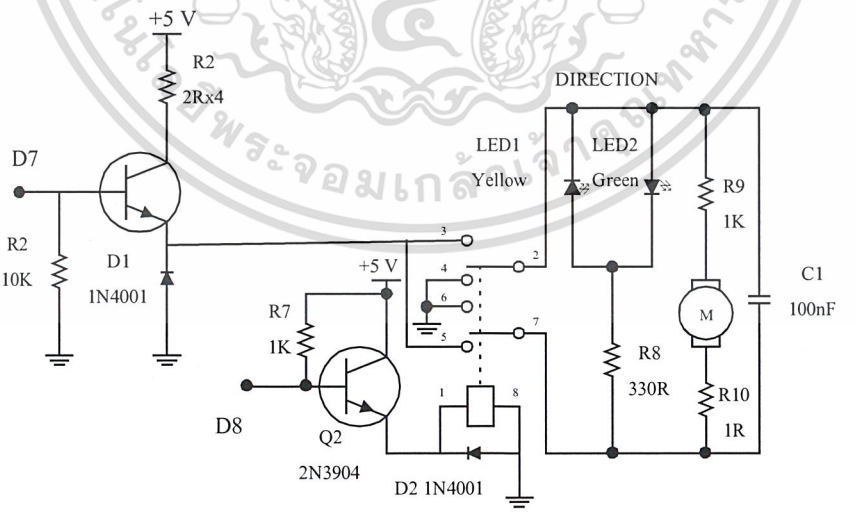
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 การจ่ายแรงดันให้กับมอเตอร์เป็นช่วงๆ

เมื่อความถี่คงที่ เราสามารถที่จะเปลี่ยนความเร็วรอบของมอเตอร์ได้โดยทำให้ช่วง “ON” มีเวลาที่มากกว่าช่วง “OFF” กระแสเฉลี่ยที่มอเตอร์ได้รับก็จะมีค่ามากขึ้นความเร็วรอบของมอเตอร์ก็จะเพิ่มขึ้นและถ้าหากต้องการลดความเร็วรอบของมอเตอร์ลง ก็ให้ลดช่วง “ON” ให้มีเวลาน้อย

สัญญาณควบคุมจะถูกส่งมาจากชุดไมโครคอนโทรลเลอร์บิต 7 (PA_7) จะเป็นบิตควบคุมการทำงานของมอเตอร์ โดยส่งแรงดันมาไบอัสทรานซิสเตอร์ BFY51 เพื่อขยายกระแสที่ใช้ขับให้มอเตอร์หมุน เมื่อสัญญาณที่จ่ายเข้ามีสภาวะเป็น “1” ทรานซิสเตอร์ BFY51 ก็จะต่อวงจรทำให้มอเตอร์หมุนและถ้าสัญญาณเป็น “0” ทรานซิสเตอร์ก็จะเปิดวงจรทำให้มอเตอร์หยุดหมุน



รูปที่ 3.14 วงจรควบคุมมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาผลประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างทรานซิสเตอร์กับมอเตอร์จะมีรีเลย์ต่อคั่นอยู่ซึ่งจะทำหน้าที่เป็นตัวเปลี่ยนทิศทางการหมุนของมอเตอร์โดยอาศัยบิต 6 (PA_6) เป็นตัวควบคุมการไบอัสทรานซิสเตอร์ 2N3904 สั่งให้

รีเลย์เปิดหรือปิด ดังนั้นวงจรควบคุมมอเตอร์ชุดนี้จะต้องมีสัญญาณมาควบคุมทั้งความเร็วและทิศทางการหมุนดังตารางที่ 3.4

ตารางที่ 3.4 บิตควบคุมการทำงานและทิศทางการหมุนของมอเตอร์

บิตที่ 7	บิตที่ 6	การทำงานของมอเตอร์	CODE
0	0	หยุดหมุน	00H
0	1	หยุดหมุน	40H
1	0	ทวนเข็มนาฬิกา	80H
1	1	ตามเข็มนาฬิกา	C0H

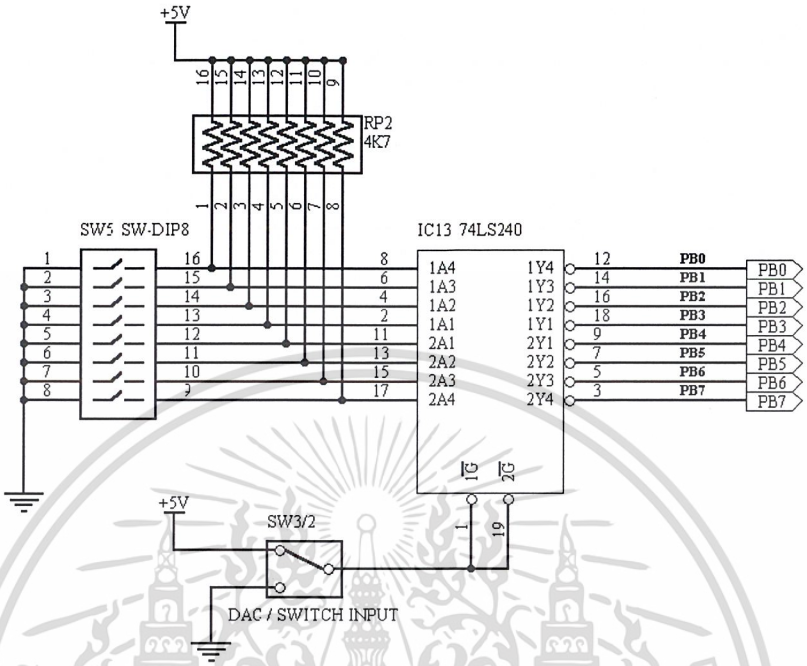
3.3.8 วงจรสร้างสภาวะลอจิกขนาด 8 บิต

มุมบนของชุดทดลองจะเป็นสวิตช์อินพุตจำนวน 8 ตัวคือ SW 1-SW 8 ซึ่งเป็นดิพสวิตช์ที่ต่อเข้ากับ IC 74LS240 ที่เป็น Octal Tri-State Inverter Buffer ดิพสวิตช์จะทำหน้าที่สร้างสภาวะลอจิก “1” หรือ “0” เมื่อดิพสวิตช์อยู่ที่ตำแหน่ง “ON” จะมีสภาวะลอจิกข้อมูลเป็น “1” (HI) ถ้าไม่อยู่ที่ตำแหน่ง “ON” จะมีสภาวะลอจิก “0” (LO)

ไอซี 74LS240 ซึ่งเป็น Octal Tri-State Inverter Buffer จะทำงานเมื่อได้รับการ “Enable” โดยต่อขา 1 และขา 19 ลงกราวด์ (SW 2) ซึ่งแต่ละแชนเนลของ 74LS240 จะให้เอาต์พุตออกมา 3 สถานะคือ

1. เอาต์พุตเป็นลอจิก “0” เมื่อได้รับอินพุตเข้ามาเป็นลอจิก “1”
2. เอาต์พุตเป็นลอจิก “1” เมื่อได้รับอินพุตเข้ามาเป็นลอจิก “0”
3. เอาต์พุตเป็นสภาวะอิมพีแดนซ์สูงเมื่อ 74LS240 ได้รับการ Disable คือ ขา 1 และขา 19 ถูกต่อเข้ากับไฟ +5V ถึงแม้อินพุตที่เข้ามาจะเป็น “1” หรือ “0” ก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 วงจรสร้างสภาวะลอจิกขนาด 8 บิต

รูปที่ 3.15 ดิพสวิตช์แต่ละตัวจะเปิดวงจรเมื่ออยู่ที่ตำแหน่ง “OFF” และถูกต่อลงกราวด์เมื่ออยู่ที่ตำแหน่ง “ON” ดิพสวิตช์ทั้ง 8 ตัว จะมีตัวต้านทาน 4.7 กิโลโอห์ม ต่อเข้ากับไฟ +5 โวลต์ (เรียกว่าการ Pull up) ที่ตำแหน่ง “OFF” ของดิพสวิตช์ทำให้อินพุตของ 74LS240 มีสภาวะลอจิกเป็น “1” และเมื่อดิพสวิตช์อยู่ที่ตำแหน่ง “ON” ทำให้อินพุตมีสข้อมูลเป็น “0” เพราะตัวต้านทานถูกต่อลงกราวด์ ซึ่งทำให้มีกระแสไหลผ่านตัวต้านทานประมาณ 1.06 มิลลิแอมป์

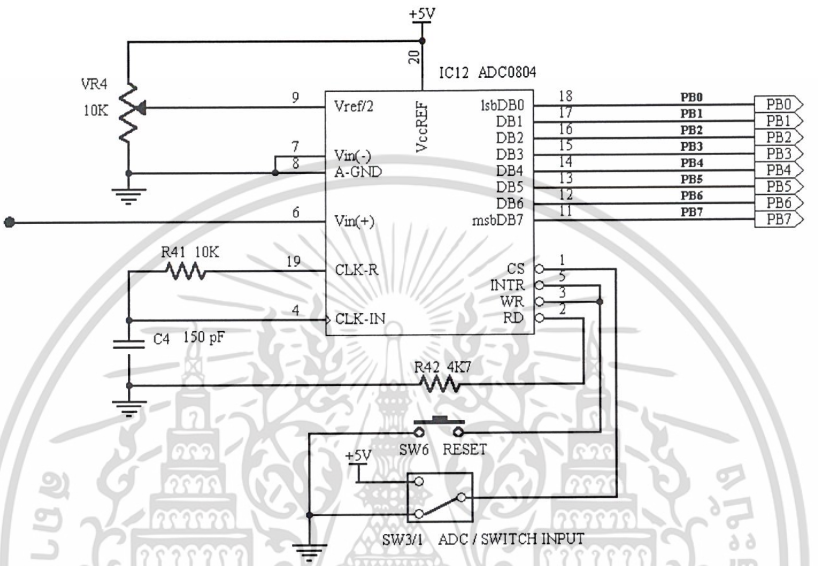
3.3.9 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ใช้ไอซีเบอร์ ADC0804 ซึ่งเป็น A/D ขนาดเอาต์พุต 8 บิต รับสัญญาณอินพุตที่เป็นแรงดันอยู่ในช่วง 0-2.55 โวลต์ จะมีค่า 1 LSB = 10 มิลลิโวลต์ สัญญาณอินพุตที่รับเข้ามาจะผ่านทางสวิตช์ SW 2 ซึ่งจะเป็นตัวเลือกของสัญญาณว่าจะเป็นสัญญาณมาจากวงจรตรวจวัดความสว่างแสงหรือจะมาจาก Potentiometer (POT) ซึ่งวงจรเหล่านี้จะให้เอาต์พุตออกมาเป็นแรงดัน 0-2.55 โวลต์

จากรูปที่ 3.16 การทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล ขา Chip-Select ของ ADC0804 ต่อเข้ากับสวิตช์ SW 3 ซึ่งจะเป็นตัวควบคุมการทำงานของ A/D ว่าจะให้ทำงานหรือหยุดทำงาน คือเมื่อเลื่อนสวิตช์ SW 3 มาอยู่ที่ตำแหน่ง A/D ขา Chip Select ก็จะถูกต่อลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราวด์ (GND) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลก็จะทำงาน แต่ถ้าเลื่อนไปอยู่ที่ตำแหน่งสวิตช์อินพุต วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลก็จะหยุดทำงานและในขณะที่วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล กำลังทำงานอยู่แล้วเกิดขัดข้องหรือไม่ทำงานตามปกติจะต้องทำการรีเซ็ต โดยการกดสวิตช์รีเซ็ต



รูปที่ 3.16 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

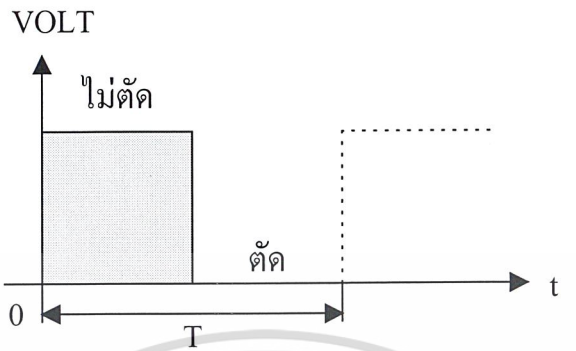
3.3.10 การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด

การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด (Infrared Sensor)

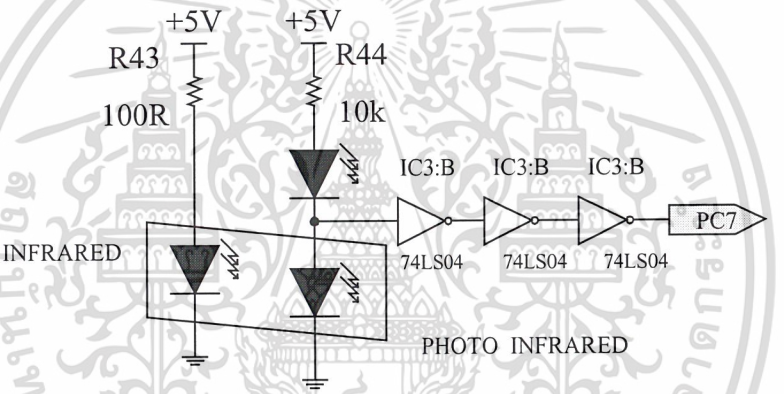
การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด อาศัยการตัดลำแสงอินฟราเรดที่ส่งจากตัวสร้างแสงอินฟราเรดไปยังตัวรับ (Photo Diode) เมื่อมีวัตถุเคลื่อนที่ผ่านลำแสง ลำแสงจะไม่ผ่านไปยังตัวรับทำให้ได้เอาต์พุตที่ขาของตัวรับเป็น “1” จากนั้นจะเข้าไปยัง NOT Gate จำนวน 3 ตัว เพื่อแปลงสัญญาณที่ได้เป็นสัญญาณ TTL เอาต์พุตที่ได้นี้จะมีค่าเป็น “0” เพื่อที่จะส่งไปยังพอร์ต PC ของ 8255 ตำแหน่งบิตที่ 7 แต่ถ้าตัวรับแสงได้รับแสงอินฟราเรดแล้วเอาต์พุตที่ได้เมื่อผ่าน NOT Gate แล้วจะมีค่าเป็น “1”

วัตถุที่เคลื่อนที่ผ่านลำแสงอินฟราเรดและเกิดการเปลี่ยนแปลงได้ ต้องเป็นวัตถุที่ทึบแสงเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 เอาต์พุตของวงจรตรวจจับการเคลื่อนที่เมื่อมอเตอร์หมุน 1 รอบ

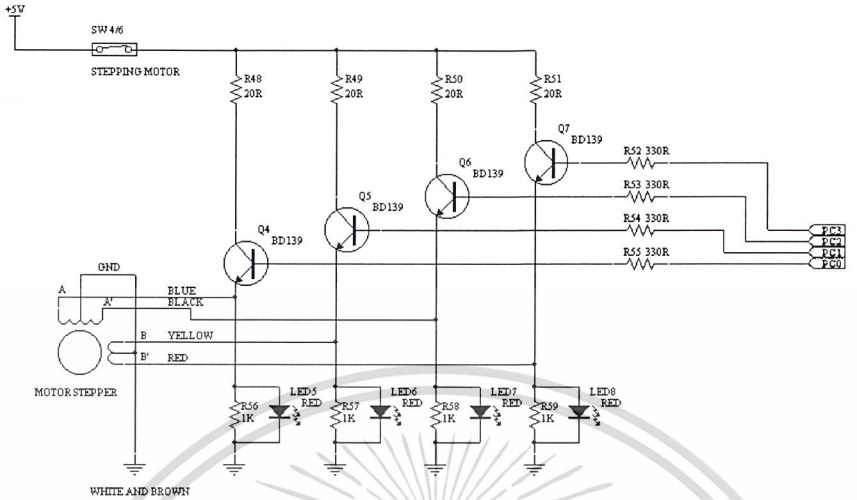


รูปที่ 3.18 วงจรตรวจจับการเคลื่อนที่

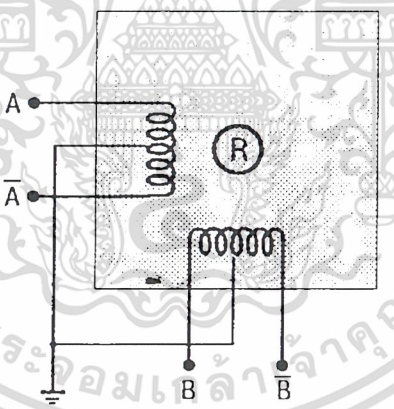
3.3.11 สเต็ปปิ้งมอเตอร์

การควบคุมสเต็ปปิ้งมอเตอร์ ก็คือการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์แต่ละขด สำหรับชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีวงจรควบคุมการจ่ายกระแสไฟไว้สำหรับสเต็ปปิ้งมอเตอร์ 2 เฟส 6 สาย ดังรูปที่ 3.20 วงจรจะจ่ายกระแสให้กับขดลวดได้สูงสุดประมาณ 800 มิลลิแอมป์ต่อเฟส วงจรควบคุมจะต่อเข้ากับพอร์ต PC_0-PC_3 โดยที่ PC_0 ต่อเข้ากับเฟส A, PC_1 ต่อเข้ากับเฟส A′, PC_2 ต่อเข้ากับเฟส B และ PC_3 ต่อเข้ากับเฟส B′ แต่ละเฟสจะมีหลอดไดโอดเปล่งแสง เพื่อแสดงสภาวะของแต่ละเฟส เมื่อเอาต์พุตของพอร์ต PC มีค่าเป็น “1” ขดลวดสเต็ปปิ้งมอเตอร์จะได้รับกระแสไฟฟ้า หลอดไดโอดเปล่งแสงจะติดสว่าง ดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



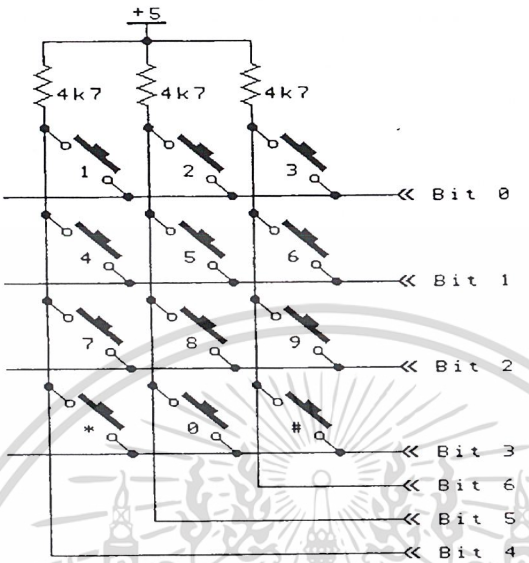
รูปที่ 3.19 วงจรควบคุมสเต็ปปิ้งมอเตอร์



รูปที่ 3.20 สเต็ปปิ้งมอเตอร์แบบ 2 เฟส 6 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.12 คีย์แพดขนาด 4X7



รูปที่ 3.21 วงจรคีย์แพดขนาด 4X7

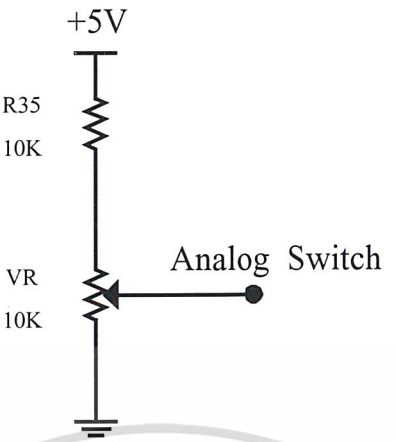
จากรูปที่ 3.21 วงจรคีย์แพดจะมีตัวต้านทาน 4.7 กิโลโอห์ม ต่อเข้ากับคอลัมน์ C1-C3 เพื่อทำให้มีสภาวะลอจิกเป็น “1” (HI) คอลัมน์ C1-C3 จะต่อเข้ากับพอร์ตอินพุตเพื่อตรวจสอบว่าคีย์ใดที่ถูกกด ส่วนแถว R1-R4 จะถูกต่อเข้ากับพอร์ตเอาต์พุต โดยสัญญาณเอาต์พุตของแต่ละแถว จะมีสภาวะลอจิกเป็น “0” (LO) ทีละแถว เมื่อแถวใดเป็น “0” แถวที่เหลือก็จะเป็น “1” โดยจะวนรอบอยู่อย่างนี้

เมื่อมีการกดคีย์ (ตามจุดในรูป) สัญญาณอินพุตของคอลัมน์ C1-C3 ที่เป็น “1” อยู่ทั้งหมด 111 จะกลายเป็น 110 เพราะที่คอลัมน์ C1 จะกลายเป็น “0” ในจังหวะที่แถว R2 มีสภาวะเป็น “0” เมื่อนำผลของอินพุตและเอาต์พุตมารวมด้วยกันก็สามารถบอกตำแหน่งของคีย์ที่ถูกกดได้

3.3.13 วงจรปรับแรงดัน 0-2.55 โวลต์

วงจรปรับแรงดัน 0-2.55 โวลต์ ใช้ POT ขนาด 10 กิโลโอห์ม ทำหน้าที่เป็นวงจรแบ่งแรงดัน ปรับแรงดันได้ในช่วง 0-2.55 โวลต์ เพื่อป้อนให้กับวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล และวงจรแสดงระดับสัญญาณ ดังรูปที่ 3.22

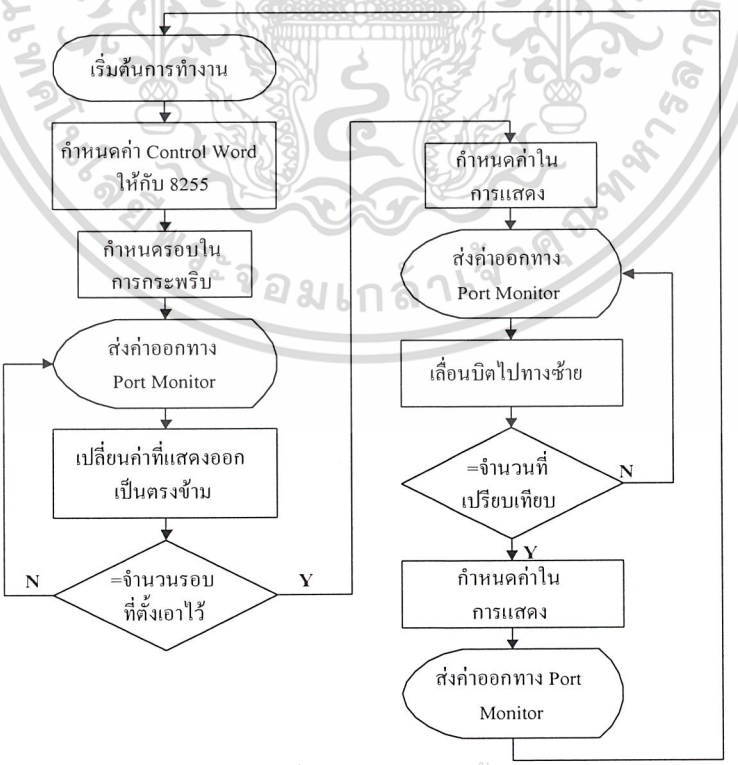
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 วงจรปรับแรงดัน 0-2.55 โวลต์

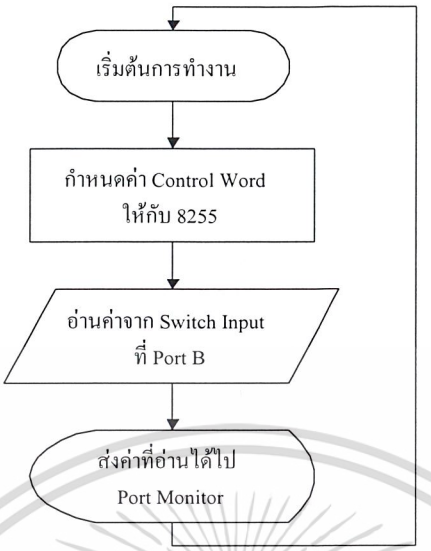
3.4 การออกแบบด้านซอฟต์แวร์

ในการออกแบบด้านซอฟต์แวร์นั้นจะเป็นตัวทดลองการทำงานของอุปกรณ์ทางด้านฮาร์ดแวร์ ซึ่งจะต้องออกแบบให้สอดคล้องกับการทำงานของอุปกรณ์นั้นๆ หลักในการออกแบบจะต้องมีการเขียนผังงานโปรแกรมของวงจรแต่ชุดเพื่อกำหนดการทำงานให้มีลำดับและขั้นตอนดังนี้

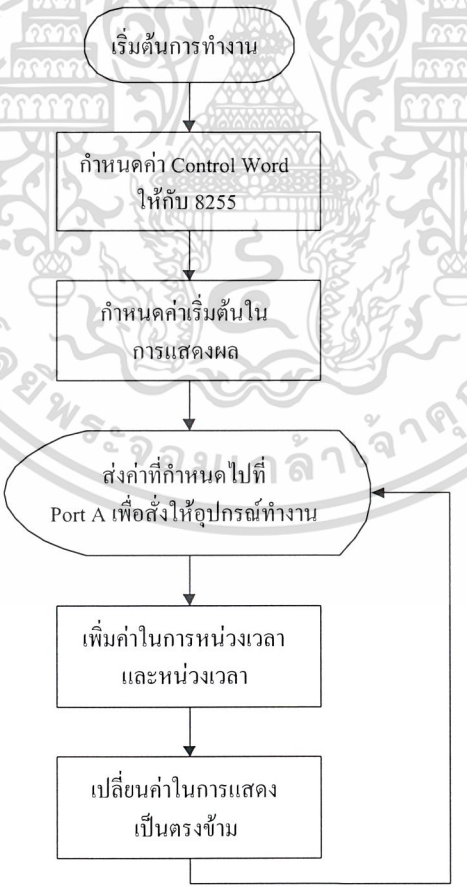


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.23 ผังงานของโปรแกรมแสดงผลข้อมูลขนาด 8 บิต

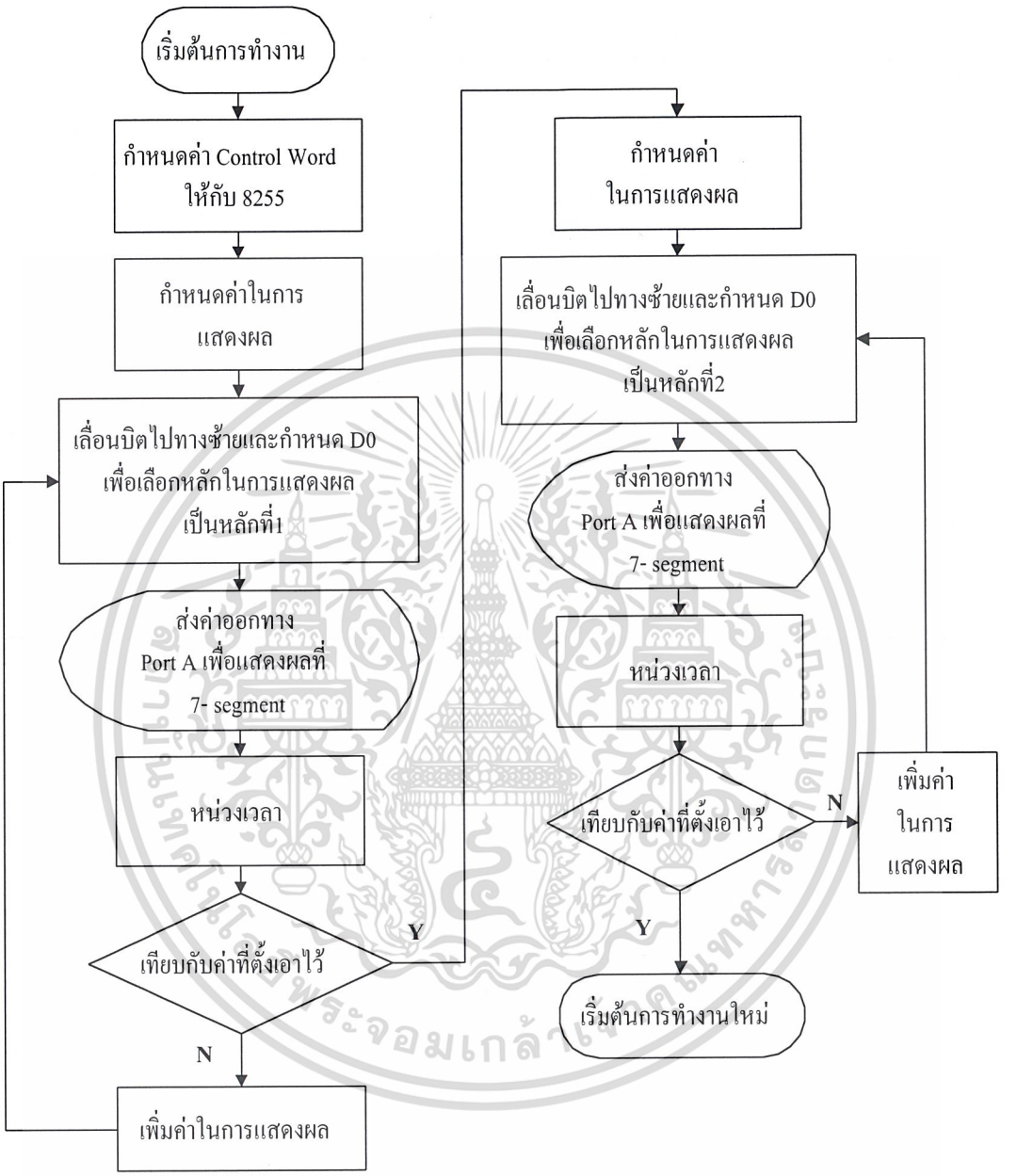


รูปที่ 3.24 ผังงานของโปรแกรมสร้างสภาวะลอจิกขนาด 8 บิต



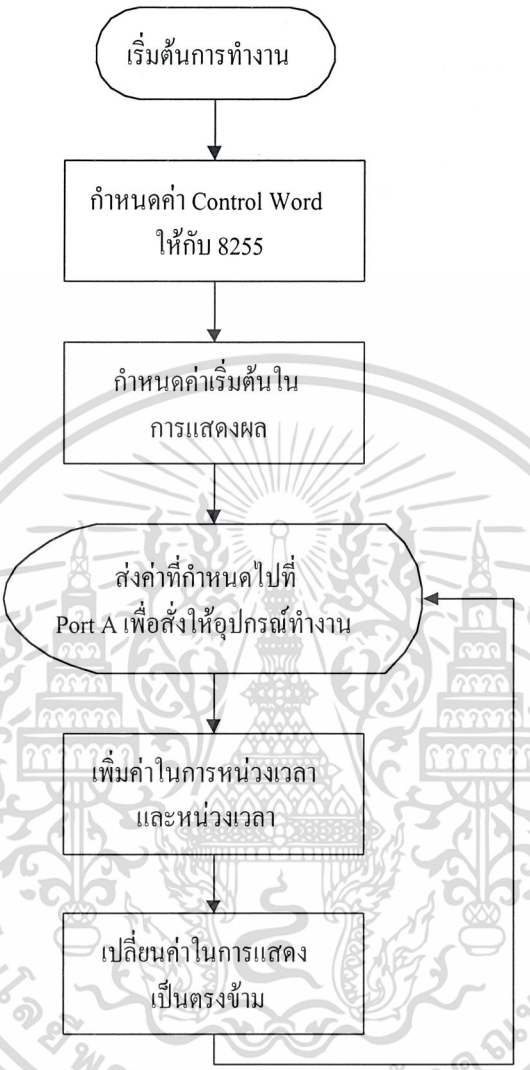
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.25 ผังงานของโปรแกรมสร้างสัญญาณเสียงด้วยบัซเซอร์



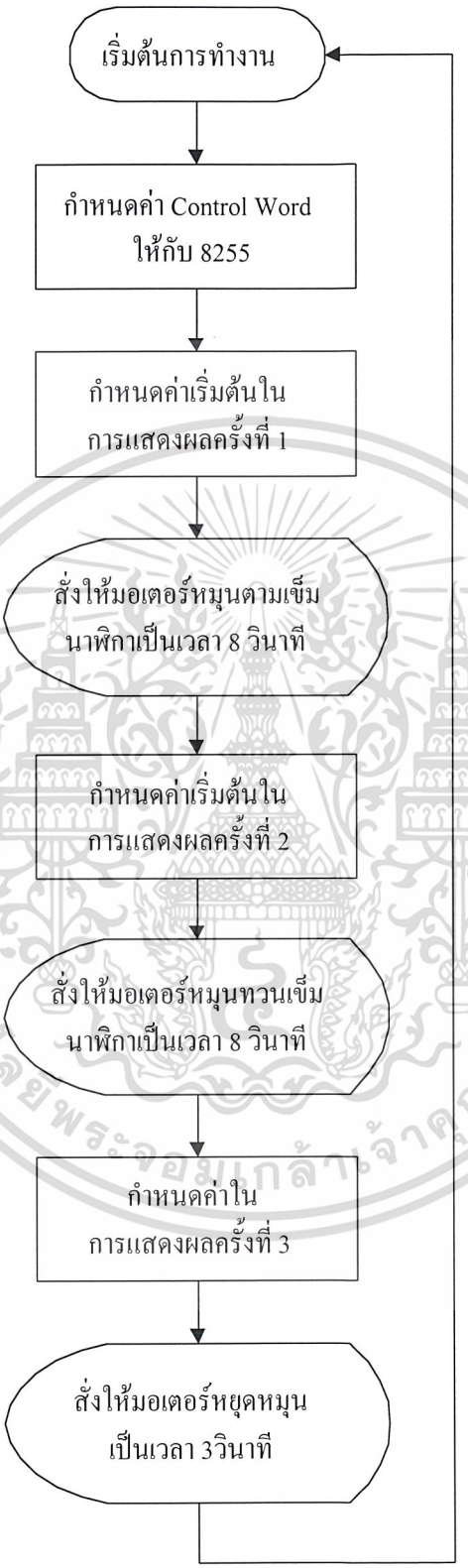
รูปที่ 3.26 ผังงานของโปรแกรมแสดงผลตัวเลข 7 ส่วนขนาด 2 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



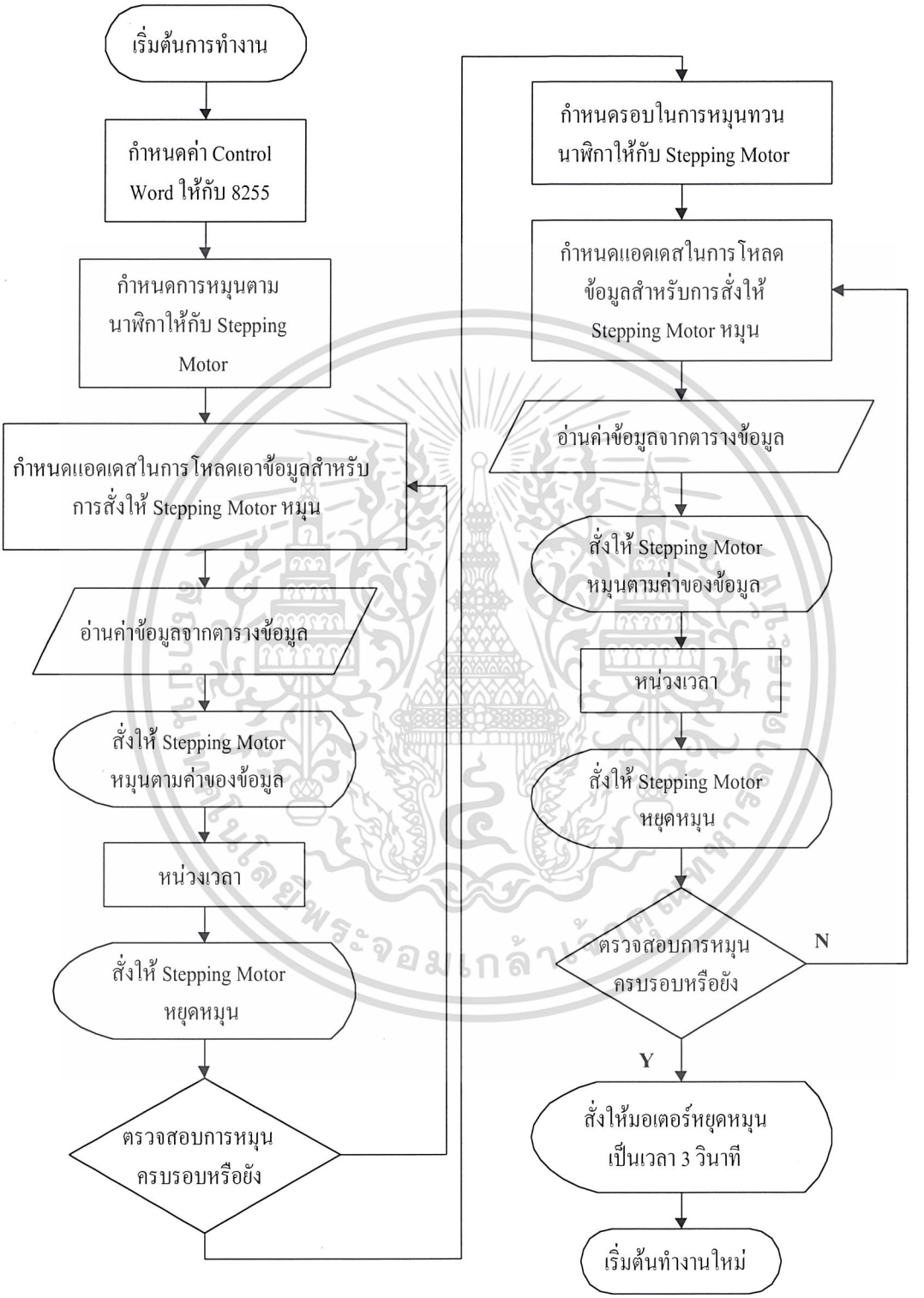
รูปที่ 3.27 ผังงานของโปรแกรมสร้างสัญญาณแสดงผลด้วย LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



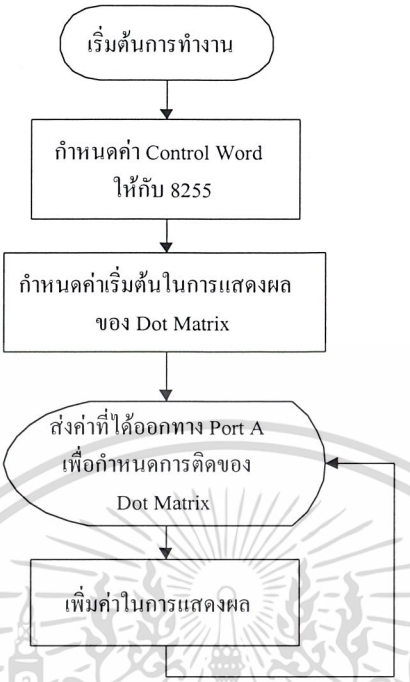
รูปที่ 3.28 ผังงานของโปรแกรมควบคุมมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

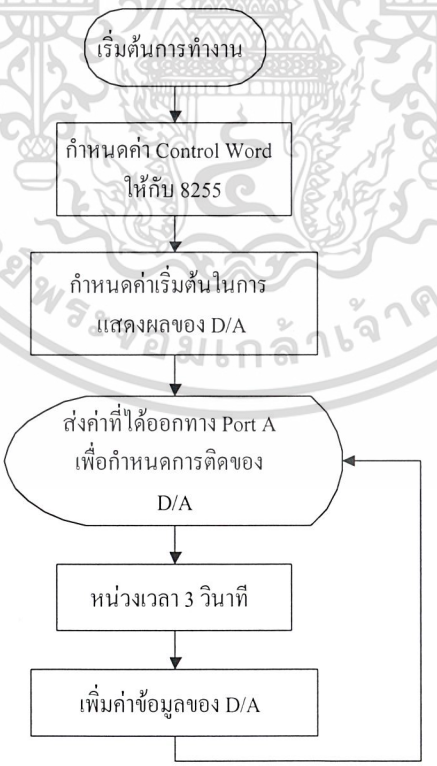


รูปที่ 3.29 ผังงานของโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

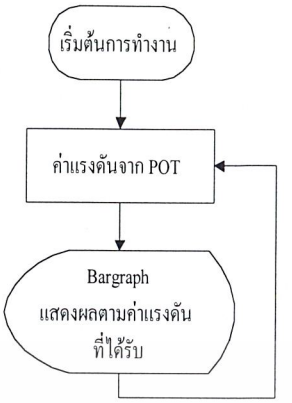


รูปที่ 3.30 ผังงานของโปรแกรมแสดงผลแบบจุดขนาด 5X7

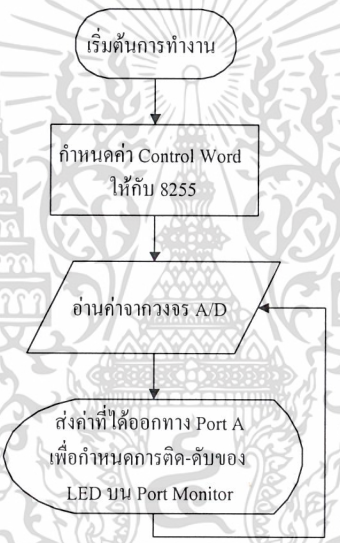


รูปที่ 3.31 ผังงานของโปรแกรมแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

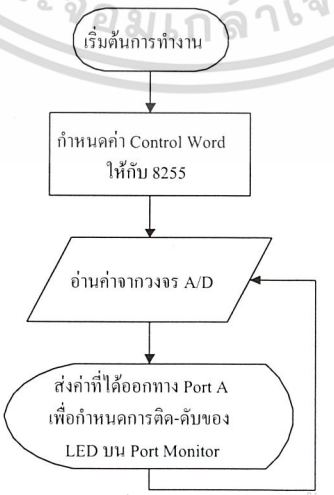
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.32 ผังงานของโปรแกรมแสดงระดับสัญญาณ

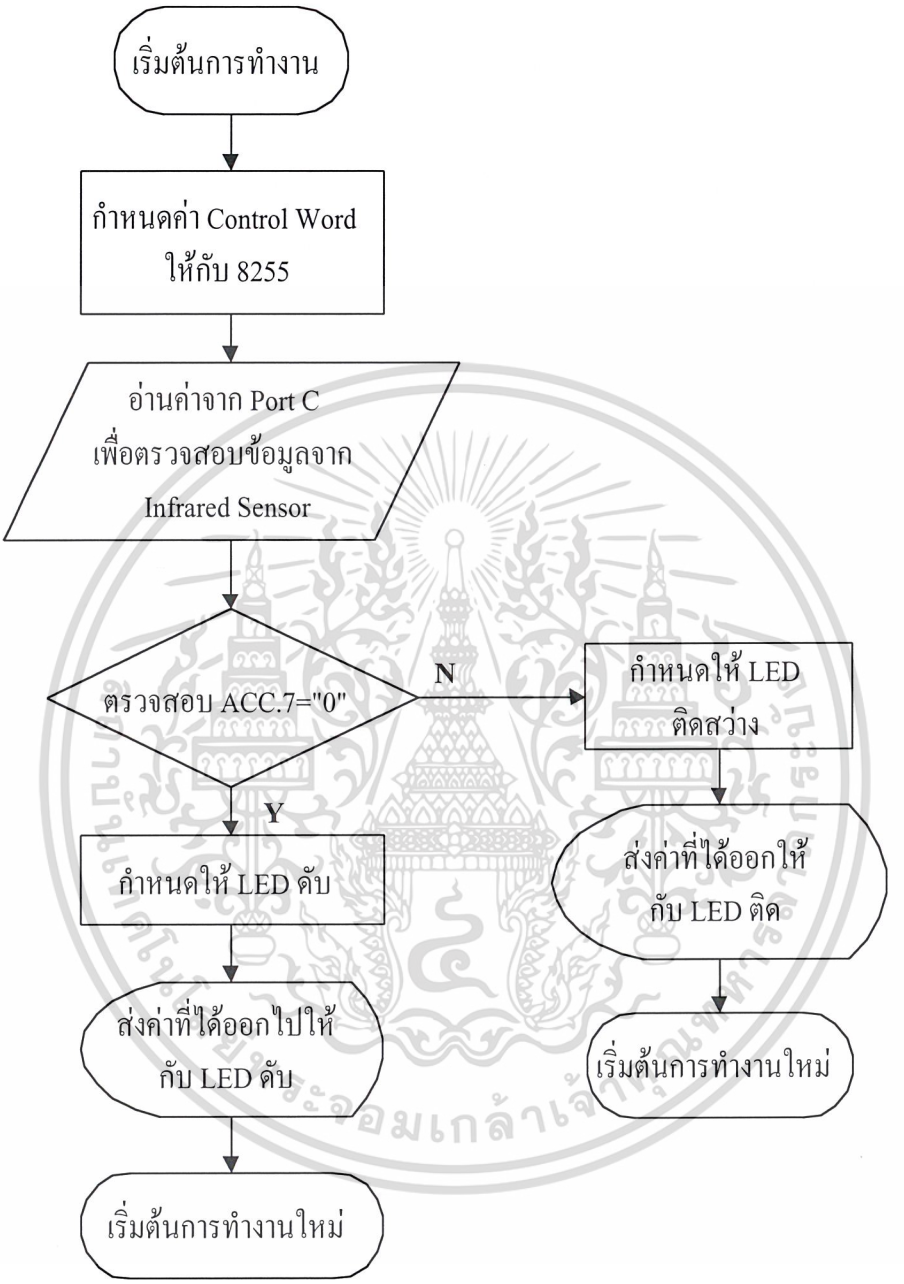


รูปที่ 3.33 ผังงานของโปรแกรมปรับแรงดัน 0-2.55 โวลต์



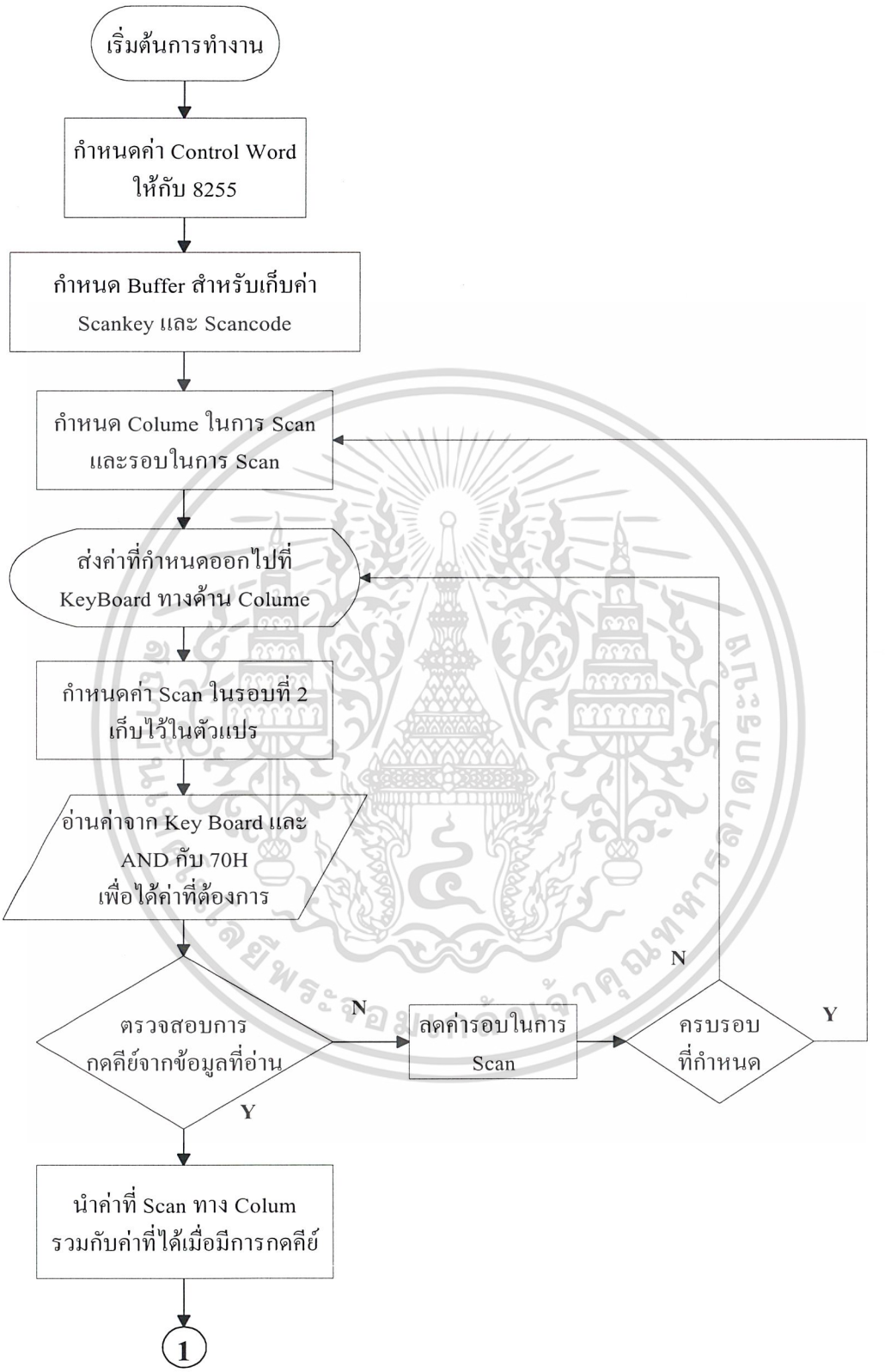
รูปที่ 3.34 ผังงานของโปรแกรมแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



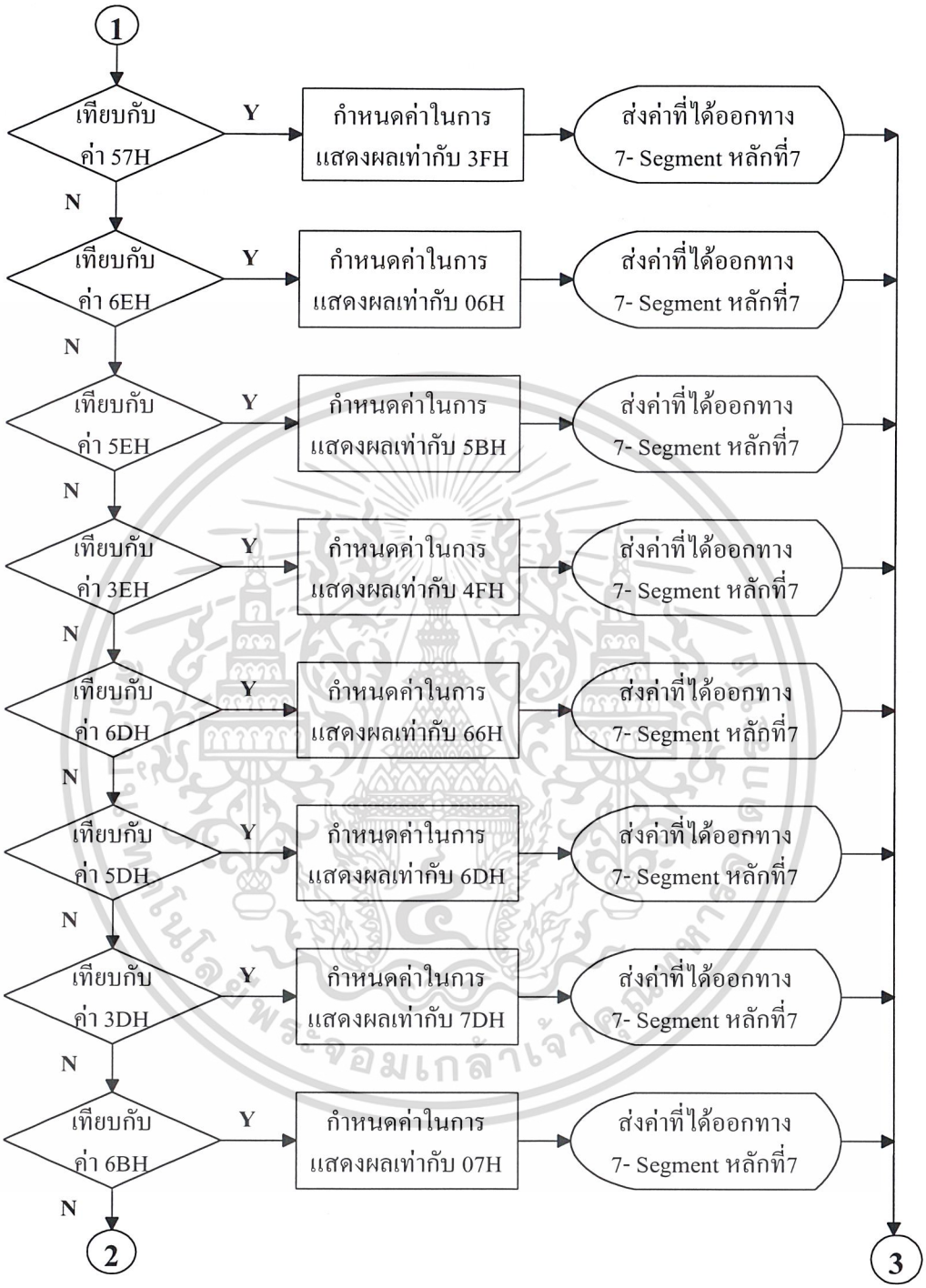
รูปที่ 3.35 ผังงานของโปรแกรมตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



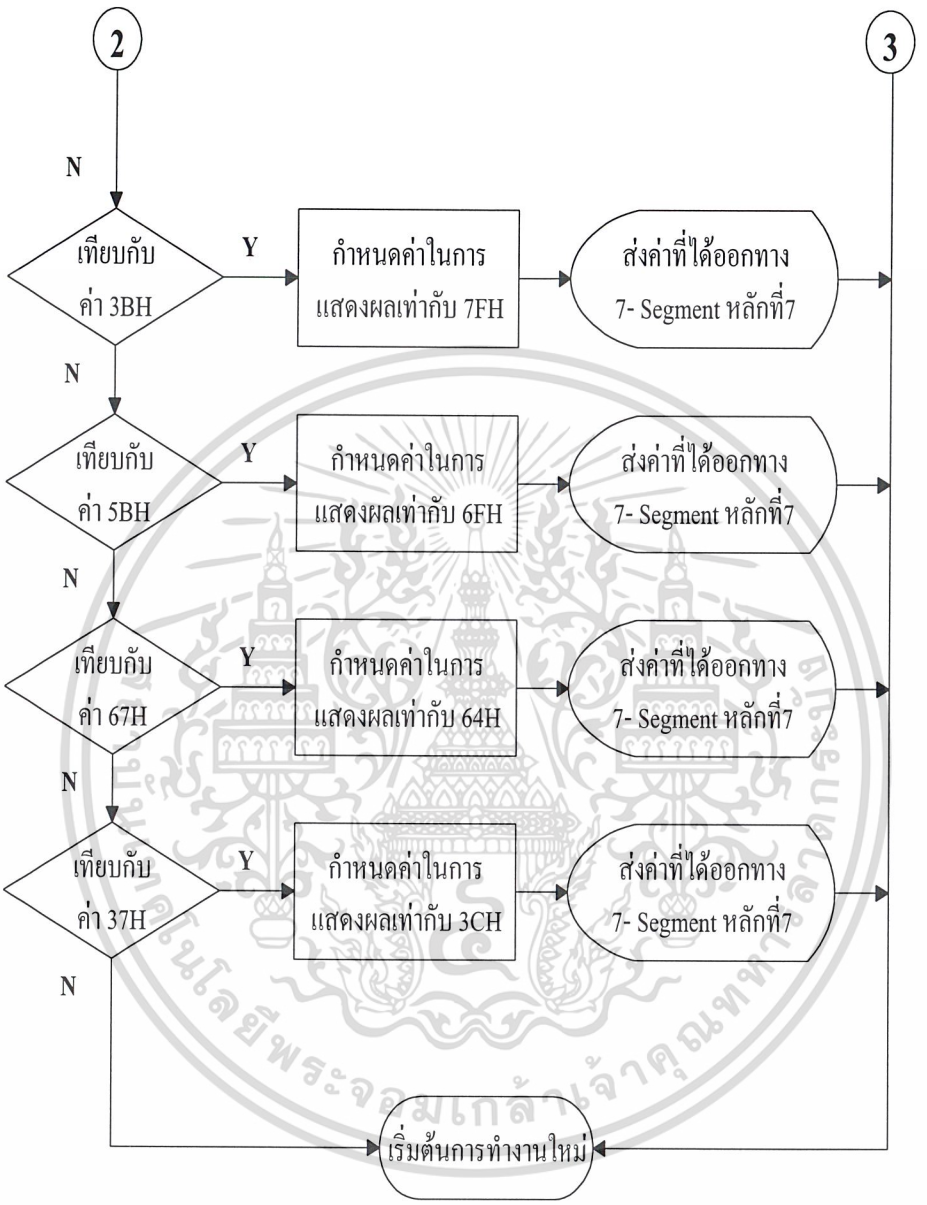
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.36 ผังงานของโปรแกรมรับข้อมูลจากคีย์แพด



รูปที่ 3.36 (ต่อ) ผังงานของโปรแกรมรับข้อมูลจากคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 (ต่อ) ผังงานของโปรแกรมรับข้อมูลจากคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

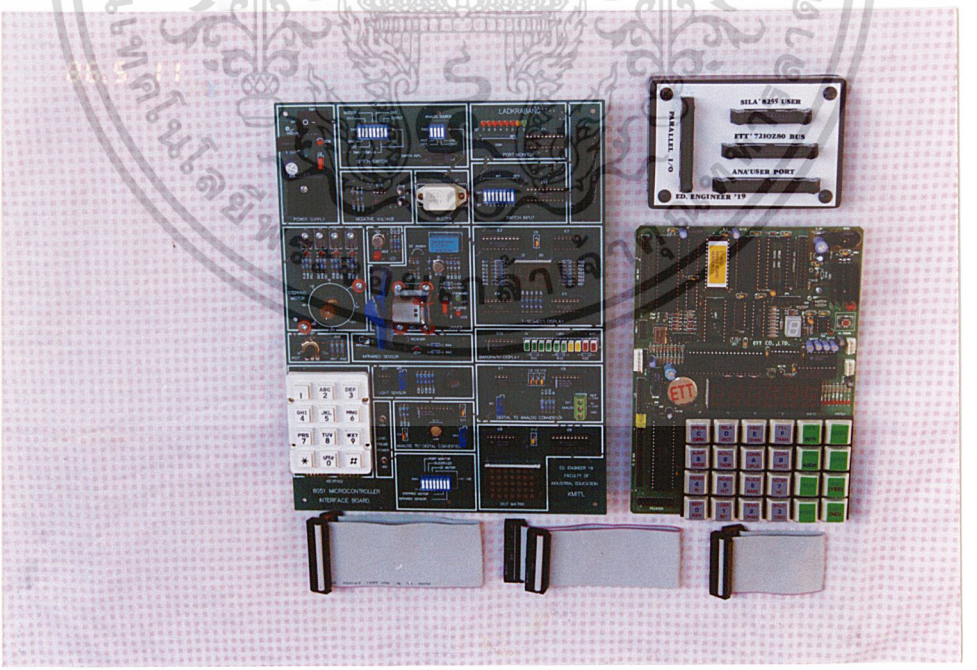
บทที่4

ผลการทดลองและทดสอบ

4.1 กล่าวนำ

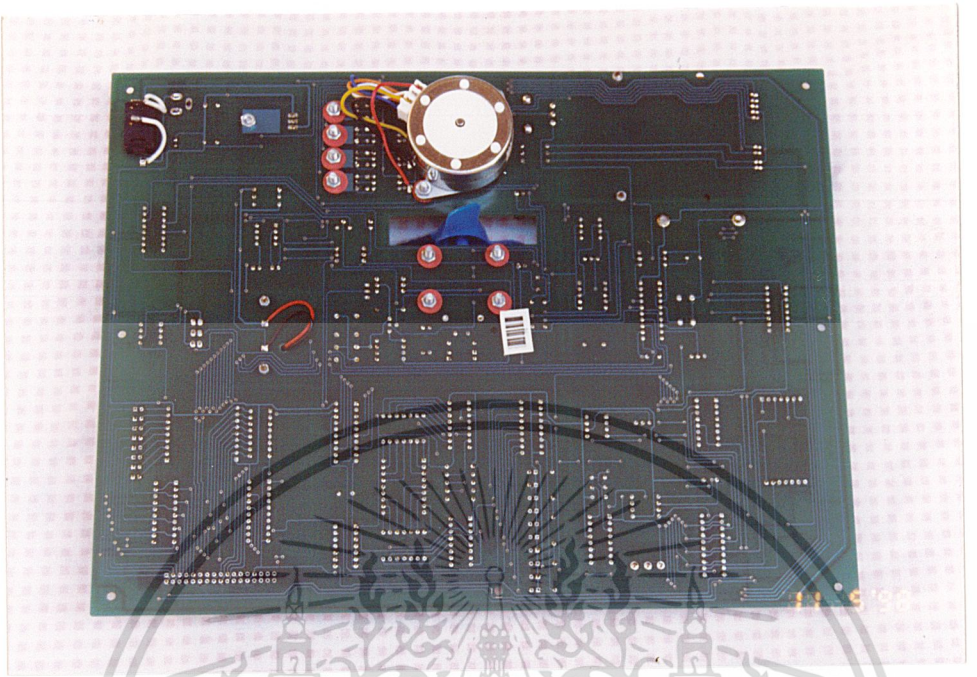
จากที่ได้กล่าวไปแล้วในบทที่ 1 ถึงบทที่ 5 เกี่ยวกับทฤษฎีและหลักการตลอดจนการออกแบบชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ในบทนี้จะกล่าวถึงการทดสอบว่า ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 สามารถทำงานได้ตามวัตถุประสงค์หรือไม่

จากวัตถุประสงค์การออกแบบได้กล่าวไปแล้วในบทที่ 3 ดังนั้นการทดสอบการทำงานในบทนี้จึงจำแนกออกเป็นหัวข้อย่อยๆ ตามใบงานที่ได้ออกแบบไว้และก่อนจะทำการทดลองการทำงานก็ต้องมาดูลักษณะการทำงานของชุดทดลองก่อน ดังรูปที่ 4.1 ภาพด้านบนของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ถึงรูปที่ 4.5 การวางตำแหน่งของคอนเน็กเตอร์บนชุดแปลงคอนเน็กเตอร์

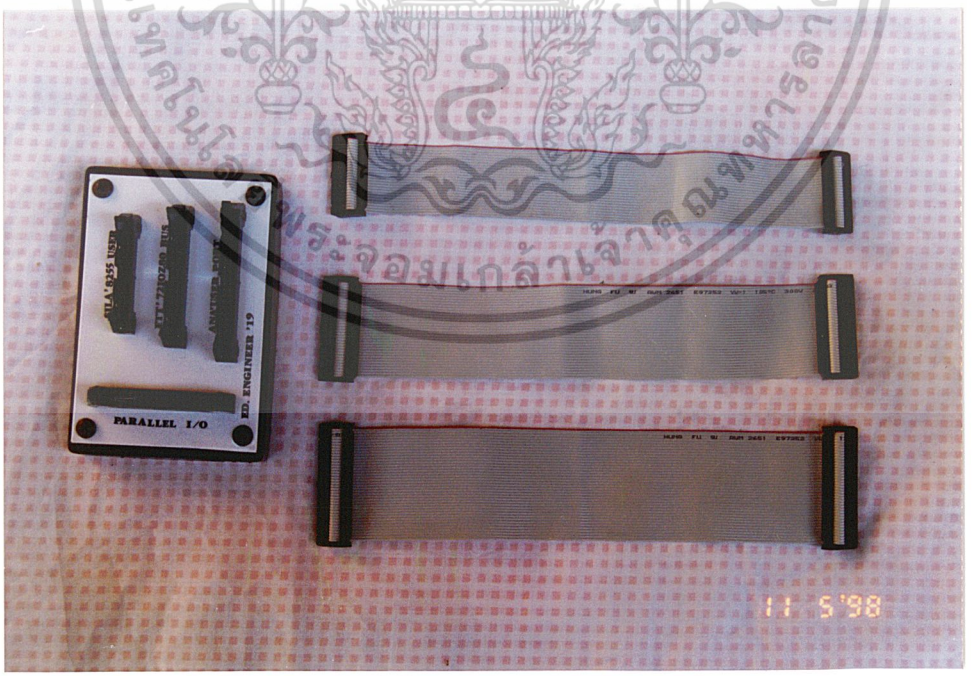


รูปที่ 4.1 ภาพด้านบนของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

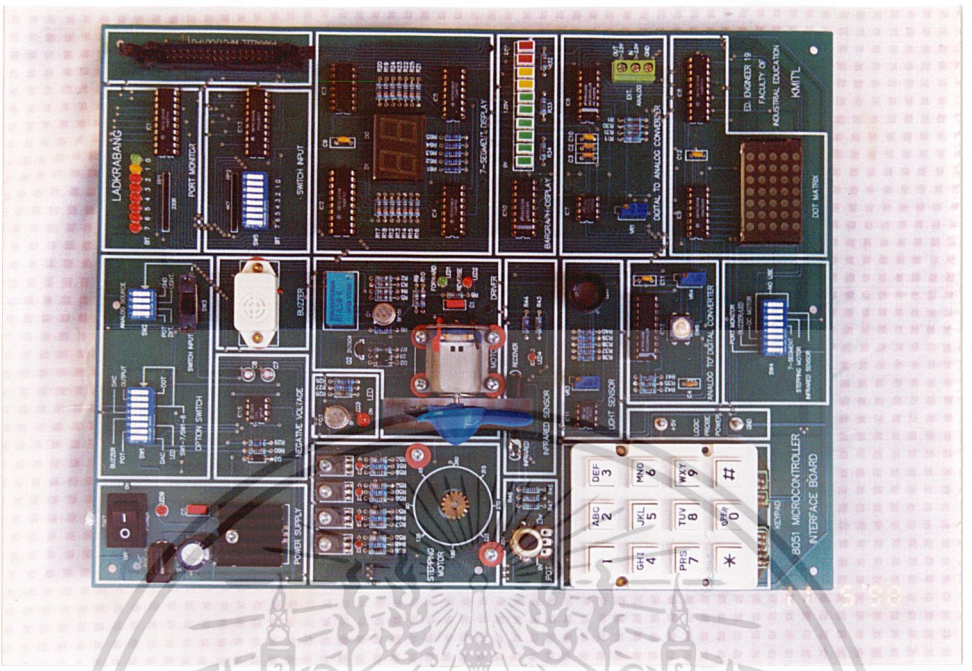


รูปที่ 4.2 ภาพด้านล่างของชุดทดลองไมโครคอนโทรลเลอร์ 8051

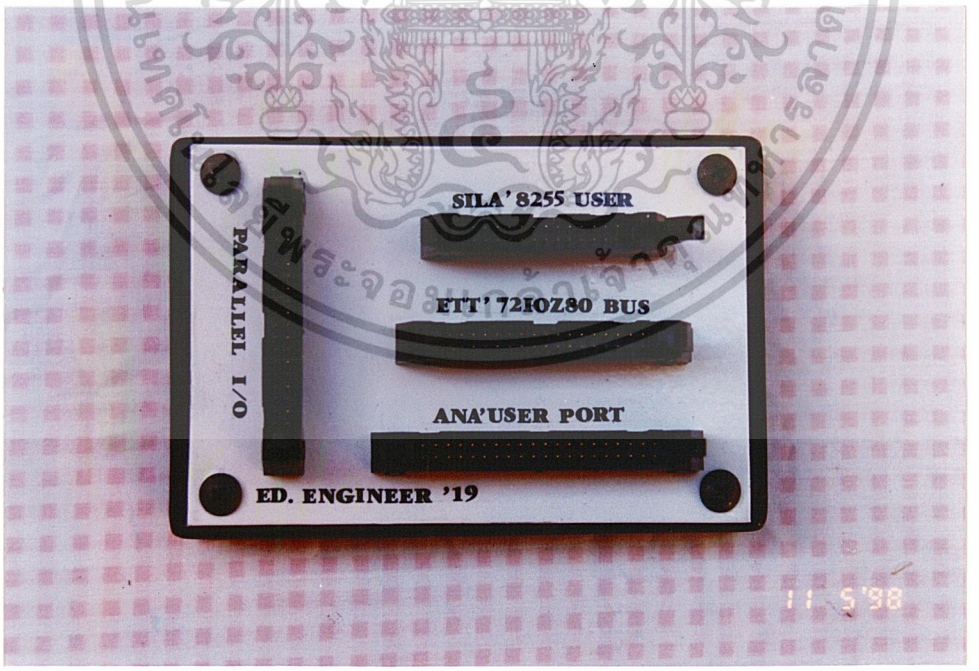


รูปที่ 4.3 ภาพด้านบนของชุดแปลงคอนเน็กเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การวางตำแหน่งของวงจรบนชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051



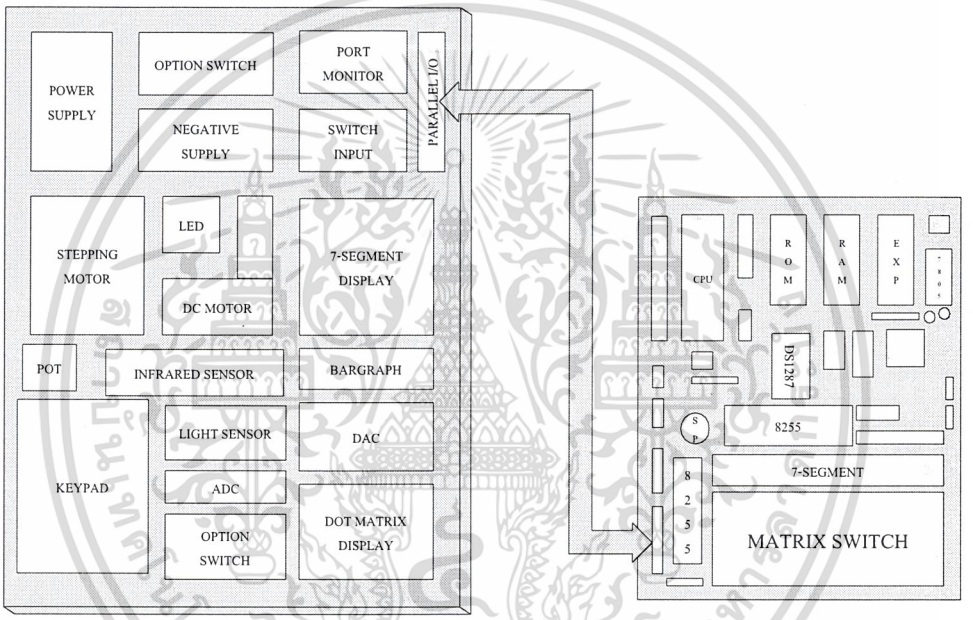
รูปที่ 4.5 การวางตำแหน่งของคอนเน็กเตอร์บนชุดแปลงคอนเน็กเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองชุดแสดงผลข้อมูลขนาด 8 บิต

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้ชุดแสดงผลข้อมูลทำงานได้
3. ทำการโหลดโปรแกรมของไดร์ฟ A ชื่อ Test01.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน



รูปที่ 4.6 การเชื่อมต่อชุดทดลองกับซิงเกิ้ลบอร์ดรุ่น ET-8032 V2.0

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานแล้ว การแสดงผลของหลอดไดโอดเปล่งแสงของชุดแสดงผลข้อมูลนั้นจะแสดงเป็นไฟกระพริบ จากนั้นเป็นไฟวิ่งและติดค้างวนไปเรื่อยๆ ทำให้เราทราบได้ว่าชุดแสดงผลข้อมูลสามารถที่จะแสดงผลข้อมูล “0” และ “1” ได้

4.3 การทดลองชุดสร้างสภาวะลอจิกขนาด 8 บิต

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปรับสวิตช์ควบคุมการทำงานให้วงจรสร้างสภาวะลอจิกขนาด 8 บิตและชุดแสดงผลข้อมูลทำงานได้

3. ทำการโหลดโปรแกรมจากไดร์ฟ A ชื่อ Test02.Hex ไปยังซิงเกิ้ลบอร์ด

4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อปรับตำแหน่งสวิตช์อินพุตตำแหน่งใดก็ตามมาอยู่ที่ตำแหน่ง "ON" แล้ว หลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูลจะติดสว่าง ณ ตำแหน่งเดียวกับตำแหน่งของสวิตช์อินพุตที่ตำแหน่ง "ON" ทำให้เราทราบว่าสวิตช์อินพุตสามารถสร้างสภาวะข้อมูล "0" และ "1"ได้

4.4 การทดลองชุดแสดงผลตัวเลข 7 ส่วนขนาด 2 หลัก

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

2. ปรับสวิตช์ควบคุมการทำงานให้ชุดแสดงผลตัวเลข 7 ส่วนทำงานได้

3. ทำการโหลดโปรแกรมจากไดร์ฟ A ชื่อ Test03.Hex ไปยังซิงเกิ้ลบอร์ด

4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากผลการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานชุดแสดงผลตัวเลข 7 ส่วน หลักที่ 0 จะนับจำนวนตั้งแต่ 0 ถึง 9 และตามด้วยการนับ 0-9 อีกครั้งในหลักที่ D1 จึงสามารถสรุปได้ว่า ชุดแสดงผลตัวเลข 7 ส่วน สามารถแสดงจำนวนนับได้ตั้งแต่ 00-99

4.5 การทดลองวงจรชุดสร้างสัญญาณเสียงด้วยบัซเซอร์

ลำดับการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

2. ปรับสวิตช์ควบคุมการทำงานให้บัซเซอร์ทำงานได้

3. ทำการโหลดโปรแกรมไดร์ฟ Aชื่อ Test04.Hex ไปยังซิงเกิ้ลบอร์ด

4. สั่งให้โปรแกรมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน บัซเซอร์จะส่งสัญญาณเสียงออกเป็นช่วงๆ และมีสัญญาณเสียงความถี่ด้วยกัน จึงสรุปได้ว่าบัซเซอร์จะมีการสร้างสัญญาณเสียงเป็นช่วงๆ เป็นความถี่หลายๆ ความถี่ ซึ่งขึ้นอยู่กับการควบคุมของไมโครคอนโทรลเลอร์

4.6 การทดลองชุดสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้หลอดไดโอดเปล่งแสงทำงานได้
3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test05.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานหลอดไดโอดเปล่งแสงจะติด-ดับ สลับกันไปและมีลักษณะการทำงานเหมือนกัน หลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูลซึ่งแสดงสภาวะข้อมูล “0” และ “1” ได้

4.7 การทดลองชุดควบคุมมอเตอร์กระแสตรง

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้มอเตอร์กระแสตรงทำงานได้
3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test06.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานมอเตอร์กระแสตรงจะหมุนตามเข็มนาฬิกาประมาณ 8 วินาที ต่อจากนั้นจะหมุนทวนเข็มนาฬิกาประมาณ 8 วินาที และหยุดหมุนเป็นเวลา 3 วินาที ทำให้สรุปได้ว่ามอเตอร์กระแสตรงสามารถควบคุมให้หมุนตามเข็มหรือทวนเข็มก็ได้ขึ้นอยู่กับการสั่งงานลงในไมโครคอนโทรลเลอร์และยังสามารถควบคุมความเร็วในการหมุนให้ช้าหรือเร็วได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 การทดลองชุดควบคุมสเต็ปปิ้งมอเตอร์

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้สเต็ปปิ้งมอเตอร์ทำงานได้
3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test07.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานสเต็ปปิ้งมอเตอร์จะหมุนไปตามทิศทางตามเข็มนาฬิกา 1 รอบ ต่อจากนั้นจะหมุนทวนเข็มนาฬิกา 1 รอบ และหยุดหมุนประมาณ 3 วินาที เมื่อเกิดการเปลี่ยนแปลงทิศทางการหมุน ในการเคลื่อนที่ของสเต็ปปิ้งมอเตอร์จะเคลื่อนที่เป็นลำดับขั้นอย่างเห็นได้ชัด ทำให้สามารถสรุปได้ว่าสเต็ปปิ้งมอเตอร์จะหมุนเป็นลำดับขั้นในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาก็ได้และความเร็วในการหมุนจะขึ้นอยู่กับการหน่วงเวลาในแต่ละลำดับขั้น

4.9 การทดลองชุดแสดงผลแบบจุดขนาด 5X7

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้ชุดแสดงผลแบบจุดทำงานได้
3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test08.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน วงจรแสดงผลแบบจุดขนาด 5×7 จะสร้างเป็นรหัสไบนารี่ ตั้งแต่ 0000 ถึง 1111 โดยแต่ละระดับของไบนารี่ให้สูงขึ้นหนึ่งระดับ ความเร็วในการแสดงผลสามารถปรับได้ที่การหน่วงเวลาในแต่ละการแสดงผลและเมื่อเราปรับความเร็วให้มากขึ้น วงจรแสดงผลแบบจุดขนาด 5×7 จะสามารถสว่างหมดทุกดวงได้

4.10 การทดลองชุดแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปรับสวิตช์ควบคุมการทำงานให้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก ทำงานและต่อกับ Ext. Analog Out

3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test09.Hex ไปยังซิงเกิ้ลบอร์ด

4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงานใช้ดิจิตอลโวลต์มิเตอร์วัดที่ขั้ว Ext. Analog Out จะมีโวลต์เตจปรากฏขึ้นและจะเพิ่มขึ้นเป็นขึ้นเป็นสเต็ปทุกๆ 3 วินาทีจนถึง 2.54 โวลต์ โดยแต่ละสเต็ปจะมีค่าแรงดันเท่ากับ 0.01 โวลต์ จำนวนสเต็ปทั้งหมด 256 สเต็ปจาก 00H-FFH

4.11 การทดลองชุดแสดงระดับสัญญาณและชุดปรับแรงดัน 0-2.55 โวลต์

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

2. ปรับสวิตช์ควบคุมการทำงานให้ POT ต่อกับวงจรแสดงระดับสัญญาณเท่านั้น

3. ทดลองหมุน POT ไป-มา

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราหมุน POT ไป-มา แล้วจะพบว่าชุดแสดงระดับสัญญาณจะมีการแสดงผลเปลี่ยนแปลงไป โดยหากหมุน POT มาทวนเข็มชุดกราฟจะไม่มีการแสดงผลใดๆ ทั้งสิ้น (หลอดไดโอดเปล่งแสงดับหมด) แต่ถ้าหากหมุน POT ไปตามเข็มเรื่อยๆ การแสดงผลของชุดแสดงระดับสัญญาณจะเปลี่ยนแปลง โดยทุกๆ ขีด (หลอดไดโอดเปล่งแสงดับหมด) ของวงจรแสดงระดับสัญญาณจะต้องมีค่าแรงดันเพิ่มขึ้น 0.25 โวลต์ และเมื่อหมุนไปตามเข็มสุด ชุดแสดงระดับสัญญาณจะสร้างครบทุกดวง

4.12 การทดลองชุดแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

2. ปรับสวิตช์ควบคุมการทำงานให้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลและวงจรแสดงผลข้อมูลทำงาน อินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล คือ POT

3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test11.Hex ไปยังซิงเกิ้ลบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน แล้วหมุน POT ไป-มาจะปรากฏการเปลี่ยนแปลงหลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูล โดยเมื่อหมุน POT ไปทวนจนสุด จะไม่มีการติดสว่างของหลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูล แต่หากปรับ POT ตามเข็มนาฬิกาไปเรื่อยๆ การเปลี่ยนแปลงของหลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูลจะติดเป็นเลขไบนารี่ระดับการเปลี่ยนแปลง 256 ระดับ ระดับละ 0.01 โวลต์ เมื่อหมุน POT ไปตามเข็มจนสุดหลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูลก็จะติดสว่างหมด

4.13 การทดลองชุดตรวจวัดความสว่างแสง

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลและชุดแสดงผลข้อมูลทำงาน อินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลคือวงจรตรวจวัดความสว่างแสง
3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test12.Hex ไปยังซิงเกิ้ลบอร์ด
4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน แล้วลองเอามือบังแสงที่ช่องรับแสงของตัวต้านทานเปลี่ยนแปลงตามแสง จะปรากฏการเปลี่ยนแปลงของหลอดไดโอดเปล่งแสงบนชุดแสดงผลข้อมูล โดยเมื่อนำนิ้วมือปิดช่องรับแสงไม่ให้มีแสงลอดผ่านลอดแสงได้ ผลที่เกิดกับหลอดไดโอดเปล่งแสงของชุดแสดงผลข้อมูลจะดับหมด แต่หากปิดช่องรับแสงให้รับแสงเต็มที่ ผลที่เกิดขึ้นบนชุดแสดงผลข้อมูลนั้นจะสร้างติดหมดทั้งดวง หรือบังแสงโดยให้มีแสงรอดไปได้บ้างการติดของหลอดไดโอดเปล่งแสงจะเท่ากับค่าแสงที่ผ่านไปได้

4.14 การทดลองชุดตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6
2. ปรับสวิตช์ควบคุมการทำงานให้วงจรตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับชุดสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสงทำงาน

3. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test13.Hexไปยังซิงเกิ้ลบอร์ด

4. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน ทดลองนำวัสดุทึบแสงไปบังลำแสงอินฟราเรดที่ส่งจากตัวส่งไปยังโฟโตไดโอด จะปรากฏว่าหลอดไดโอดเปล่งแสงจะสว่างขึ้นมาทันที แต่หากเรานำวัสดุนั้นออกจากทิศทางของลำแสงหลอดไดโอดเปล่งแสงจะดับ ดังนั้นสรุปได้ว่าหากมีวัสดุทึบแสงติดผ่านลำแสงอินฟราเรดหลอดไดโอดเปล่งแสงจะติด หากเคลื่อนผ่านไปแล้วหลอดไดโอดเปล่งแสงจะติด หากเคลื่อนผ่านไปแล้วหลอดไดโอดเปล่งแสงจะดับ

4.15 การทดลองคีย์แพด

ลำดับขั้นการทดลอง

1. ต่อชุดทดลองกับซิงเกิ้ลบอร์ดดังรูปที่ 4.6

2. ทำการโปรแกรมจากไดร์ฟ A ชื่อ Test14.Hexไปยังซิงเกิ้ลบอร์ด

3. สั่งให้โปรแกรมทำงาน

ผลการทดลอง

จากการทดลองพบว่าเมื่อเราสั่งให้โปรแกรมทำงาน แล้วทดลองกดคีย์ทั้ง 12 คีย์ดูสังเกตการเปลี่ยนแปลงบนชุดแสดงผลตัวเลข 7 ส่วน ของซิงเกิ้ลบอร์ดจะเป็นเลขตามที่เรากด ส่วนเครื่องหมาย * และ # จะแสดงเป็น ㄣ และ □

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป อภิปรายและข้อเสนอแนะ

5.1 บทสรุป

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 สร้างขึ้นเพื่อได้ศึกษาหลักการทฤษฎี รวมทั้งการประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ซึ่งมีขอบเขตของโครงงานนี้คือ

5.1.1 วงจรของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

1. วงจรชุดแสดงผลข้อมูลขนาด 8 บิต
2. วงจรควบคุมมอเตอร์กระแสตรง
3. วงจรแสดงผลตัวเลข 7 ส่วนขนาด 2 หลัก
4. วงจรชุดสร้างสัญญาณเสียงด้วยบัซเซอร์
5. วงจรชุดสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง
6. วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก
7. วงจรแสดงผลแบบจุดขนาด 5×7
8. วงจรแสดงระดับสัญญาณ
9. วงจรปรับแรงดันขนาด 0 ถึง 2.55 โวลต์
10. วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล
11. วงจรสร้างสภาวะลอจิกขนาด 8 บิต
12. วงจรตรวจวัดความสว่างของแสง
13. วงจรตรวจจับความเร็ว
14. วงจรควบคุมสเต็ปปิ้งมอเตอร์
15. วงจรคีย์แพดขนาด 4×3
16. วงจรผลิตแรงดันคงที่ +5 VDC
17. วงจรผลิตแรงดันคงที่ -5 VDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 ใบงานการทดลองทั้งหมด 21 ใบงาน

1. การโปรแกรม 8255
2. การเขียนโปรแกรมหน่วงเวลาสำหรับไมโครคอนโทรลเลอร์ 8051
3. การแสดงผลข้อมูลขนาด 8 บิต
4. การอ่านค่าจากสวิตช์อินพุต
5. การแสดงผลตัวเลข 7 ส่วน
6. การชุดสร้างสัญญาณเสียงด้วยบัซเซอร์
7. การสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง
8. การควบคุมมอเตอร์กระแสตรง
9. การควบคุมสเต็ปปิ้งมอเตอร์
10. การแสดงผลแบบจุดขนาด 5×7
11. การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก
12. การแสดงระดับสัญญาณและการปรับแรงดัน 0-2.55 โวลต์
13. การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล
14. การตรวจวัดความสว่างของแสง
15. ตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด
16. การรับข้อมูลจากคีย์แพด
17. การควบคุมความเร็วมอเตอร์แบบ Pulse With Modulation
18. การวัดค่าความเร็วของมอเตอร์
19. การส่งชุดคำสั่งผ่านช่องสื่อสารอนุกรม RS–232
20. การพัฒนาโปรแกรมด้วย EPROM Emulator
21. การโปรแกรม EPROM

โดยใบงานทั้งหมดนี้มีประโยชน์อย่างมาก ในการศึกษาและพัฒนาทางด้านไมโครคอนโทรลเลอร์ 8051 ต่อไป

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ปัญหาในส่วนของฮาร์ดแวร์

ปัญหาที่ 1 แรงดันจากวงจร POT ไม่ได้ตามค่า 0-2.55 โวลต์ อันเนื่องมาจากอุปกรณ์ที่ใช้ปรับค่ามีค่าความผิดพลาดมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางแก้ไข ใช้อุปกรณ์ที่นำมาใช้ปรับค่าให้มีค่าความผิดพลาดน้อยที่สุด

ปัญหาที่ 2 ความผิดพลาดของการออกแบบลายวงจรพิมพ์ ตำแหน่งเส้นอินพุตของสัญญาณแอนะลอกกับสัญญาณเอาต์พุตของอินฟราเรด (ที่ขาโฟโตไดโอด) โดยสองเส้นนี้จะติดกันอยู่ทางด้าน Bottom Layer ตำแหน่งรูยึดน็อตของมอเตอร์ไฟตรง

แนวทางแก้ไข ทำการตัดลายวงจรพิมพ์ส่วนสัญญาณเอาต์พุตของอินฟราเรด (ส่วนที่ออกมาจากโฟโตไดโอด) ออกเป็นสองส่วน โดยที่ทั้งสองไม่สัมผัสกับเส้นสัญญาณอินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล และใช้สายไฟขนาดเล็กเชื่อมต่อทั้งสองส่วนที่ถูกตัดเข้าด้วยกัน

ปัญหาที่ 3 ตำแหน่งขาของการใส่หลอดไดโอดเปล่งแสง เพื่อแสดงสถานะการหมุนของมอเตอร์ทั้งหมุนตามเข็มและทวนเข็มนาฬิกา ถ้าใส่ตามลายวงจรพิมพ์จะให้การแสดงทิศทางการหมุนของมอเตอร์ผิดพลาด

แนวทางแก้ไข การใส่ขาของหลอดไดโอดเปล่งแสงสลับกับในรูปที่เห็นบนแผ่นวงจรพิมพ์

ปัญหาที่ 4 ในการลงอุปกรณ์บางตัว ขนาดขาของอุปกรณ์จะมีขนาดใหญ่กว่ารูบนแผ่นวงจรพิมพ์ ทำให้ไม่สามารถใส่อุปกรณ์ลงไปได้

แนวทางแก้ไข นำขาของอุปกรณ์ที่มีขนาดใหญ่กว่ารูบนแผ่นวงจรพิมพ์มาตะไบหรือขัดให้มีขนาดที่สามารถใส่ลงไปในรูเพื่อให้บัดกรีได้

5.2.2 ปัญหาและแนวทางแก้ไขในส่วนของซอฟต์แวร์

ปัญหาที่ 1 การศึกษาการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ 8051 เป็นไปได้ช้า เนื่องจากเป็นเรื่องที่ยากมากพอควร

แนวทางแก้ไข ต้องใช้ระยะเวลาในการศึกษาและจัดทำทางด้านซอฟต์แวร์เป็นเวลานาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

1. การออกแบบวงจรสามารถเพิ่มเติมวงจรได้มากขึ้น เพราะสวิตช์ที่ใช้ในการควบคุมยังไม่ได้ถูกใช้งานอีกจำนวนหนึ่ง และสามารถเพิ่มเติมในส่วนของใบงานให้หลากหลายมากขึ้น

2. อุปกรณ์ที่ใช้ เช่น Potentiometer หรืออุปกรณ์ที่เกี่ยวกับทรานสดิวเซอร์นั้นควรมีค่าความความผิดพลาดน้อยที่สุด

3. ควรออกแบบแผ่นวงจรพิมพ์ให้มีขนาดเล็กกว่าเดิม เพื่อความประหยัดและสะดวกในการเคลื่อนย้ายและการใช้งาน

4. ควรเขียนโปรแกรมทดลองและโปรแกรมที่ใช้ในใบงาน ให้มีลักษณะการทำงานแบบเดิมแต่ใช้จำนวนคำสั่งน้อยกว่าเดิม

5. ควรเพิ่มเติมอุปกรณ์เชื่อมต่อกับซิงเกิ้ลบอร์ดให้อยู่บนชุดทดลอง เพื่อที่จะได้เชื่อมต่อกับซิงเกิ้ลบอร์ดได้หลายๆ บริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
โครงสร้างและคู่มือการใช้งานชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

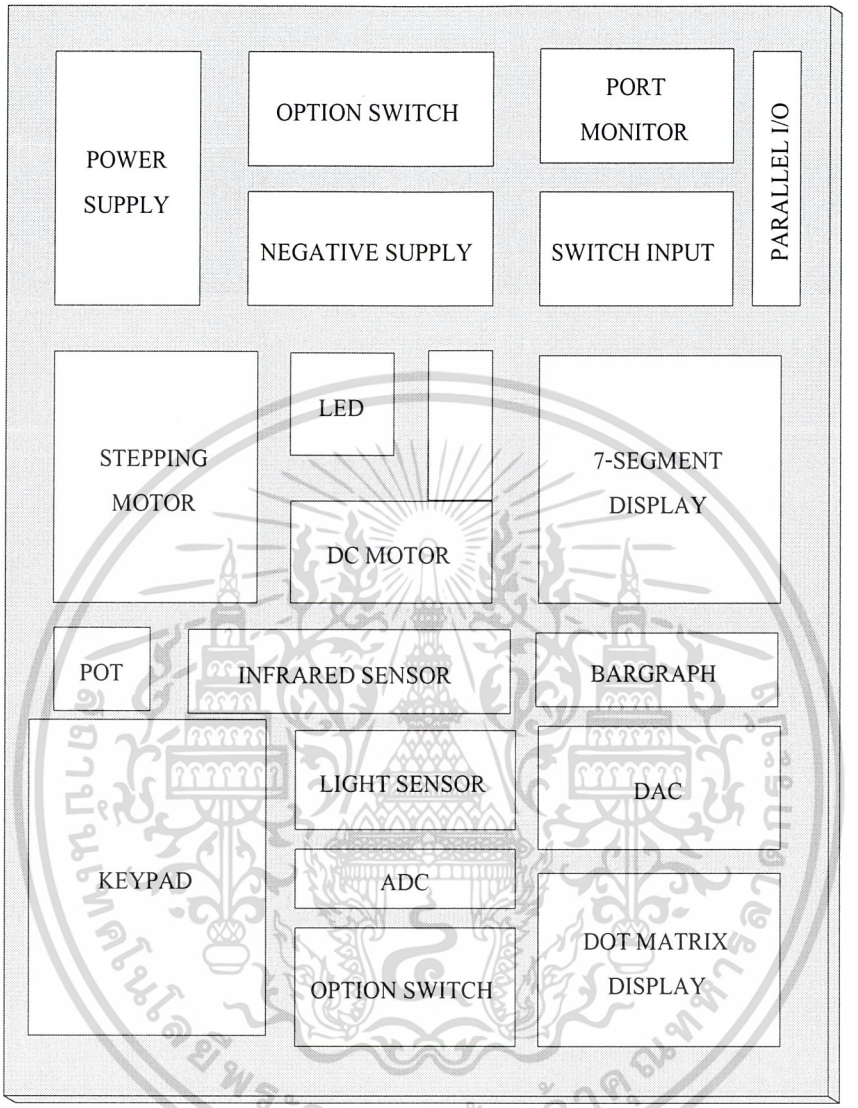
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างและคู่มือการใช้งานชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 เป็นชุดทดลองสำหรับทำการทดลองและควบคุมอุปกรณ์ต่างๆ ให้ทำงานโดยอาศัยซีพียูเป็นตัวควบคุม โดยซีพียูจะสั่งงานผ่านทางพอร์ตแบบขนาน (Parallel Port) ในที่นี้เราจะใช้ไอซี 8255 ซึ่งเป็นโปรแกรมเมเบิลอินพุต/เอาต์พุตพอร์ต (Programmable Peripheral I/O Port) เป็นตัวกลางระหว่างซีพียูกับชุดทดลอง จำนวนพอร์ตที่ใช้ จะใช้ทั้งหมด 3 พอร์ต หรือทั้งหมด 24 บิต คือ พอร์ต PA, พอร์ต PB และ พอร์ต PC

สำหรับซีพียูจะเป็นไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์ในตระกูล 68HC11, 8031, 8032, 8751, 8752, Z-80 ฯลฯ หรืออาจจะใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมก็สามารถที่จะทำได้ โดยใช้การเชื่อมต่อผ่านไอซี 8255 และได้ออกแบบให้ต่อกับพอร์ตของบริษัท ETT ที่เป็นแบบ 72IOZ80 แต่ก็สามารถที่จะต่อร่วมกับ User Port ของบริษัทอื่นๆ ได้ด้วย โดยมีกล่องสำหรับแปลงพอร์ตอีก 1 กล่อง นั่นคือ User Port ของบริษัท ANA-DIGI และ 8255 User ของบริษัท SILA หรือบริษัทอื่นๆ เพียงแต่ต้องจัดตำแหน่งของพอร์ตให้ตรงกันกับพอร์ตของชุดทดลองที่เป็นแบบ 72IOZ80 ของบริษัท ETT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ส่วนต่างๆ ของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

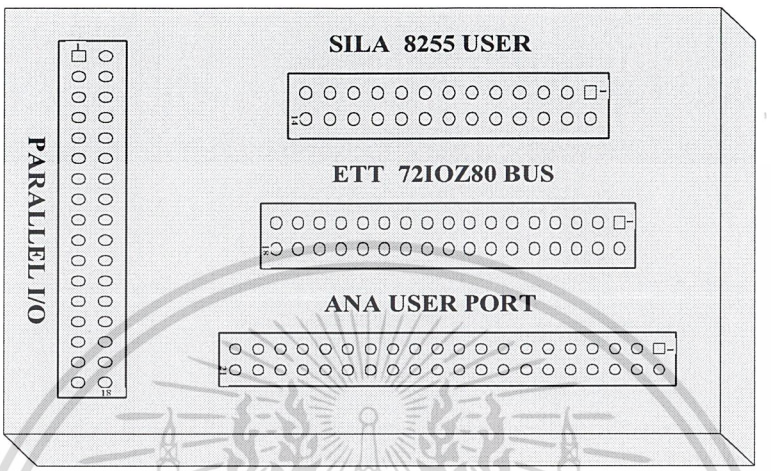
1.1 การต่อชุดทดลองเข้ากับซิงเกิ้ลบอร์ด

โดยปกติแล้วชุดทดลองจะสามารถต่อเข้ากับพอร์ตของ 8255 ของบริษัท ETT (72IOZ80) ได้เลย แต่เพื่อความสะดวกของผู้ใช้งาน หรือผู้ที่มีซิงเกิ้ลบอร์ดของบริษัทอื่น ก็สามารถต่อเข้ากับชุดทดลองได้ โดยต่อให้ตรงกับช่องของแต่ละพอร์ตผ่านทางกล่องแปลงพอร์ต

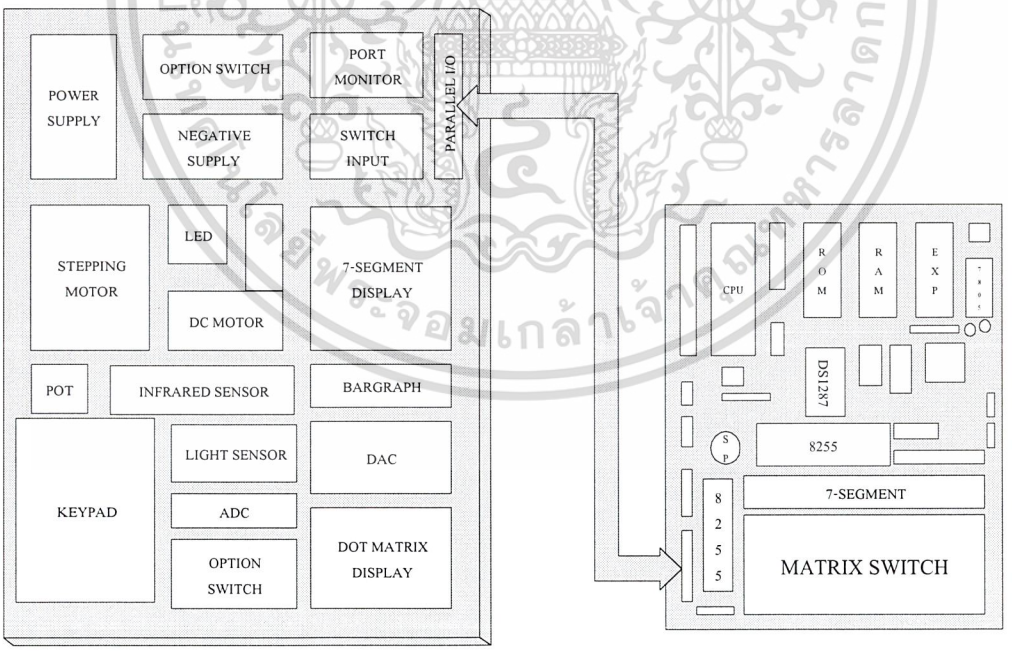
1. บริษัท ETT ใช้พอร์ต 8255 แบบ 72IOZ80 ขนาด 32 Pin
2. บริษัท SILA ใช้พอร์ต 8255 แบบ 8255 User ขนาด 26 Pin
3. บริษัท ANA-DIGI ใช้พอร์ต 8255 แบบ User Port ขนาด 40 Pin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งพอร์ตต่างๆ เหล่านี้จะประกอบด้วยพอร์ต PA_0-PA_7, พอร์ต PB_0-PB_7, พอร์ต PC_0-PC_7, Vcc และ Gnd

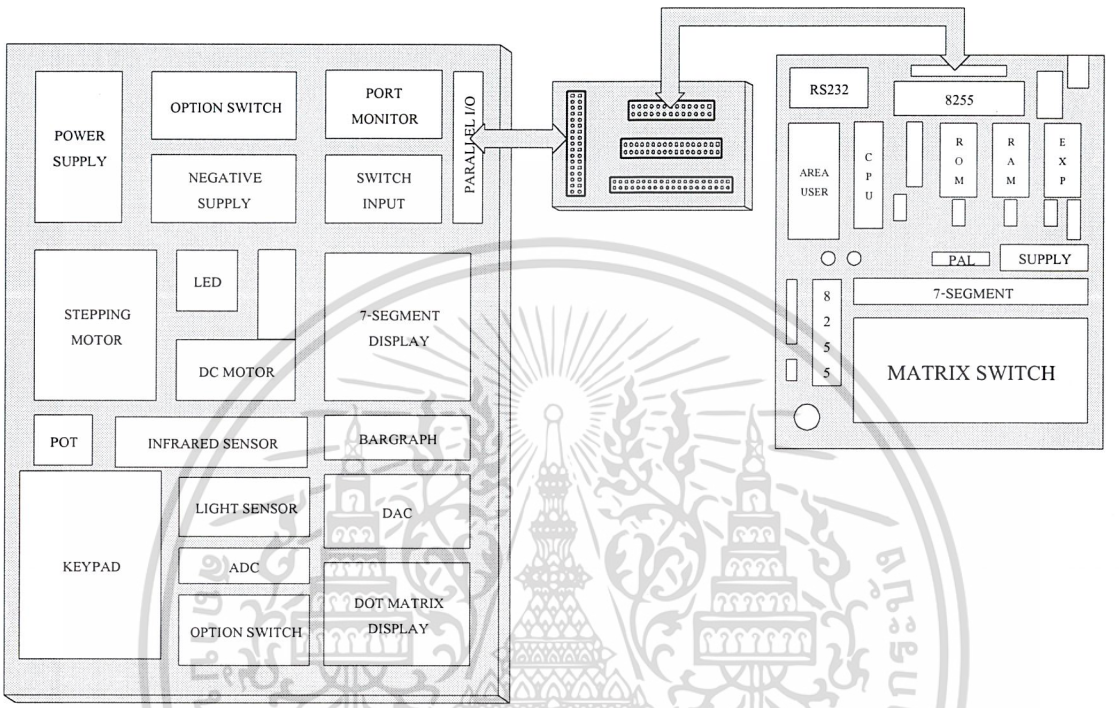


รูปที่ ก.2 กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ดบริษัทอื่น



รูปที่ ก.3 การเชื่อมต่อชุดทดลองเข้ากับซิงเกิ้ลบอร์ดรุ่น ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 การเชื่อมต่อชุดทดลองเข้ากับซิงเกิ้ลบอร์ดรุ่นอื่นๆ

วงจรของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จัดแบ่งตามการต่อใช้งานเข้ากับพอร์ตของ 8255 เป็น 3 ส่วนด้วยกันคือ

1.1.1 พอร์ต PA เป็นพอร์ตที่รับข้อมูลมาจาก 8255 เพื่อควบคุมการทำงานของวงจรต่างๆ ที่ต่อร่วมกันอยู่ ซึ่งประกอบด้วย

1. Port Monitor เป็นตัวแสดงสภาวะลอจิกของพอร์ต PA ทั้ง 8 บิต
2. DC Motor เป็นวงจรควบคุมการทำงานของมอเตอร์ให้หมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา
3. Electronic Buzzer เป็นตัวสร้างสัญญาณความถี่เสียง
4. LED เป็นตัวแสดงสภาวะลอจิกขนาด 1 บิต
5. 7-Segment Display เป็นตัวแสดงผลแบบตัวเลข 0-9 จำนวน 2 หลัก
6. Digital to Analog Converter แปลงสัญญาณดิจิตอลขนาด 8 บิตให้เป็นแรงดันไฟฟ้าขนาด 0-2.55 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Dot Matrix Display เป็นตัวแสดงผลขนาด 5×7 มีลักษณะเป็นหลอดกลมๆ เดี่ยวๆ สำหรับแสดงสัญลักษณ์หรือตัวอักษรต่างๆ
8. External Analog OUT เป็นตัวส่งสัญญาณแอนะล็อกขนาด 0-2.55 โวลต์ออกจากชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051
9. Bargraph Display เป็นวงจรที่ใช้ในการแสดงผลของระดับแรงดันไฟฟ้าขนาด 0-2.55 โวลต์ แบ่งระดับการแสดงผลเป็น 10 ระดับ

1.1.2 พอร์ต PB เป็นพอร์ตที่ส่งข้อมูลให้กับ 8255 โดยรับข้อมูลจากตัวเซนเซอร์ต่างๆ ผ่านทางวงจร แปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

1. Analog to Digital converter แปลงสัญญาณแอนะล็อกที่เป็นแรงดันไฟฟ้า 0-2.55 โวลต์ ให้เป็นสัญญาณดิจิตอลขนาด 8 บิต
2. Switch Input สำหรับสร้างสภาวะลอจิกขนาด 8 บิต
3. Light Sensor สำหรับตรวจวัดความสว่างของแสงให้เป็นแรงดันไฟฟ้า 0-2.55 โวลต์
4. Potentiometer (POT) เป็นวงจรสร้างแรงดัน 0-2.55 โวลต์
5. External Analog IN เป็นส่วนที่รับสัญญาณแอนะล็อกจากภายนอกขนาด 0-2.55 โวลต์ เข้ามายังชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

1.1.3 พอร์ต PC เป็นพอร์ตที่ทำหน้าที่ทั้งรับและส่งข้อมูลกับ 8255 ในเวลาเดียวกันจะแยกเป็นพอร์ต PC_0-PC_3 เป็นตัวรับข้อมูล และ PC_4-PC_7 เป็นตัวส่งข้อมูล

1. Infrared Sensor เป็นวงจรวัดความเร็วรอบของมอเตอร์ โดยอาศัยตัวอินฟราเรดและเป็นวงจรตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด
2. Stepping Motor เป็นวงจรควบคุมการหมุนของ Stepping Motor
3. Keypad ประกอบด้วยคีย์ตัวเลขจำนวน 10 ปุ่มกด (0-9) และคีย์พิเศษอีก 2 ปุ่ม (*, #)

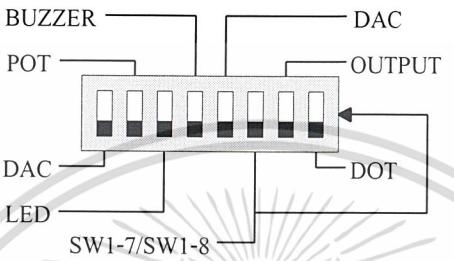
1.2 การปรับสวิตช์ควบคุมการทำงานของชุดทดลอง

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 เป็นชุดทดลองที่สามารถทำการทดลองในลักษณะแบบแยกทีละวงจร เพื่อศึกษารายละเอียดในแต่ละเรื่อง หรือทำการทดลองเป็นระบบโดยให้แต่ละวงจรทำงานร่วมกัน ดังนั้นจึงจำเป็นที่จะต้องมีการตัดต่อทางเดินของสัญญาณ หรือสิ่งที่ทำให้วงจรแต่ละส่วนทำงานแตกต่างกันออกไป ก่อนที่จะเริ่มการทดลองควรศึกษา รายละเอียดของสวิตช์ต่างๆ เหล่านี้ให้ดีเสียก่อน จะช่วยให้มีความเข้าใจ และทำการทดลองได้อย่างถูกต้องมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.1 สวิตช์ SW 1 ขนาด 8 ตำแหน่ง

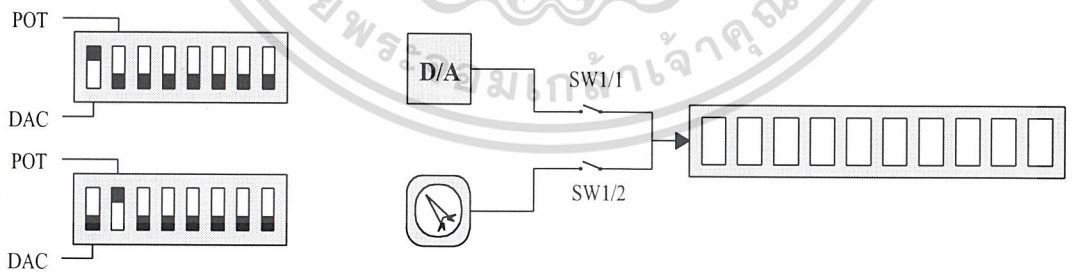
เป็นสวิตช์ควบคุมการต่อพอร์ต PA เข้ากับวงจรต่างๆ ว่าจะเลือกให้วงจรใดทำงาน ซึ่งประกอบด้วยสวิตช์ย่อยๆ อีก 8 ตัว โดยกำหนดเป็น SW 1/1 ถึง SW 1/8 ซึ่งสวิตช์แต่ละตัวก็จะควบคุมวงจรเพียงวงจรเดียวเท่านั้น



รูปที่ ก.5 รูปแบบของสวิตช์ SW 1

SW 1/1 เป็นสวิตช์เลือกสัญญาณอินพุตของวงจรแสดงระดับสัญญาณ (Bargraph) เมื่อสวิตช์นี้อยู่ที่ตำแหน่ง "ON" จะเป็นการรับสัญญาณจากวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

SW 1/2 เป็นสวิตช์เลือกสัญญาณอินพุตของวงจรแสดงระดับสัญญาณ (Bargraph) เช่นเดียวกันกับ SW 1/1 แต่แตกต่างกันที่ เมื่อสวิตช์นี้อยู่ที่ตำแหน่ง "ON" จะเป็นการรับสัญญาณจาก วงจรปรับแรงดัน 0-2.55 โวลต์

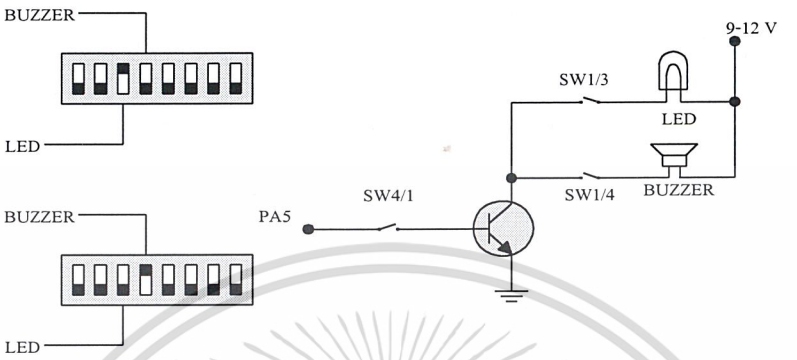


รูปที่ ก.6 การใช้งานสวิตช์ SW 1/1 และ SW 1/2

SW 1/3 เป็นสวิตช์เลือกให้ LED ทำงาน เมื่อให้สวิตช์นี้อยู่ที่ตำแหน่ง "ON" วงจรจะถูกควบคุมโดยบิตที่ 5 ของพอร์ต PA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

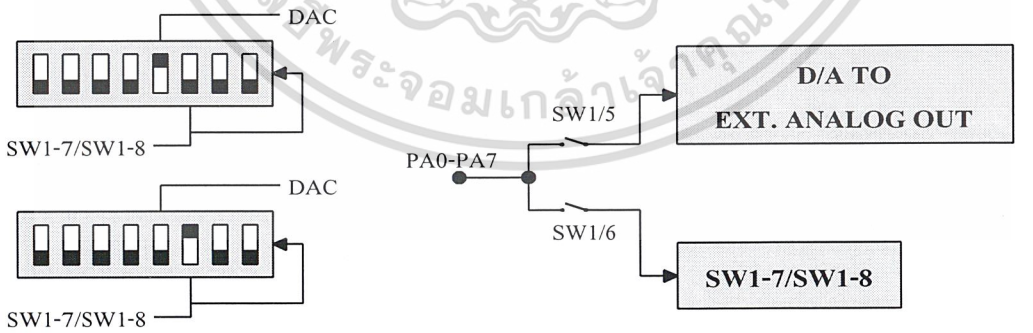
SW 1/4 เป็นสวิตช์เลือกให้บัซเซอร์ทำงาน เมื่อให้สวิตช์นี้อยู่ที่ตำแหน่ง “ON” บัซเซอร์จะถูกควบคุมโดยบิตที่ 5 ของพอร์ต PA เช่นเดียวกับ LED



รูปที่ ก.7 การใช้งานสวิตช์ SW 1/3 และ SW 1/4

SW 1/5 เป็นสวิตช์เลือกให้เอาต์พุตของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกต่อเข้ากับ External Analog OUT เพื่อควบคุมอุปกรณ์ภายนอกชุดทดลอง เมื่อสวิตช์นี้อยู่ที่ตำแหน่ง “ON”

SW 1/6 เป็นสวิตช์เลือกให้เอาต์พุตของพอร์ต PA ต่อเข้ากับวงจรที่กำหนดจาก SW 1/7 และ SW 1/8 เมื่อสวิตช์ SW 1/6 อยู่ที่ตำแหน่ง “ON” เป็นการควบคุม IC2 เบอร์ 74LS244 หรือ ควบคุม IC8 เบอร์ 74LS244

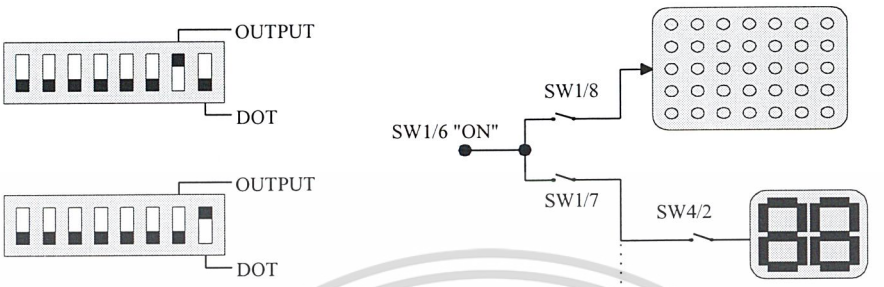


รูปที่ ก.8 การใช้งานสวิตช์ SW 1/5 และ SW 1/6

SW 1/7 เป็นสวิตช์เลือกให้พอร์ต PA ต่อเข้ากับ IC2 เบอร์ 74LS244 เพื่อให้พอร์ต PA เป็นข้อมูลของวงจรควบคุมมอเตอร์กระแสตรง, วงจรสร้างสัญญาณเสียงด้วยบัซเซอร์, วงจรสร้างสัญญาณแสดงผลด้วย LEDและวงจรแสดงผลตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SW 1/8 เป็นสวิตช์เลือกว่าให้พอร์ต PA ต่อเข้ากับ IC8 เบอร์ 74LS244 เมื่อสวิตช์นี้อยู่ที่ตำแหน่ง “ON” เพื่อให้พอร์ต PA เป็นข้อมูลของวงจรแสดงผลแบบจุดขนาด 5X7



รูปที่ ก.9 การใช้งานสวิตช์ SW 1/7 และ SW 1/8

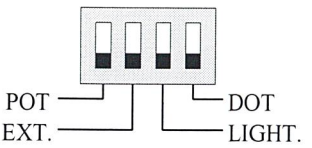
หมายเหตุ การกำหนดเลือกใช้สวิตช์ควรเลือกสวิตช์ที่ใช้ตามลำดับขั้นตอนการทดลองในใบงานเท่านั้น เพื่อผลการทดลองที่ถูกต้อง และป้องกันอุปกรณ์เสียหาย

1.2.2 สวิตช์ SW 2 ขนาด 4 ตำแหน่ง

เป็นสวิตช์เลือกสัญญาณอินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลว่าจะเลือกสัญญาณแอนะล็อกมาจากแหล่งใด ซึ่งแต่ละแหล่งก็จะให้สัญญาณเป็นแรงดันออกมาอยู่ในช่วง 0-2.55 โวลต์ SW 2 เป็นสวิตช์เลือกเช่นเดียวกันกับสวิตช์ SW 1 แต่มีขนาดเล็กกว่าเพียง 4 ตำแหน่งเท่านั้น เมื่อต้องการเลือกสัญญาณแอนะล็อกจากแหล่งใดก็ตาม ให้เลื่อนสวิตช์นั้นไปอยู่ที่ตำแหน่ง “ON” โดยสวิตช์ SW 2 จะใช้งานก็ต่อเมื่อสวิตช์ SW 3 อยู่ที่ตำแหน่ง ADC เท่านั้น

1. SW 2/1 เมื่ออยู่ที่ตำแหน่ง “ON” จะเป็นการเลือกสัญญาณแอนะล็อกจากวงจรปรับแรงดัน 0-2.55 โวลต์
2. SW 2/2 เมื่ออยู่ที่ตำแหน่ง “ON” จะเป็นการเลือกสัญญาณแอนะล็อกจาก External Analog IN
3. SW 2/3 เมื่ออยู่ที่ตำแหน่ง “ON” จะเป็นการเลือกสัญญาณแอนะล็อกจากวงจรตรวจวัดความสว่างแสง
4. SW 2/4 ไม่มีการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.10 การใช้งานสวิตช์ SW 2

หมายเหตุ ในการเลือกสัญญาณแอนะล็อกด้วยสวิตช์ SW 2 ควรเลือกสัญญาณแอนะล็อกเพียงสัญญาณเดียวเท่านั้น เพื่อการทดลองที่ถูกต้องและป้องกันอุปกรณ์เสียหาย

1.2.3 สวิตช์ SW 3 เป็นสวิตช์เลือก 2 ทาง

สวิตช์ SW 3 เป็นสวิตช์เลือกการทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลหรือวงจรแสดงสภาวะลอจิกโดยหากวงจรใดไม่ได้เลือกให้ทำงาน เอาต์พุตของวงจรนั้นจะมีสภาวะเป็น High Impedance

รูปที่ ก.11 การใช้งานสวิตช์ SW 3

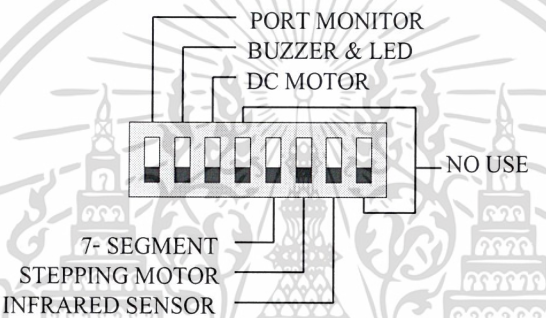
1.2.4 สวิตช์ SW 4 ขนาด 8 บิต

สวิตช์ SW 4 จะต่างจากสวิตช์ SW 1, SW 2, SW 3 เพราะสวิตช์ SW 4 จะเป็นตัวควบคุมการทำงานของแต่ละวงจรอีกทีหนึ่ง ถึงแม้ว่าเราจะปรับสวิตช์ SW 1, SW 2 และ SW 3 มายังตำแหน่งของวงจรที่เราต้องการแล้วก็ตาม แต่ถ้าไม่ปรับสวิตช์ SW 4 ให้วงจรนั้นๆ ทำงาน การควบคุมก็ไม่สามารถที่จะทำได้ เหตุผลที่ต้องมี SW 4 เพราะแต่ละวงจรจะต่อใช้งานพอร์ตร่วมกันอยู่ ดังนั้นวงจรที่ยังไม่จำเป็นต้องใช้ก็สามารถที่จะสั่งให้หยุดทำงานไปก่อนได้ ทำให้ไม่เกิดความสับสนในขณะที่ทำการทดลอง

สวิตช์ SW 4 แบ่งออกเป็นสวิตช์ย่อยๆ จำนวน 8 ตัวด้วยกัน คือ SW 4/1 ถึง SW 4/8 เมื่อเราให้สวิตช์ย่อย ณ. ตำแหน่งใดอยู่ที่ตำแหน่ง “ON” วงจรนั้นก็จะทำงาน การปรับสวิตช์ย่อยนี้ต้องปรับให้สวิตช์ย่อยอยู่ที่ตำแหน่ง “ON” เพียงตำแหน่งเดียวหรือตามแต่ในใบงานจะกำหนดมาเท่านั้น เพื่อป้องกันความสับสนขณะทำการทดลองและป้องกันอุปกรณ์เสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สวิตช์ SW 4/1 เป็นสวิตช์เลือกใช้วงจร Port Monitor ทำงาน
2. สวิตช์ SW 4/2 เป็นสวิตช์เลือกใช้วงจร LED หรือ Buzzer ทำงาน
3. สวิตช์ SW 4/3 เป็นสวิตช์เลือกใช้วงจร DC Motor ทำงาน
4. สวิตช์ SW 4/4 ไม่มีการใช้งาน
5. สวิตช์ SW 4/5 เป็นสวิตช์เลือกใช้วงจร 7-Segment Display ทำงาน
6. สวิตช์ SW 4/6 เป็นสวิตช์เลือกใช้วงจร Stepping Motor ทำงาน
7. สวิตช์ SW 4/7 เป็นสวิตช์เลือกใช้วงจร Infrared Sensor ทำงาน
8. สวิตช์ SW 4/8 ไม่มีการใช้งาน

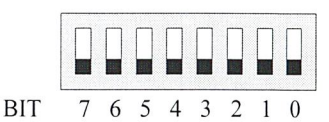


รูปที่ ก.12 การใช้งานสวิตช์ SW 4

หมายเหตุ จะเห็นได้ว่าทั้งสวิตช์ SW 1, SW 2, SW 3 และ SW 4 แต่ละตัวล้วนมีความสำคัญด้วยกันทุกตัว ดังนั้นการทดลองทุกๆ การทดลอง จะประสบผลสำเร็จหรือไม่นั้น ไม่ขึ้นอยู่กับการเขียนโปรแกรมเพียงอย่างเดียว แต่การปรับสวิตช์ควบคุมเหล่านี้ให้ถูกต้องก็มีความสำคัญเช่นกัน

1.2.5 สวิตช์ SW 5 ขนาด 8 ตำแหน่ง

สวิตช์ SW 5 เป็นสวิตช์สร้างสภาวะลอจิกขนาด 8 บิต ให้กับพอร์ต PB โดยการสร้างสภาวะนี้ขึ้นอยู่กับการเลือกสวิตช์ SW 3 ด้วย



รูปที่ ก.13 ลักษณะตำแหน่งของสวิตช์ SW 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.6 สวิตช์ SW 6 กดติด-ปล่อยดับ

สวิตช์ SW 6 เป็นสวิตช์แบบกดติด-ปล่อยดับ ใช้สำหรับทำการรีเซ็ต IC เบอร์ ADC0804 ซึ่งเป็นไอซีแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล สวิตช์นี้จะกดเมื่อการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลเกิดการผิดเพี้ยนหรือไม่ทำงาน

1.2.7 สวิตช์ SW 7 กดติด-กดดับ

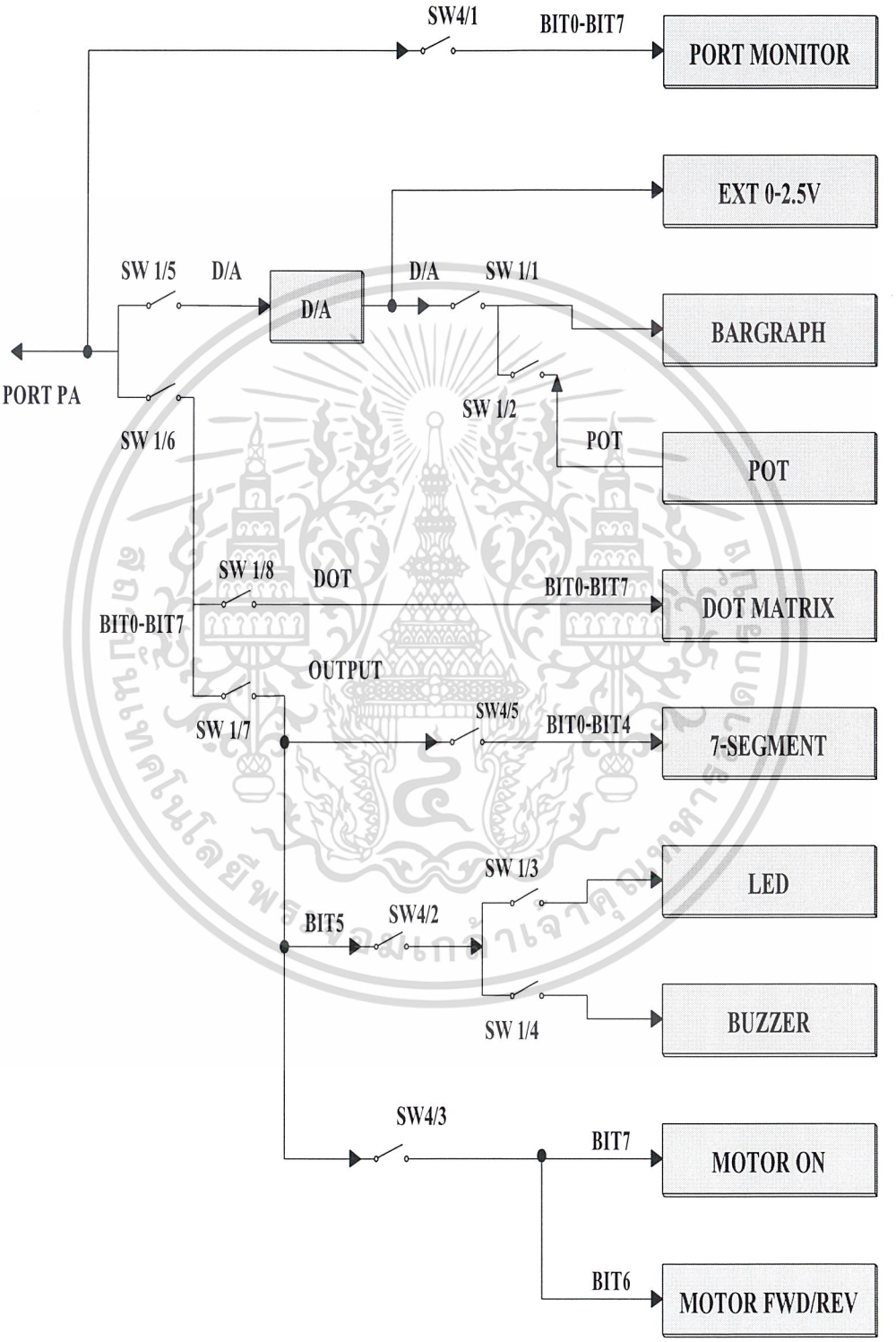
สวิตช์ SW 7 เป็นสวิตช์เปิด-ปิด สำหรับจ่ายไฟเลี้ยงให้กับชุดทดลอง (ก่อนทำการทดลองควรจะอยู่ที่ตำแหน่ง "OFF" และเมื่อต้องการทำการทดลองให้กดมายังตำแหน่ง "ON")

1.3 การปรับแต่งชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 มีทริมพ็อต (Trimpot) อยู่จำนวน 3 ตัวด้วยกันคือ VR1, VR3 และ VR4 ส่วน VR2 เป็น Potentiometer สำหรับปรับแรงดัน 0-2.55 โวลต์ เพื่อเป็นอินพุตให้แก่วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล ซึ่งทริมพ็อตแต่ละตัวจะถูกออกแบบให้กับวงจรที่ต่อร่วมด้วยทำงานได้อย่างถูกต้อง แต่ถ้าหากมีการคลาดเคลื่อนเกิดขึ้นก็สามารถปรับแต่งใหม่ เพื่อให้วงจรที่ต่อร่วมด้วยทำงานได้อย่างถูกต้อง

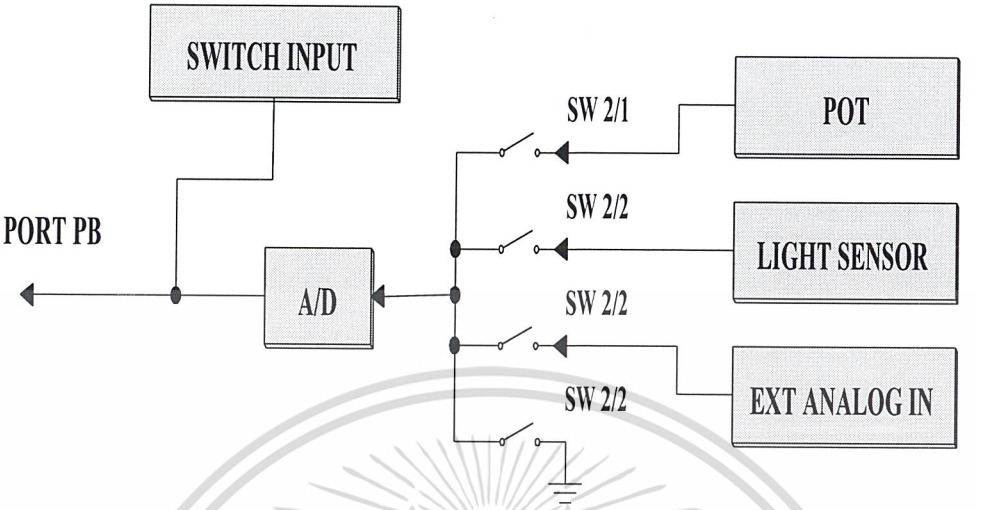
1. VR1 ค่า 5 กิโลโอห์ม อยู่ในส่วนของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะ-ล็อก การปรับให้ส่งข้อมูลเข้ามายังวงจรนี้เป็น 0FFH แล้วใช้โวลต์มิเตอร์วัดแรงดันที่ External Analog OUT ปรับให้ได้แรงดัน 2.55 โวลต์
2. VR3 ค่า 50 กิโลโอห์ม อยู่ในส่วนของวงจรตรวจวัดความสว่างแสง การปรับให้ใช้โวลต์มิเตอร์วัดที่ขา 6 ของ IC11 เบอร์ 741 แล้วปรับให้ได้แรงดัน 0 โวลต์ เมื่อใช้นิ้วปิดที่ช่องรับแสงของ LDR
3. VR4 ค่า 10 กิโลโอห์ม อยู่ในส่วนของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลการปรับให้ใช้โวลต์มิเตอร์วัดที่ขา 9 ของ IC12 เบอร์ ADC0804 ปรับแรงดันให้ได้ขนาด 1.275 โวลต์ หรือป้อนแรงดันเข้าที่อินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลขนาด 2.55 โวลต์ แล้วบิตที่ 0 ถึง บิตที่ 7 ของเอาต์พุตวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลต้องเป็น "1" ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

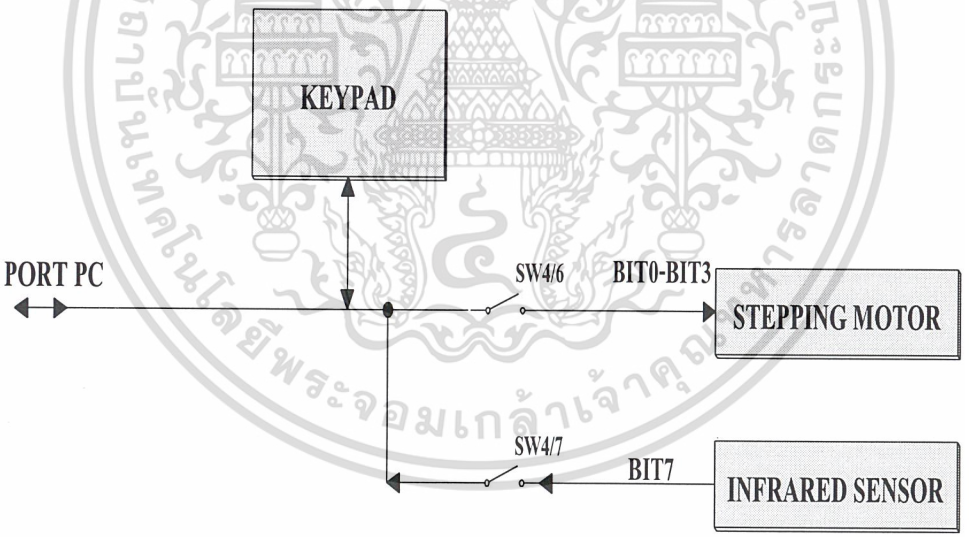


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.14 แผนผังการทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051



รูปที่ ก.14 (ต่อ) แผนผังการทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051



รูปที่ ก.14 (ต่อ) แผนผังการทำงานของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1

การโปรแกรม 8255 (Programmable Peripheral Interface)

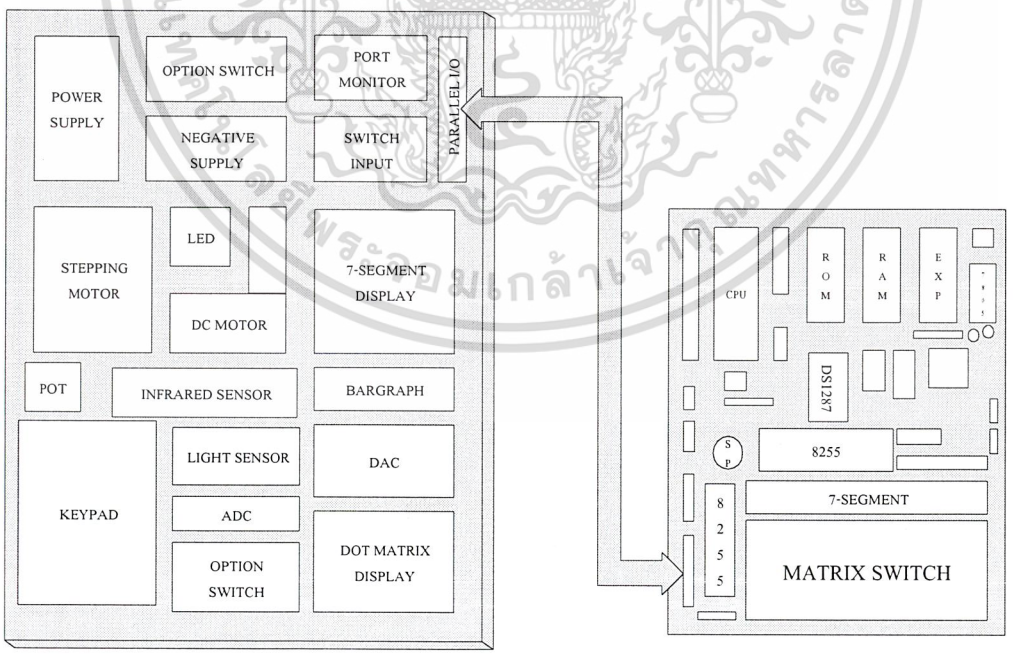
วัตถุประสงค์

1. เพื่อศึกษาการเขียน Command Word ให้กับ 8255
2. เพื่อสามารถกำหนด Command Word ให้ถูกต้องกับการใช้งานของชุดทดลองเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 1 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 1 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การติดต่อสั่งงานของไมโครคอนโทรลเลอร์ 8051 กับชุดทดลอง จะสั่งผ่านมาทาง 8255 PPI (Programmable Peripheral Interface) ซึ่งทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุต 8255 เป็นไอซีประเภท LSI (Large Scale Integrated Circuit) บรรจุอยู่ใน Package แบบ DIP (Dual In-line Package) เป็นไอซีที่สามารถโปรแกรมได้ ซึ่งถูกสร้างขึ้นมาใช้งานกับ 8080 แต่ในภายหลังได้มีการนำเอา 8255 ไปประยุกต์ใช้งานกับ ไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์เบอร์ต่างๆ เช่น Z-80, MCS-48, MCS-51 เป็นต้น

จากที่ได้ศึกษามาแล้วว่า 8255 ประกอบด้วยพอร์ต 4 พอร์ต คือ

1. พอร์ต A มีขนาด 8 บิต (PA_0-PA_7)
2. พอร์ต B มีขนาด 8 บิต (PB_0-PB_7)
3. พอร์ต C มีขนาด 8 บิต (PC_0-PC_7) สามารถที่จะแบ่ง เป็น 2 กลุ่มอีกคือ
   3.1. 4 บิตล่าง คือ (PC_0-PC_3)
   3.2. 4 บิตบนคือ (PC_4-PC_7)
4. พอร์ต Control มีหน้าที่ในการควบคุมหน้าที่การทำงานของพอร์ต A, พอร์ต B และพอร์ต C

ดังนั้นการใช้งานของ 8255 จึงจำเป็นต้องมีการเขียนคำสั่งควบคุม (Command Word) ส่งมาที่พอร์ตควบคุมทุกครั้งก่อนที่จะมีการใช้งาน 8255 ซึ่งคำสั่งนี้จะคงอยู่ตลอดไป จนกว่าจะมีการรีเซ็ต (Reset) เมื่อกำหนดรูปแบบการใช้งานให้กับ 8255 ได้แล้ว ก็สามารถที่จะนำเอาพอร์ต PA, PB และ PC มาใช้งานเป็นอินพุตและเอาต์พุตตามที่ต้องการได้

สำหรับชุดทดลองนี้ ได้กำหนดไว้ว่า พอร์ต PA ให้เป็นอินพุต, พอร์ต PB เป็นเอาต์พุต, พอร์ต PC_0-PC_3 เป็นอินพุต และพอร์ต PC_4-PC_7 เป็นเอาต์พุต ดังนั้นจึงจะต้องกำหนดพอร์ตของ 8255 ให้ทำงานตรงกันกับชุดทดลองคือ ให้พอร์ต ของ 8255 เป็นตามนี้ พอร์ต PA เป็นเอาต์พุต, พอร์ต PB เป็นอินพุต, พอร์ต PC_0-PC_3 เป็นเอาต์พุต และพอร์ต PC_4-PC_7 เป็นอินพุต ดังนั้นจึงต้องสั่งให้ 8255 ที่อยู่บน Single Board (ET-8032 Version 2.0) หรือบอร์ดไมโครคอนโทรลเลอร์รุ่นอื่นๆ ที่มี 8255 อยู่ ให้แต่ละพอร์ตมีการทำงานให้ตรงกับที่กำหนดไว้ ถึงสามารถควบคุมการทำงานของชุดทดลองได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข. 1 การกำหนดค่าให้กับพอร์ตต่างๆ ที่ใช้ในชุดทดลองและซิงเกิ้ลบอร์ด

พอร์ต	บอร์ดชุดทดลอง	8255	พอร์ตของซิงเกิ้ลบอร์ด
PA	อินพุต	เอาต์พุต	E020H*
PB	เอาต์พุต	อินพุต	E021H*
PC_0-PC_3	อินพุต	เอาต์พุต	E022H*
PC_4-PC_7	เอาต์พุต	อินพุต	E022H*
Control Port	-	-	E023H*

*ขึ้นอยู่กับการกำหนดหมายเลขพอร์ตให้กับ 8255 บนซิงเกิ้ลบอร์ด

การกำหนดการทำงานของพอร์ตต่างๆ ใน 8255 จะแบ่งการทำงานออกเป็น 3 โหมดคือ

1. โหมด 0 เป็นอินพุต-เอาต์พุตเบื้องต้น (Basic input/output)
2. โหมด 1 เป็นอินพุต-เอาต์พุตแบบสโตรบ (Strobe input/output)
3. โหมด 2 เป็นบัสสองทาง (Bidirection Bus)

ซึ่งในชุดทดลองนี้ใช้เฉพาะโหมด 0 เท่านั้น (ดังนั้นรายละเอียดจะไม่กล่าวถึง สามารถที่จะศึกษาได้จาก คู่มือของ 8255) ในการควบคุม 8255 จะต้องส่งรหัสควบคุม (Command Word) ไปยังคอนโทรลพอร์ตเพื่อควบคุมการทำงานของ 8255 ว่าให้มีการทำงานในโหมดใด แต่ละพอร์ตเป็นอินพุตหรือเอาต์พุต

ความหมายของบิตต่างๆ ของรหัสควบคุม

D7 แสดงถึงรหัสควบคุมให้เริ่มการทำงาน (“1” ทำงาน) คือถ้าทำให้บิตนี้เป็น “1” จึงจะมีผลทำให้ 8255 รับรู้สิ่งต่อไปในบิตอื่นๆ ที่กำหนดให้ ดังนั้นเวลาจะสั่งงานหรือกำหนดหน้าที่ให้กับ 8255 บิตนี้จะต้องเป็น “1” เสมอ

D6 และ D5 เป็นการเลือกโหมดในการทำงานของพอร์ตในกลุ่ม A ซึ่งมี 3 โหมด

00 โหมด 0

01 โหมด 1

XX โหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D4 กำหนดให้พอร์ต PA เป็นอินพุตหรือเอาต์พุตโดย

"0" เป็นเอาต์พุตพอร์ต

"1" เป็นอินพุตพอร์ต

D3 กำหนดให้พอร์ต PC บน (PC_4-PC_7) เป็นอินพุตหรือเอาต์พุตโดย

"0" เป็นเอาต์พุตพอร์ต

"1" เป็นอินพุตพอร์ต

D2 เป็นการเลือกโหมด ในการทำงานให้กับพอร์ตในกลุ่ม B ซึ่งมี 2 โหมดคือ

"0" เป็นโหมด 0

"1" เป็นโหมด 1

D1 กำหนดให้พอร์ต PB เป็นอินพุตหรือเอาต์พุต โดย

"0" เป็นเอาต์พุตพอร์ต

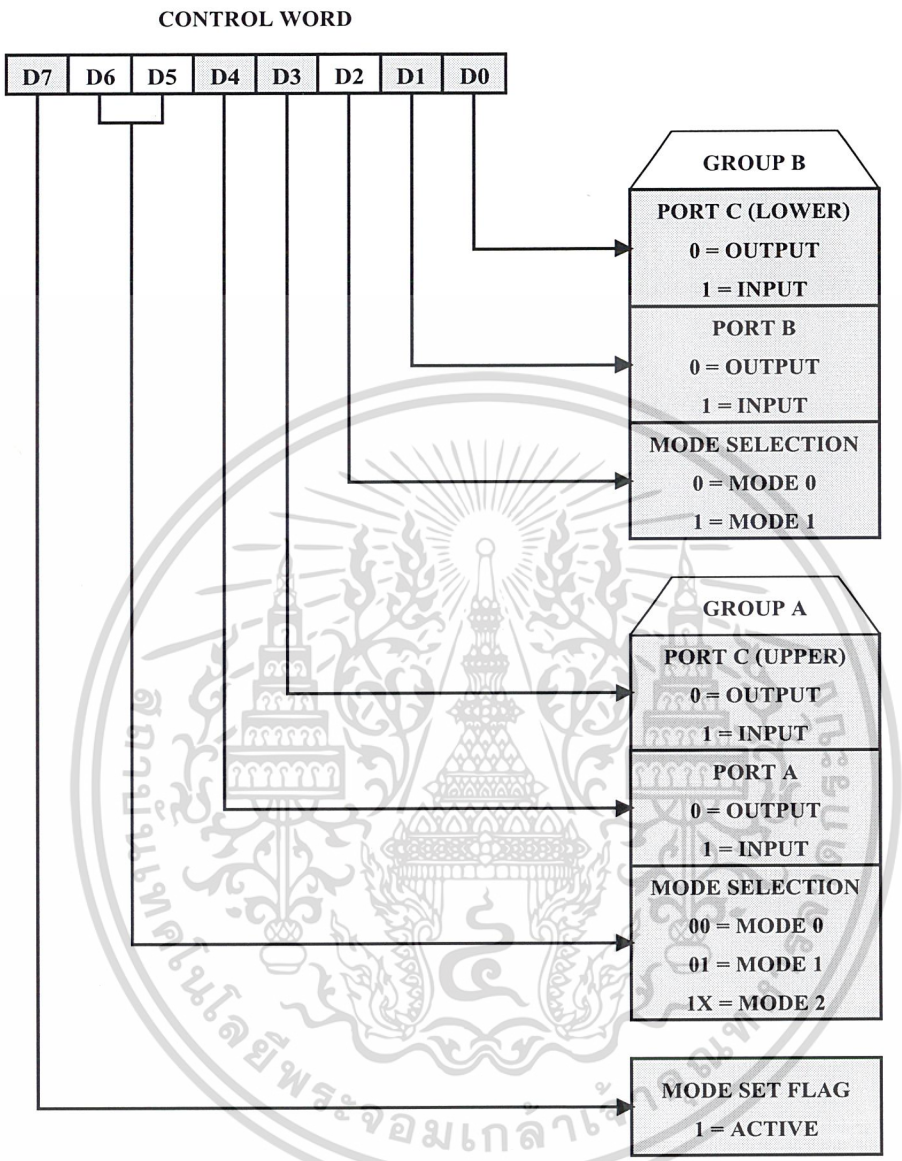
"1" เป็นอินพุตพอร์ต

D0 กำหนดให้พอร์ต PC ล่าง (PC_0-PC_3) เป็นอินพุตหรือเอาต์พุตโดย

"0" เป็นเอาต์พุตพอร์ต

"1" เป็นอินพุตพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 2 การกำหนด Command Word ให้กับ 8255

จากข้อกำหนดการใช้งานของพอร์ตบนบอร์ด นำมาเขียนเป็น Command Word ได้เป็น 8AH

BIT 7							BIT 0
1	0	0	0	1	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉะนั้นทุกครั้งที่จะต้องมีการเขียนโปรแกรมติดต่อกับชุดทดลองในส่วนต้นของโปรแกรมต้องกำหนด Command Word ให้กับ 8255 หรือที่เรียกว่า "Initial" ซึ่งการ Initial 8255 จะทำเพียงครั้งแรกเท่านั้น ถ้าโปรแกรมมีการทำงานซ้ำก็สามารถที่จะข้ามการ Initial ในตอนแรกได้

ในกรณีที่โปรแกรมของเราเริ่มต้นที่ ADDR 0000H ก่อนที่จะมีการ Initial ให้กับ 8255 ที่ส่วนต้นของโปรแกรม จำเป็นต้องมีการหน่วงเวลาก่อน เพื่อให้ 8255 ทำการรีเซ็ตตัวเองและพร้อมรับคำสั่งต่อไป

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "OFF"
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง "OFF" ยกเว้น SW 1/2 "ON"
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "ON"
5. ป้อนโปรแกรมที่ 1.1 ลงบนซิงเกิ้ลบอร์ดแล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 1.1 การ Initial 8255

```
                    ORG   8000H
8000 : 9E023  INIT: MOV   DPTR,#0E023H
8003 : 748A         MOV   A,#08AH       ; กำหนดค่า Command Word = 8AH
8005 : F0           MOVX  @DPTR,A       ; ส่งค่าไปยัง Control Port ของ 8255
8006 : 80FE         SJMP  $
                    END
```

6. ทดลองเปลี่ยน Command Word เป็นดังนี้ พอร์ต PA เป็นอินพุต, พอร์ต PB เป็นอินพุต, พอร์ต $PC_0 - PC_3$ เป็นเอาต์พุต และพอร์ต $PC_4 - PC_7$ เป็นเอาต์พุต
7. ทดลองเปลี่ยน Command Word เป็นดังนี้ พอร์ต PA เป็นอินพุต, พอร์ต PB เป็นเอาต์พุต, พอร์ต $PC_0 - PC_3$ เป็นอินพุต และพอร์ต $PC_4 - PC_7$ เป็นเอาต์พุต
8. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

9. จงอธิบายการทำงานของโปรแกรมในข้อที่ 3 และข้อที่ 4
10. จงอธิบายการกำหนดค่า Command Word ให้กับ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

การเขียนโปรแกรมหน่วงเวลาสำหรับ MCS-51

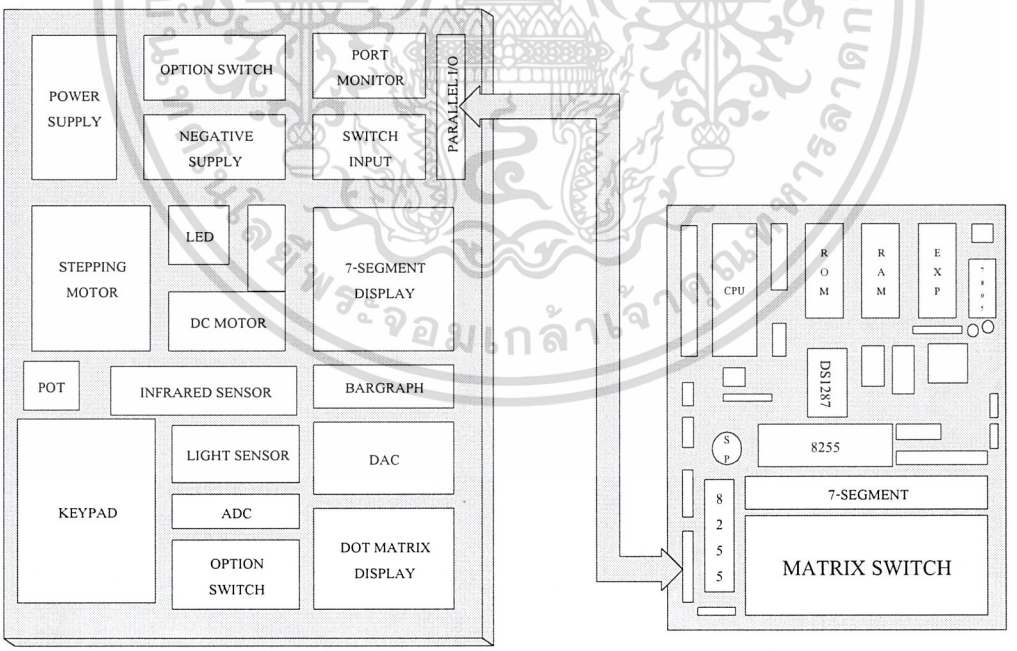
วัตถุประสงค์

1. เพื่อศึกษาการเขียนโปรแกรมหน่วงเวลาสำหรับ MCS-51
2. เพื่อสามารถกำหนดการหน่วงเวลาให้เหมาะกับโปรแกรมแต่ละโปรแกรมได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 2 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 3 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การติดต่อสั่งงานของไมโครคอนโทรลเลอร์ 8051 กับชุดทดลอง โดยใช้โปรแกรมที่พัฒนาขึ้นมาเองนั้น บางที่การแสดงผลที่ชุดทดลอง ผู้ทดลองต้องการให้ผลลัพธ์ที่แสดงนั้นมีความชัดเจนและไม่เร็วจนไม่สามารถที่จะสังเกตเห็นการเปลี่ยนแปลงได้ หรือต้องการศึกษาผลของคำสั่งแต่ละคำสั่งว่าเมื่อไมโครคอนโทรลเลอร์กระทำคำสั่งนั้นๆ ไปแล้วผลที่เกิดขึ้นจะเป็นอย่างไร

โปรแกรมที่สามารถทำให้ไมโครคอนโทรลเลอร์เสียเวลาทำงานมากที่สุด และมีผลให้ผลลัพธ์ของการกระทำคำสั่งที่กระทำก่อนหน้านี้ ต้องแสดงค้างไว้นั้นก็คือ โปรแกรมหน่วงเวลา โปรแกรมส่วนนี้มีความจำเป็นอย่างยิ่งในงานด้านต่างๆ เช่น Scankey and Display สามารถที่จะเขียนและพัฒนาขึ้นมาได้หลายรูปแบบด้วยกัน ใช้หลักการของการทำงานเป็นวงรอบ รอจนถึงเวลาที่ต้องการแล้วออกจากการทำงานในวงรอบนั้น ในการเขียนโปรแกรมในส่วนนี้สามารถเขียนเป็นโปรแกรมย่อยได้เลย และสามารถเรียกใช้งานได้เหมือนฟังก์ชันๆ หนึ่ง

1. โปรแกรมหน่วงเวลาแบบ Loop เดียว

จะใช้รีจิสเตอร์ตัวเดียวเป็นตัวกำหนดค่าเริ่มต้น แล้วใช้วิธีลดค่าลงหนึ่งแล้วทำการตรวจสอบถ้าไม่เท่ากับศูนย์ ให้ทำการลดค่าลงอีกครั้งแล้วทำการตรวจสอบใหม่จนกว่าค่าในรีจิสเตอร์นั้นจะมีค่าเป็นศูนย์ ในการเขียนโปรแกรมในส่วนนี้ได้กล่าวไปแล้วว่าควรจะเขียนให้อยู่ในรูปแบบของโปรแกรมย่อย ดังนั้น ต้องปิดท้ายโปรแกรมหน่วงเวลานี้ด้วยคำสั่ง RET (Return) เสมอดังตัวอย่าง

ตัวอย่าง โปรแกรมหน่วงเวลาที่มีเวลาหน่วงน้อยที่สุด โดยใช้รีจิสเตอร์ R7 เมื่อกำหนด MCS-51 ทำงานที่ความถี่ 12 MHz และ 11.0592 MHz

ที่ 12 MHz 1 Machine Cycle ใช้ Clock 12 ลูก ฉะนั้นใช้เวลาทั้งหมด $12x(1/12x10^6) = 1\ \mu S$

ที่ 11.0592 MHz 1 Machine Cycle ใช้ Clock 12 ลูก ฉะนั้นใช้เวลาทั้งหมด $12x(1/11.0592x10^6) = 1.082\ \mu S$

```
; Subroutine for Delay Time          Machine Cycle
DELAY:    MOV   R7, #01H             [1]    ; กำหนดค่า R7 = 01H
DELAY1:   DJNZ  R7, DELAY1           [2]    ; ลดค่า R7 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                            ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
          RET                        [2]
```

รวม Machine Cycle = MOV R7,#01H[1] + DJNZ R7,DELAY1[2] + RET[2]

= [1]+[2]+[2]

= 5 Machine Cycle

ฉะนั้นใช้เวลา = 5x1 μS = 5 μS ที่ความถี่ 12 MHz

และ = 5x1.082 μS = 5.3 μS ที่ความถี่ 11.0592 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>สรุป</u> *สามารถที่จะสรุปเป็นสูตรในการคำนวณหาค่าเวลาที่หน่วงคือ*

(3+R7(2))x12(1/ความถี่ใช้งาน)

2. โปรแกรมหน่วงเวลาแบบ 2 Loop

โปรแกรมหน่วงเวลาแบบ 2 Loop จะคิดเวลาที่หน่วงไปจาก Loop ที่อยู่ในสุดคูณด้วย Loop ที่อยู่ถัดออกมา ถ้ากำหนด R6 เป็นจำนวนรอบการทำงานวงรอบใน และ R7 เป็นจำนวนรอบการทำงานวงรอบนอก เราสามารถหาค่าเวลาที่หน่วงไปได้

```
; Subroutine for Delay Time          Machine Cycle
DELAY:    MOV   R7, #02H             [1]    ; กำหนดค่า R7 = 02H
DELAY1:   MOV   R6, #03H             [1]    ; กำหนดค่า R6 = 03H
DELAY2:   DJNZ  R6, DELAY2           [2]    ; ลดค่า R6 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                            ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
          DJNZ  R7, DELAY1           [2]    ; ลดค่า R7 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                            ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
          RET                        [2]
รวม Machine Cycle  = 1+[(2+2+2)+2+1]+ [(2+2+2)+2+1]+2
                   = 21 Machine Cycle
ฉะนั้นใช้เวลา      = 21x1 μs      = 21 μs      ที่ความถี่ 12 MHz
และ                = 21x1.082 μs  = 22.722 μs  ที่ความถี่ 11.0592 MHz
```

<u>สรุป</u> *สามารถที่จะสรุปเป็นสูตรในการคำนวณหาค่าเวลาที่หน่วงคือ*

(3+R7(3+R6(2))x12(1/ความถี่ใช้งาน)

3. โปรแกรมหน่วงเวลาแบบ 3 Loop

ในการหน่วงเวลาที่ต้องการการหน่วงเวลาที่มาก ต้องเพิ่มวงรอบการทำงานอีกหนึ่งวงรอบเป็น 3 วงรอบ ในโปรแกรมส่วนนี้จะใช้ R6 เป็นจำนวนรอบการทำงานวงรอบใน, R7 เป็นจำนวนรอบการทำงานวงรอบนอก และจะมี R5 เป็นการกำหนดรอบทั้งหมดอีกที ในการกำหนดค่า R5 นั้นจะกำหนดที่โปรแกรมหลักก่อนการเรียกใช้โปรแกรมย่อยหน่วงเวลา เพื่อให้โปรแกรมสามารถหน่วงเวลาได้ตามความต้องการของผู้เรียกใช้ โดยกำหนดระยะเวลาที่ R5 และโปรแกรมหน่วงเวลาควรเป็นค่าเวลาที่เป็นจำนวนเต็มเพื่อสะดวกในการกำหนดค่าเวลาที่ต้องการหน่วงเช่น 10 μS, 500 μS, 10 ms เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ; โปรแกรมหลักกำหนดค่า R5 = 02H
        ; Subroutine for Delay Time          Machine Cycle
DELAY:  MOV   R7, #02H          [1]   ; กำหนดค่า R7 = 02H
DELAY1: MOV   R6, #03H          [1]   ; กำหนดค่า R6 = 03H
DELAY2: DJNZ  R6, DELAY2        [2]   ; ลดค่า R6 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                      ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
DELAY3: DJNZ  R7, DELAY1        [2]   ; ลดค่า R7 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                      ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
        DJNZ  R5, DELAY         [2]   ; ลดค่า R5 ลงหนึ่ง ตรวจสอบถ้าไม่เท่า
                                      ; กับศูนย์ให้ทำซ้ำจนกว่าจะเท่ากับศูนย์
        RET                     [2]
```

รวม Machine Cycle = {[(2+2+2)+2+1]+ [(2+2+2)+2+1]+2+1}+{[(2+2+2)+2+1]
+ [(2+2+2)+2+1]+2+1}+2
= 44 Machine Cycle

ฉะนั้นใช้เวลา = 44x1 μs = 44 μs ที่ความถี่ 12 MHz

และ = 44x1.082 μs = 47.608 μs ที่ความถี่ 11.0592 MHz

สรุป *สามารถที่จะสรุปเป็นสูตรในการคำนวณหาค่าเวลาที่หน่วงคือ*

{R5[3+R7(3+R6(2))]+2}x12(1/ความถี่ใช้งาน)

หมายเหตุ ในการคิดเวลาที่หน่วงไปนั้น สามารถที่จะใช้สูตรในการคำนวณได้เลย แต่ค่าของรีจิสเตอร์นั้นต้องใช้ค่าที่เป็นเลขฐานสิบเท่านั้น 256-255 หรือ 00H-FFH และหากของค่ารีจิสเตอร์เป็น 00H แล้วค่าที่ใช้ในการคำนวณคือ 256 เพราะเมื่อลดค่า 00H ลงหนึ่งแล้วจะเท่ากับ 0FFH

ถ้าต้องการเพิ่ม Machine Cycle ให้มากขึ้น ก็สามารถทำได้โดยการเพิ่มคำสั่งเข้าไปอีก แต่ต้องคำนึงถึงว่าคำสั่งที่เพิ่มเข้าไปจะไม่ไปเป็นปัญหาในส่วนของโปรแกรมหน่วงเวลาหรือโปรแกรมหลัก ในชุดคำสั่งของ Z-80 หรือ MCS-51จะมีอยู่หนึ่งคำสั่งที่ไม่มีผลต่อโปรแกรมส่วนใดเลยคือ NOP (No Operation) เพียงแต่เพิ่มการทำงานให้กับไมโครคอนโทรลเลอร์ 1 Machine Cycle เท่านั้น โดยส่วนมากจะใส่ในวงรอบการทำงานในสุด

และยังมีคำสั่งอีกคำสั่งหนึ่งที่จะทำให้โปรแกรมทำงานในวงรอบตลอดไม่สามารถจบการทำงานได้ เหมาะสำหรับให้ไมโครคอนโทรลเลอร์ไม่ไปทำงานในโปรแกรมใดๆ เลย (เหมือนคำสั่ง HALT ของZ-80) มีรูปแบบคำสั่งดังนี้ SJMP $ รูปแบบคำสั่งนี้จะทำงานเฉพาะคำสั่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งเดียว โดยเมื่อแปลงเป็น Machine Code แล้วจะได้ 80 FE นั่นคือเมื่อสิ้นสุดการกระทำคำสั่งนี้แล้ว โปรแกรมเคาน์เตอร์จะชี้ที่ตำแหน่งเริ่มต้นของคำสั่งนี้

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/3, SW 1/6, SW 1/7 และ SW 4/2 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
6. ป้อนโปรแกรมที่ 2.1 แล้วสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ LED

โปรแกรมที่ 2.1 โปรแกรมหน่วงเวลาแบบวงรอบเดียว

```
                PORT_A  EQU    0E020H
                PORT_B  EQU    PORT_A+1
                PORT_C  EQU    PORT_A+2
                C_PORT  EQU    PORT_A+3
                        ORG    8000H
8000 : 90E023   INIT:   MOV    DPTR,#C_PORT
8003 : 748A             MOV    A,#08AH
8005 : F0               MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV    DPTR,#PORT_A
8009 : E4               CLR    A
800A : F0       LOOP:   MOVX   @DPTR,A          ; โปรแกรมไฟกระพริบ
800B : 1111             ACALL  DELAY
800D : B2E5             CPL    ACC.5
800F : 80F9             SJMP   LOOP

8011 : 7F80     DELAY:  MOV    R7,#80H          ; โปรแกรมหน่วงเวลาแบบ 1 วงรอบ
8013 : DFFE     DELAY1: DJNZ   R7,DELAY1
8015 : 22               RET
                        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. อธิบายการทำงานของโปรแกรมที่ 2.1

8. ป้อนโปรแกรมที่ 2.2 แล้วสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ LED

โปรแกรมที่ 2.2 โปรแกรมหน่วงเวลาแบบ 2 วงรอบ

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A         ; โปรแกรมไฟกระพริบ
800B : 1111             ACALL   DELAY
800D : B2E5             CPL     ACC.5
800F : 80F9             SJMP    LOOP

8011 : 7F80     DELAY:  MOV     R7,#80H         ; โปรแกรมหน่วงเวลาแบบ 2 วงรอบ
8013 : 7EFF     DELAY1: MOV     R6,#0FFH
8015 : DEFE     DELAY2: DJNZ    R6,DELAY2
8017 : DFFA             DJNZ    R7,DELAY1
8019 : 22               RET
                        END
```

9. อธิบายการทำงานของโปรแกรมที่ 2.2

10. ป้อนโปรแกรมที่ 2.3 แล้วสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ LED

โปรแกรมที่ 2.3 โปรแกรมหน่วงเวลาแบบ 3 วงรอบ

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A         ; โปรแกรมไฟกระพริบ
800B : 7D0A             MOV     R5,#0AH
800D : 1113             ACALL   DELAY
800F : B2E5             CPL     ACC.5
8011 : 80F7             SJMP    LOOP

8013 : 7F80     DELAY:  MOV     R7,#80H         ; โปรแกรมหน่วงเวลาแบบ 3 วงรอบ
8015 : 7EFF     DELAY1: MOV     R6,#0FFH
8017 : DEFE     DELAY2: DJNZ    R6,DELAY2
8019 : DFFA     DELAY3: DJNZ    R7,DELAY1
801B : DDF6             DJNZ    R5,DELAY
801D : 22               RET
                        END
```

11. อธิบายการทำงานของโปรแกรมที่ 2.3
12. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

13. ในแต่ละโปรแกรมที่ทดลอง หากต้องการเพิ่มหรือลดการหน่วงเวลาจะต้องแก้ไขโปรแกรมอย่างไร
14. จงเขียนโปรแกรมหน่วงเวลาโดยหน่วงเวลาไว้ 500 ไมโครวินาทีและ 1 วินาที พร้อมทั้งแสดงวิธีการคำนวณเวลามาอย่างละเอียด
15. จงเขียนโปรแกรมให้ไดโอดเปล่งแสงติดเป็นเวลา 5 วินาที และดับเป็นเวลา 7 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

การแสดงผลข้อมูลขนาด 8 บิต (Port Monitor)

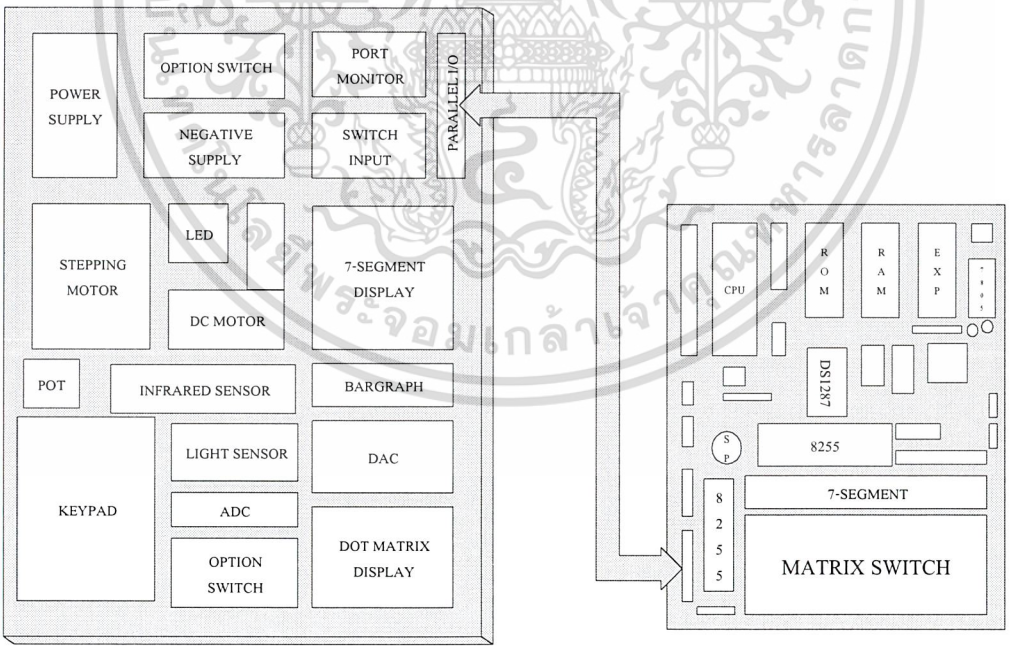
วัตถุประสงค์

1. เพื่อศึกษาการส่งข้อมูลขนาด 8 บิตผ่านทางพอร์ต PA ของ 8255
2. เพื่อสามารถใช้ไมโครคอนโทรลเลอร์ควบคุมการส่งข้อมูลทางพอร์ต PA ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 3 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



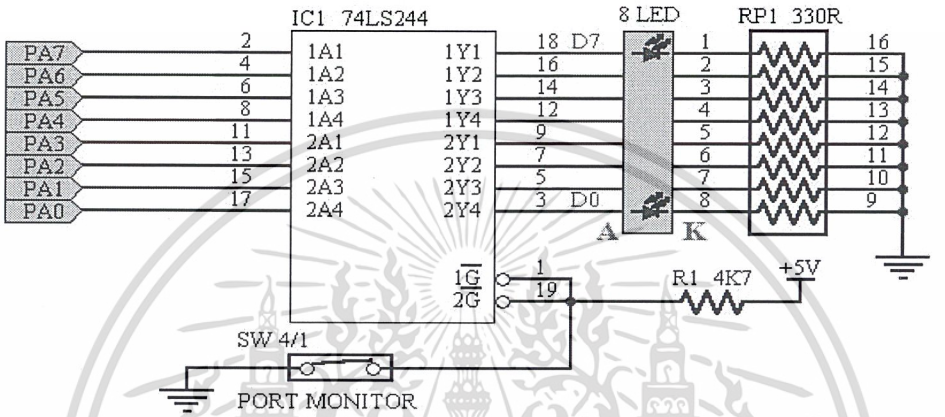
รูปที่ ข. 4 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

วงจรแสดงผลข้อมูลขนาด 8 บิต (Port Monitor)

หลอดไดโอดเปล่งแสง (Light Emitting Diode) จะทำหน้าที่เป็นตัวแสดงสภาวะลอจิกในแต่ละบิต (Bit) ของพอร์ตเอาต์พุต วงจรจะประกอบด้วยหลอดไดโอดเปล่งแสง จำนวน 8 หลอด



รูปที่ ข. 5 วงจรแสดงผลข้อมูลขนาด 8 บิต (Port Monitor)

ไอซี 74LS244 จะต่อเป็นบัฟเฟอร์ (Buffer) ระหว่างพอร์ตกับหลอดไดโอดเปล่งแสง เมื่อทำให้ 74LS244 อยู่ในสภาวะ Enable (ปรับให้สวิตช์ SW 4/1 อยู่ที่ตำแหน่ง "ON") หลอดไดโอดเปล่งแสงจะติดสว่าง เมื่อมีอินพุตลอจิกเป็น "1" เอาต์พุตของ 74LS244 ก็จะเป็น "1" ทำให้เกิดความต่างศักย์ของแรงดันเอาต์พุตของ 74LS244 กับกราวด์ เป็นผลให้มีกระแสจำนวนหนึ่งไหลผ่านตัวต้านทาน 330 โอห์ม และหลอดไดโอดเปล่งแสง ทำให้หลอดไดโอดเปล่งแสงติดสว่าง

ถ้าอินพุตมีลอจิกเป็น "0" เอาต์พุตของ 74LS244 ก็จะมีลอจิกเป็น "0" ทำให้ไม่เกิดความต่างศักย์ของแรงดันขึ้นระหว่างเอาต์พุตของ 74LS244 กับกราวด์ ทำให้ไม่เกิดกระแสไหลผ่านตัวต้านทานมายังหลอดไดโอดเปล่งแสง ดังนั้นหลอดไดโอดเปล่งแสง จึงไม่ติดสว่าง

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "OFF"
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง "OFF" ยกเว้น SW 1/2 "ON"
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 4/1 มาที่ตำแหน่ง "ON"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จ่ายไฟเลี้ยงให้กับชิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
6. ป้อนโปรแกรมที่ 3.1 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกการติด-ดับ ของหลอดไดโอดเปล่งแสงของ Port Monitor

โปรแกรมที่ 3.1 โปรแกรมสั่งงานการติด-ดับของ LED ของ Port Monitor

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH          ; Command Word
8005 : F0               MOVX  @DPTR,A          ; ส่ง Command Word ให้ 8255
8006 : 90E020   START:  MOV   DPTR,#PORT_A
8009 : 743C             MOV   A,#3CH           ; ข้อมูลที่ส่งออกทางพอร์ต PA
800B : F0               MOVX  @DPTR,A
800C : 80FE             SJMP  $
                        END
```

7. ทดลองเปลี่ยนค่าของข้อมูลที่ส่งออกบันทึกผลการติด-ดับของไดโอดเปล่งแสงลงในตาราง

	PORT MONITOR							
ข้อมูลของพอร์ต PA	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0FFH	O	O	O	O	O	O	O	O
0FH	O	O	O	O	O	O	O	O
00H	O	O	O	O	O	O	O	O
96H	O	O	O	O	O	O	O	O
55H	O	O	O	O	O	O	O	O
0AAH	O	O	O	O	O	O	O	O

● ติด

O ดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. จงอธิบายความสัมพันธ์ระหว่างข้อมูลที่ส่งออกมา กับการติด-ดับของหลอด LED
9. ทดลองโปรแกรมที่ 3.2 โปรแกรมไฟกระพริบบนหลอด LED ของ Port Monitor

โปรแกรมที่ 3.2 ไฟกระพริบบน Port Monitor

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7400             MOV     A,#00H
800B : F0       LOOP:   MOVX    @DPTR,A         ; โปรแกรมไฟกระพริบ
800C : 1111             ACALL   DELAY
800E : F4               CPL     A
800F : 80FA             SJMP    LOOP

8011 : 7E00     DELAY:  MOV     R6,#00H         ; โปรแกรมหน่วงเวลา
8013 : 7F00     DELAY1: MOV     R7,#00H
8015 : 00       DELAY2: NOP
8016 : 00               NOP
8017 : DFFC             DJNZ    R7,DELAY2
8019 : DEF8             DJNZ    R6,DELAY1
801B : 22               RET
                        END
```

10. จงอธิบายการทำงานและผลการทดลองในโปรแกรมที่ 3.2
11. ทดลองโปรแกรมที่ 3.3 โปรแกรมไฟวิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.3 โปรแกรมไฟวิ่ง

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7400             MOV     A,#00H
800B : F0       LOOP1:  MOVX    @DPTR,A         ; LED ติดตั้งแต่หลัก LSB TO MSB
800C : 111D             ACALL   DELAY
800E : D3               SETB    C
800F : 33               RLC     A
8010 : B4FFF8           CJNE    A,#0FFH,LOOP1
8013 : F0       LOOP2:  MOVX    @DPTR,A         ; LED ดับตั้งแต่หลัก LSB TO MSB
8014 : 111D             ACALL   DELAY
8016 : C3               CLR     C
8017 : 33               RLC     A
8018 : B400F8           CJNE    A,#00H,LOOP2
801B : 80EE             SJMP    LOOP1

801D : 7E00     DELAY:  MOV     R6,#00H         ; โปรแกรมหน่วงเวลา
801F : 7F00     DELAY1: MOV     R7,#00H
8021 : 00       DELAY2: NOP
8022 : 00               NOP
8023 : DFFC             DJNZ    R7,DELAY2
8025 : DEF8             DJNZ    R6,DELAY1
8027 : 22               RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

12. จงอธิบายการทำงานและผลการทดลองในโปรแกรมที่ 3.3
13. จงแก้ไขโปรแกรมที่ 3.3 ให้เลื่อนจาก หลัก MSB ไป LSB
14. ทดลองโปรแกรมที่ 3.4 โปรแกรมควบคุมไฟจราจรโดยใช้หลอด LED สีเขียว, สีเหลืองและสีแดง เป็นตัวแสดงผล

โปรแกรม 3.4 โปรแกรมไฟจราจร

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7401     LOOP1:  MOV     A,#01H
800B : F0               MOVX    @DPTR,A         ; หลอด LED สีเขียวติด
800C : 7D32             MOV     R5,#32H         ; หน่วงเวลา 5 วินาที
800E : 111E             ACALL   DELAY
8010 : 23               RL      A
8011 : F0               MOVX    @DPTR,A         ; หลอด LED สีเหลืองติด
8012 : 7D14             MOV     R5,#14H         ; หน่วงเวลา 2 วินาที
8014 : 111E             ACALL   DELAY
8016 : 23               RL      A
8017 : F0               MOVX    @DPTR,A         ; หลอด LED สีแดงติด
8018 : 7D32             MOV     R5,#32H         ; หน่วงเวลา 5 วินาที
801A : 111E             ACALL   DELAY
801C : 80EB             SJMP    LOOP1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
801E : 7EBE    DELAY:  MOV   R6,#190          ; เวลา = 12*(1/11.0592*10^6)*
8020 : 7FF1    DELAY1: MOV   R7,#241          ;        (R5(1+R6(1+R7(2)+2)
8022 : DFFE    DELAY2: DJNZ  R7,DELAY2        ;        +2)+2)
8024 : DEFA            DJNZ  R6,DELAY1        ; โปรแกรมหน่วงเวลา 100 มิลลิวินาที
8026 : DDF6            DJNZ  R5,DELAY
8028 : 22              RET
                       END
```

15. จงอธิบายการทำงานและผลการทดลองในโปรแกรมที่ 3.4
16. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

17. ถ้าต้องการให้หลอด LED กระพริบเร็วขึ้นหรือช้าลง จะต้องแก้โปรแกรมอย่างไร
18. จงเขียนโปรแกรมไฟวิ่งจากด้านขวา (LSB) ไปด้านซ้าย (MSB) และวิ่งจากด้านซ้าย (MSB) ไปด้านขวา (LSB)
19. จงเขียนโปรแกรมไฟวิ่งจากขอบทั้งสองข้างเข้าสู่ตรงกลาง และวิ่งจากตรงกลางออกสู่ขอบทั้งสองข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

การอ่านข้อมูลจากสวิตช์อินพุต (Switch Input)

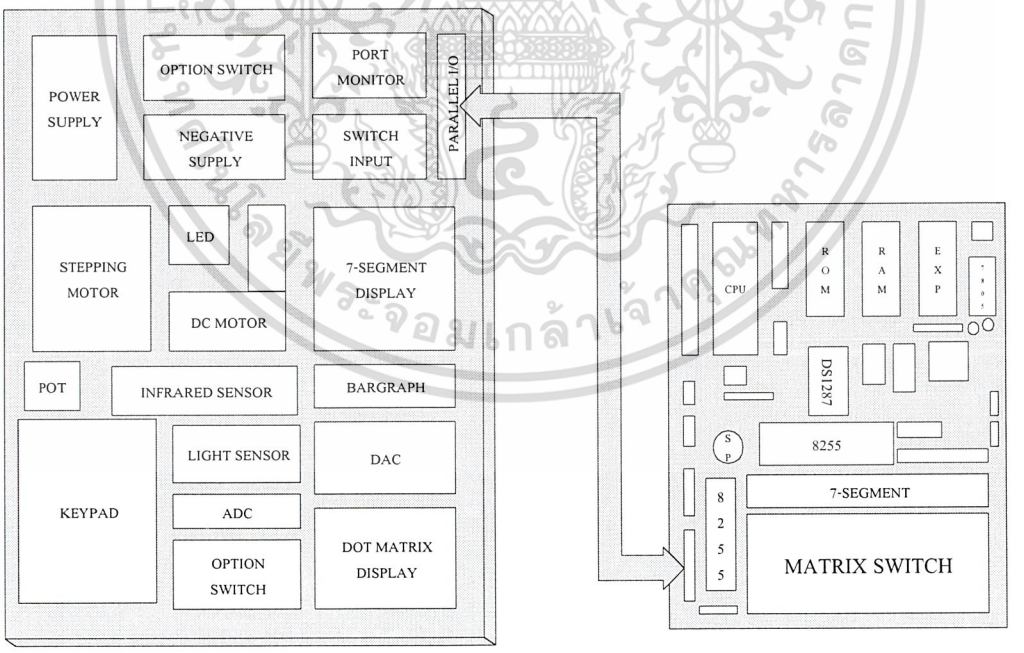
วัตถุประสงค์

1. เพื่อศึกษาการอ่านข้อมูลขนาด 8 บิตเข้ามาทางพอร์ต PB ของ 8255
2. เพื่อสามารถใช้ไมโครคอนโทรลเลอร์อ่านข้อมูลขนาด 8 บิตเข้ามาทางพอร์ต PB ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 4 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 6 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

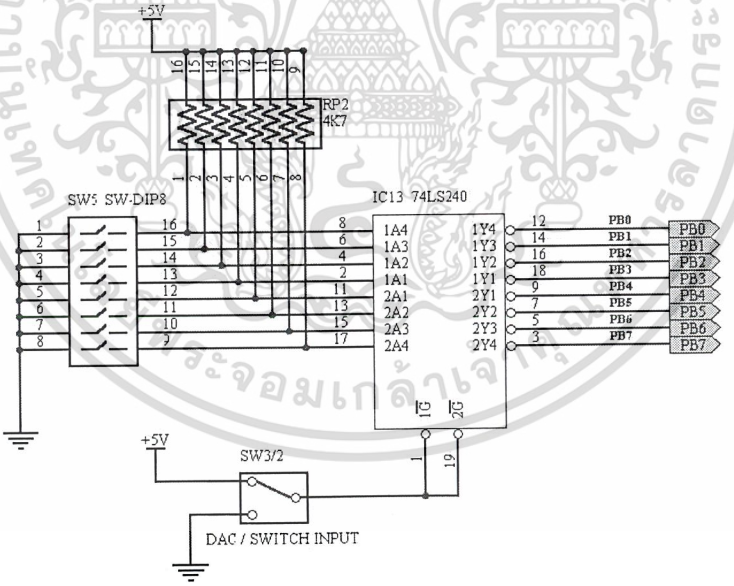
ทฤษฎีเบื้องต้น

การอ่านข้อมูลจากสวิตช์อินพุต (Switch Input)

มุมบนขวาของชุดทดลองจะเป็นส่วนของวงจร Switch Input จำนวน 8 ตัวคือ SW 1 ถึง SW 8 ซึ่งเป็นดิพสวิตช์ที่ต่อเข้ากับ IC13 เบอร์ 74LS240 ภายในเป็น Octal Tri-State Inverter Buffer ดิพสวิตช์จะทำหน้าที่สร้างสภาวะลอจิก (Logic) “1” หรือ “0” เมื่อดิพสวิตช์อยู่ที่ตำแหน่ง “ON” จะมีสภาวะลอจิกเป็น “1” (HI) ถ้าอยู่ที่ตำแหน่ง “OFF” จะมีสภาวะลอจิก “0” (LO)

ไอซี 74LS240 ซึ่งเป็น Octal Tri-State Inverter Buffer จะทำงานเมื่อได้รับการ “Enable” โดยต่อขา 1 และขา 19 ลงกราวด์ (SW 3) ซึ่งแต่ละเอาต์พุตของ 74LS240 จะให้เอาต์พุตออกมา 3 สถานะคือ

1. เอาต์พุตเป็นลอจิก “0” เมื่อได้รับอินพุตเข้ามาเป็นลอจิก “1”
2. เอาต์พุตเป็นลอจิก “1” เมื่อได้รับอินพุตเข้ามาเป็นลอจิก “0”
3. เอาต์พุตเป็นสภาวะ High Impedance เมื่อ 74LS240 ได้รับการ Disable คือ ขา 1 และขา 19 ถูกต่อเข้ากับไฟ +5V ถึงแม้อินพุตที่เข้ามาจะเป็น “1” หรือ “0” ก็ตาม



รูปที่ ข. 7 วงจรสร้างสภาวะลอจิกขนาด 8 บิต (Switch Input)

ดิพสวิตช์แต่ละตัวจะเปิดวงจรเมื่ออยู่ที่ตำแหน่ง “OFF” และถูกต่อลงกราวด์เมื่ออยู่ที่ตำแหน่ง “ON” ดิพสวิตช์ทั้ง 8 ตัวจะมีตัวต้านทาน 4.7 กิโลโอห์มต่อเข้ากับไฟ +5V (เรียกว่าการ Pull up) ที่ตำแหน่ง “OFF” ของดิพสวิตช์ทำให้อินพุตของ 74LS240 มีสภาวะลอจิกเป็น “1” และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อดิพสวิตช์อยู่ที่ตำแหน่ง “ON” ทำให้อินพุตมีสภาวะลอจิกเป็น “0” เพราะตัวต้านทานถูกต่อลงกราวด์ ซึ่งทำให้มีกระแสไหลผ่านตัวต้านทานประมาณ 1.06 mA

ลำดับขั้นตอนการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 4/1 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “SWITCH INPUT”
6. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 4.1 แล้วสั่งให้โปรแกรมทำงาน
8. ทดลองปรับสวิตช์ “SW 5” แล้วสังเกตและบันทึกการติด-ดับของหลอด LED ที่ Port Monitor

โปรแกรมที่ 4.1 การอ่านข้อมูลไปแสดงที่ Port Monitor

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E021   START:  MOV     DPTR,#PORT_B
8009 : E0               MOVX    A,@DPTR         ; อ่านข้อมูลจาก “Switch Input”
800A : 90E020           MOV     DPTR,#PORT_A
800D : F0               MOVX    @DPTR,A         ; นำข้อมูลไปแสดงผลที่ Port Monitor
800E : 80F6             SJMP    START           ; กลับไปอ่านข้อมูลใหม่
                        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ทดลองเปลี่ยนค่าของข้อมูลที่ SW 5 บันทึกผลการติด-ดับของหลอด LED ลงในตาราง

SWITCH INPUT								PORT MONITOR							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
OFF	OFF	OFF	OFF	ON	ON	ON	ON	O	O	O	O	O	O	O	O
OFF	OFF	ON	ON	OFF	OFF	ON	ON	O	O	O	O	O	O	O	O
ON	OFF	ON	OFF	ON	OFF	ON	OFF	O	O	O	O	O	O	O	O
OFF	OFF	ON	ON	ON	ON	OFF	OFF	O	O	O	O	O	O	O	O
ON	OFF	ON	ON	OFF	OFF	OFF	ON	O	O	O	O	O	O	O	O
ON	ON	OFF	OFF	OFF	ON	ON	OFF	O	O	O	O	O	O	O	O

● ติด

O ดับ

10. จงอธิบายความสัมพันธ์ระหว่างข้อมูลจาก SW 5 กับการติด-ดับของหลอด LED
11. จงเขียนโปรแกรมอ่านข้อมูลจากสวิตช์ เมื่อ Switch Input อยู่ที่ตำแหน่ง “ON” ให้หลอด LED ดับ (ทำงานตรงกันข้ามกับโปรแกรมที่ 4.1)
12. ทดลองโปรแกรมที่ 4.2 โปรแกรมควบคุมไฟกระพริบบนหลอด LED ของ Port Monitor โดยใช้บิตที่ 0 ของ SW 5 เป็นสวิตช์ควบคุม

โปรแกรมที่ 4.2 ควบคุมไฟกระพริบบน Port Monitor

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH
8005 : F0               MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E021   START:  MOV   DPTR,#PORT_B
8009 : E0       LOOP1:  MOVX  A,@DPTR
800A : 30E0FC           JNB   ACC.0,LOOP1      ; ตรวจสอบสวิตช์ ตำแหน่งที่ 0
800D : 90E020           MOV   DPTR,#PORT_A
```

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8010 : 7400              MOV    A,#00H
8012 : F0       LOOP2:   MOVX   @DPTR,A          ; โปรแกรมไฟกระพริบ
8013 : 111B              ACALL  DELAY
8015 : F4                CPL    A
8016 : F0                MOVX   @DPTR,A
8017 : 111B              ACALL  DELAY
8019 : 80EB              SJMP   START

801B : 7E00     DELAY:   MOV    R6,#00H          ; โปรแกรมหน่วงเวลา
801D : 7F00     DELAY1:  MOV    R7,#00H
801F : 00       DELAY2:  NOP
8020 : 00                NOP
8021 : DFFC              DJNZ   R7,DELAY2
8023 : DEF8              DJNZ   R6,DELAY1
8025 : 22                RET
                         END
```

13. จงอธิบายการทำงานและผลการทดลองในโปรแกรมที่ 4.2
14. ทดลองโปรแกรมที่ 4.3 โปรแกรมควบคุมความเร็วในการกระพริบของหลอด LED โดยใช้ บิตที่ 0-7 ของ SW 5 เป็นสวิตช์ควบคุม

โปรแกรม 4.3 โปรแกรมควบคุมความเร็วในการกระพริบของหลอด LED

```
          PORT_A  EQU    0E020H
          PORT_B  EQU    PORT_A+1
          PORT_C  EQU    PORT_A+2
          C_PORT  EQU    PORT_A+3
                  ORG    8000H
8000 : 90E023 INIT:      MOV    DPTR,#C_PORT
8003 : 748A              MOV    A,#08AH
8005 : F0                MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 7400     START:   MOV    A,#00H
8008 : 90E020   LOOP2:   MOV    DPTR,#PORT_A     ; โปรแกรมไฟกระพริบ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
800B : F0              MOVX   @DPTR,A
800C : 1113            ACALL  ROUND
800E : 111D            ACALL  DELAY
8010 : F4              CPL    A
8011 : 80F5            SJMP   LOOP2
8013 : C0E0   ROUND:   PUSH   ACC          ; โปรแกรมตรวจสอบรอบความเร็ว
8015 : 90E021          MOV    DPTR,#PORT_B  ; จากสวิตช์
8018 : E0              MOVX   A,@DPTR
8019 : FD              MOV    R5,A
801A : D0E0            POP    ACC
801C : 22              RET

801D : 7E63   DELAY:   MOV    R6,#99        ; เวลา = 12*(1/11.0592*10^6)*
801F : 7F2D   DELAY1:  MOV    R7,#45        ;        (R5(1+R6(1+R7(2)+2)
8021 : DFFE   DELAY2:  DJNZ   R7,DELAY2     ;          +2)+2)
8023 : DEFA            DJNZ   R6,DELAY1     ;โปรแกรมหน่วงเวลา 10 มิลลิวินาที
8025 : DDF6            DJNZ   R5,DELAY
8027 : 22              RET
                       END
```

15. จงอธิบายการทำงานและผลการทดลองในโปรแกรมที่ 4.3
16. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

17. จงเขียนโปรแกรมไฟวิ่งที่ควบคุมรูปแบบการวิ่งโดย SW 5 ดังนี้

รูปแบบการวิ่ง	SW 5 บิตที่ 1	SW 5 บิตที่ 0
0000000X ←	OFF	OFF
X0000000 →	OFF	ON
000XX000 ←→	ON	OFF
X000000X →←	ON	ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. จงเขียนโปรแกรมอ่านข้อมูลจากสวิตช์ SW 5 แล้วนำไปแสดงผลที่จอ 7-Segment หลักที่ 4 และหลักที่ 5 ของ Single Board โดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H

-0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter

-0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment

-0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch

-E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

การแสดงผลตัวเลข 7 ส่วน (7-Segment Display)

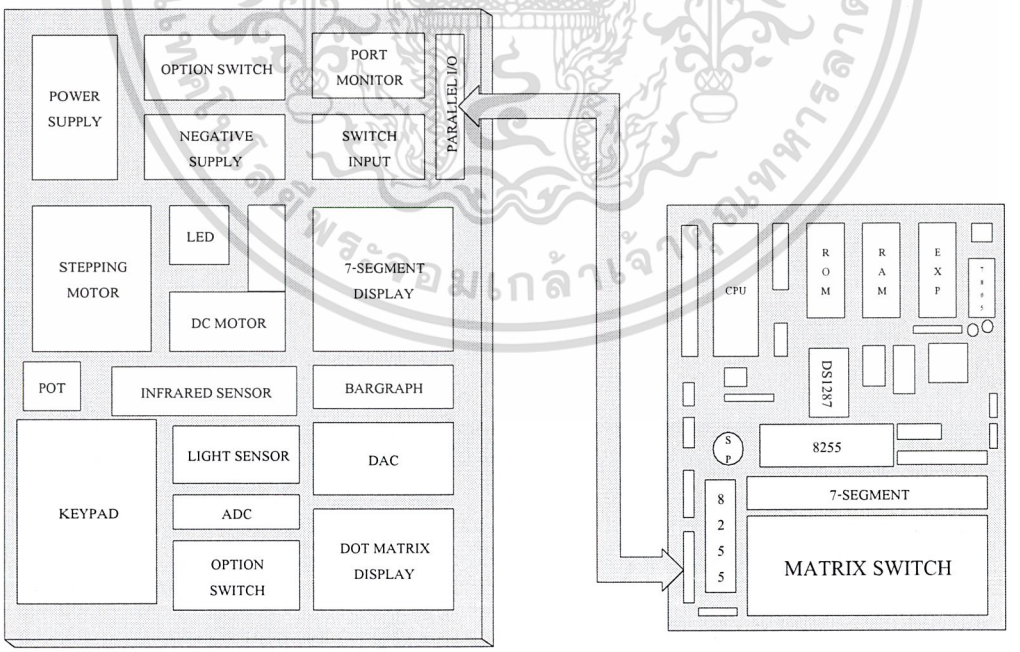
วัตถุประสงค์

1. เพื่อศึกษาการแสดงผลของชุดแสดงผลตัวเลข 7 ส่วน (7-Segment Display) โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม
2. เพื่อสามารถควบคุมการแสดงผลของชุดแสดงผลตัวเลข 7 ส่วน (7-Segment Display)ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 5 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 8 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

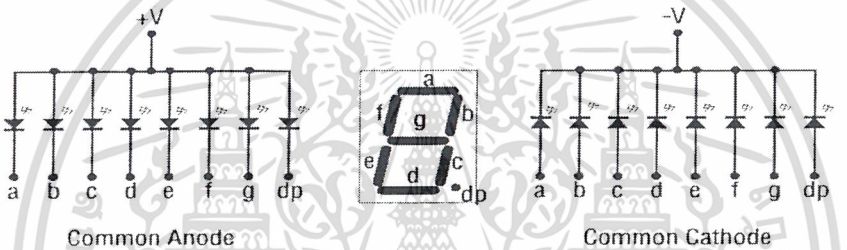
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การแสดงผลตัวเลข 7 ส่วน (7-Segment Display)

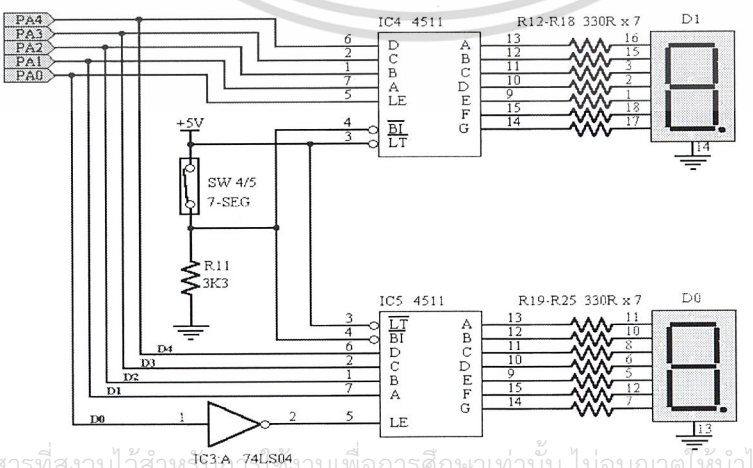
ชุดแสดงผลตัวเลข 7 ส่วน (7-Segment Display) เป็นตัวแสดงผลอีกแบบหนึ่งที่แสดงผลออกมาเป็นตัวเลข ภายในจะประกอบไปด้วยหลอด LED จำนวน 8 ดวงเป็นเซกเมนต์ a, b, c, d, e, f, g และเป็นจุดทศนิยม (dp) LED แต่ละดวงจะต่อเข้ากันอยู่หนึ่งขา ซึ่งอาจจะเป็นขาแคโทด (Cathode) หรือขาแอโนด (Anode) ดังนั้นเราจึงแบ่ง 7-Segment Display ออกเป็นประเภทตามลักษณะตามการต่อขาภายในร่วมกันเป็น

1. แบบแคโทดร่วม (Common Cathode)
2. แบบแอโนดร่วม (Common Anode)



รูปที่ ข. 9 การต่อขา 7-Segment Display

การแสดงผลของ 7-Segment Display จำเป็นต้องอาศัยวงจรถอดรหัส (Decoder) และวงจรขับเพื่อรับอินพุตเป็น BCD มาแสดงผลเป็นตัวเลขฐานสิบ ในปัจจุบันจะทำเป็นไอซีสำเร็จรูปอย่างเช่น 7447, 4511 ซึ่งจะเป็น BCD to 7-Segment ชนิด 4 อินพุต จะเห็นได้ว่าการควบคุม 7-Segment Display แต่ละหลักจะต้องใช้สัญญาณอินพุตจำนวน 4 บิต ดังนั้นถ้าหากต้องการแสดงผลหลายๆ หลัก ก็จะต้องใช้จำนวนบิตของอินพุตมากขึ้นตามไปด้วย



รูปที่ ข. 10 วงจรชุดแสดงผลตัวเลข 7 ส่วน (7-Segment Display)

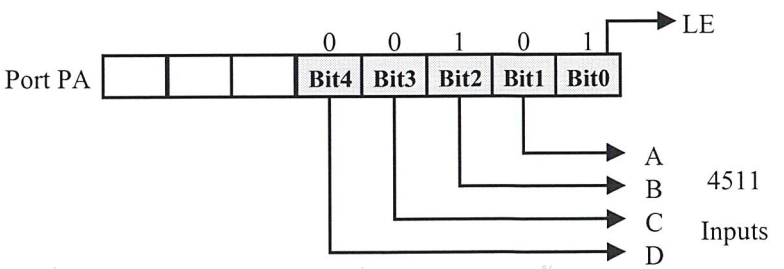
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7-Segment Display ของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีอยู่ 2 หลัก คือ D0 และ D1 โดยทั้ง 2 หลักจะใช้สัญญาณอินพุตรวมกันแค่ 5 บิต ต่อเข้ากับ 4511 จำนวน 2 ตัว 7-Segment Display ที่ใช้จะเป็นแบบแคโทดร่วม (Common Cathode)

ตารางที่ ข. 2 การทำงานของ 4511

INPUTS							OUTPUTS							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X								

จากวงจรของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051จะเห็นว่าบิต 0 จะต่อเข้ากับขา LE ทำหน้าที่ควบคุมว่าจะให้หลัก D0 หรือ D1 ทำงาน ขา LE ของทั้งสองหลักจะมีสภาวะที่ตรงข้ามกันตลอด ถ้าหลักใดที่ขา LE เป็น “0” หลักนั้นก็จะทำงาน



รูปที่ ข. 11 บิตที่ใช้กับวงจร 7-Segment Display

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น อินพุตของ 4511 เป็น 00011 บิต 0 เป็น “1” แสดงว่าหลัก D1 จะทำงานแสดงผลออกมาเป็นตัวเลข 1 ที่ 7-Segment Display

ตารางที่ ข. 3 อินพุตของ D0 และ D1

Bit 4	Bit3	Bit2	Bit1	Bit0	CODE	7-Segment D1	7-Segment D0
0	0	0	0	0	00H	0	-
0	0	0	0	1	01H	-	0
0	0	0	1	0	02H	1	-
0	0	0	1	1	03H	-	1
0	0	1	0	0	04H	2	-
0	0	1	0	1	05H	-	2
0	0	1	1	0	06H	3	-
0	0	1	1	1	07H	-	3
0	1	0	0	0	08H	4	-
0	1	0	0	1	09H	-	4
0	1	0	1	0	0AH	5	-
0	1	0	1	1	0BH	-	5
0	1	1	0	0	0CH	6	-
0	1	1	0	1	0DH	-	6
0	1	1	1	0	0EH	7	-
0	1	1	1	1	0FH	-	7
1	0	0	0	0	10H	8	-
1	0	0	0	1	11H	-	8
1	0	0	1	0	12H	9	-
1	0	0	1	1	13H	-	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6, SW 1/7, SW 4/1 และ SW4/5 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “SWITCH INPUT”
6. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 5.1 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลที่ 7-Segment Display

โปรแกรมที่ 5.1 โปรแกรมแสดงตัวเลขบน 7-Segment Display

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7411             MOV     A,#11H
800B : F0               MOVX    @DPTR,A         ; กำหนดรหัสให้กับ 7-Segment
800C : 80FE             SJMP    $
                        END
```

8. ทดลองเปลี่ยนรหัสตามตารางข้างล่างและบันทึกตัวเลขที่ปรากฏบน 7-segment Display

รหัส	7-segment Display หลัก D0
01H	
03H	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัส	7-segment Display หลัก D0
05H	
07H	
09H	
0BH	
0DH	
0FH	
11H	
13H	

9. ทดลองเปลี่ยนรหัสตามตารางข้างล่างและบันทึกตัวเลขที่ปรากฏบน 7-segment Display

รหัส	7-segment Display หลัก D1
00H	
02H	
04H	
06H	
08H	
0AH	
0CH	
0EH	
10H	
12H	

10. จงอธิบายการทำงานของโปรแกรมที่ 5.1
11. ทดลองโปรแกรมที่ 5.2 ให้ 7-Segment Display หลัก D0 นับเลขตั้งแต่ 0-9 สังเกตและบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 5.2 โปรแกรมนับเลข 0-9 หลัก D0

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 7A00     START:  MOV     R2,#00H
8008 : EA       LOOP:   MOV     A,R2
8009 : 23               RL      A               ; เปลี่ยนรหัสในการในแสดงผล
800A : D2E0             SETB    ACC.0           ; เลือกหลักในการแสดงผล
800C : 90E020           MOV     DPTR,#PORT_A
800F : F0               MOVX    @DPTR,A
8010 : 7D05             MOV     R5,#05H         ; กำหนดรอบในการหน่วงเวลา
8012 : 111A             ACALL   DELAY
8014 : 0A               INC     R2
8015 : B413F0           CJNE    A,#13H,LOOP
8018 : 80EC             SJMP    START

801A : 7EBE     DELAY:  MOV     R6,#190         ; เวลา = 12*(1/11.0592*10^6)*
801C : 7FF1     DELAY1: MOV     R7,#241         ;        (R5(1+R6(1+R7(2)+2)
801E : DFFE     DELAY2: DJNZ    R7,DELAY2       ;        +2)+2)
8020 : DEFA             DJNZ    R6,DELAY1       ;โปรแกรมหน่วงเวลา 100 มิลลิวินาที
8022 : DDF6             DJNZ    R5,DELAY
8024 : 22               RET
                        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. จงแก้ไขโปรแกรมที่ 5.2 โดยให้ 7-Segment Display หลัก D1 นับ 0-9 แทนหลัก D0
13. จงแก้ไขโปรแกรมที่ 5.2 โดยให้ 7-Segment Display หลัก D1 และ D0 นับ 0-9 พร้อมกัน
14. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

15. จงเขียนโปรแกรมนับเลข 0-99 (เพิ่มขึ้นทีละ 1) มาแสดงผลที่ 7-Segment Display
16. จงเขียนโปรแกรมนับเลข 99-0 (ลดลงทีละ 1) มาแสดงผลที่ 7-Segment Display
17. จงเขียนโปรแกรมอ่านค่าจาก Switch Input แล้วนำมาแสดงที่ 7-Segment Display โดยกำหนดให้หลัก D1 แสดงค่าสวิตช์ตำแหน่งที่ 4 ถึง 7 และหลัก D0 แสดงค่าสวิตช์ตำแหน่งที่ 0 ถึง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

การสร้างสัญญาณเสียงด้วย Buzzer

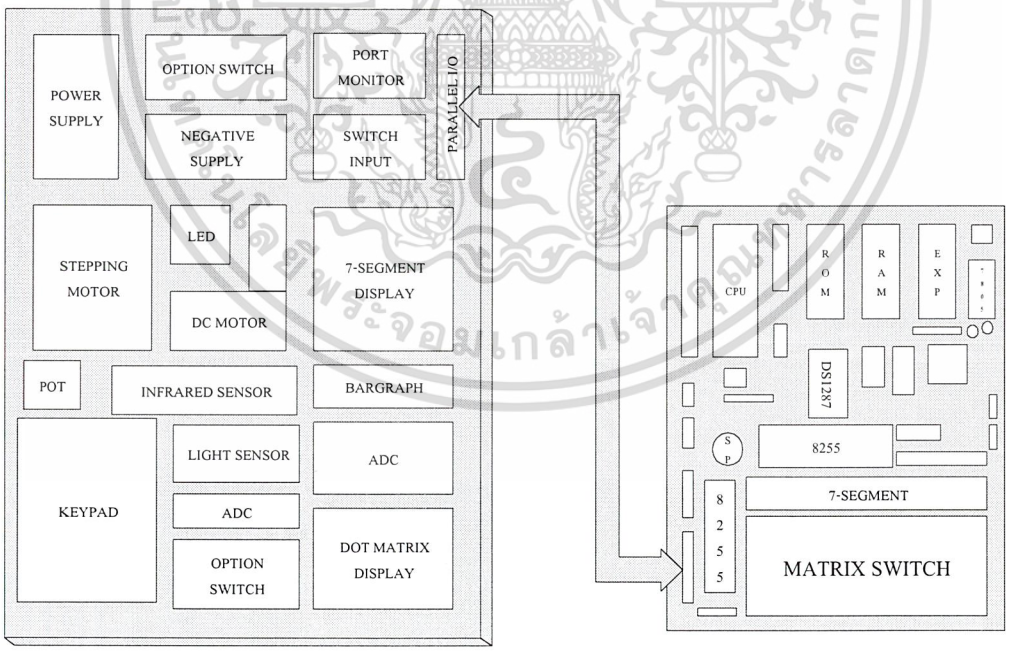
วัตถุประสงค์

1. เพื่อศึกษาวิธีการสร้างสัญญาณเสียงด้วย Buzzer
2. เพื่อสามารถใช้ไมโครคอนโทรลเลอร์ควบคุม Buzzer ให้สร้างสัญญาณเสียงได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 6 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 12 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

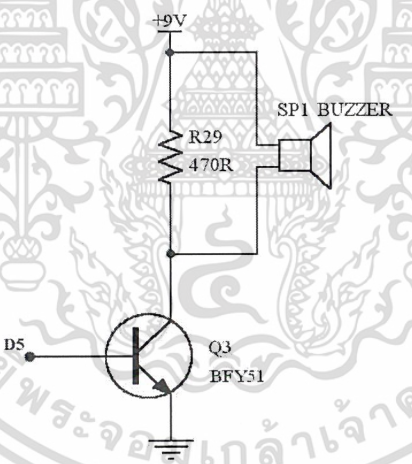
ทฤษฎีเบื้องต้น

บัซเซอร์ (Buzzer)

บัซเซอร์เป็นอุปกรณ์ที่สามารถสร้างสัญญาณเสียงออกมาได้ บัซเซอร์ที่ใช้อยู่ทั่วไปแบ่งออกเป็น 2 ชนิด คือ

1. TYPE A เป็นแบบที่ทำงานอิสระ เมื่อป้อนแรงดันไฟฟ้าให้กับตัวมันก็จะสามารถสร้างสัญญาณเสียงออกมาได้ สัญญาณที่ออกมาไม่สามารถเปลี่ยนแปลงได้
2. TYPE B เป็นแบบที่จะต้องสร้างสัญญาณพัลส์ (Pulse) ที่มีความถี่แตกต่างกันป้อนให้กับตัวบัซเซอร์ จึงจะสามารถสร้างสัญญาณเสียงออกมาได้ตามความถี่ที่ป้อนให้ บัซเซอร์ชนิดนี้โดยทั่วไปจะเรียกว่า “เปียซโซ” (Piezo)

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ใช้บัซเซอร์แบบ TYPE B เพราะฉะนั้นการที่จะสั่งให้ตัวบัซเซอร์สร้างสัญญาณเสียงออกมาได้จะต้องสร้างสัญญาณพัลส์ (Pulse) ที่มีความถี่ที่แตกต่างกันป้อนให้กับบัซเซอร์นี้



รูปที่ ข. 13 วงจรสร้างสัญญาณเสียงด้วย Buzzer

ลำดับขั้นตอนการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/4, SW 1/6, SW 1/7 และ SW 4/2 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “SWITCH INPUT”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 6.1 และสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ Buzzer

โปรแกรมที่ 6.1 โปรแกรมควบคุม Buzzer

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A
800B : 1111             ACALL   DELAY
800D : B2E5             CPL     ACC.5
800F : 80F9             SJMP    LOOP

8011 : 7FFF     DELAY:  MOV     R7,#0FFH        ; โปรแกรมหน่วงเวลา
8013 : 7EFF     DELAY1: MOV     R6,#0FFH
8015 : DEFE     DELAY2: DJNZ    R6,DELAY2
8017 : DFFA             DJNZ    R7,DELAY1
8019 : 22               RET
                        END
```

8. อธิบายการทำงานของโปรแกรมที่ 6.1

โปรแกรมที่ 6.2 โปรแกรมควบคุมการสร้างคลื่นเสียงของ Buzzer โดยใช้ Switch Input ตำแหน่งที่ 3 เป็นตัวควบคุมการสร้างคลื่นเสียง

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A
800B : 111D             ACALL   DELAY
800D : B2E5             CPL     ACC.5
800F : 0582             INC     DPL
8011 : C0E0             PUSH    ACC
8013 : E0       LOOP2:  MOVX    A,@DPTR         ; ตรวจสอบ Switch Input
8014 :30E3FC            JNB     ACC.3,LOOP2
8017 : D0E0             POP     ACC
8019 : 1582             DEC     DPL
801B : 80ED             SJMP    LOOP

801D : 7F80     DELAY:  MOV     R7,#80H         ; โปรแกรมหน่วงเวลา
801F : 7EFF     DELAY1: MOV     R6,#0FFH
8021 : DEFE     DELAY2: DJNZ    R6,DELAY2
8023 : DFFA             DJNZ    R7,DELAY1
8025 : 22               RET
                        END
```

9. บันทึกผลการทดลองและอธิบายการทำงานของโปรแกรมที่ 6.2
10. จงแก้ไขโปรแกรมที่ 6.2 โดยให้ Buzzer ส่งเสียง 5 วินาที และหยุดส่งเสียง 3 วินาที
11. ป้อนโปรแกรมที่ 6.3 และสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ Buzzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 6.3 โปรแกรมสร้างคลื่นเสียงความถี่ 500 Hz

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : E4               CLR     A
8007 : 90E020   START:  MOV     DPTR,#PORT_A
800A : F0       LOOP:   MOVX    @DPTR,A
800B : 7D0A             MOV     R5,#0AH
800D : 1113             ACALL   DELAY
800F : B2E5             CPL     ACC.5
8011 : 80F7             SJMP    LOOP

8013 : 7F2C     DELAY:  MOV     R7,#44          ; เวลา = 12*(1/11.0592*10^6)*
8015 : DFFE     DELAY1: DJNZ    R7,DELAY1       ;        (R5(1+R7(2)+2))
8017 : DDFA             DJNZ    R5,DELAY        ; โปรแกรมหน่วงเวลา 10 ไมโครวินาที
8019 : 22               RET
                        END
```

12. อธิบายการทำงานของโปรแกรมที่ 6.3
13. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

14. จงเขียนโปรแกรมสร้างคลื่นเสียงดังนี้
    14.1 ความถี่ 10 kHz
    14.2 ความถี่ 1 kHz
    14.3 คาบเวลา 5 มิลลิวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. จงเขียนโปรแกรมสร้างคลื่นเสียง โดยควบคุมการส่งเสียงจาก Switch Input ตำแหน่งที่ 7 และตำแหน่งที่ 6 มีลักษณะการควบคุมดังนี้

รูปแบบ	SW 5 บิตที่ 7	SW 5 บิตที่ 6
คงสภาวะเดิม	OFF	OFF
ดับ	OFF	ON
ติด	ON	OFF
คงสภาวะเดิม	ON	ON

16. จงเขียนโปรแกรมสร้างคลื่นเสียง โดยควบคุมการส่งเสียงจาก Switch Input ดังนี้
    16.1 Switch Input ตำแหน่งที่ 0-3 ควบคุมเวลาในการส่งเสียง
    16.2 Switch Input ตำแหน่งที่ 4-7 ควบคุมเวลาในการหยุดส่งเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7

การสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง

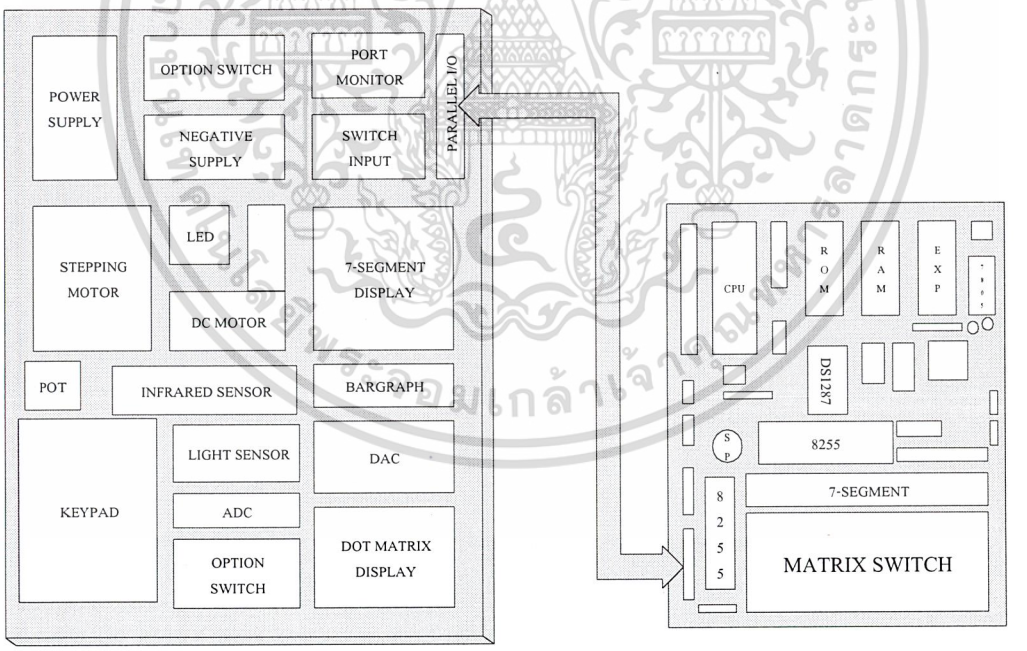
วัตถุประสงค์

1. สามารถใช้ไมโครคอนโทรลเลอร์ควบคุมการสร้างสัญญาณด้วยไดโอดเปล่งแสงได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 7 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



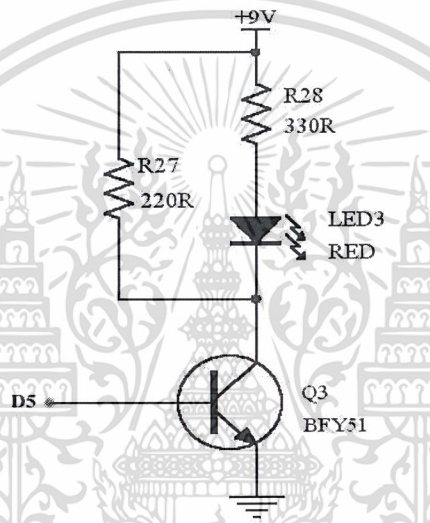
รูปที่ ข. 14 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

ชุดสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง

อุปกรณ์ที่ทำหน้าที่เปล่งแสงในที่นี้จะใช้หลอดไดโอดเปล่งแสง หรือ Light Emitting Diode โดยมีทรานซิสเตอร์ BFY51 เป็นตัวเปิด-ปิดวงจรที่ต่อเข้ากับแหล่งจ่ายไฟ 9 โวลต์ เมื่อให้บิตที่ 5 ของพอร์ต PA เป็น "1" ทรานซิสเตอร์จะได้รับไบแอสทำให้มีกระแสไหลผ่านหลอดไดโอดเปล่งแสง เมื่อมีกระแสไหลผ่านตัวหลอดไดโอดเปล่งแสงจะทำให้หลอดไดโอดเปล่งแสงเปล่งแสงสว่างขึ้น ซึ่งจะคล้ายกับการทำงานของหลอดไดโอดเปล่งแสงบน Port Monitor



รูปที่ ข. 15 วงจรชุดสร้างสัญญาณแสดงผลด้วยหลอดไดโอดเปล่งแสง

ลำดับขั้นตอนการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "OFF"
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง "OFF" ยกเว้น SW 1/2 "ON"
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/3, SW 1/6, SW 1/7 และ SW 4/2 มาที่ตำแหน่ง "ON"
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง "SWITCH INPUT"
6. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "ON"
7. ป้อนโปรแกรมที่ 7.1 และสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 7.1 โปรแกรมควบคุมการติดและดับของ LED

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A
800B : 1111             ACALL   DELAY
800D : B2E5             CPL     ACC.5
800F : 80F9             SJMP    LOOP

8011 : 7F80     DELAY:  MOV     R7,#80H         ; โปรแกรมหน่วงเวลา
8013 : 7EFF     DELAY1: MOV     R6,#0FFH
8015 : DEFE     DELAY2: DJNZ    R6,DELAY2
8017 : DFFA             DJNZ    R7,DELAY1
8019 : 22               RET
                        END
```

8. อธิบายการทำงานของโปรแกรมที่ 7.1
9. จงแก้ไขโปรแกรมที่ 7.1 โดยให้ LED ติด 8 วินาที และดับ 4 วินาที
10. ป้อนโปรแกรมที่ 7.2 และสั่งให้โปรแกรมทำงาน และบันทึกผลที่เกิดขึ้นกับ LED

โปรแกรมที่ 7.2 โปรแกรมควบคุมการติดและดับของ LED โดย Switch Input ตำแหน่งที่ 6

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : E4               CLR     A
800A : F0       LOOP:   MOVX    @DPTR,A
800B : 111D             ACALL   DELAY
800D : B2E5             CPL     ACC.5
800F : 0582             INC     DPL
8011 : C0E0             PUSH    ACC
8013 : E0       LOOP2:  MOVX    A,@DPTR
8014 : 30E6FC           JNB     ACC.6,LOOP2
8017 : D0E0             POP     ACC
8019 : 1582             DEC     DPL
801B : 80ED             SJMP    LOOP

801D : 7F80     DELAY:  MOV     R7,#80H         ; โปรแกรมหน่วงเวลา
801F : 7EFF     DELAY1: MOV     R6,#0FFH
8021 : DEFE     DELAY2: DJNZ    R6,DELAY2
8023 : DFFA             DJNZ    R7,DELAY1
8025 : 22               RET
                        END
```

11. อธิบายการทำงานของโปรแกรมที่ 7.2
12. สรุปและวิจารณ์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฝึกหัด

13. จงเขียนโปรแกรมควบคุมหลอดไดโอดเปล่งแสง โดยมีรูปแบบดังนี้

รูปแบบ	SW 5 บิตที่ 7	SW 5 บิตที่ 0
ดับ	OFF	OFF
ติดนานกว่าดับ	OFF	ON
ดับนานกว่าติด	ON	OFF
ติด	ON	ON

14. จงเขียนโปรแกรมควบคุมหลอดไดโอดเปล่งแสง จาก Switch Input ดังนี้

14.1 Switch Input ตำแหน่งที่ 0-3 ควบคุมเวลาในการติดของหลอดไดโอดเปล่งแสง

14.2 Switch Input ตำแหน่งที่ 4-7 ควบคุมเวลาในการดับของหลอดไดโอดเปล่งแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 8

การควบคุมมอเตอร์กระแสตรง (DC Motor)

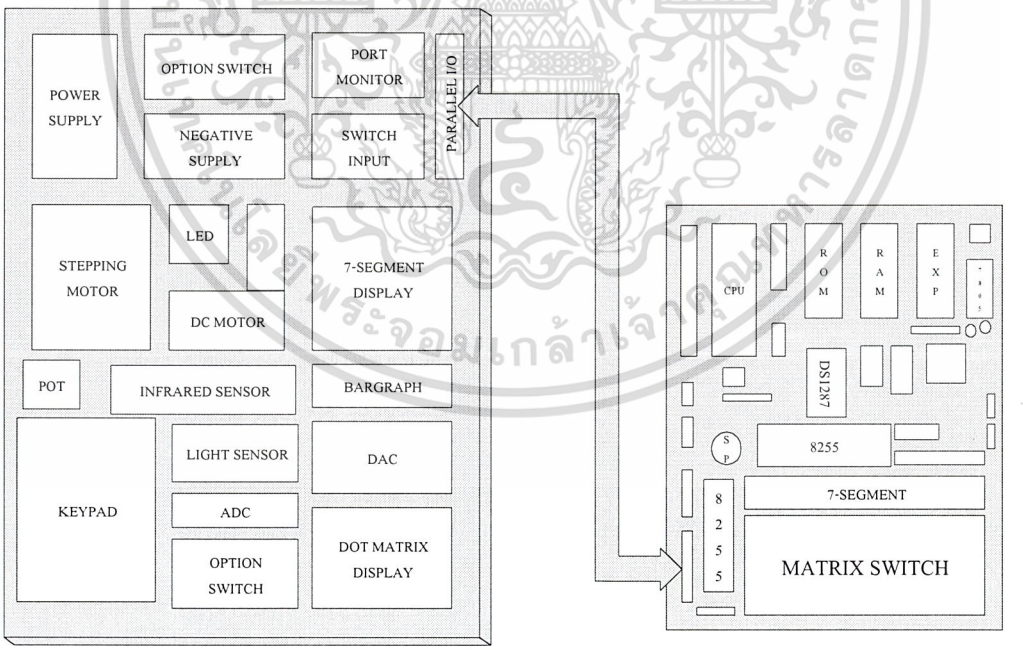
วัตถุประสงค์

1. เพื่อศึกษาการควบคุมมอเตอร์กระแสตรง ด้วยไมโครคอนโทรลเลอร์
2. เพื่อสามารถใช้ไมโครคอนโทรลเลอร์ควบคุมมอเตอร์กระแสตรงได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 8 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



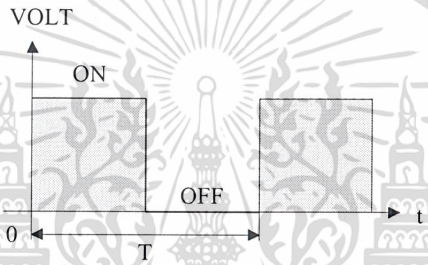
รูปที่ ข. 16 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

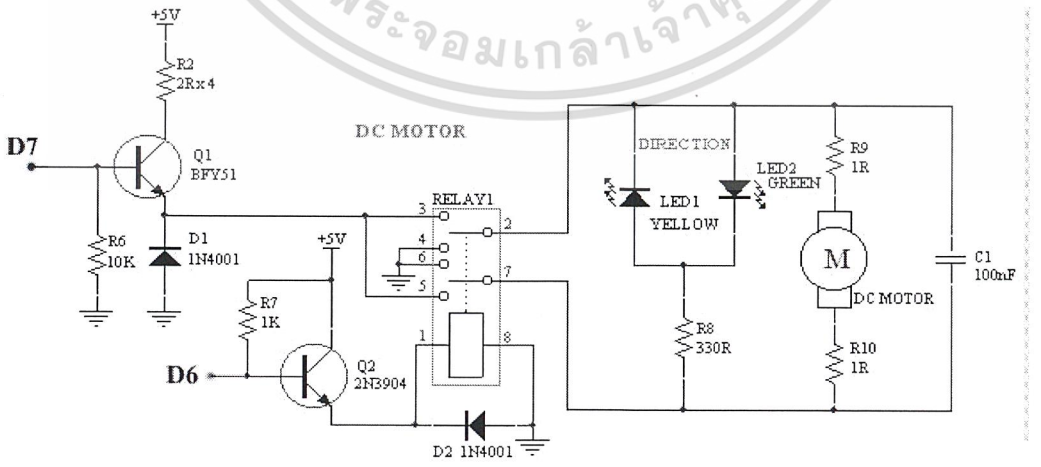
มอเตอร์กระแสตรง (DC Motor)

การควบคุมความเร็วของมอเตอร์ ให้ความเร็วคงที่ตามที่เราต้องการเป็นเรื่องที่ทำได้ค่อนข้างยาก ถ้าใช้การควบคุมแบบธรรมดาทั่วไป คือ การควบคุมแรงดันที่จ่ายให้กับมอเตอร์ การจะควบคุมความเร็วของมอเตอร์ให้ได้ความเร็วที่คงที่ ต้องควบคุมกระแสที่จ่ายให้กับมอเตอร์ ซึ่งวิธีนี้เรียกว่า การควบคุมแบบ Pulse Width Modulation (PWM) เป็นการควบคุมกระแสที่จ่ายให้กับมอเตอร์ โดยการจ่ายแรงดันให้กับมอเตอร์เป็นช่วง (รูปที่ ข. 17) กระแสไฟฟ้าที่มอเตอร์ได้รับจะเป็นค่าเฉลี่ยที่เกิดขึ้นในช่วง “ON” โดยความถี่ของการ “ON” และ “OFF” คงที่



รูปที่ ข. 17 การจ่ายแรงดันให้กับมอเตอร์เป็นช่วงๆ

เมื่อความถี่คงที่ เราสามารถที่จะเปลี่ยนความเร็วรอบของมอเตอร์ได้โดยทำให้ช่วง “ON” มีเวลาที่มากกว่าช่วง “OFF” กระแสเฉลี่ยที่มอเตอร์ได้รับก็จะมีค่ามากขึ้น ความเร็วรอบของมอเตอร์ก็จะเพิ่มขึ้น และถ้าหากต้องการลดความเร็วรอบของมอเตอร์ลง ก็ให้ลดช่วง “ON” ให้มีเวลาน้อยดังแสดงในรูป



รูปที่ ข. 18 วงจรควบคุมมอเตอร์กระแสตรง (DC Motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณควบคุมจะถูกส่งมาจากชุดไมโครคอนโทรลเลอร์ บิตที่ 7 (PA_7) จะเป็นบิตควบคุมการทำงานของมอเตอร์ โดยส่งแรงดันมาไบแอสทรานซิสเตอร์ BFY51 เพื่อขยายกระแสที่ใช้ขับให้มอเตอร์หมุน เมื่อสัญญาณที่จ่ายเข้ามีสภาวะเป็น "1" ทรานซิสเตอร์ BFY51 ก็จะต่อวงจรทำให้มอเตอร์หมุน และถ้าสัญญาณเป็น "0" ทรานซิสเตอร์ก็จะเปิดวงจรทำให้มอเตอร์หยุดหมุน

ระหว่างทรานซิสเตอร์กับมอเตอร์ จะมีรีเลย์ต่อคั่นอยู่ซึ่งจะทำหน้าที่เป็นตัวเปลี่ยนทิศทางการหมุนของมอเตอร์โดยอาศัยบิต 6 (PA_6) เป็นตัวควบคุมการไบแอสทรานซิสเตอร์ 2N3904 สั่งให้รีเลย์เปิดหรือปิด ดังนั้นวงจรควบคุมมอเตอร์ชุดนี้จะต้องมีสัญญาณมาควบคุมทั้งความเร็ว และทิศทางการหมุนมอเตอร์

ตารางที่ ข. 4 บิตควบคุมการทำงานและทิศทางการหมุนของมอเตอร์

บิตที่ 7	บิตที่ 6	การทำงานของมอเตอร์	CODE
0	0	หยุดหมุน	00H
0	1	หยุดหมุน	40H
1	0	ทวนเข็มนาฬิกา	80H
1	1	ตามเข็มนาฬิกา	C0H

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "OFF"
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง "OFF" ยกเว้น SW 1/2 "ON"
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6, SW 1/7 และ SW 4/3 มาที่ตำแหน่ง "ON"
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง "SWITCH INPUT"
6. จ่ายไฟเลี้ยงให้กับซิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "ON"
7. ป้อนโปรแกรมที่ 8.1 แล้วสั่งโปรแกรมทำงานและสังเกตทิศทางการหมุนของมอเตอร์

โปรแกรมที่ 8.1 การควบคุมทิศทางของมอเตอร์

```
PORT_A  EQU   0E020H
PORT_B  EQU   PORT_A+1
PORT_C  EQU   PORT_A+2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            C_PORT EQU   PORT_A+3
                   ORG    8000H
8000 : 90E023  INIT:  MOV    DPTR,#C_PORT
8003 : 748A           MOV    A,#08AH
8005 : F0             MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E020  START: MOV    DPTR,#PORT_A
8009 : 7480           MOV    A,#80H           ; สั่งให้มอเตอร์ทำงาน
800B : F0             MOVX   @DPTR,A
800C : 80FE           SJMP   $
                      END
```

8. จงเพิ่มโปรแกรมเพื่อให้มอเตอร์หยุดหมุน เมื่อมอเตอร์หมุนได้ระยะเวลาหนึ่ง
9. จงแก้ไขโปรแกรมที่ทำให้มอเตอร์หมุนในทิศทางที่ตรงข้ามกับโปรแกรมข้อที่ 8
10. สามารถสรุปได้ว่า บิต_____ เป็นบิตควบคุมให้มอเตอร์ทำงาน
    บิต_____ เป็นบิตควบคุมการหมุนของมอเตอร์
11. ป้อนโปรแกรมที่ 8.2 แล้วสั่งโปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง

โปรแกรมที่ 8.2 การควบคุมมอเตอร์ด้วย Switch Input ตำแหน่งที่ 0 และตำแหน่งที่ 1

```
            PORT_A EQU   0E020H
            PORT_B EQU   PORT_A+1
            PORT_C EQU   PORT_A+2
            C_PORT EQU   PORT_A+3
                   ORG    8000H
8000 : 90E023  INIT:  MOV    DPTR,#C_PORT
8003 : 748A           MOV    A,#08AH
8005 : F0             MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E021  START: MOV    DPTR,#PORT_B
8009 : E0             MOVX   A,@DPTR
800A : 1582           DEC    DPL
800C : 03             RR     A                ; เลื่อนบิตเพื่อให้ตรงกับบิตควบคุม
800D : 03             RR     A
800E : F0             MOVX   @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
800F : 80F5          SJMP    START
                     END
```

12. จงอธิบายการทำงานของโปรแกรมที่ 8.2
13. สามารถสรุปได้ว่า บิต _____เป็นบิตควบคุมให้มอเตอร์ทำงาน
    บิต _____ เป็นบิตควบคุมการหมุนของมอเตอร์
14. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

15. จงเขียนโปรแกรมสั่งให้มอเตอร์หมุนตามเข็มและทวนเข็มนาฬิกา ครั้งละประมาณ 5 วินาที
16. จงเพิ่มเติมโปรแกรมในข้อที่ 10. โดยมีการให้มอเตอร์หยุดหมุนก่อนที่จะมีการเปลี่ยนทิศทางการหมุนประมาณ 3 วินาที
17. จงเขียนโปรแกรมควบคุมมอเตอร์จาก Switch Input โดยมีลักษณะการควบคุมดังนี้

BIT 7	BIT 6	การทำงานของมอเตอร์
OFF	OFF	หยุดทำงาน
OFF	ON	หยุดทำงาน
ON	OFF	หมุนทวนเข็มนาฬิกา
ON	ON	หมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 9

การควบคุมสเต็ปปิ้งมอเตอร์ (Stepping Motor)

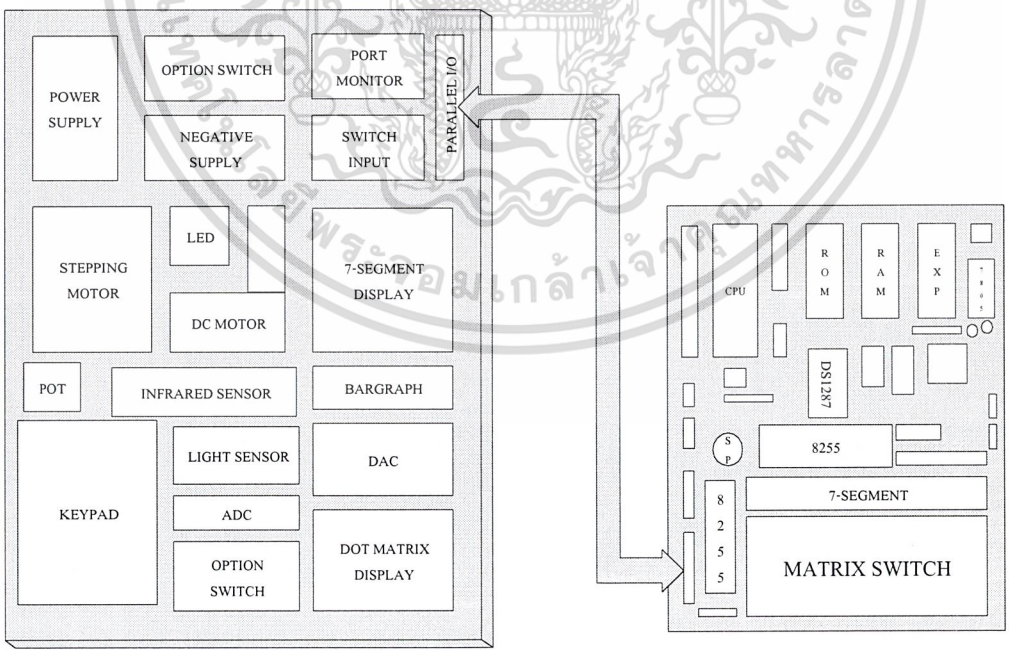
วัตถุประสงค์

1. เพื่อศึกษาการกระตุ้นเฟสของสเต็ปปิ้งมอเตอร์
2. เพื่อศึกษาการควบคุมสเต็ปปิ้งมอเตอร์ ด้วยไมโครคอนโทรลเลอร์
3. เพื่อให้สามารถใช้ไมโครคอนโทรลเลอร์ควบคุมสเต็ปปิ้งมอเตอร์ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 9 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 19 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

สเต็ปปิ้งมอเตอร์ (Stepping Motor)

สเต็ปปิ้งมอเตอร์ เป็นมอเตอร์ที่เหมาะสมสำหรับงานควบคุมการหมุน ที่ต้องการตำแหน่งที่แน่นอน สเต็ปปิ้งมอเตอร์ต่างจากดีซีมอเตอร์ตรงที่การหมุนไปทีละสเต็ป อาจเป็นสเต็ปละ 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ตามแต่ชนิด แต่ดีซีมอเตอร์จะหมุนแบบต่อเนื่อง ไม่สามารถหมุนเป็นสเต็ปได้

ข้อดีของสเต็ปปิ้งมอเตอร์เมื่อเปรียบเทียบกับดีซีมอเตอร์

1. การควบคุมสามารถที่จะควบคุมแบบลูปเปิด (Open loop) ได้โดยไม่ต้องอาศัยตัวตรวจจับการหมุน
2. ไม่ต้องใช้แปรงถ่าน ทำให้ไม่มีส่วนที่สึกหรอ และปัญหาเรื่องการสปาร์คที่ทำให้เกิดสัญญาณรบกวน
3. การควบคุมโดยไมโครคอนโทรลเลอร์ทำได้ง่าย

สเต็ปปิ้งมอเตอร์แบ่งออกเป็น 3 ชนิด คือ

1. Variable Reluctance (VR) เป็นการหมุนได้อิสระในขณะที่ไม่ได้จ่ายไฟให้ โรเตอร์จะทำด้วยเหล็กอ่อน มีลักษณะเป็นฟันเลื่อยรูปทรงกระบอก จะมี Step Angle ขนาด 7.5 องศา หรือ 15 องศา
2. Permanent Magnet (PM) โรเตอร์จะทำด้วยแม่เหล็กถาวร เป็นรูปทรงกระบอกฟันเลื่อย ที่ตัวสเตเตอร์ (Stator) จะมีขดลวดพันอยู่ ตัวโรเตอร์จะหมุนไม่เป็นอิสระในขณะที่ไม่ได้ป้อนกระแสไฟฟ้า เพราะแรงดูดจากแม่เหล็กถาวร
3. Hybrid เป็นสเต็ปปิ้งมอเตอร์ที่นำเอาข้อดีของแบบ VR และ PM เข้ามารวมกันไว้จะมีตัว สเตเตอร์คล้ายกับแบบ VR ที่โรเตอร์จะมีแม่เหล็กกำลังสูงติดอยู่ ทำให้แรงบิดสูง มีความแม่นยำในการหมุนสูง และมีขนาดกระทัดรัด ขนาดของ Step Angle จะเป็น 0.9 องศา และ 1.8 องศา

การกระตุ้นเฟส (Phase Excitation)

สเต็ปปิ้งมอเตอร์จะทำงานได้ต้องอาศัยการจ่ายกระแสไฟฟ้าให้กับขดลวดสเตเตอร์ของแต่ละเฟสซึ่งมีลักษณะเป็นพัลส์ (Pulse) ซึ่งเรียกว่า “การกระตุ้นเฟส”

การกระตุ้นเฟสแบ่งเป็น 3 แบบ คือ

1. การกระตุ้นเฟสเดียว (1 Phase Excitation) เป็นการจ่ายกระแสไฟฟ้าให้กับขดลวดสเตเตอร์ครั้งละเฟสสลับกันไป แบบนี้กำลังงานสูญเสียที่เกิดขึ้นจะต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การกระตุ้นสองเฟส (2 Phase Excitation) เป็นการจ่ายกระแสไฟฟ้าให้กับขดลวดสเตเตอร์ครั้งละสองเฟสเท่ากัน แบบนี้กำลังงานสูญเสียที่เกิดขึ้นจะเป็นสองเท่าของแบบกระตุ้นเฟสเดียว
3. การกระตุ้นหนึ่งเฟสและสองเฟส (1-2 Phase Excitation) เป็นการจ่ายกระแสไฟฟ้าให้กับขดลวดสเตเตอร์ทีละ 1 และ 2 เฟสสลับกันไป แบบนี้จะให้การหมุนเป็นแบบ Half Step

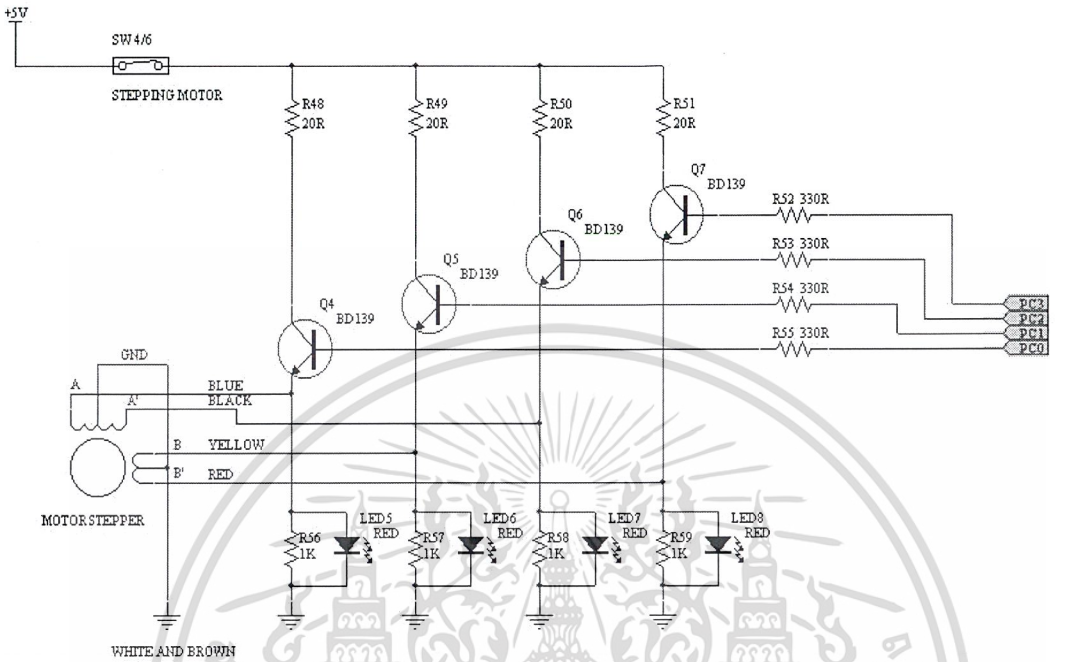
ตารางที่ ข. 5 การกระตุ้นเฟสทั้ง 3 แบบ

Phase Driving Method	Phase-1 Excitation				Phase-2 Excitation				Phase 1-2 Excitation							
	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8
Phase A	1				1	1			1	1	1					
Phase B		1				1	1				1	1	1			
Phase A'			1				1	1					1	1	1	
Phase B'				1	1			1	1						1	1

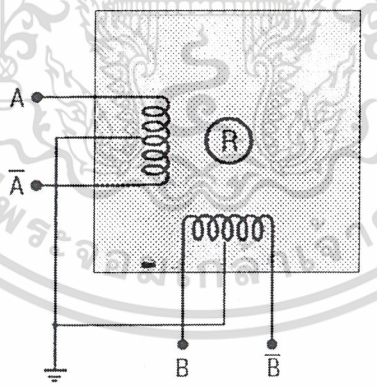
<u>การควบคุมสเต็ปปิ้งมอเตอร์</u>

การควบคุมเต็ปปิ้งมอเตอร์ ก็คือการควบคุมกระแสไฟที่จ่ายให้กับขดลวดสเตเตอร์แต่ละขด สำหรับชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีวงจรควบคุมการจ่ายกระแสไฟไว้สำหรับสเต็ปปิ้งมอเตอร์ 2 Phase 6 Wire วงจรจะจ่ายกระแสให้กับขดลวดได้สูงสุดประมาณ 800 mA ต่อเฟส วงจรควบคุมจะต่อเข้ากับพอร์ต PC_0-PC_3 โดยที่ PC_0 ต่อเข้ากับเฟส A, PC_1 ต่อเข้ากับเฟส A', PC_2 ต่อเข้ากับเฟส B และ PC_3 ต่อเข้ากับเฟส B' แต่ละเฟสจะมี LED เพื่อแสดงสภาวะของแต่ละเฟส เมื่อเอาต์พุตของพอร์ต PC มีค่าเป็น "1" ขดลวดสเตเตอร์ของสเต็ปปิ้งมอเตอร์จะได้รับกระแสไฟฟ้า หลอดไดโอดเปล่งแสงจะติดสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 20 วงจรควบคุมสเต็ปปิ้งมอเตอร์ (Stepping Motor)



รูปที่ ข. 21 ลักษณะภายในของสเต็ปปิ้งมอเตอร์แบบ 2 Phase 6 Wire

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ปรับสวิตช์ SW 4/6 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “SWITCH INPUT”
6. จ่ายไฟเลี้ยงให้กับชิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 9.1 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการหมุนของสเต็ปปิ้งมอเตอร์

โปรแกรมที่ 9.1 การกระตุ้นทีละเฟส (1 Phase Excitation)

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 7A00     START:  MOV     R2,#00H
8008 : 908100   LOOP1:  MOV     DPTR,#TABLE
800B : 8A82             MOV     DPL,R2
800D : E0               MOVX    A,@DPTR
800E : 0A               INC     R2
800F : 90E022           MOV     DPTR,#PORT_C
8012 : F0               MOVX    @DPTR,A         ; สั่งให้มอเตอร์หมุนไปตามขั้น
8013 : 111C             ACALL   DELAY
8015 : E4               CLR     A
8016 : F0               MOVX    @DPTR,A         ; หยุดการหมุนของ Stepping Motor
8017 : BA04EE           CJNE    R2,#04H,LOOP1
801A : 80EA             SJMP    START

801C : 7E00     DELAY:  MOV     R6,#00          ; โปรแกรมหน่วงเวลา
801E : 7F00     DELAY1: MOV     R7,#00
8020 : DFFE     DELAY2: DJNZ    R7,DELAY2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8022 : DEFA              DJNZ   R6,DELAY1
8024 : 22                RET

                         ORG    8100H
8100 : 01020408  TABLE:  DB     01H,02H,04H,08H
                         END
```

8. จงอธิบายการทำงานของโปรแกรมที่ 9.2
9. ป้อนโปรแกรมที่ 9.2 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการหมุนของสเต็ปปิ้งมอเตอร์

โปรแกรมที่ 9.2 การกระตุ้นทีละ 2 เฟส (2 Phase Excitation)

```
          PORT_A  EQU    0E020H
          PORT_B  EQU    PORT_A+1
          PORT_C  EQU    PORT_A+2
          C_PORT  EQU    PORT_A+3
                  ORG    8000H
8000 : 90E023  INIT:  MOV    DPTR,#C_PORT
8003 : 748A           MOV    A,#08AH
8005 : F0             MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 7A00    START: MOV    R2,#00H
8008 : 908100  LOOP1: MOV    DPTR,#TABLE
800B : 8A82           MOV    DPL,R2
800D : E0             MOVX   A,@DPTR
800E : 0A             INC    R2
800F : 90E022         MOV    DPTR,#PORT_C
8012 : F0             MOVX   @DPTR,A          ; สั่งให้มอเตอร์หมุนไปตามขั้น
8013 : 111C           ACALL  DELAY
8015 : E4             CLR    A
8016 : F0             MOVX   @DPTR,A          ; หยุดการหมุนของ Stepping Motor
8017 : BA04EE         CJNE   R2,#04H,LOOP1
801A : 80EA           SJMP   START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
801C : 7E00     DELAY:  MOV   R6,#00          ; โปรแกรมหน่วงเวลา
801E : 7F00     DELAY1: MOV   R7,#00
8020 : DFFE     DELAY2: DJNZ  R7,DELAY2
8022 : DEFA             DJNZ  R6,DELAY1
8024 : 22               RET

                        ORG   8100H
8100 : 01020408 TABLE:  DB    03H,06H,0CH,09H
                        END
```

10. จงอธิบายการทำงานของโปรแกรมที่ 9.2
11. จงแก้ไขให้สเต็ปปิ้งมอเตอร์หมุนตรงข้ามกับโปรแกรมที่ 9.2
12. ทดลองแก้ไขค่าใน TABLE (แอดเดรส 8100H) เป็น

    01H,03H,02H,06H,04H,0CH,08H,09H

    และเปลี่ยนคำสั่ง CJNE R2,#04H,LOOP1 เป็น

    CJNE R2,#08H,LOOP1
13. ถ้าต้องการให้สเต็ปปิ้งมอเตอร์ในโปรแกรมข้อที่ 12. หมุนตรงข้ามกัน จะต้องแก้ไขโปรแกรมอย่างไร
14. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

15. จงเขียนโปรแกรม เพื่อกำหนดให้สเต็ปปิ้งมอเตอร์ หมุนตามเข็มนาฬิกาจำนวน 5 รอบ แล้วหยุด (กระตุ้นทีละ 2 เฟส)
16. จงเขียนโปรแกรม เพื่อกำหนดให้สเต็ปปิ้งมอเตอร์ หมุนทวนเข็มนาฬิกาจำนวน 10 รอบแล้วหมุนตามเข็มนาฬิกา 5 รอบ และหยุดเป็นเวลา 3 วินาที (กระตุ้นเฟสแบบข้อที่ 12.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 10

การแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display)

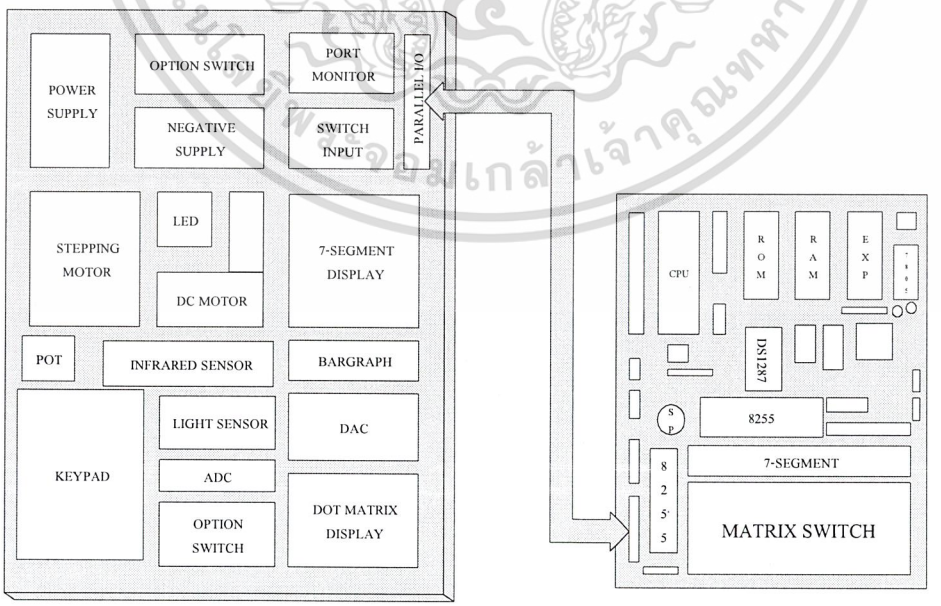
วัตถุประสงค์

1. เพื่อศึกษาการใช้งานชุดแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display)ในการแสดงผล
2. เพื่อศึกษาการใช้ไมโครคอนโทรลเลอร์มาควบคุมการแสดงผล โดยใช้ชุดแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display)
3. เพื่อสามารถใช้ไมโครคอนโทรลเลอร์ควบคุมการแสดงผล โดยใช้ชุดแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display) ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 10 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

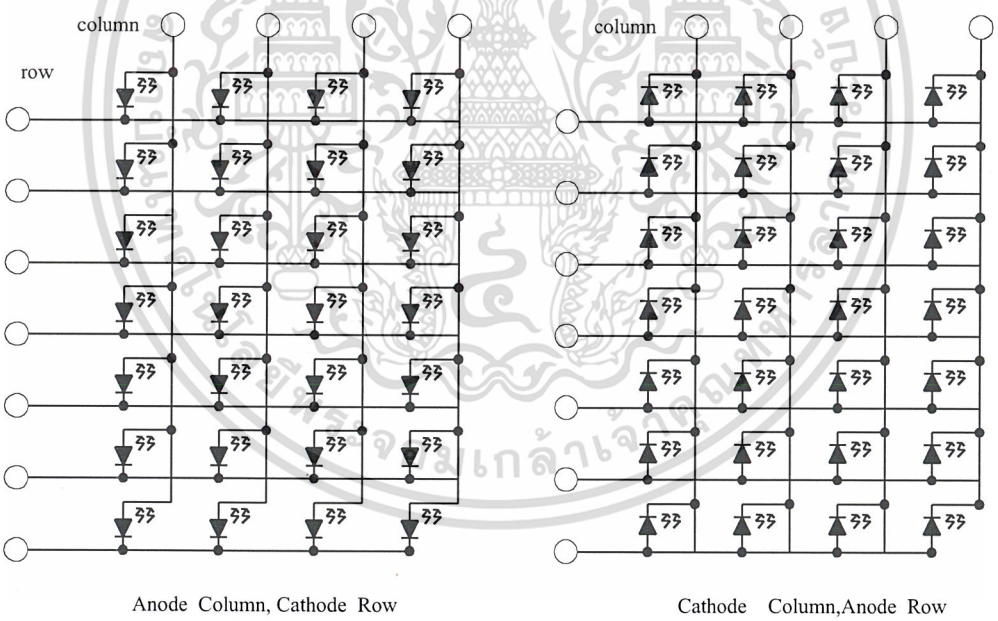
รูปที่ ข. 22 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

ทฤษฎีเบื้องต้น

การแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display)

หลอดหลอดไดโอดเปล่งแสงที่เห็นอยู่ทั่วไปจะมีลักษณะเป็นหลอดเดี่ยวๆ จะใช้เป็นตัวแสดงผลแบบติด-ดับ (ON-OFF) แต่ถ้านำเอาหลอดหลอดไดโอดเปล่งแสงเหล่านี้มารวมกันให้เป็นกลุ่มแถวในแนวนอน และแนวตั้งแล้ว รูปแบบของการแสดงผลก็จะเปลี่ยนไป การติด-ดับของหลอดหลอดไดโอดเปล่งแสงสามารถที่จะสร้างให้เกิดเป็นแบบจุดขนาด 5X7 หรือรูปลักษณ์ต่างๆ ตามที่เราต้องการได้

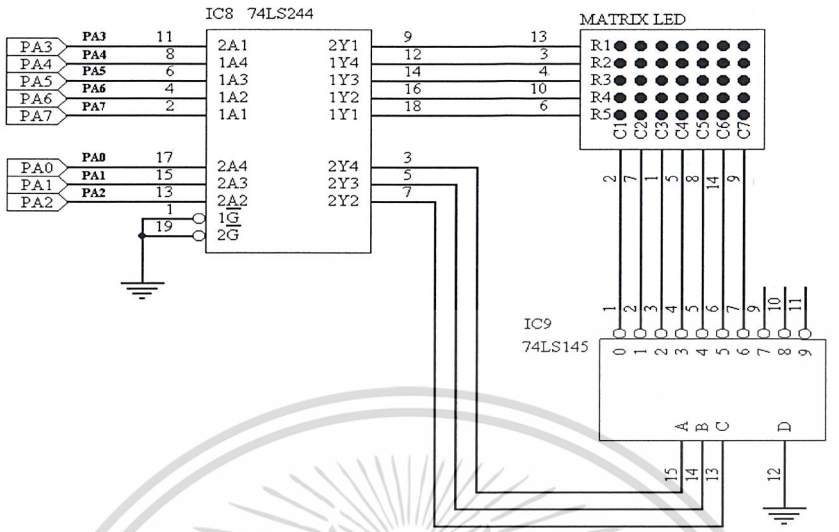
กลุ่มของหลอดที่ว่านี้มีชื่อเรียกว่าชุดแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display) ซึ่งจะมีอยู่ด้วยกันหลายแบบหลายขนาดเช่น 4x4, 5x7, 5x8, 7x5, 8x8, 16x16 มีทั้งแบบสีเดียว (Single Color) หรือแบบหลายสี (Multi Color) และยังแบ่งตามการใช้งานเป็นแบบ Anode Column, Cathode Row และ Cathode Column, Anode Row



รูปที่ ข. 23 ชุดแสดงผลแบบจุดขนาด 5X7

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ใช้ชุดแสดงผลแบบจุดขนาด 5X7 (DOT Matrix Display) แบบ Cathode Column, Anode Row สีเขียวจะให้ค่ากำลังส่องสว่างประมาณ 400 μcd แต่ถ้าเป็นสีแดงจะให้ค่ากำลังส่องสว่างประมาณ 1200 μcd DOT Matrix จะต่อเข้ากับพอร์ต PA โดยบิต 0-2 เป็นบิตกำหนด Column บิต 3-7 เป็นบิตกำหนด Row

เอกสารนี้มีลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 24 วงจรชุดแสดงผลแบบจุดขนาด 5X7

จากวงจรบิต 0-7 จะต่อผ่านบัฟเฟอร์ (74LS244) แยกบิต 0-2 เข้า 74LS145 ซึ่งเป็นไอซี BCD to Decimal Decoder (4 to 10) แต่จะป้อนอินพุตให้ที่ขา A, B และ C ส่วนขา D จะต่อลงกราวด์เอาไว้ทำให้ 74LS145 นับแค่ 0-6 เท่านั้น 74LS145 จะทำหน้าที่เหมือนสวิตช์คอยต่อให้ Column ของ DOT Matrix ลงกราวด์ทีละ Column ตามอินพุตที่ป้อนเข้ามา

ตารางที่ ข. 6 การกำหนด Column

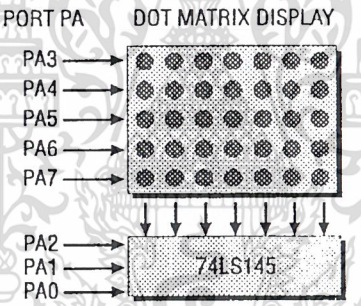
อินพุตของ 74LS145			Column ที่ถูกต่อลงกราวด์								CODE
C	B	A									
PA2	PA1	PA0	C1	C2	C3	C4	C5	C6	C7	C8	
0	0	0	⊗								00
0	0	1		⊗							01
0	1	0			⊗						02
0	1	1				⊗					03
1	0	0					⊗				04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ข. 6 การกำหนด Column (ต่อ)

อินพุตของ 74LS145			Column ที่ถูกต่อลงกราวด์								CODE
C	B	A									
PA2	PA1	PA0	C1	C2	C3	C4	C5	C6	C7	C8	
1	0	1						⊗			05
1	1	0							⊗		06
1	1	1								⊗	07

ส่วนบิต 3-7 เป็นบิตข้อมูลของ Row เมื่อต้องการให้ Row ใดติดก็กำหนดให้บิตของ Row นั้นเป็น "1" นำรหัสที่ได้มารวมกันเพื่อใช้เป็นข้อมูลที่จะส่งออกมาทางพอร์ต PA



รูปที่ ข. 25 ลักษณะการต่อพอร์ตกับชุดแสดงผลแบบจุดขนาด 5X7

เช่นต้องการให้ Column 5, Row 1-5 ติดสว่าง

Column 5 ที่จะต้องถูกต่อลงกราวด์ รหัส = 04H

Row 1-5 ติดหมด รหัส = F8H (1 1 1 1 1 X X X)

ข้อมูลที่จะถูกส่งออกมาที่พอร์ต PA คือ 04H + F8H = FCH

ด้วยวิธีนี้จะเห็นได้ว่าเราสามารถที่จะสั่งให้หลอดไดโอดเปล่งแสงหลอดใดหลอดหนึ่งในจำนวน 35 หลอด ที่มีอยู่ติดสว่างได้ตามต้องการ แต่ไม่สามารถที่จะสั่งให้หลอดไดโอดเปล่งแสงทุกหลอดติดสว่างพร้อมๆ กันได้ จะต้องอาศัยเทคนิคในการสั่งให้ติดทีละ Column ต่อเนื่องกันไปด้วยความเร็วที่มากกว่า 24 ครั้งต่อวินาที จะทำให้ดูเหมือนว่าหลอดหลอดไดโอดเปล่งแสงทุกหลอดติดสว่างพร้อมกันหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6 และ SW 1/8 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับชิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
6. ป้อนโปรแกรมที่ 10.1 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง

โปรแกรมที่ 10.1 การสั่งชุดแสดงผลแบบจุดขนาด 5X7 ให้ติด-ดับ

```
                PORT_A  EQU    0E020H
                PORT_B  EQU    PORT_A+1
                PORT_C  EQU    PORT_A+2
                C_PORT  EQU    PORT_A+3
                        ORG    8000H
8000 : 90E023   INIT:   MOV    DPTR,#C_PORT
8003 : 748A             MOV    A,#08AH
8005 : F0               MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV    DPTR,#PORT_A
8009 : E4       LOOP:   CLR    A
800A : F0               MOVX   @DPTR,A
800B : 1114             ACALL  DELAY
800D : D2E3             SETB   ACC.3
800F : F0               MOVX   @DPTR,A          ; ส่งค่าที่ต้องการให้จุดสว่าง
8010 : 1114             ACALL  DELAY
8012 : 80F5             SJMP   LOOP

8014 : 7E00     DELAY:  MOV    R6,#00           ; โปรแกรมหน่วงเวลา
8016 : 7F00     DELAY1: MOV    R7,#00
8018 : DFFE     DELAY2: DJNZ   R7,DELAY2
801A : DEFA             DJNZ   R6,DELAY1
801C : 22               RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    END
```

7. จงเขียนโปรแกรมสั่งให้จุดด้านขวา-ตรงกลางติด - ดับ
8. จงเขียนโปรแกรมสั่งให้จุดที่อยู่ตรงกลางติด - ดับ
9. ป้อนโปรแกรมที่ 10.2 สั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง

โปรแกรมที่ 10.2 การควบคุมชุดแสดงผลแบบจุดขนาด 5X7

```
                PORT_A  EQU    0E020H
                PORT_B  EQU    PORT_A+1
                PORT_C  EQU    PORT_A+2
                C_PORT  EQU    PORT_A+3
                        ORG    8000H
8000 : 90E023   INIT:   MOV    DPTR,#C_PORT
8003 : 748A             MOV    A,#08AH
8005 : F0               MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV    DPTR,#PORT_A
8009 : 7410     LOOP1:  MOV    A,#00010000B
800B : F0       LOOP2:  MOVX   @DPTR,A
800C : 1114             ACALL  DELAY
800E : 04               INC    A
800F : B417F9           CJNE   A,#00010111B,LOOP2
8012 : 80F5             SJMP   LOOP1

8014 : 7E00     DELAY:  MOV    R6,#00           ; โปรแกรมหน่วงเวลา
8016 : 7F00     DELAY1: MOV    R7,#00
8018 : DFFE     DELAY2: DJNZ   R7,DELAY2
801A : DEFA             DJNZ   R6,DELAY1
801C : 22               RET
                        END
```

10. จงแก้ไขโปรแกรมที่ 10.2 ให้ชุดแสดงผลแบบจุดขนาด 5X7 แถวที่ 1, 3 และ 5 วิ่งพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ป้อนโปรแกรมที่ 10.3 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง

โปรแกรมที่ 10.3 โปรแกรมแสดงตัวอักษร

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 7A00     START:  MOV     R2,#00H
8008 : 908100   LOOP1:  MOV     DPTR,#TABLE
800B : 8A82             MOV     DPL,R2
800D : E0               MOVX    A,@DPTR
800E : 0A               INC     R2
800F : 90E020           MOV     DPTR,#PORT_A
8012 : F0               MOVX    @DPTR,A
8013 : 111A             ACALL   DELAY
8015 : BA07F0           CJNE    R2,#07H,LOOP1
8018 : 80EC             SJMP    START

801A : 7E04     DELAY:  MOV     R6,#4           ; โปรแกรมหน่วงเวลา
801C : 7FFF     DELAY1: MOV     R7,#255
801E : DFFE     DELAY2: DJNZ    R7,DELAY2
8020 : DEFA             DJNZ    R6,DELAY1
8022 : 22               RET

                        ORG     8100H
8100 : 00F9FAAB TABLE:  DB      00H,0F9H,0FAH,0ABH,0ACH,55H,06H
8104 : AC5506
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

12. จงแก้ไขโปรแกรมโดยแสดงตัวอักษร A

13. ป้อนโปรแกรมที่ 10.4 แล้วสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง

โปรแกรมที่ 10.4 อักษรวิ่งบนชุดแสดงผลแบบจุดขนาด 5X7

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH
8005 : F0               MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 7A00     START:  MOV   R2,#00H
8008 : 7BFF             MOV   R3,#0FFH
800A : 7C00             MOV   R4,#00H
800C : 908100   LOOP1:  MOV   DPTR,#TABLE
800F : 8A82             MOV   DPL,R2
8011 : E0               MOVX  A,@DPTR
8012 : 0A               INC   R2               ; เพิ่มตำแหน่งของข้อมูลในชุดข้อมูล
8013 : 0B               INC   R3
8014 : 90E020           MOV   DPTR,#PORT_A
8017 : 2B               ADD   A,R3             ; รวมตำแหน่งหลักที่ต้องการแสดง
8018 : F0               MOVX  @DPTR,A
8019 : 1128             ACALL DELAY
801B : BB07EE           CJNE  R3,#07H,LOOP1
801E : 0C               INC   R4               ; เลื่อนชุดของข้อมูล
801F : 7BFF             MOV   R3,#0FFH
8021 : AA04             MOV   R2,04
8023 : BC1FE6           CJNE  R4,#1FH,LOOP1
8026 : 80DE             SJMP  START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8028 : 7E19     DELAY:  MOV    R6,#25              ; โปรแกรมหน่วงเวลา
802A : 7FC8     DELAY1: MOV    R7,#200
802C : DFFE     DELAY2: DJNZ   R7,DELAY2
802E : DEFA             DJNZ   R6,DELAY1
8030 : 22               RET

                        ORG    8100H
8100 : 00000000 TABLE:  DB     00H,00H,00H,00H,00H,00H,00H      ; ไม่มีการแสดงผล
8104 : 000000
8107 : F8205088         DB     0F8H,20H,50H,88H,00H             ; K
810B : 00
810C : F8081008         DB     0F8H,08H,10H,08H,0F8H,00H        ; M
8110 : F800
8112 : F800             DB     0F8H,00H                         ; I
8114 : 08F80800         DB     08H,0F8H,08H,00H                 ; T
8118 : 100800           DB     10H,08H,00H                      ; '
811B : F8808080         DB     0F8H,80H,80H,80H,00H             ; L
811F : 00
8120 : 00000000         DB     00H,00H,00H,00H,00H,00H          ; ไม่มีการแสดงผล
8124 : 0000
                        END
```

14. จงอธิบายการทำงานของโปรแกรมที่ 10.4
15. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

16. จงเขียนโปรแกรมแสดงจุดวิ่งจากมุมบน-ซ้ายไปสู่มุมล่าง-ขวาของ Matrix Display
17. จงเขียนโปรแกรมแสดงตัวเลขตั้งแต่ 0-9 บนชุดแสดงผลแบบจุดขนาด 5X7
18. จงเขียนโปรแกรมอักษรวิ่งบนชุดแสดงผลแบบจุดขนาด 5X7 โดยเป็นชื่อของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 11

การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก (Digital to Analog Converter)

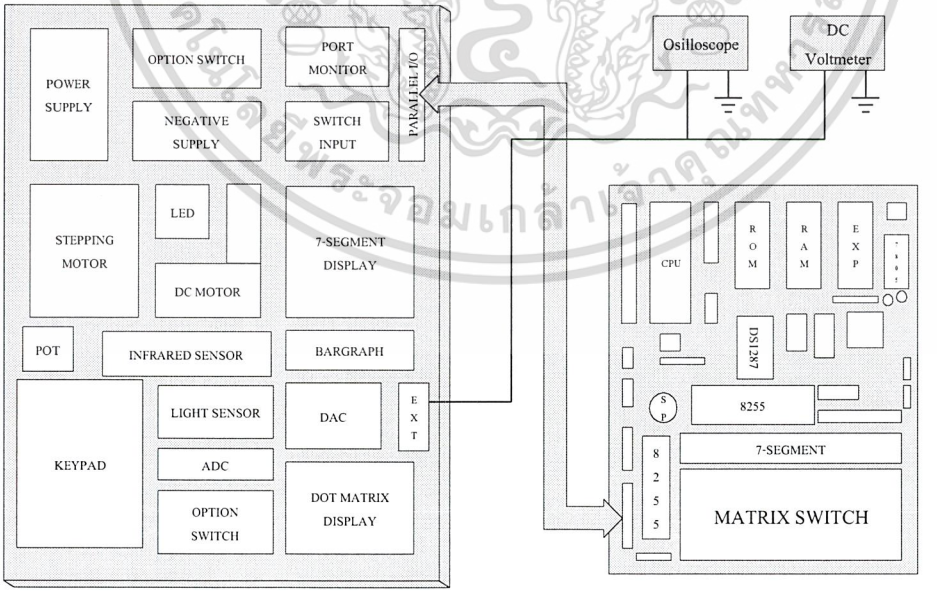
วัตถุประสงค์

1. เพื่อศึกษาการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก (Digital to Analog Converter)
2. เพื่อสามารถใช้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกสร้างรูปคลื่นต่างๆ ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 1 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. ออสซิลโลสโคป 1 เครื่อง
5. โวลต์มิเตอร์วัดไฟกระแสตรง 1 ตัว
6. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



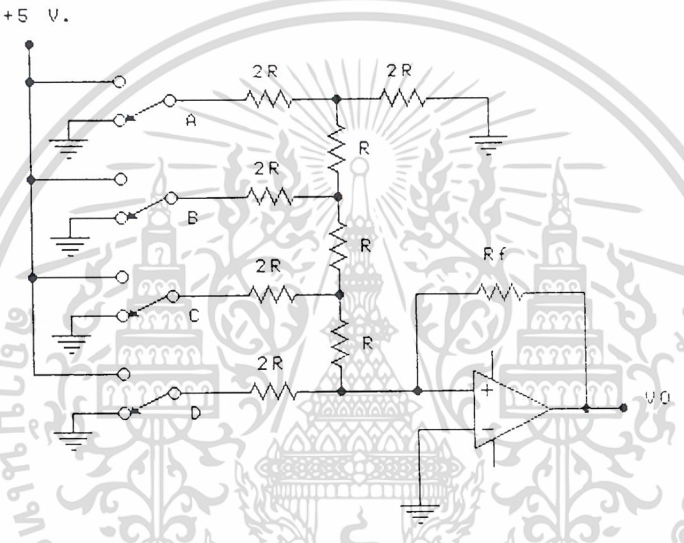
รูปที่ ข. 26 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก (Digital to Analog Converter)

การแปลงกลุ่มของสัญญาณดิจิตอล (Digital) ให้กลายเป็นปริมาณของแรงดันหรือกระแส เราเรียกวิธีการนี้ว่า Digital to Analog Converter วงจรพื้นฐานของการแปลงสัญญาณดิจิตอลให้เป็น สัญญาณแอนะล็อก จะใช้ออปแอมป์ (Operational Amplifier) เป็นตัวรวบรวมสัญญาณ และอาศัยเทคนิคในการต่อวงจรที่อินพุตของออปแอมป์ ซึ่งวงจรพื้นฐานที่จะกล่าวถึงนี้คือ วงจร R-2R Ladder ดังแสดงในรูป



รูปที่ ข. 27 วงจร R-2R Ladder

วงจร R-2R Ladder มีสมการของแรงดันเอาต์พุต

Vo = (Vref / 16) x D โวลต์

เมื่อ Vref = แรงดันอ้างอิง 2.55 โวลต์

D = เลขฐานสิบที่แปลงมาจากสัญญาณอินพุต

<u>ตัวอย่าง</u> มีอินพุตป้อนเข้าวงจร R-2R เป็น 0101 โดยที่ Vref = 2.55 Volt สามารถคำนวณหาค่า Vo ได้โดย

จากสมการ Vo = (Vref/16) x D

Vref = 2.55 โวลต์

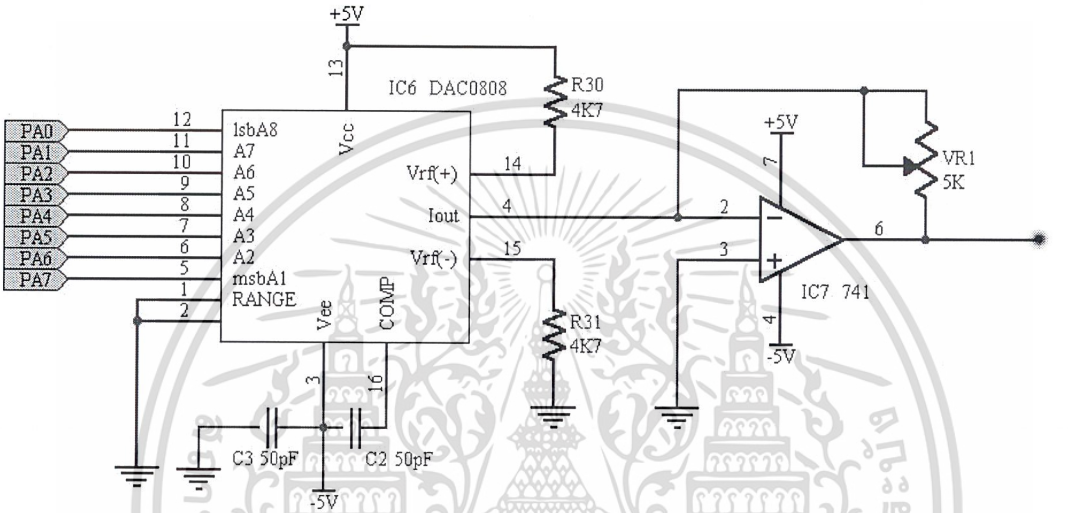
D = 0101 = 5 ในฐานสิบ

Vo = (2.55/16) x 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

= 0.7968 โวลต์

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 จะมีไอซี DAC0808LCN เป็นตัวแปลงสัญญาณดิจิตอล 8 บิต ให้เป็นกระแส แล้วใช้ออปแอมป์ 741 แปลงกระแสที่ได้ให้เป็นแรงดันให้อยู่ในช่วง 0-2.55 V ซึ่งวงจรแสดงดังในรูป



รูปที่ ข. 28 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

จากวงจรสามารถที่จะคำนวณหาค่าแรงดัน Vo ได้โดยอาศัยสมการ

เมื่อ Vref = แรงดันอ้างอิง (กำหนดให้เท่ากับ 5 โวลต์)

Rf = ตัวต้านทานที่ต่อระหว่างขา 13 และ 14 ของ ADC0808LCN

Ro = ตัวต้านทานป้อนกลับของออปแอมป์

Do = ลอจิกของสัญญาณดิจิตอล

<u>ตัวอย่าง</u> ป้อนสัญญาณดิจิตอลให้กับวงจรเป็น 10101010 จงคำนวณหาค่าแรงดัน Vo

จากสมการ $$Vo = \frac{Vref}{Rref} \times Ro \times \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right)$$

Vref = 5 โวลต์

Rref = 4.7 kΩ

Ro = 2.35 kΩ

$$Vo = \frac{5}{4700} \times 2350 \times \left(\frac{1}{2} + \frac{0}{4} + \frac{1}{8} + \frac{0}{16} + \frac{1}{32} + \frac{0}{64} + \frac{1}{128} + \frac{0}{256} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

= 2.55 x (0.5 + 0.125 + 0.03125 + 0.0078125)

∴ Vo = 1.6932 โวลต์

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/5 และ SW 4/1 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับชิงเกิ้ลบอร์ดและปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”

รูปที่ ข. 29 แผนผังการทำงานของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

6. ป้อนโปรแกรมที่ 11.1 แล้วสั่งให้โปรแกรมทำงาน ใช้โวลต์มิเตอร์วัดแรงดันที่จุด EXT. ANALOG OUT ของวงจร D/A บันทึกในตาราง

โปรแกรมที่ 11.1 โปรแกรมแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH
8005 : F0               MOVX  @DPTR,A      ; กำหนด Command Word ให้กับ 8255
8006 : 90E020   START:  MOV   DPTR,#PORT_A
8009 : 747F             MOV   A,#7FH
800B : F0               MOVX  @DPTR,A
800C : 80FE             SJMP  $
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

7. หยุดโปรแกรมการทดลองเปลี่ยนค่าดิจิตอลใหม่ แล้วสั่งให้โปรแกรมทำงาน

ปริมาณดิจิตอล	แรงดันที่วัดได้ (โวลต์)	แรงดันจากการคำนวณ (โวลต์)
00H		
19H		
32H		
4BH		
64H		
7DH		
96H		
0AFH		
0C8H		
0E1H		
0FFH		

8. จงเปรียบเทียบผลของแรงดันที่ได้จากการวัดและจากการคำนวณ
9. ป้อนโปรแกรมที่ 11.2 แล้วสั่งให้โปรแกรมทำงาน ใช้ออสซิลโลสโคปวัดรูปสัญญาณที่จุด EXT. ANALOG OUT ของวงจร D/A บันทึกในตาราง

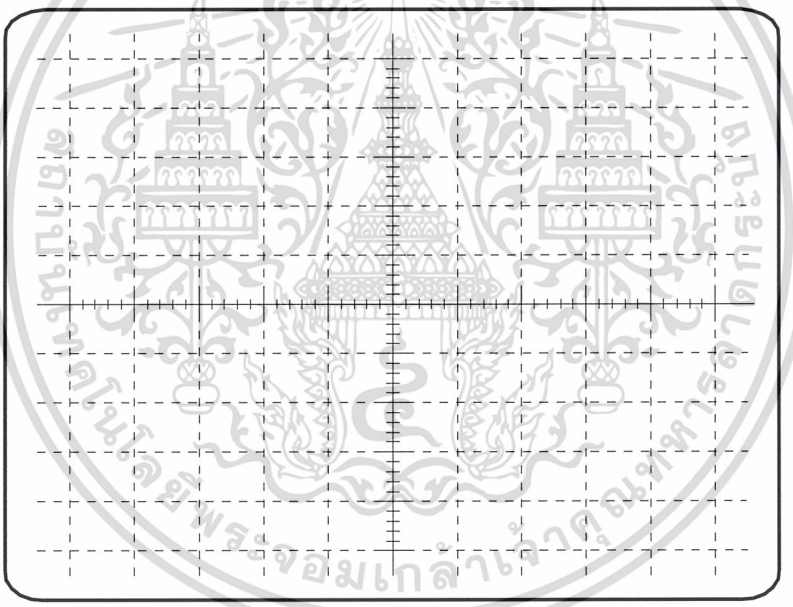
โปรแกรมที่ 11.2 โปรแกรมสร้างสัญญาณรูปฟันเลื่อย (Saw Tooth)

```
                PORT_A  EQU    0E020H
                PORT_B  EQU    PORT_A+1
                PORT_C  EQU    PORT_A+2
                C_PORT  EQU    PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A          ; กำหนด Command Word ให้ 8255
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7400             MOV     A,#00H
800B : F0       LOOP:   MOVX    @DPTR,A
800C : D8FE             DJNZ    R0,$            ; หน่วงเวลา
800E : 04               INC     A
800F : 80FA             SJMP    LOOP
                        END
```

ตารางบันทึกรูปคลื่น



10. สัญญาณรูปฟันเลื่อยที่ได้มีความถี่_____Hz

11. ป้อนโปรแกรมที่ 11.3 แล้วสั่งให้โปรแกรมทำงาน ใช้ออสซิลโลสโคปวัดรูปสัญญาณที่จุด EXT. ANALOG OUT ของวงจร D/A บันทึกในตาราง

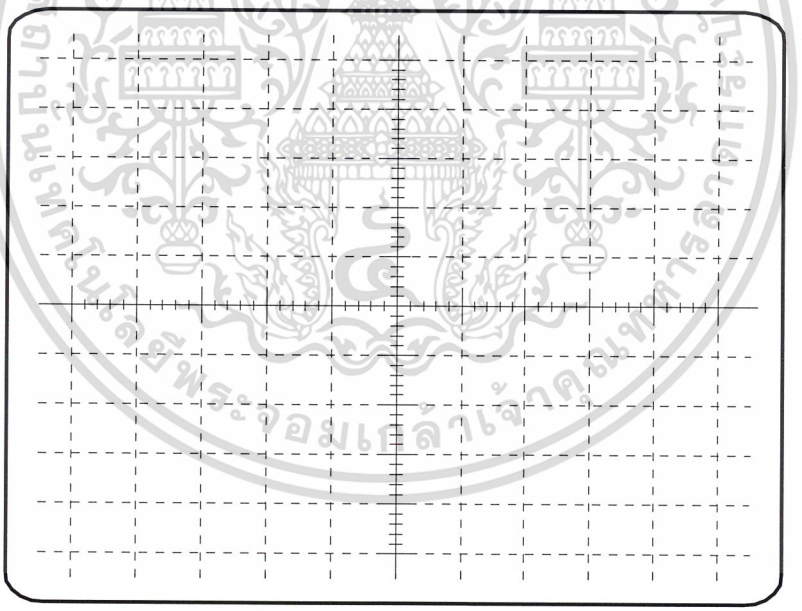
โปรแกรมที่ 11.3 โปรแกรมสร้างสัญญาณรูปสี่เหลี่ยม (Square Wave)

```
PORT_A  EQU     0E020H
PORT_B  EQU     PORT_A+1
PORT_C  EQU     PORT_A+2
C_PORT  EQU     PORT_A+3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7400             MOV     A,#00H
800B : F0       LOOP:   MOVX    @DPTR,A
800C : D8FE             DJNZ    R0,$            ; หน่วงเวลา
800E : F4               CPL     A
800F : 80FA             SJMP    LOOP
                        END
```

ตารางบันทึกรูปคลื่น



12. สัญญาณรูปฟันเลื่อยที่ได้มีความถี่ _____ Hz
13. สรุปและวิจารณ์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฝึกหัด

14. จงเขียนโปรแกรมสั่งให้วงจร D/A สร้างระดับแรงดันที่ 1.86 โวลต์ (หรือใกล้เคียง) พร้อมทั้งเปรียบเทียบปริมาณดิจิตอลที่ป้อนให้กับวงจร D/A กับการคำนวณ
15. จงอธิบายการทำงานของโปรแกรมที่ 11.2 และโปรแกรมที่ 11.3
16. จงอธิบายถึงขนาดความถี่ของรูปคลื่นสัญญาณว่าเกิดขึ้นได้อย่างไร และสามารถเปลี่ยนแปลงโดยแก้ไขส่วนใดของโปรแกรม
17. จงเขียนโปรแกรมสร้างสัญญาณรูปสามเหลี่ยม (Triangle Wave) จากวงจร D/A พร้อมทั้งบันทึกรูปสัญญาณด้วย (กำหนดความถี่เอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 12

การแสดงระดับสัญญาณและการปรับแรงดัน 0-2.55 โวลต์

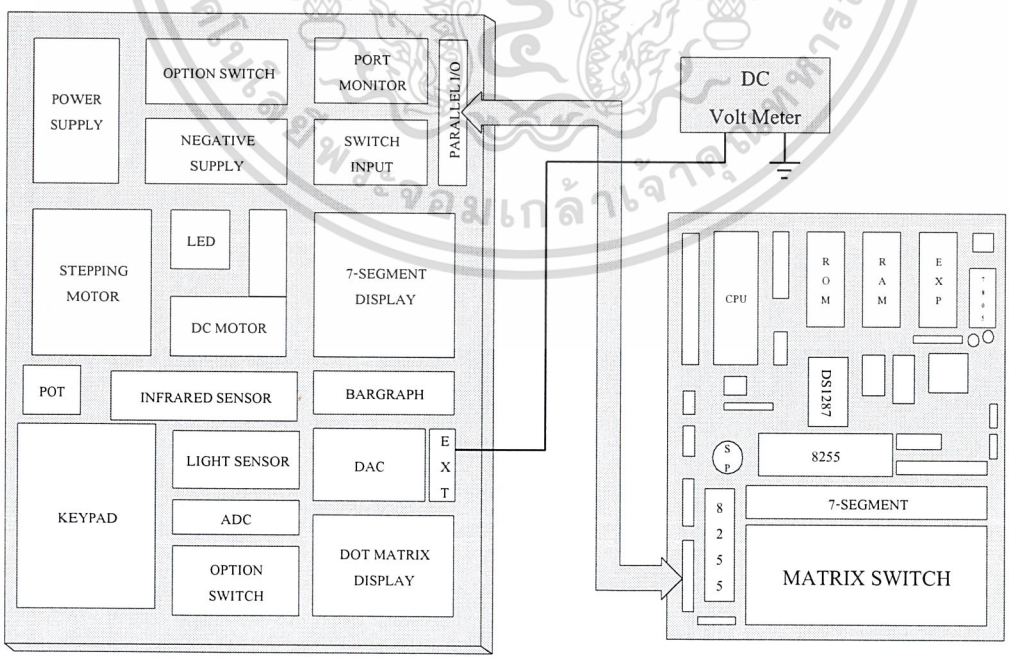
วัตถุประสงค์

1. เพื่อให้สามารถควบคุมการทำงานของวงจรแสดงระดับสัญญาณ ด้วยไมโครคอนโทรลเลอร์ได้
2. เพื่อให้สามารถควบคุมการทำงานของวงจรแสดงระดับสัญญาณด้วยวงจรปรับแรงดัน 0-2.55 โวลต์ ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 1 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. โวลต์มิเตอร์วัดไฟกระแสตรง 1 ตัว
5. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



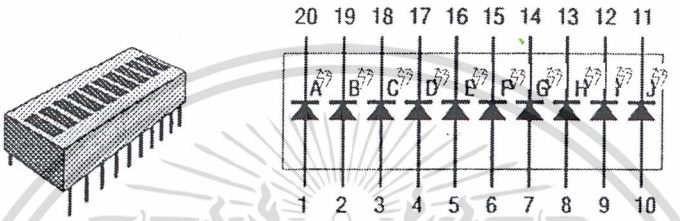
รูปที่ ข. 30 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การแสดงระดับสัญญาณ (Bargraph Display)

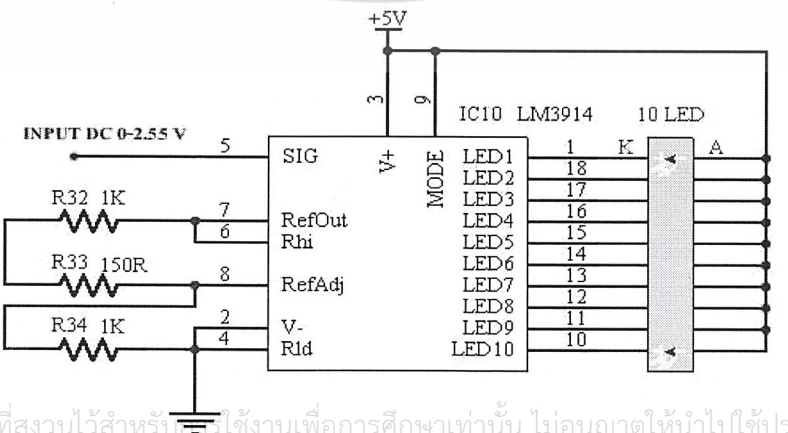
วงจรแสดงระดับสัญญาณจัดเป็นตัวแสดงผลอีกแบบหนึ่ง ที่เกิดจากการนำเอาหลอดหลอดไดโอดเปล่งแสงสี่เหลี่ยมมาเรียงติดกันเป็นแถว มีตั้งแต่ 2 หลอด ไปจนถึง 10 หลอด แล้วถูกยึดเป็นชิ้นเดียวกัน แต่ละหลอดจะต่อขาออกมีทั้งแบบแถวเดี่ยวและแถวคู่



รูปที่ ข. 31 โครงสร้างของตัวแสดงระดับสัญญาณ

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 นำเอาตัวแสดงระดับสัญญาณมาเป็นหน่วยแสดงผลโดยต่อเข้ากับไอซี LM3914 ซึ่งเป็น Dot/Bar Display Driver ภายในจะประกอบไปด้วยออปแอมป์จำนวน 10 ตัว ทำหน้าที่เป็นตัวเปรียบเทียบ (Comparator) ระหว่างแรงดันอินพุตกับแรงดันอ้างอิง แรงดันอินพุตมีค่ามากกว่าแรงดันอ้างอิงของออปแอมป์แต่ละตัว เอาต์พุตของออป-แอมป์แต่ละตัวจะเป็นศูนย์ ทำให้หลอดหลอดไดโอดเปล่งแสงที่ต่ออยู่ติดสว่าง อินพุตของ LM3914 จะรับมาจากวงจรปรับแรงดัน 0-2.55 โวลต์ หรือ D/A โดยที่ปรับที่ SW 1/1 และ SW 1/2

แรงดันอินพุตจะอยู่ในช่วง 0-2.55 โวลต์ LM3914 สามารถแสดงผลได้ 10 ช่อง ฉะนั้นแรงดันอินพุตของวงจรแสดงระดับสัญญาณจะมีค่าเท่ากับ 0.255 V/Bar

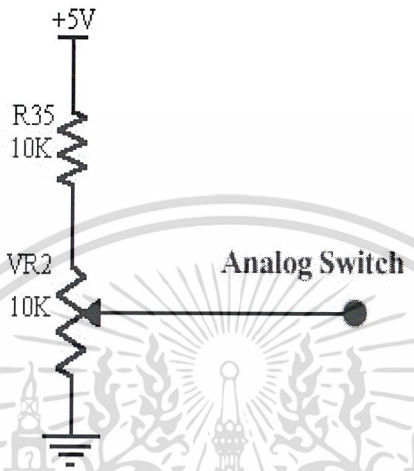


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ ข. 32 วงจรแสดงผลแบบวงจรแสดงระดับสัญญาณ (Bargraph Display)

การปรับแรงดัน 0-2.55 โวลต์ (Potentiometer)

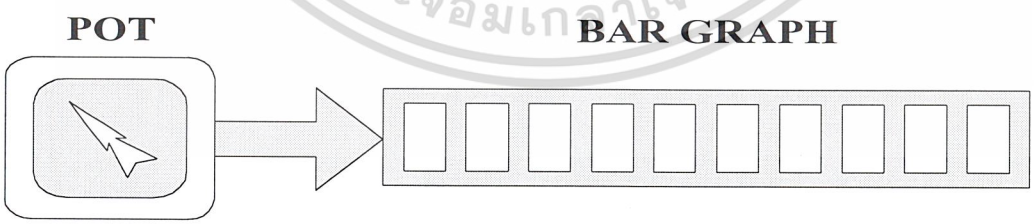
วงจรปรับแรงดัน 0-2.55 โวลต์ ใช้ POT ขนาด 10 KΩ ทำหน้าที่เป็น Voltage Divider ปรับแรงดันได้ในช่วง 0-2.55 โวลต์ เพื่อป้อนให้กับวงจร A/D และวงจร วงจรแสดงระดับสัญญาณ



รูปที่ ข. 33 วงจรปรับแรงดัน 0-2.55 โวลต์

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "OFF"
2. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง "SWITCH INPUT"
3. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง "OFF" ยกเว้น SW 1/2 "ON"



รูปที่ ข. 34 แผนผังการทำงานของวงจรแสดงระดับสัญญาณร่วมกับวงจรปรับแรงดัน 0-2.55 โวลต์

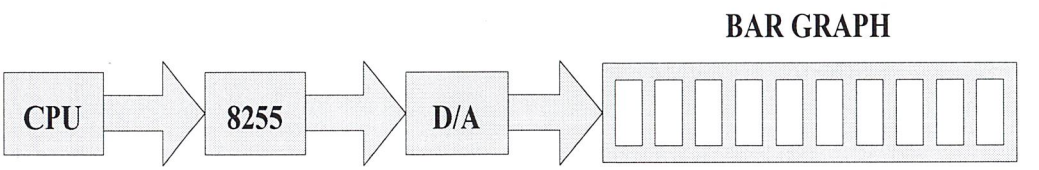
4. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง "ON"
5. ใช้โวลต์มิเตอร์วัดที่แรงดันที่ขากลางของ POT (VR2) เพื่อหาค่าแรงดันด้านอินพุตของวงจรแสดงระดับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ปรับ POT ให้วงจรแสดงระดับสัญญาณดับหมด (ตำแหน่ง 0 โวลต์) แล้วค่อยๆ ปรับ POT ให้แสดงระดับสัญญาณติดสว่างที่ละขีด บันทึกค่าแรงดัน

แสดงระดับสัญญาณ	แรงดันอินพุตของวงจรแสดงระดับสัญญาณ
ดับหมด	
ติด 1 ขีด	
ติด 2 ขีด	
ติด 3 ขีด	
ติด 4 ขีด	
ติด 5 ขีด	
ติด 6 ขีด	
ติด 7 ขีด	
ติด 8 ขีด	
ติด 9 ขีด	
ติด 10 ขีด	

7. วงจรแสดงระดับสัญญาณติดสว่างหมดต้องการแรงดันอินพุต _____ โวลต์ แต่ละขีดของวงจรแสดงระดับสัญญาณต้องการแรงดันอินพุตต่างกัน_____ โวลต์
8. ปรับสวิตช์ SW 1/2 มาที่ตำแหน่ง “OFF” และปรับสวิตช์ SW 1/1 และ SW 1/5 มาที่ตำแหน่ง “ON”
9. ทดลองควบคุมการแสดงผลของวงจรแสดงระดับสัญญาณด้วย D/A โดยป้อนโปรแกรมที่ 12.1 และสั่งให้โปรแกรมทำงาน สังเกตและบันทึกผลการทดลอง



รูปที่ ข. 35 แผนผังการทำงานของวงจรแสดงระดับสัญญาณร่วมกับวงจร D/A Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 12.1 โปรแกรมควบคุมวงจรแสดงระดับสัญญาณด้วยวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อก

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A      ; กำหนด Command Word ให้กับ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 747F             MOV     A,#7FH
800B : F0               MOVX    @DPTR,A
800C : 80FE             SJMP    $
                        END
```

10. ทดลองเปลี่ยนข้อมูลที่ส่งให้วงจร D/A แล้วบันทึกผลการทดลองการติดสว่างของวงจรแสดงระดับสัญญาณ และแรงดันอินพุตของวงจรแสดงระดับสัญญาณที่ EXT. Analog IN

ข้อมูลของ D/A	แสดงระดับสัญญาณ	แรงดันอินพุตของวงจรแสดงระดับสัญญาณ
00H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
19H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
32H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
4BH	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
64H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
80H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	
96H	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลของ D/A	แสดงระดับสัญญาณ	แรงดันอินพุตของวงจรแสดงระดับสัญญาณ
0AFH		
0C8H		
0E1H		
0FFH		

□ = ดับ

■ = ติด

11. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

12. จงเขียนโปรแกรมไฟวิ่งบนวงจรแสดงระดับสัญญาณ ให้ติดเพิ่มขึ้นทีละขีดจนครบแล้วกลับมาเริ่มต้นใหม่ โดยใช้ D/A เป็นตัวอินพุต
13. จงเขียนโปรแกรมควบคุมการติดของวงจรแสดงระดับสัญญาณ โดยใช้วงจร D/A มีอินพุตเป็น Switch Input ขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 13

การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล (Analog to Digital Converter)

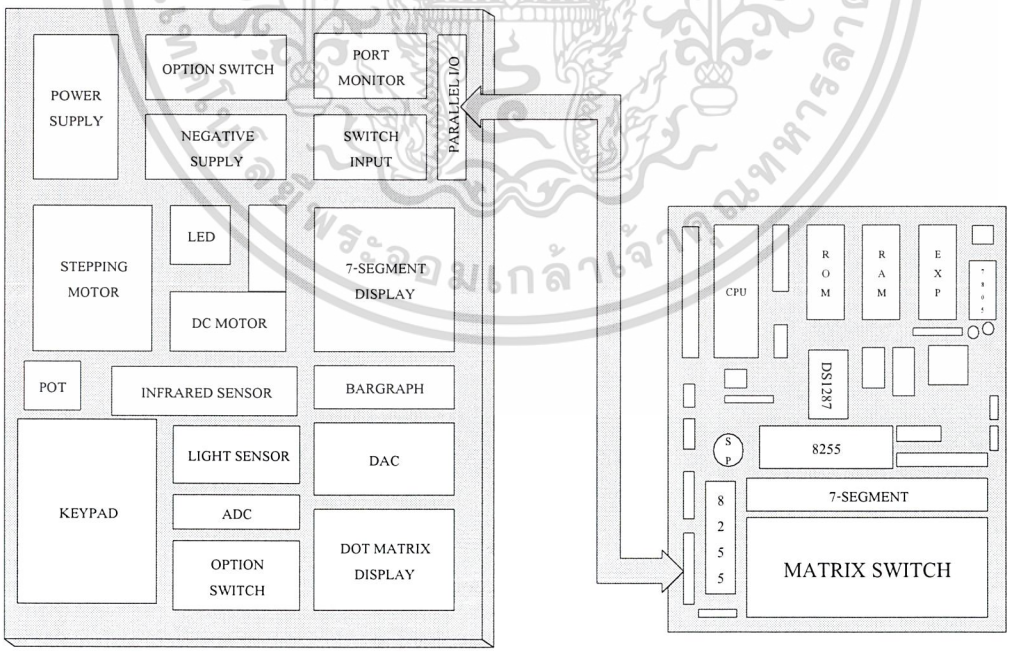
วัตถุประสงค์

1. เพื่อศึกษาวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล
2. เพื่อสามารถควบคุมวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลด้วยไมโครคอนโทรลเลอร์ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 13 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 36 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

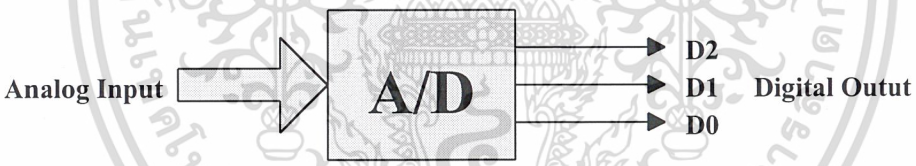
การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล (A/D) คือการเปลี่ยนสัญญาณแอนะล็อก ได้แก่ สัญญาณที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง เช่น ระดับของแรงดันไฟฟ้าหรือปริมาณของกระแสไฟฟ้า ให้กลายเป็นสัญญาณดิจิตอลที่อยู่ในรูปของเลขฐานสอง คือ “0” กับ “1” ซึ่งเป็นสัญญาณที่ไม่ขึ้นอยู่กับฟังก์ชันของเวลา

การเปลี่ยนสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิตอลนั้นจะต้องคำนึงถึง

1. ความละเอียดของ A/D

ความละเอียดของ A/D ขึ้นอยู่กับจำนวนบิตของเอาต์พุต ยิ่งมีจำนวนบิตของเอาต์พุตมากเท่าไร ความละเอียดของ A/D ก็จะยิ่งมากขึ้น ทั้งนี้เพราะว่าแรงดันอินพุตที่ป้อนเข้ามา ถ้าหากมีการเปลี่ยนแปลงอยู่ในช่วง +1/2 LSB ถึง -1/2 LSB จะไม่มีผลอะไรต่อเอาต์พุตของ A/D เลย

<u>ตัวอย่าง</u> เช่น การแปลงแรงดันไฟฟ้าให้เป็นสัญญาณดิจิตอลขนาด 3 บิต เอาต์พุตจะประกอบด้วย D0, D1 และ D2 โดยที่ D0 เป็นบิตที่มีความสำคัญต่ำสุด (Least Significant Bit (LSB)) และบิต D2 เป็นบิตที่มีความสำคัญสูงสุด (Most Significant Bit (MSB))

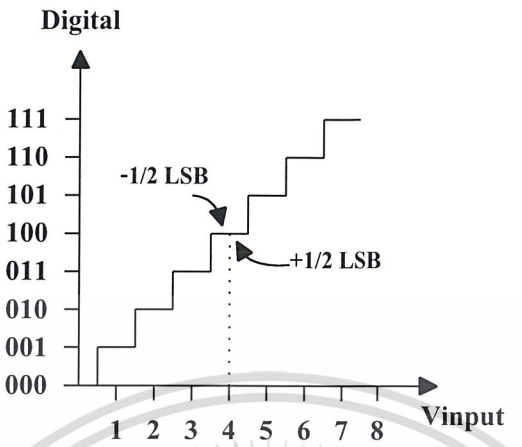


รูปที่ ข. 37 แผนผังการทำงานของวงจรAnalog to Digital Converter ขนาด 3 บิต

คุณลักษณะในทางอุดมคติของการแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิตอล 3 บิต จะให้ 1 LSB เทียบเท่ากับแรงดัน 1 โวลต์ ดังนั้นเมื่อป้อนอินพุตแรงดันเข้ามาเป็น 0, 1 และ 2 โวลต์ เอาต์-พุตที่ได้จะเป็น 000, 001 และ 010 ตามลำดับถ้าหากแรงดันอินพุตที่ป้อนเข้ามามีค่าเป็นเศษส่วน เช่น มีแรงดันอินพุตเท่ากับ 1.25 โวลต์ เอาต์พุตที่ได้ก็ยังคงเป็น 001 อยู่ ที่เป็นอย่างนี้ก็เพราะว่าแรงดันอินพุตที่ป้อนเข้ามามีค่าอยู่ในช่วง +1/2 LSB ถึง -1/2 LSB จะไม่มีผลต่อเอาต์พุต

ดังนั้นจะเห็นได้ว่าแรงดันที่ป้อนเข้ามาอยู่ในช่วง 0.5-1.5 โวลต์ เอาต์พุตของ A/D ก็ยังคงมีค่าเป็น 001 อยู่ตลอด แต่ถ้าหากแรงดันอินพุตมีค่าเท่ากับ 1.6 โวลต์ เอาต์พุตที่ได้จะเป็น 010 ซึ่งความผิดพลาดของวงจร A/D ที่เราจะต้องคำนึงถึงในการที่จะออกแบบวงจร A/D เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 38 ความผิดพลาดของวงจร Analog to Digital Converter

จากรูปแสดงให้เห็นว่า A/D ไม่สามารถแยกความแตกต่างได้ถ้าแรงดันอินพุตมีค่าความแตกต่างของแรงดันน้อยกว่า 1 LSB แต่เพื่อที่จะทำให้ A/D แยกความแตกต่างของแรงดันอินพุตได้ดีขึ้นจะต้องทำให้ 1 LSB มีช่วงแรงดันที่น้อยที่สุด โดยการเพิ่มจำนวนบิตของเอาต์พุตของ A/D ให้มากขึ้น

2. เวลาการเปลี่ยนแปลง (Conversion Time)

การแปลงสัญญาณทางสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิตอลนั้น ไม่ได้เกิดขึ้นในทันทีทันใด ต้องมีการผ่านกระบวนการต่างๆ ดังนั้นจึงต้องใช้ระยะเวลาหนึ่งจึงจะได้สัญญาณแอนะล็อกออกมา ซึ่งระยะเวลาที่ถูกใช้ไปนี้เรียกว่า "เวลาการแปลง" (Conversion Time) โดยปกติจะมีค่าเป็นไมโคร- วินาที (Microsecond) ซึ่งค่านี้ดูได้จากข้อมูลของไอซีที่เราใช้ เช่น เบอร์ ADC0804 ที่ใช้ในชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 นี้มีค่าเวลาการแปลงประมาณ 100 μsec

3. วิธีการแปลงสัญญาณ

จะมีอยู่ด้วยกันหลายวิธี แต่วิธีที่เป็นที่นิยม มีความเร็วในการแปลงสัญญาณสูง ราคาไม่สูงมากนัก มีความละเอียดพอสมควร เราเรียกวิธีการแปลงสัญญาณแบบนี้ว่า "การแปลงสัญญาณแบบประมาณค่า" ซึ่งเป็นแบบที่ใช้อยู่กับชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051

ขั้นตอนการแปลงเริ่มต้นเมื่อสัญญาณแอนะล็อกถูกป้อนเข้าที่อินพุต แล้วมีการจ่ายพัลส์ SOC (Start of Conversion) เข้าที่รีจิสเตอร์ SAR (Successive Approximation Register) พัลส์สัญญาณนาฬิกาลูกแรกที่ป้อนให้กับตัว SAR จะ "ON" เอาต์พุตของบิตนัยสำคัญสูงสุด

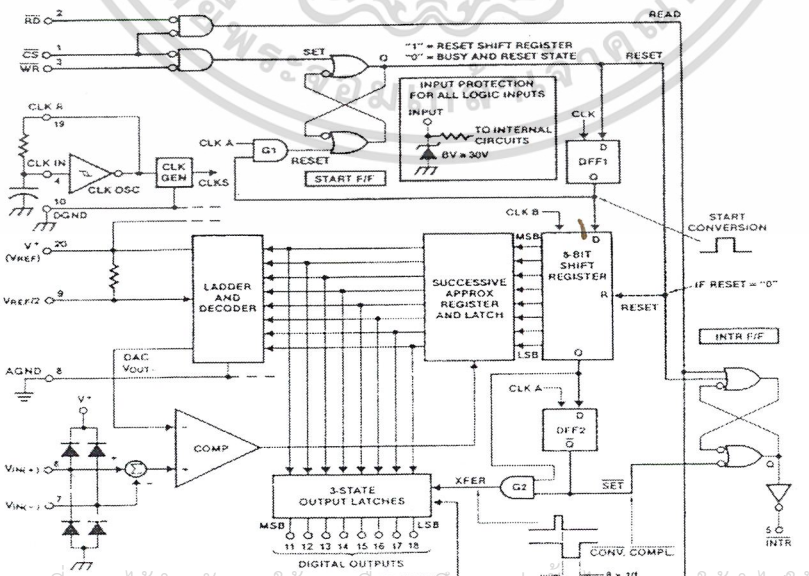
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงเป็นการปรับให้เอาต์พุตของ ADC เป็น 50% ของแรงดันเอาต์พุตของอินเวอร์เตอร์ ตัว SAR จะมองไปยังเอาต์พุตของวงจรเปรียบเทียบว่า เอาต์พุตของ DAC มีค่ามากกว่าหรือน้อยกว่าสัญญาณแอนะล็อกทางอินพุต ถ้าแรงดันของ DAC มีค่ามากกว่าวงจรเปรียบเทียบจะยังคงอยู่ในสภาวะ “OFF” ดังนั้นตัว SAR จะ “OFF” บิตนัยสำคัญสูงสุดและให้ชื่อว่าสภาวะ “0”

ถ้าแรงดันของ DAC มีค่าน้อยกว่าสัญญาณแอนะล็อกทางอินพุตวงจรเปรียบเทียบจะยังคงทำงานอยู่ดังนั้น SAR จะยังคงปล่อยให้บิตนัยสำคัญสูงสุด “ON” อยู่ และเราเรียกสภาวะนี้ว่า “1” ซึ่งสภาวะ “1” หรือ “0” นี้จะกระทำภายในพัลส์ของสัญญาณนาฬิกาเพียงพัลส์เดียว สัญญาณนาฬิกาถัดไป ตัว SAR จะ “ON” บิตนัยสำคัญสูงสุดอันดับ 2 และทำการตรวจสอบผลลัพธ์ที่ได้อีกครั้งหนึ่งจากวงจรเปรียบเทียบ ถ้าสัญญาณจาก DAC ครั้งใหม่นี้มีค่ามากกว่าแรงดันอินพุตเอาต์พุตของวงจรเปรียบเทียบจะยังไม่มีดังนั้นตัว SAR จะ “OFF” บิตนัยสำคัญสูงสุดอันดับที่ 2 ทิ้ง

แต่ถ้าสัญญาณจาก DAC มีค่าน้อยกว่า วงจรเปรียบเทียบจะยังคงทำงานและตัว SAR จะปล่อยให้บิตนัยสำคัญสูงสุดอันดับ 2 นี้ “ON” ตัว SAR จะพิจารณาแต่ละบิตด้วยวิธีเดียวกัน (บิตนัยสำคัญสูงสุดถึงบิตนัยสำคัญต่ำสุด) จนกระทั่งทุกๆ บิตถูกพิจารณาหมดเนื่องจาก 1 บิต ถูกหาค่าภายใน 1 พัลส์ ADC ขนาด 8 บิตจึงใช้สัญญาณนาฬิกาเพียง 8 พัลส์ ก็สามารถทำการแปลงได้จนจบกระบวนการ

เมื่อบิตนัยสำคัญต่ำสุดถูกพิจารณาเสร็จสิ้นแล้ว SAR จะส่งสัญญาณสิ้นสุดการแปลง (End of Conversion; EOC) ไปทำการค้างผลลัพธ์ที่ได้ ซึ่งเป็นเลขฐานสองทางเอาต์พุตไว้ ADC ที่ใช้เทคนิคแบบนี้สามารถทำการแปลงให้ได้ข้อมูลขนาด 12 บิต ได้โดยใช้เวลาน้อยกว่า 10 μsec

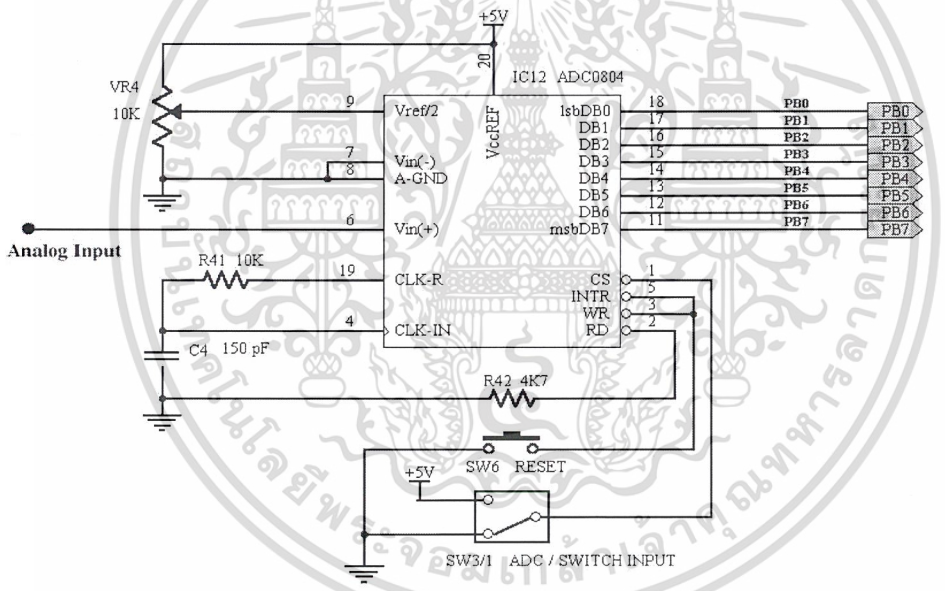


รูปที่ ข. 39 การแปลงสัญญาณแบบประมาณค่าหลายครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ใช้ไอซีเบอร์ ADC0804 ซึ่งเป็น A/D ขนาดเอาต์พุต 8 บิต รับสัญญาณอินพุตที่เป็นแรงดันอยู่ในช่วง 0-2.55 โวลต์ จะมีค่า 1 LSB = 10 mV สัญญาณอินพุตที่รับเข้ามาจะผ่านทางสวิตช์ SW 2 ซึ่งจะเป็นตัวเลือกของสัญญาณว่าจะเป็นสัญญาณมาจากวงจร Light Sensor หรือจะมาจาก Potentiometer (POT) ซึ่งวงจรเหล่านี้จะให้เอาต์พุตออกมาเป็นแรงดัน 0-2.55 โวลต์

การทำงานของวงจร A/D ขา CS ของ ADC0804 ต่อเข้ากับสวิตช์ SW 3 ซึ่งจะเป็นตัวควบคุมการทำงานของ A/D ว่าจะให้ทำงานหรือหยุดทำงาน คือเมื่อเลื่อนสวิตช์ SW 3 มาอยู่ที่ตำแหน่ง ADC ขา CS ก็จะถูกต่อลงกราวด์ (GND) วงจร A/D ก็จะทำงาน แต่ถ้าเลื่อนไปอยู่ที่ตำแหน่ง Switch Input วงจร A/D ก็จะหยุดทำงาน และในขณะที่ A/D กำลังทำงานอยู่แล้วเกิดขัดข้องหรือไม่ทำงานตามปกติ จะต้องทำการรีเซ็ต (Reset) โดยการกดสวิตช์ RESET



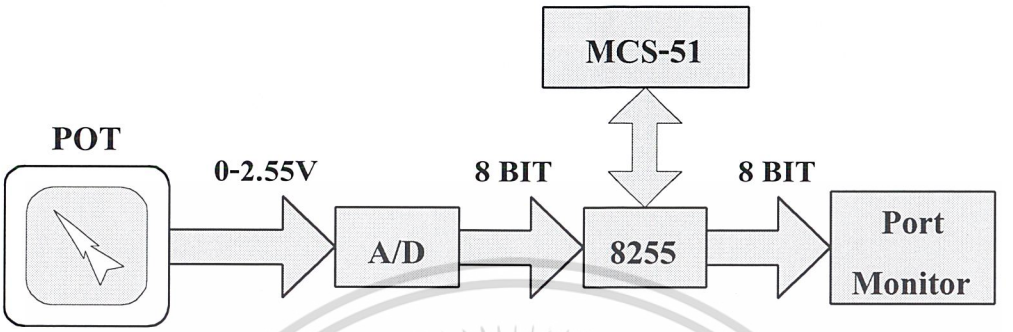
รูปที่ ข. 40 วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “ADC”
5. ปรับสวิตช์ SW 2/1 และ SW 4/1 มาที่ตำแหน่ง “ON”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 13.1 แล้วสั่งให้โปรแกรมทำงาน



รูปที่ ข. 41 แผนผังการทำงานของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอล

โปรแกรมที่ 13.1 อ่านค่าสัญญาณดิจิตอลจากวงจร A/D ที่ Port Monitor

```
PORT_A  EQU   0E020H
PORT_B  EQU   PORT_A+1
PORT_C  EQU   PORT_A+2
C_PORT  EQU   PORT_A+3
        ORG   8000H
8000 : 90E023  INIT:  MOV   DPTR,#C_PORT
8003 : 748A           MOV   A,#08AH
8005 : F0             MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E021  START: MOV   DPTR,#PORT_B
8009 : E0             MOVX  A,@DPTR
800A : 90E020         MOV   DPTR,#PORT_A
800D : F0             MOVX  @DPTR,A          ; ส่งออก Port Monitor
800E : 80F6           SJMP  START
        END
```

8. ทดลองปรับ POT เพื่อป้อนแรงดันอินพุตให้กับวงจร A/D (วัดที่ขากลางของ POT) บันทึกผลการติดดับของหลอด LED ของ Port Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันอินพุต A/D	PORT MONITOR	ข้อมูล
0V	○○○○○○○○	
0.25V	○○○○○○○○	
0.50V	○○○○○○○○	
0.75V	○○○○○○○○	
1.00V	○○○○○○○○	
1.25V	○○○○○○○○	
1.50V	○○○○○○○○	
1.75V	○○○○○○○○	
2.00V	○○○○○○○○	
2.25V	○○○○○○○○	
2.50V	○○○○○○○○	
2.55V	○○○○○○○○	

○ = ดับ

● = ติด

9. ปรับสวิตช์ SW 2/1 มาที่ตำแหน่ง “OFF”
10. ปรับสวิตช์ SW 2/2 มาที่ตำแหน่ง “ON”
11. ทดลองป้อนแรงดัน 0-2.55 โวลต์ที่ EXT. Analog IN แล้วสังเกตการติด-ดับของ LED บน Port Monitor และบันทึกผลการทดลอง
12. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

13. จากผลการทดลองโปรแกรมที่ 13.1 จงอธิบายการติด-ดับ ของหลอด LED หมายถึงอะไร
14. ปรับสวิตช์ SW 1/4, SW 1/6, SW 1/7 และ SW 4/2 มาที่ตำแหน่ง “ON”
15. จงเขียนโปรแกรมที่ตรวจสอบแรงดันจาก POT หากมีแรงดันมากกว่า 2.0 โวลต์ ให้ Buzzer ทำงาน แต่ถ้าแรงดันลดลงต่ำกว่า 2.0 โวลต์ให้ Buzzer หยุดทำงาน
16. จงเขียนโปรแกรมนำค่าจากวงจร A/D ที่มีอินพุตเป็น POT ไปแสดงผลที่จอ 7-Segment หลักที่ 5 และหลักที่ 6 ของ Single Board ในรูปแบบของเลขฐานสิบหกโดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter

-0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment

-0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch

-E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)

17. จงยกตัวอย่างการใช้วงจร A/D มา 5 วงจรพร้อมทั้งอธิบายการใช้งานมาพอสังเขป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 14

การตรวจวัดความสว่างของแสง (Light Sensor)

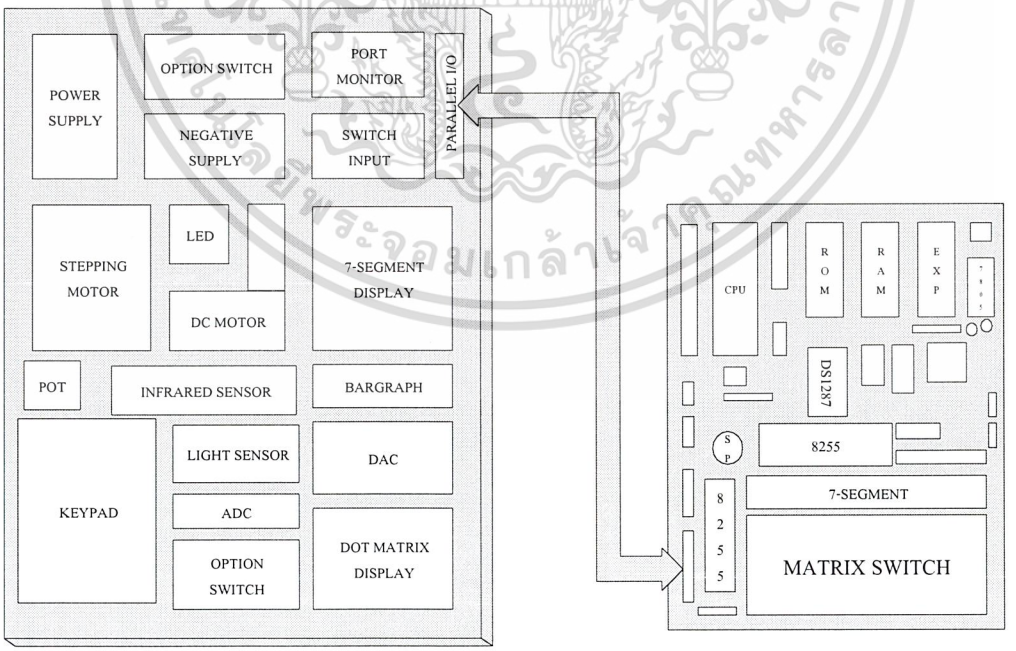
วัตถุประสงค์

1. เพื่อศึกษาตัวตรวจจับความสว่างของแสง (LDR)
2. เพื่อสามารถเชื่อมต่อวงจรตรวจตรวจวัดความสว่างของแสงกับไมโครคอนโทรลเลอร์และอุปกรณ์อื่นๆ ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 14 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



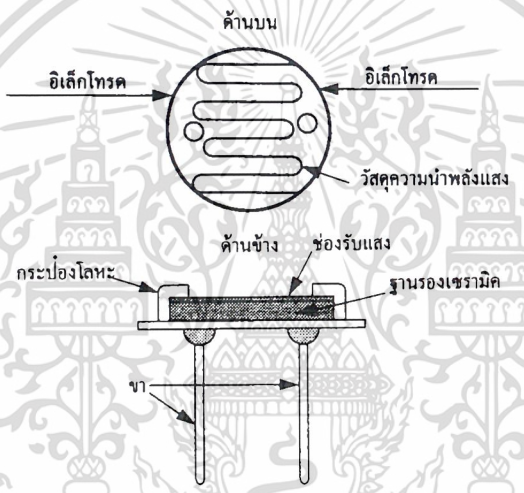
รูปที่ ข. 42 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

เซลล์การนำพลังแสง

เซลล์การนำพลังแสง (Photoconductive Cell) หรือบางครั้งเรียกว่า "ตัวต้านทานพลังแสง" (Photoresistors) ซึ่งจะมีค่าความต้านทานขึ้นอยู่กับความเข้มของแสง (Light Dependent Resistor) หรือที่เรียกย่อๆ ว่า "แอล ดี อาร์" (LDR) กล่าวคือ สภาพความนำทางไฟฟ้า (Conductivity) สามารถเปลี่ยนแปลงได้เมื่อมีแสงมาตกกระทบตัวมัน ตัวความต้านทานพลังแสงจะจัดเป็นตัวแปลงแบบเฉื่อยงาน(Passive Transducer) ที่ต้องมีแหล่งจ่ายไฟภายนอกมาต่อเข้ากับตัว LDR สำหรับการใช้งาน โครงสร้างสามารถดูได้จากรูปที่ ข. 43



รูปที่ ข. 43 โครงสร้างของตัวต้านทานพลังแสง

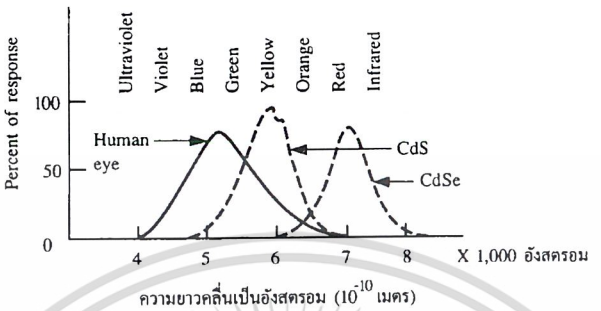
ตัวต้านทานพลังแสงหรือ LDR ทำมาจากสารกึ่งตัวนำชนิดแคดเมียมซัลไฟด์ (Cadmium Sulfide: Cds) หรือแคดเมียมซีลีไนด์ (Cadmium Selenide: CdSe) โดยการนำสารกึ่งตัวนำดังกล่าวมาฉาบลงบนเซรามิค (Ceramic) ที่ใช้เป็นฐานรองแล้ว ทำการบัดกรีตัวถังไม่ให้มีอากาศเข้าไปได้ ซึ่งจะมีขาต่อออกมาใช้งาน 2 ขา และมีช่องสำหรับรับแสงที่ทำด้วยพลาสติกใสหรือกระจก

ขณะเมื่อมีแสงตกกระทบที่วัสดุความต้านทานไวแสงของ LDR อันจะทำให้เกิดอิเล็กตรอนอิสระ ซึ่งมีผลทำให้กระแสไหลได้ค่าความต้านทานของ LDR จะลดลง และถ้ามีแสงมาตกกระทบตัว LDR มากขึ้นจะมีผลทำให้ค่าความต้านทานของ LDR ลดลงอย่างมาก

ผลการตอบสนองต่อแสงที่มาตกกระทบบนสารกึ่งตัวนำที่นำมาใช่ทำ LDR คือ ถ้าเป็นแคดเมียมซัลไฟด์ (Cds) จะอยู่ในช่วงระหว่าง 5,000 ถึง 7,000 กว่าอังสตรอม ซึ่งจะอยู่ในช่วงการตอบสนองต่อแสงของสายตามนุษย์ (Human Eye) ที่อยู่ในช่วง 4,000 ถึง 7,000 กว่าอังสตรอม และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

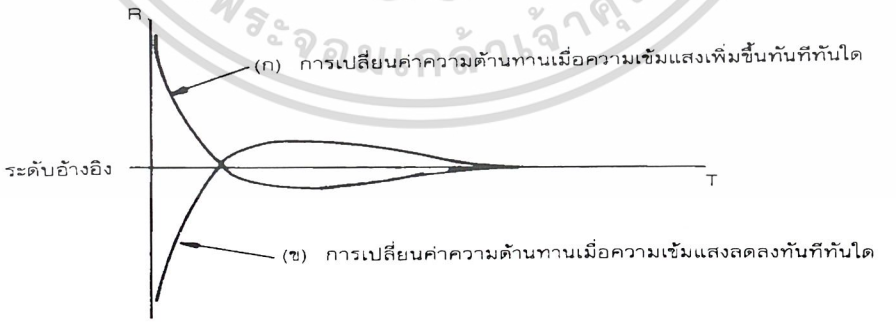
ถ้าเป็นแคดเมียมซีลีไนด์ (CdSe) จะมีผลการตอบสนองต่อแสงในช่วงประมาณ 6,000 ถึง 8,000 กว่าอังสตรอม โดยค่อนไปทางอินฟราเรด (Infrared) ดังแสดงในรูปที่ ข. 44



รูปที่ ข. 44 ความยาวคลื่นเป็นอังสตรอม (10^{-1} เมตร)

อัตราส่วนระหว่างค่าความต้านทานของ LDR ในขณะที่ไม่มีแสง (Dark) กับขณะที่รับแสง (Light) ซึ่งจะมีพิสัยการใช้งานตั้งแต่ 100 ต่อ 1 ถึง 10,000 ต่อ 1 เท่า แต่โดยทั่วไปค่าความต้านทานของ LDR ขณะไม่มีแสงมาตกกระทบจะมีค่าประมาณ 0.5 เมกะโอห์มขึ้นไป และขณะในที่มืดสนิทจะมีค่าประมาณ 2 เมกะโอห์มขึ้นไป

คุณลักษณะของ LDR ที่สำคัญอีกประการหนึ่งคือ ปรากฏการณ์การนำเนื่องจากพลังแสง (Photoconductive Effect) ที่เปลี่ยนแปลงอย่างฉับพลัน ซึ่งสามารถพิจารณาได้จากเส้นกราฟความเปลี่ยนแปลงระหว่างความต้านทานกับเวลา ดังแสดงในรูปที่ ข. 45



รูปที่ ข. 45 ปรากฏการนำเนื่องจากพลังแสง (Photoconductive Effect)

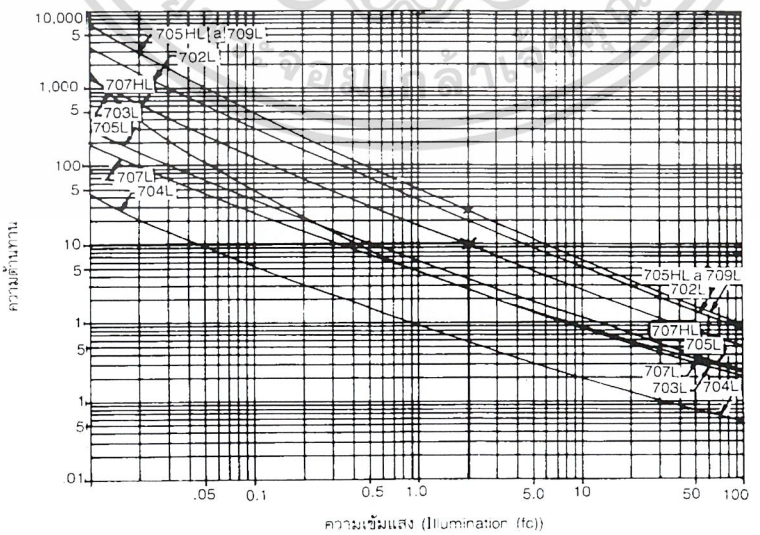
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ข. 45 บนเส้นโค้ง (ก) แสดงให้เห็นว่า เมื่อ LDR ได้รับแสงในทันทีทันใด โดยค่าความต้านทานของมันจะค่อยๆ ลดลงเข้าสู่ระดับอ้างอิง แต่ที่ค่าความต้านทานของ LDR จะหยุดอยู่ที่ระดับอ้างอิง มันจะเลยไปอีกเล็กน้อยแล้วค่อยๆ วกกลับสู่ระดับอ้างอิง

และจากรูปที่ ข. 45 บนเส้นโค้ง (ข) แสดงให้เห็นว่า เมื่อ LDR ไม่ได้รับแสงอย่างฉับพลัน ซึ่งจะมีผลการเปลี่ยนแปลงค่าความต้านทานเพิ่มขึ้นที่ตรงข้ามกับเส้นโค้ง (ก) จากปรากฏการณ์ที่เกิดขึ้นเราจะเห็นได้ว่าความเร็วในการเปลี่ยนแปลงระดับค่าความต้านทานของ LDR จะเป็นไปได้ช้ามากโดย LDR ที่ทำมาจากสารแคดเมียมซัลไฟด์ (Cds) จะใช้เวลาประมาณ 100 ms ส่วนสารแคดเมียมซิลิไนด์ (CdSe) จะใช้เวลาประมาณ 10 ms ซึ่งเร็วกว่าสารแคดเมียมซัลไฟด์ (Cds)

ดังนั้นการนำเอา LDR ไปใช้งานจึงเหมาะกับประเภทความถี่ต่ำๆ เช่น ใช้ LDR ทำเป็นเครื่องตรวจวัดความเข้มของแสง, วงจรสวิตช์ปิด-เปิดหลอดไฟที่ควบคุมด้วยแสงหรือการต่อใช้งานเป็นวงจรแบ่งแรงดัน ดังรูปที่ ข. 46

รูปที่ ข. 46 วงจรตัวแบ่งแรงดันที่ใช้ LDR

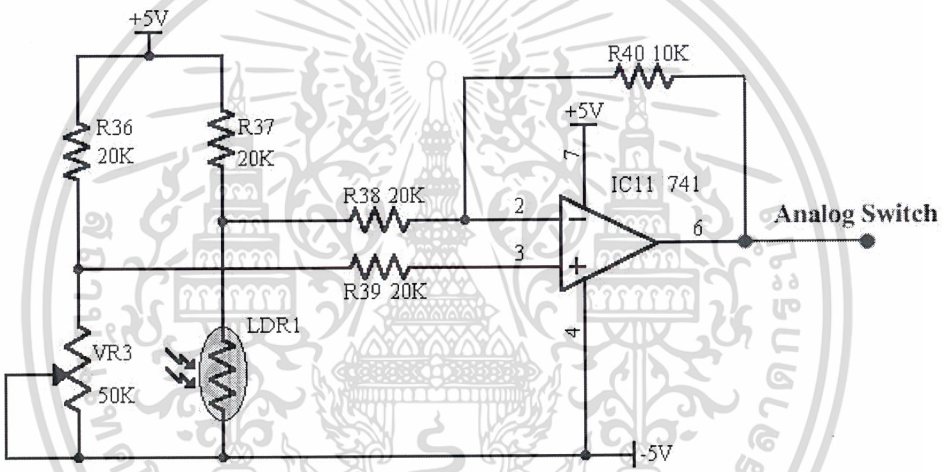


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น เพื่อการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ ข. 47 เส้นแสดงความเปลี่ยนแปลงระหว่างความต้านทานกับความเข้มของแสงของตัว LDR

ในชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 นี้ใช้ LDR ที่ทำมาจากสารแคดเมียมซัลไฟด์ (Cds) และต่อเป็นวงจรแบ่งแรงดัน แรงดันก็จะเพิ่มตามด้วยอัตราส่วนที่คงที่ แรงดันที่ได้จะส่งไปเป็นอินพุตให้กับออปแอมป์ 741 ซึ่งต่อเป็นวงจรดิฟเฟอร์เรนเชียล (Differential Amplifier) ทำให้ได้เอาต์พุต Vo เป็นแรงดันที่เปลี่ยนแปลงอยู่ในช่วง 0-2.55 โวลต์ ซึ่งแรงดัน Vo นี้จะต่อเข้ากับวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล เพื่อเปลี่ยนแรงดัน 0-2.55 โวลต์ ให้เป็นสัญญาณดิจิตอลขนาด 8 บิต

เมื่อมีแสงมาตกกระทบตัว LDR เพิ่มขึ้น แรงดันตกคร่อมตัว LDR ก็จะมากขึ้น ทำให้ผลต่างแรงดันที่ขา 2 และขา 3 ของ 741 มีมากขึ้น แรงดันเอาต์พุต ของวงจรก็จะเพิ่มขึ้นจนถึง 2.55 โวลต์

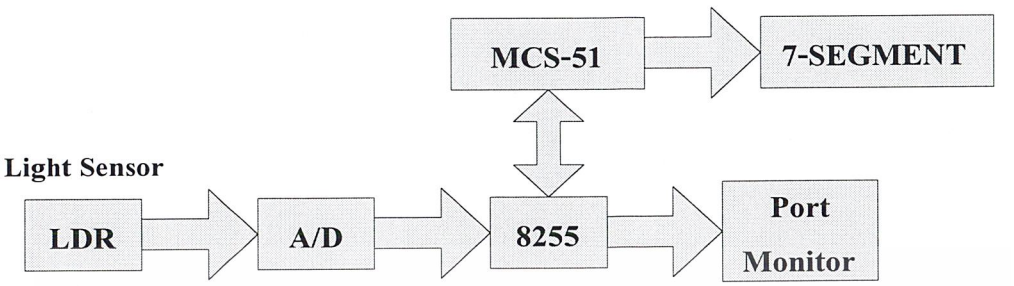


รูปที่ ข. 48 วงจรตรวจวัดค่าความสว่างของแสง

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 2/3 และ SW 4/1 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 49 แผนผังการทำงานของวงจรตรวจวัดค่าความสว่างของแสง

6. ป้อนโปรแกรมที่ 14.1 แล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 14.1 อ่านค่าดิจิตอลจากวงจร A/D ที่ Port Monitor

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH
8005 : F0               MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E021   START:  MOV   DPTR,#PORT_B
8009 : E0               MOVX  A,@DPTR
800A : 90E020           MOV   DPTR,#PORT_A     ; ส่งออก Port Monitor
800D : F0               MOVX  @DPTR,A
800E : 80F6             SJMP  START
                        END
```

7. ทดลองปิด-เปิดช่องรับแสงของตัวตรวจวัดแสงบันทึกผลการติดดับของหลอด LED ของ Port Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการรับแสง	PORT MONITOR	ข้อมูล
ปิดช่องรับแสงสนิท	OOOOOOOO	
เปิดช่องรับแสงเล็กน้อย	OOOOOOOO	
ใช้มือบังช่องรับแสง	OOOOOOOO	
เปิดช่องรับแสง	OOOOOOOO	
ใช้แสงไฟส่องไปที่ช่องรับแสง	OOOOOOOO	

O = ดับ

● = ติด

8. จงอธิบายโปรแกรมที่ 14.1
9. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

10. จากผลการทดลองโปรแกรมที่ 14.1 จงอธิบายการติด-ดับ ของหลอด LED หมายถึงอะไร
11. ปรับสวิตช์ SW 1/4, SW 1/6, SW 1/7, SW 4/2 มาที่ตำแหน่ง “ON”
12. จงเขียนโปรแกรมที่ตรวจสอบแรงดันจาก Light Sensor หากมีแรงดันมากกว่า 1.75 โวลต์ ให้ Buzzer ทำงาน แต่ถ้าแรงดันลดลงต่ำกว่า 1.75 โวลต์ให้ Buzzer หยุดทำงาน
13. จงเขียนโปรแกรมนำค่าจากวงจร A/D ที่มีอินพุตเป็นวงจรตรวจวัดความสว่างของแสงไปแสดงผลที่จอ 7-Segment หลักที่ 1 และหลักที่ 2 ของ Single Board ในรูปแบบของเลขฐานสิบหก โดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H
    - 0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter
    - 0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment
    - 0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch
    - 0E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)
14. จงยกตัวอย่างการใช้วงจรประเภทตรวจวัดมา 5 วงจรพร้อมทั้งอธิบายการใช้งานมาพอสังเขป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 15

การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด (Infrared Sensor)

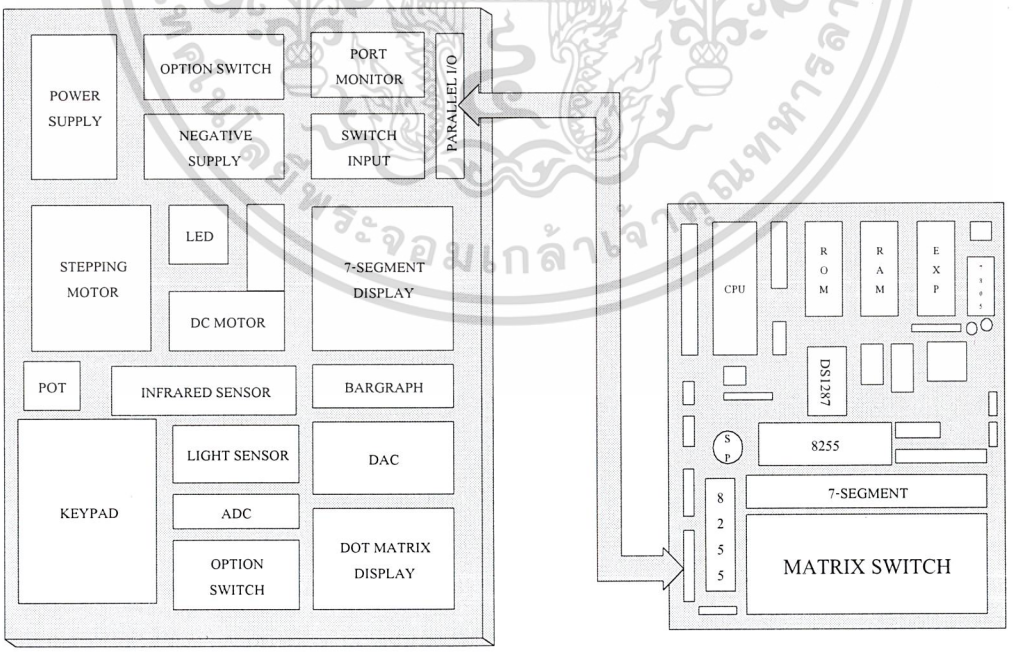
วัตถุประสงค์

1. เพื่อศึกษาตัวตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด (Infrared Sensor)
2. เพื่อสามารถเชื่อมต่อวงจรตรวจจับการเคลื่อนที่ด้วยแสงอินฟราเรด (Infrared Sensor) กับไมโครคอนโทรลเลอร์และอุปกรณ์อื่นๆ ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 15 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 50 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

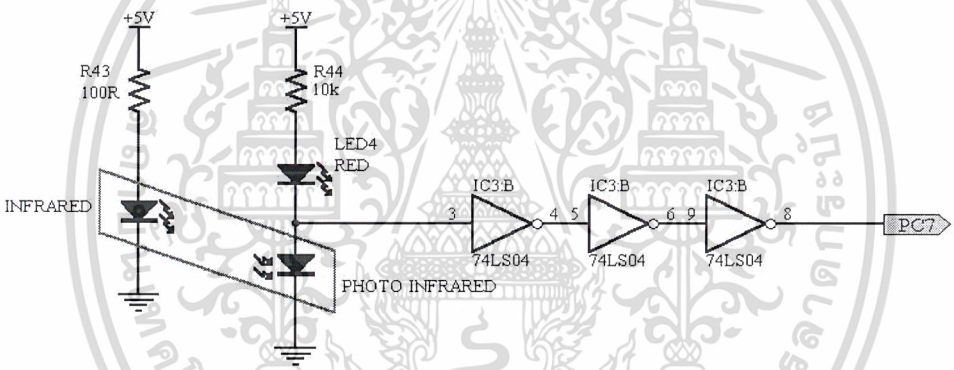
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด (Infrared Sensor)

การตรวจจับการเคลื่อนที่ของวัตถุด้วยแสงอินฟราเรด อาศัยการตัดลำแสงอินฟราเรดที่ส่งจากตัวสร้างแสงอินฟราเรดไปยังตัวรับ (Photo Diode) เมื่อมีวัตถุเคลื่อนที่ผ่านลำแสง ลำแสงจะไม่ผ่านไปยังตัวรับทำให้ได้เอาต์พุตที่ขาของตัวรับเป็น “1” จากนั้นจะเข้าไปยัง NOT Gate จำนวน 3 ตัวเพื่อแปลงสัญญาณที่ได้เป็นสัญญาณ TTL เอาต์พุตที่ได้นี้จะมีค่าเป็น “0” เพื่อที่จะส่งไปยังพอร์ต PC ของ 8255 ตำแหน่งบิตที่ 7 แต่ถ้าตัวรับแสงได้รับแสงอินฟราเรดแล้วเอาต์พุตที่ได้เมื่อผ่าน NOT Gate แล้วจะมีค่าเป็น “1”

วัตถุที่เคลื่อนที่ผ่านลำแสงอินฟราเรดและเกิดการเปลี่ยนแปลงได้ ต้องเป็นวัตถุที่ทึบแสงเท่านั้น

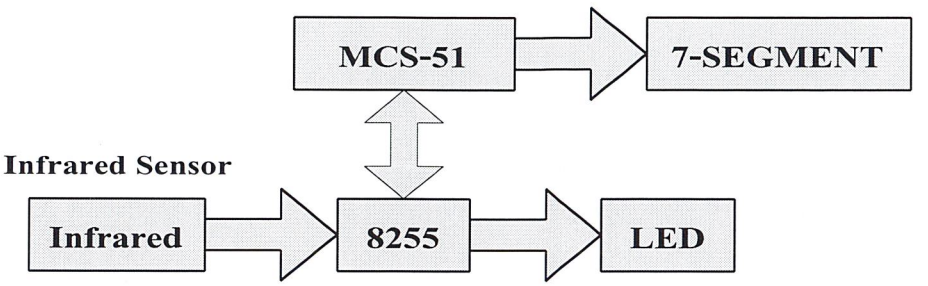


รูปที่ ข. 51 วงจรตรวจจับการเคลื่อนที่ด้วยแสงอินฟราเรด (Infrared Sensor)

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/3, SW 1/6, SW 1/7, SW 4/2 และ SW 4/7 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 52 แผนผังการทำงานของวงจรตรวจจับการเคลื่อนที่ด้วยแสงอินฟราเรด

6. ป้อนโปรแกรมที่ 15.1 แล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 15.1 สังเกตการติด-ดับของหลอด LED

```
          PORT_A  EQU   0E020H
          PORT_B  EQU   PORT_A+1
          PORT_C  EQU   PORT_A+2
          C_PORT  EQU   PORT_A+3
                  ORG   8000H
8000 90E023  INIT:  MOV   DPTR,#C_PORT
8003 : 748A         MOV   A,#08AH
8005 : F0           MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 90E022 START: MOV  DPTR,#PORT_C
8009 : E0           MOVX  A,@DPTR
800A : 20E704        JB   ACC.7,LOOP; ตรวจสอบการตัดลำแสง
800D : D2E5         SETB  ACC.5
800F : 8002         SJMP  END
8011 : C2E5  LOOP:  CLR   ACC.5
8013 : 90E020 END:  MOV   DPTR,#PORT_A     ; ส่งผลที่ได้ไปยัง Port Monitor
8016 : F0           MOVX  @DPTR,A
8017 : 80ED         SJMP  START
                    END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทดลองปิด-เปิดช่องรับแสงของตัวรับแสงอินฟราเรดบันทึกผลการติดดับของหลอด LED

ลักษณะการรับแสง	LED	ข้อมูล
ปิดช่องรับแสงสนิท	O	
เปิดช่องรับแสง	O	

O = ดับ

● = ติด

8. ปรับสวิตช์ SW 4/2 มาที่ตำแหน่ง “OFF”
9. ปรับสวิตช์ SW 4/1 มาที่ตำแหน่ง “ON”
10. ป้อนโปรแกรมที่ 15.2 แล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 15.2 สังเกตการติด-ดับของหลอด LED บน Port Monitor

```
                PORT_A  EQU   0E020H
                PORT_B  EQU   PORT_A+1
                PORT_C  EQU   PORT_A+2
                C_PORT  EQU   PORT_A+3
                        ORG   8000H
8000 : 90E023   INIT:   MOV   DPTR,#C_PORT
8003 : 748A             MOV   A,#08AH
8005 : F0               MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 7A00             MOV   R2,#00H
8008 : 90E022   START:  MOV   DPTR,#PORT_C
800B : E0       LOOP:   MOVX  A,@DPTR
800C : 20E7FC           JB    ACC.7,LOOP       ; ตรวจสอบการตัดลำแสง
800F : E0       LOOP1:  MOVX  A,@DPTR
8010 : 30E7FC           JNB   ACC.7,LOOP1
8013 : 0A               INC   R2               ; เพิ่มค่าที่นับได้
8014 : 90E020           MOV   DPTR,#PORT_A
8017 : EA               MOV   A,R2
8018 : F0               MOVX  @DPTR,A          ; ส่งค่าที่นับได้ออกทาง Port Monitor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
8019 : 80ED          SJMP    START
                     END
```

11. ทดลองปิด-เปิดช่องรับแสงของตัวรับแสงอินฟราเรดบันทึกผลการติดดับของหลอดบน Port Monitor
12. จงอธิบายโปรแกรมที่ 15.1 และโปรแกรมที่ 15.2 มาพอสังเขป
13. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

14. จากผลการทดลองโปรแกรมที่ 15.1 จงอธิบายการติด-ดับ ของหลอด LED หมายถึงอะไร
15. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง "SWITCH INPUT"
16. จงเขียนโปรแกรมตรวจจับการเคลื่อนที่ของวัตถุที่ตัดผ่านช่องรับแสง โดยเมื่อมีการตัดลำแสงเกิดขึ้นให้ Buzzer ทำงานตลอด และสามารถรีเซตได้โดยการ "ON" Switch Input ณ. ตำแหน่งใดก็ได้
17. ปรับสวิตช์ SW 4/2 มาที่ตำแหน่ง "ON" และปรับสวิตช์ SW 4/1 มาที่ตำแหน่ง "OFF"
18. จงเขียนโปรแกรมตรวจนับการตัดช่องรับแสงอินฟราเรด โดยการตัดลำแสง 1 ครั้งจะทำให้ LED กระพริบ 1 ครั้ง และเพิ่มจำนวนการนับที่แสดงผลที่จอ 7-Segment หลักที่ 2 ของ Single Board ในรูปแบบของเลขฐานสิบ เมื่อนับครบถึง 9 ให้มาเริ่มต้นนับใหม่ (การตัด ลำแสง 1 ครั้ง หมายถึงการปิดกั้นทางเดินของลำแสงและการเปิดทางเดินของลำแสง) โดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H

    -0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter

    -0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment

    -0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch

    -E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 16

การรับข้อมูลจากคีย์แพด (Keypad)

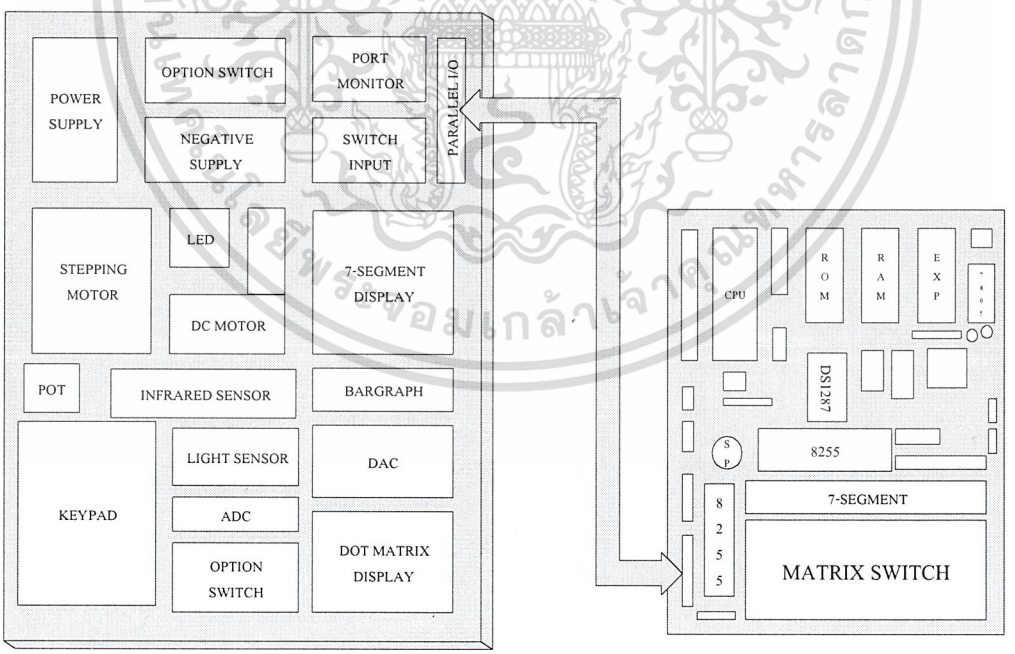
วัตถุประสงค์

1. เพื่อศึกษาวิธีการสแกนคีย์แพด (Keypad)
2. เพื่อให้สามารถนำคีย์แพด (Keypad) ไปประยุกต์ใช้งานได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 16 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



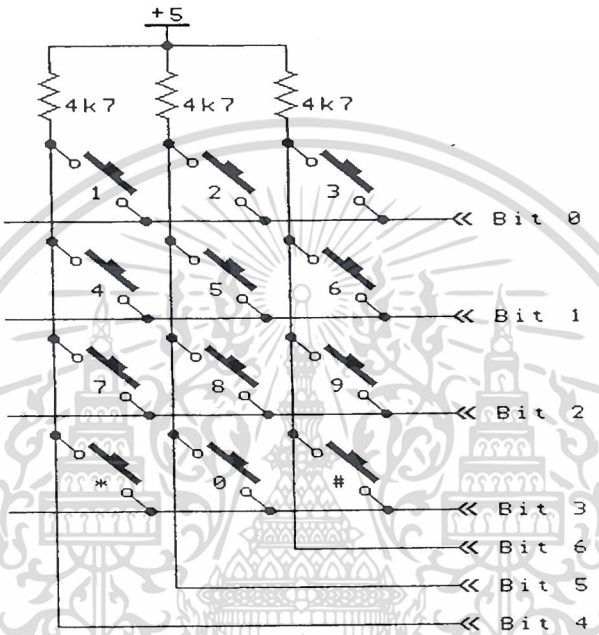
รูปที่ ข. 53 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

คีย์แพด (Keypad)

คีย์แพดของชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 เป็นตัวอย่างแสดงถึงหลักการทำงานของคีย์แพดในระบบไมโครคอมพิวเตอร์ซึ่งจะเป็นพื้นฐานในการประยุกต์ใช้งาน



รูปที่ ข. 54 วงจรภายในของคีย์แพด (Keypad)

จากวงจรคีย์แพดจะมีตัวต้านทาน 4.7 กิโลโอห์ม ต่อเข้ากับคอลัมน์ (Column) C1-C3 เพื่อทำให้มีสภาวะลอจิกเป็น “1” (HI) คอลัมน์ C1-C3 จะต่อเข้ากับพอร์ตอินพุตเพื่อตรวจสอบว่าคีย์ใดที่ถูกกด ส่วนแถว R1-R4 จะถูกต่อเข้ากับพอร์ตเอาต์พุต โดยสัญญาณเอาต์พุตของแต่ละแถว จะมีสภาวะลอจิกเป็น “0” (LO) ทีละแถวเมื่อแถวใดเป็น “0” แถวที่เหลือก็จะเป็น “1” โดยจะวนรอบอยู่อย่างนี้

เมื่อมีการกดคีย์ (ตามจุดในรูปที่ ข. 54) สัญญาณอินพุตของคอลัมน์ C1-C3 ที่เป็น “1” อยู่ทั้งหมด 111 จะกลายเป็น 110 เพราะที่คอลัมน์ C1 จะกลายเป็น “0” ในจังหวะที่แถว R2 มีสภาวะเป็น “0” เมื่อนำผลของอินพุตและเอาต์พุตมารวมด้วยกันก็สามารถบอกตำแหน่งของคีย์ที่ถูกกดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6, SW 1/7 และ SW 4/1 มาที่ตำแหน่ง “ON”
5. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
6. ป้อนโปรแกรมที่ 16.1 แล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 16.1 การหาค่าประจำคีย์แต่ละตัว

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 7831     START:  MOV     R0,#31H
8008 : 7930             MOV     R1,#30H
800A : 1112     LOOP:   ACALL   SCANKEY         ; เรียกใช้โปรแกรมย่อย SCANKEY
800C : 90E020           MOV     DPTR,#PORT_A
800F : F0               MOVX    @DPTR,A
8010 : 80F8             SJMP    LOOP

8012 : 90E022   SCANKEY:  MOV     DPTR,#PORT_C
8015 : 7A0E               MOV     R2,#0EH
8017 : 7B04               MOV     R3,#04H       ; กำหนดรอบในการ Scan
8019 : EA       SCAN1:    MOV     A,R2          ; กำหนดแถวที่ต้องการ Scan
801A : F0                 MOVX    @DPTR,A
801B : D3                 SETB    C
801C : 33                 RLC     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
801D : FA                MOV     R2,A              ; กำหนดแถวที่จะ Scan ต่อไป
801E : E0                MOVX    A,@DPTR
801F : 5470              ANL     A,#01110000B  ; ต้องการค่าที่ได้จากการกดคีย์
8021 : B47004            CJNE    A,#01110000B,SCAN2  ; เปรียบเทียบค่าที่ยังไม่ถูกกด
8024 : DBF3              DJNZ    R3,SCAN1
8026 : 80EA              SJMP    SCANKEY
8028 : F7        SCAN2:  MOV     @R1,A             ; นำค่าที่กดคีย์เก็บไว้ใน Buffer
8029 : EA                MOV     A,R2
802A : 03                RR      A
802B : F6                MOV     @R0,A
802C : E7                MOV     A,@R1
802D : D6                XCHD    A,@R0             ; สลับเปลี่ยนค่าที่ Scan กับค่าที่กดคีย์
802E : 22                RET
                         END
```

7. ทดลองกดคีย์ตัวเลขแต่ละตัว สังเกตการติด-ดับ ของ LED บน Port Monitor บันทึกผลการทดลองในตาราง

KEY	PORT MONITOR	CODE
0	OOOOOOOO	57H
1	OOOOOOOO	
2	OOOOOOOO	
3	OOOOOOOO	
4	OOOOOOOO	
5	OOOOOOOO	
6	OOOOOOOO	
7	OOOOOOOO	
8	OOOOOOOO	
9	OOOOOOOO	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEY	PORT MONITOR	CODE
*	OOOOOOOO	
#	OOOOOOOO	

O = ดับ

● = ติด

8. จากการทดลองโปรแกรมที่ 16.1 สามารถที่จะหาโค้ดประจำคีย์แต่ละตัวได้ จงอธิบายความหมายของโค้ดแต่ละตัวว่าได้มาอย่างไร
9. ทดลองรับค่าจากคีย์แพดมาเป็นเงื่อนไขกำหนดการทำงานของมอเตอร์ เมื่อกดคีย์เลข 1 ให้มอเตอร์หมุนตามเข็มนาฬิกา กดเลข 2 ให้หมุนทวนเข็มนาฬิกาและกดเลข 3 ให้หยุดหมุน
10. ปรับสวิตช์ SW 4/3 ให้อยู่ที่ตำแหน่ง “ON” และ SW 4/1 ให้อยู่ที่ตำแหน่ง “OFF”
11. ป้อนโปรแกรมที่ 16.2 แล้วสั่งให้โปรแกรมทำงานทดลองกดคีย์ หมายเลข 1, 2 และ 3 สังเกตการทำงานของมอเตอร์และบันทึกผลการทดลอง

โปรแกรมที่ 16.2 โปรแกรมควบคุมการทำงานของมอเตอร์ด้วยคีย์แพด

```
PORT_A  EQU   0E020H
PORT_B  EQU   PORT_A+1
PORT_C  EQU   PORT_A+2
C_PORT  EQU   PORT_A+3
        ORG    8000H
8000 : 90E023  INIT:   MOV   DPTR,#C_PORT
8003 : 748A            MOV   A,#08AH
8005 : F0              MOVX  @DPTR,A          ; กำหนด Command Word ให้ 8255
8006 : 7831    START:  MOV   R0,#31H
8008 : 7930            MOV   R1,#30H
800A : 112D    LOOP:   ACALL SCANKEY         ; เรียกใช้โปรแกรมย่อย SCANKEY
800C : 90E020          MOV   DPTR,#PORT_A
800F : B46E02  CHK1:   CJNE  A,#6EH,CHK2      ; เปรียบเทียบคีย์ที่กดกับค่าคีย์ “1”
8012 : 800A            SJMP  FORWARD
8014 : B45E02  CHK2:   CJNE  A,#5EH,CHK3      ; เปรียบเทียบคีย์ที่กดกับค่าคีย์ “2”
8017 : 800A            SJMP  REVERSE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8019 : B43EEE  CHK3:  CJNE    A,#3EH,LOOP        ; เปรียบเทียบคีย์ที่กดกับค่าคีย์ "3"
801C : 800A           SJMP    STOP

801E : 74C0    FORWARD:  MOV   A,#11000000B
8020 : F0                MOVX  @DPTR,A      ; สั่งให้มอเตอร์หมุนตามเข็มนาฬิกา
8021 : 80E7              SJMP  LOOP
8023 : 7480    REVERSE:  MOV   A,#10000000B
8025 : F0                MOVX  @DPTR,A      ; สั่งให้มอเตอร์หมุนทวนเข็มนาฬิกา
8026 : 80E2              SJMP  LOOP
8028 : 7400    STOP:     MOV   A,#00000000B
802A : F0                MOVX  @DPTR,A      ; สั่งให้มอเตอร์หยุดหมุน
802B : 80DD              SJMP  LOOP

802D : 90E022  SCANKEY:  MOV   DPTR,#PORT_C
8030 : 7A0E              MOV   R2,#0EH
8032 : 7B04              MOV   R3,#04H      ; กำหนดรอบในการ Scan
8034 : EA      SCAN1:    MOV   A,R2         ; กำหนดแถวที่ต้องการ Scan
8035 : F0                MOVX  @DPTR,A
8036 : D3                SETB  C
8037 : 33                RLC   A
8038 : FA                MOV   R2,A         ; กำหนดแถวที่จะ Scan ต่อไป
8039 : E0                MOVX  A,@DPTR
803A : 5470              ANL   A,#01110000B ; ต้องการค่าที่ได้จากการกดคีย์
803C : B47004            CJNE  A,#01110000B,SCAN2  ; เปรียบเทียบค่าที่ยังไม่ถูกกด
803F : DBF3              DJNZ  R3,SCAN1
8041 : 80EA              SJMP  SCANKEY
8043 : F7      SCAN2:    MOV   @R1,A        ; นำค่าที่กดคีย์เก็บไว้ใน Buffer
8044 : EA                MOV   A,R2
8045 : 03                RR    A
8046 : F6                MOV   @R0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8047 : E7          MOV   A,@R1
8048 : D6          XCHD  A,@R0        ; สลับเปลี่ยนค่าที่ Scan กับค่าที่กดคีย์
8049 : 22          RET
                   END
```

12. จงอธิบายการทำงานของโปรแกรมที่ 16.2 มาพอสังเขป
13. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

14. จงอธิบายหลักในการ Scankey มาพอสังเขป
15. จงเขียนโปรแกรมรับค่าการกดคีย์ และนำไปแสดงผลที่ 7-Segment ของ Single Board หลักที่ 6 โดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H
    -0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter
    -0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment
    -0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch
    -0E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)
16. ปรับสวิตช์ SW 4/1 ให้อยู่ที่ตำแหน่ง “ON” และ SW 4/3 ให้อยู่ที่ตำแหน่ง “OFF”
17. จากโปรแกรมที่ 16.2 ให้เปลี่ยนการควบคุมเป็นไฟวิ่ง เมื่อกดคีย์เลข 1 ให้ไฟวิ่งไปทางซ้าย กดเลข 2 ให้ไฟวิ่งไปทางขวาและกดเลข 3 ให้หยุดวิ่ง (ลักษณะการวิ่งของ LED จะต่อเนื่องกัน ไม่มีการเริ่มต้นใหม่เมื่อมีการเปลี่ยนค่าคีย์)
18. จงเขียนโปรแกรมรับค่าการกดคีย์จำนวน 1 ค่า นำไปแสดงผลที่ 7-Segment ของ Single Board หลักที่ 1 ซึ่งเป็นจำนวนรอบในการแสดงไฟวิ่ง ค่าที่รับมาจะต้องลดลงหนึ่งเมื่อครบการแสดงผลในหนึ่งรอบและจะเริ่มลดลงเมื่อมีการกดคีย์ # เท่านั้น เมื่อเป็นศูนย์ให้รอรับค่าคีย์ที่กดใหม่ โดย 8255 ที่ใช้เพิ่มเติมอยู่ที่ตำแหน่ง 0E000H-0E003H
    -0E000H เป็นเอาต์พุต ส่งข้อมูลให้แก่วงจร Digital to Analog Converter
    -0E001H เป็นเอาต์พุต ส่งข้อมูลให้แก่ 7-Segment
    -0E002H Upper เป็นอินพุต รับข้อมูลจาก Matrix Switch
    -0E002H Lower เป็นเอาต์พุต ส่งข้อมูลเลือกหลักในการแสดงผล (00H-06H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 17

การควบคุมความเร็วมอเตอร์แบบ Pulse Width Modulation

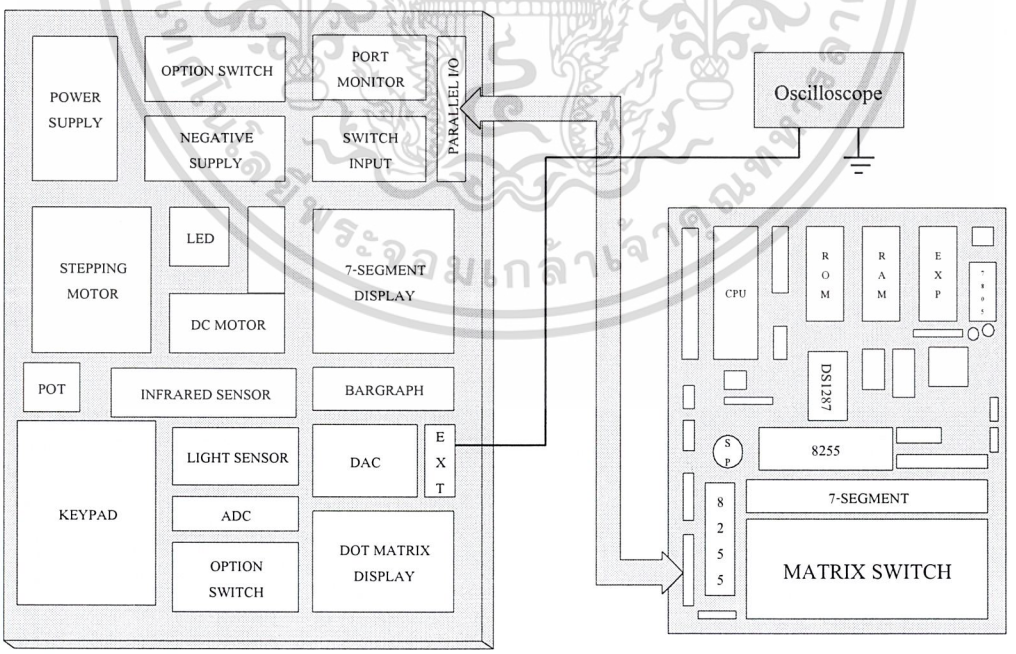
วัตถุประสงค์

1. เพื่อศึกษาการควบคุมความเร็วมอเตอร์ด้วยวิธี Pulse Width Modulation
2. เพื่อให้สามารถควบคุมความเร็วมอเตอร์ด้วยวิธี Pulse Width Modulation ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 17 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. ออสซิลโลสโคป 1 เครื่อง
5. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 55 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

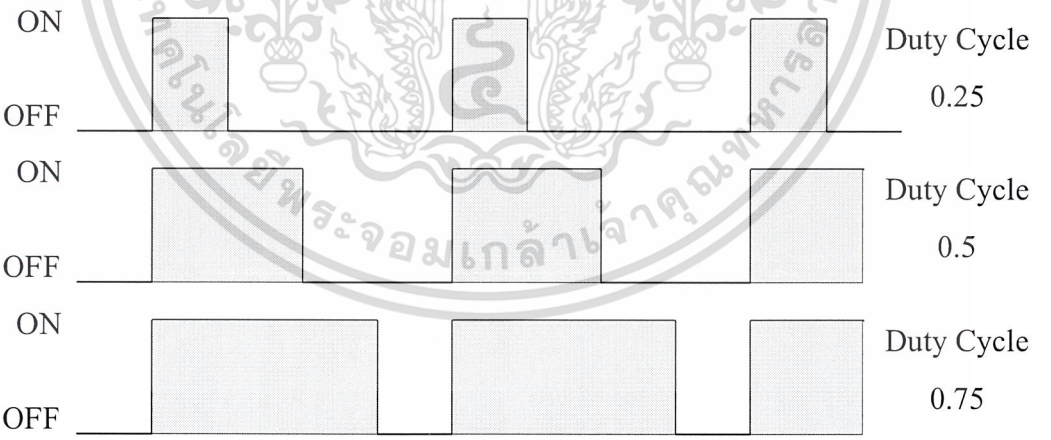
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การควบคุมความเร็วมอเตอร์แบบ Pulse Width Modulation

โดยปกติแล้วเมื่อจ่ายไฟกระแสตรงให้กับมอเตอร์ จะมีผลทำให้มอเตอร์หมุนด้วยความเร็วสูงสุด ดังนั้นการควบคุมความเร็วการหมุนของมอเตอร์วิธีหนึ่งอาจทำได้โดยการปรับค่าของตัวต้านทานซึ่งเชื่อมต่ออยู่กับแหล่งจ่ายไฟกระแสตรง เพื่อปรับระดับของแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ แต่วิธีการนี้จะทำให้เกิดข้อเสียในด้านของการสูญเสียกำลังไฟฟ้า (Power Consumption) และความร้อนที่เกิดขึ้นมากตลอดเวลา ดังนั้นจึงมักเลี่ยงไปใช้การควบคุมแบบสวิตช์ (Switch Type Control) เพื่อจ่ายกระแสไฟฟ้าเป็นช่วงๆ ให้กับมอเตอร์แทน ซึ่งตามหลักการแล้วหากความเร็วในการสวิตช์นั้นมากเพียงพอก็สามารถทำให้มอเตอร์หมุนด้วยความเร็วต่างๆ ได้ เทคนิคที่นิยมใช้งานมี 2 วิธีดังนี้ คือ

1. แบบ Pulse Width Modulation (PWM) เป็นการใช้ความถี่ที่คงที่ ในการสร้างจังหวะในการเปิด-ปิดสวิตช์ ซึ่งจะควบคุมการจ่ายกระแสไฟฟ้าให้กับมอเตอร์อีกต่อหนึ่ง สำหรับการควบคุมระดับไฟฟ้านั้น ใช้การปรับช่วงเวลาของการเปิดซึ่งจะสัมพันธ์กับช่วงเวลาของการปิดสวิตช์แทน (ซึ่งอัตราส่วนนี้เรียกว่า ดิวตี้ไซเคิล (Duty Cycle)) ดังแสดงในรูปที่ ข. 56 ดังนั้นจึงมีชื่อเรียกอีกอย่างหนึ่งว่า “การควบคุมดิวตี้ไซเคิล”

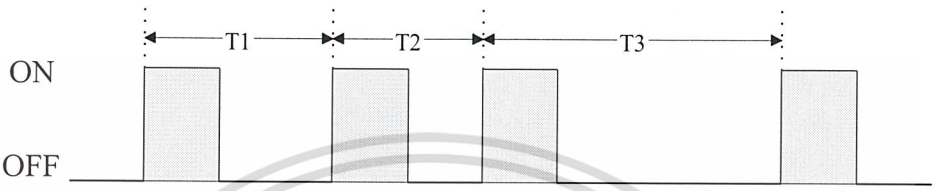


รูปที่ ข. 56 สัญญาณพัลส์ที่ได้จากการจังหวะการเปิด-ปิดสวิตช์
เพื่อควบคุมความเร็วของมอเตอร์แบบ PWM

2. แบบ Pulse Frequency Modulation (PFW) เป็นการควบคุมจังหวะเวลาการเปิด-ปิดที่ควบคุมการจ่ายกระแสไฟฟ้าให้กับมอเตอร์ มอเตอร์จึงจะเปิด-ปิดตามจังหวะเวลานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกันดังแสดงในรูปที่ ข. 57 ดังนั้นความถี่การควบคุมจังหวะเปิด-ปิดจะมีผลต่อการควบคุมระดับโวลต์เตจที่จะเปลี่ยนแปลงความเร็วการหมุนของมอเตอร์นั่นเอง หรือจะกล่าวในอีกลักษณะหนึ่งว่า อัตราเร็วในการหมุนของมอเตอร์ จะมีความสัมพันธ์โดยตรงกับอัตราพัลส์การเปิด-ปิดสวิตช์นี้

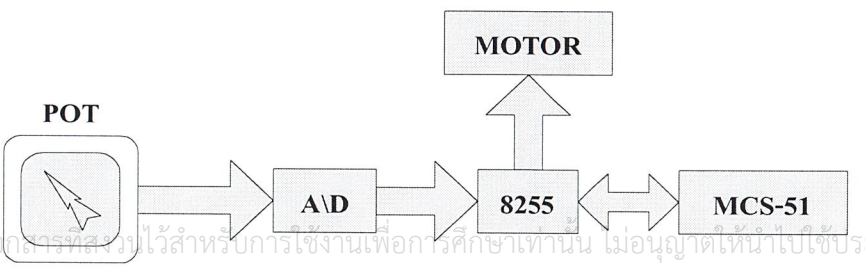


รูปที่ ข. 57 สัญญาณพัลส์ที่ได้จากการจังหวะการเปิด-ปิดสวิตช์
เพื่อควบคุมความเร็วของมอเตอร์แบบ PFM

นอกจากนี้ได้มีการติดตั้งวงจรตรวจจับความเร็วแบบใช้แสง โดยจัดวางอยู่ในตำแหน่งด้านหน้าและด้านหลังในแนวการหมุนใบพัดของมอเตอร์ เมื่อใบพัดหมุนจะเกิดการตัดลำแสงจากตัวส่งแสงอินฟราเรดไปยังตัวรับ ทำให้เกิดการเปลี่ยนสภาวะสัญญาณที่วงจรทางด้านรับ จากปกติเป็นสภาวะ “1” ไปเป็นสภาวะ “0” ดังนั้นเมื่อเชื่อมต่อสัญญาณนี้เข้ากับไมโครคอนโทรลเลอร์ จึงทำให้ทราบสภาวะ ของวงจรตรวจจับความเร็วได้

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6, SW 1/7, SW 2/1 และ SW 4/3 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “ADC”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข. 58 แผนผังการทำงานการควบคุมความเร็วมอเตอร์แบบ Pulse Width Modulation

6. จ่ายไฟเลี้ยงให้กับ Single Board และรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 17.1 แล้วสั่งโปรแกรมทำงาน

โปรแกรมที่ 17.1 ใช้ POT ควบคุมความเร็วของมอเตอร์

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E021   START:  MOV     DPTR,#PORT_B
8009 : E0               MOVX    A,@DPTR
800A : FA               MOV     R2,A            ; กำหนดช่วงของการหยุดหมุน
800B : B40002           CJNE    A,#00H,LOOP1
800E : 801B             SJMP    ON
8010 : B4FF02   LOOP1:  CJNE    A,#0FFH,NEXT
8013 : 801E             SJMP    OFF
8015 : 74FF     NEXT:   MOV     A,#0FFH
8017 : 9A               SUBB    A,R2            ; หาค่าของการมหนุนของมอเตอร์
8018 : FB               MOV     R3,A            ; กำหนดช่วงของการหมุน

8019 : 7480     DC_HIGH: MOV    A,#80H
801B : 90E020            MOV    DPTR,#PORT_A
801E : F0                MOVX   @DPTR,A         ; สั่งให้มอเตอร์หมุนตามค่าที่ได้
801F : DBF8              DJNZ   R3,DC_HIGH
8021 : 7400     DC_LOW:  MOV    A,#00H
8023 : 90E020            MOV    DPTR,#PORT_A
8026 : F0                MOVX   @DPTR,A         ; สั่งให้มอเตอร์หยุดหมุนตามค่าที่ได้
8027 : DAF8              DJNZ   R2,DC_LOW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

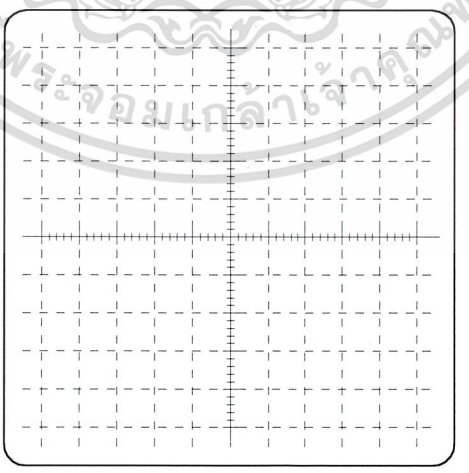
```
8029 : 80DB              SJMP    START

802B : 90E020   ON:      MOV     DPTR,#PORT_A      ; ช่วงที่มอเตอร์หมุนเร็วที่สุด
802E : 7480              MOV     A,#80H
8030 : F0                MOVX    @DPTR,A
8031 : 80D3              SJMP    START

8033 : 90E020   OFF:     MOV     DPTR,#PORT_A      ; ช่วงที่มอเตอร์หยุดหมุน
8036 : 7400              MOV     A,#00H
8038 : F0                MOVX    @DPTR,A
8039 : 80CB              SJMP    START
                         END
```

8. ทดลองปรับ POT แล้วสังเกตความเร็วของมอเตอร์และบันทึกผลการทดลอง
9. ใช้ออสซิลโลสโคปวัดสัญญาณที่ขา 18 ของ IC2 บันทึกรูปสัญญาณเมื่อปรับ POT ไว้ที่ตำแหน่งต่างๆ กัน

ตารางบันทึกรูปคลื่น



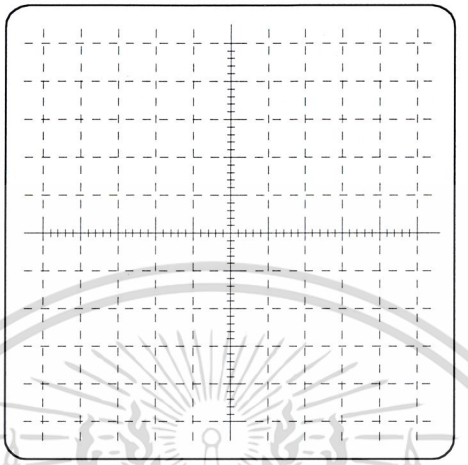
ความเร็วต่ำ

Volt/div = _____

Time/div = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางบันทึกรูปคลื่น

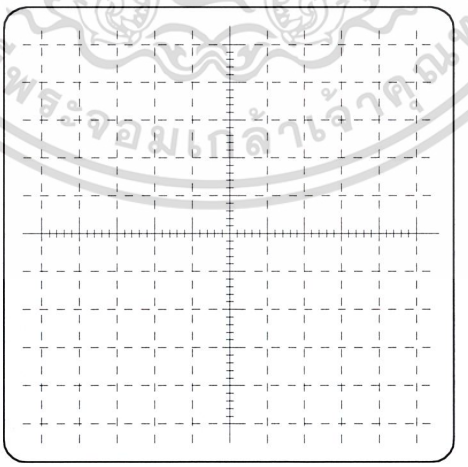


ความเร็ว 50%

Volt/div = _____

Time/div = _____

ตารางบันทึกรูปคลื่น



ความเร็วสูงสุด

Volt/div = _____

Time/div = _____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. จงอธิบายสาเหตุที่มอเตอร์หมุนช้า-เร็ว เมื่อปรับ POT ให้มีค่าต่างกัน
11. ป้อนโปรแกรมที่ 17.2 แล้วสั่งให้โปรแกรมทำงาน สังเกตการหมุนของมอเตอร์และบันทึกผลการทดลอง

โปรแกรมที่ 17.2 โปรแกรมควบคุมการทำงานของมอเตอร์แบบ PWM

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     8000H
8000 : 90E023   INIT:   MOV     DPTR,#C_PORT
8003 : 748A             MOV     A,#08AH
8005 : F0               MOVX    @DPTR,A         ; กำหนด Command Word ให้ 8255
8006 : 90E020   START:  MOV     DPTR,#PORT_A
8009 : 7480             MOV     A,#80H
800B : F0               MOVX    @DPTR,A         ; สั่งให้มอเตอร์หมุน
800C : 7D04             MOV     R5,#04H
800E : 1118             ACALL   DELAY
8010 : E4               CLR     A
8011 : F0               MOVX    @DPTR,A         ; สั่งให้มอเตอร์หยุดหมุน
8012 : 7D01             MOV     R5,#01H
8014 : 1118             ACALL   DELAY
8016 : 80EE             SJMP    START

8018 : 7EBE     DELAY:  MOV     R6,#190         ; เวลา=12*(1/11.0592*10^6)*
801A : 7FF1     DELAY1: MOV     R7,#241         ;       (R5(1+R6(1+R7(2)+2)
801C : DFFE     DELAY2: DJNZ    R7,DELAY2       ;       +2)+2)
801E : DEFA             DJNZ    R6,DELAY1       ;100 มิลลิวินาที
8020 : DDF6             DJNZ    R5,DELAY
8022 : 22               RET
                        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. จงแก้ไขโปรแกรมที่ 17.2 โดยให้มีช่วงการ OFF มากกว่าช่วง ON
13. สั่งให้โปรแกรมทำงาน สังเกตการหมุนของมอเตอร์และบันทึกผลการทดลอง
14. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

15. จงอธิบายหลักในการ Pulse Width Modulation มาพอสังเขป
16. จงอธิบายผลที่เกิดขึ้นกับมอเตอร์ เมื่อใช้โปรแกรมข้อที่ 11 และ 12
17. จากโปรแกรมที่ 17.1 ลองเปลี่ยนอินพุตของวงจร A/D เป็น Light Sensor สังเกตการหมุนของมอเตอร์และบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 18

การวัดค่าความเร็วรอบของมอเตอร์ (Speed Sensor)

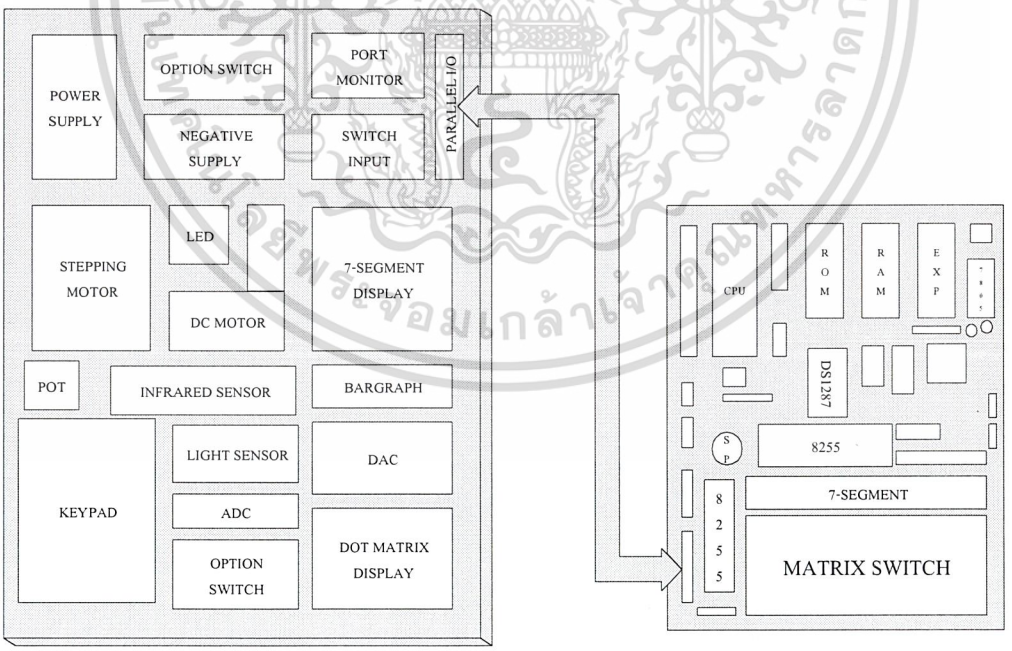
วัตถุประสงค์

1. เพื่อศึกษาการวัดความเร็วรอบของมอเตอร์ด้วยแสงอินฟราเรด
2. เพื่อให้สามารถวัดความเร็วรอบของมอเตอร์ด้วยแสงอินฟราเรดได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 18 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 59 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

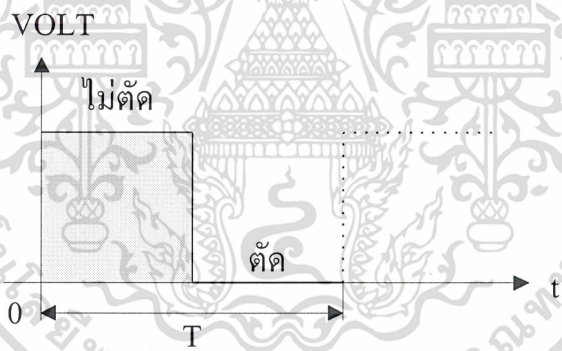
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การตรวจจับความเร็ว (Speed Sensor)

การตรวจจับความเร็วรอบของมอเตอร์ (Speed Sensor) จะอาศัยใบพัดของมอเตอร์หมุนตัดผ่านแสงอินฟราเรด (Infrared) ที่ส่องมายังโฟโตไดโอด (Photo Diode) เมื่อลำแสงอินฟราเรดถูกบังคับโดยใบพัดของมอเตอร์ วงจรตรวจจับความเร็วก็จะส่งเอาต์พุตออกมาเป็น "0" เมื่อใบพัดของมอเตอร์เลื่อนผ่านไป โฟโตไดโอดก็สามารถที่จะรับแสงจากหลอดอินฟาเรดก็จะมีผลทำให้เอาต์พุตเป็น "1"

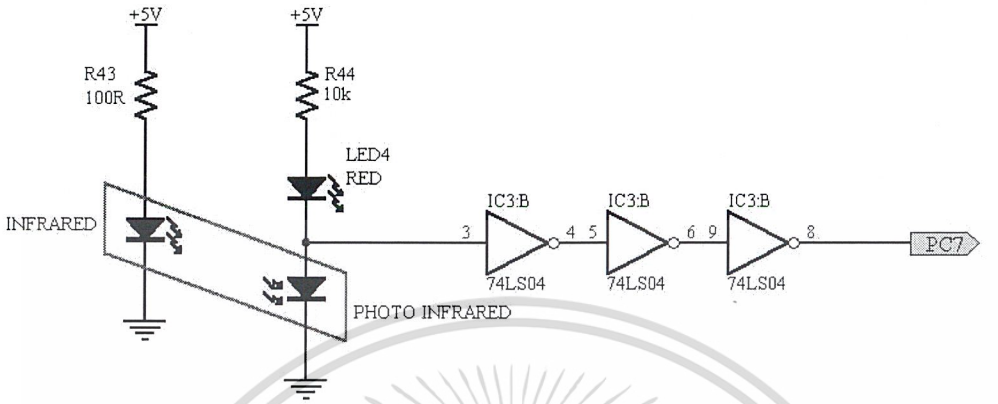
ใบพัดของมอเตอร์เป็นแบบ 3 ใบพัด ดังนั้นเมื่อมอเตอร์หมุนไป 1 รอบ ใบพัดก็จะไปตัดกับแสงอินฟราเรด อันที่จริงเอาต์พุตของวงจรตรวจจับความเร็วจะมีสภาวะเป็น "1" และ "0" อย่างละ 1 ครั้งสลับกันจะเร็วหรือช้า ขึ้นอยู่กับความเร็วของมอเตอร์ แต่ใบพัดที่ใช้นั้นไม่มีความทึบแสงพอที่จะตัด ลำแสงอินฟราเรดได้ จึงต้องหาอุปกรณ์ทึบแสงมาปิดทับใบพัดหนึ่งใบ ดังนั้นเมื่อใบพัดเคลื่อนที่ตัดลำแสงหนึ่งรอบ เอาต์พุตจะมีสภาวะเป็น "1" และ "0" เพียง 1 ครั้งเท่านั้น



รูปที่ ข. 60 เอาต์พุตของวงจรตรวจจับความเร็วเมื่อมอเตอร์หมุน 1 รอบ

หลอดอินฟราเรดจะส่งแสงอินฟราเรดออกมาอยู่ตลอดเวลา โดยมีตัวต้านทาน 100 โอห์มเป็นตัวจำกัดกระแสให้หลอดอินฟราเรด เมื่อโฟโตไดโอดได้รับแสงอินฟราเรดทำให้เอาต์พุตที่ขาของโฟโตไดโอดมีสภาวะเป็น "0" เมื่อผ่าน NOT Gate แล้วจะมีค่าเป็น "1" ในทางกลับกัน เมื่อมีวัตถุเคลื่อนที่ผ่านลำแสง ลำแสงจะไม่ผ่านไปยังตัวรับทำให้ได้เอาต์พุตที่ขาของตัวรับเป็น "1" จากนั้นจะเข้าไปยัง NOT Gate จำนวน 3 ตัวเพื่อแปลงสัญญาณที่ได้เป็นสัญญาณ TTL เอาต์พุตที่ได้นี้จะมีค่าเป็น "0" เพื่อที่จะส่งไปยังพอร์ต PC ของ 8255 ตำแหน่งบิตที่ 7

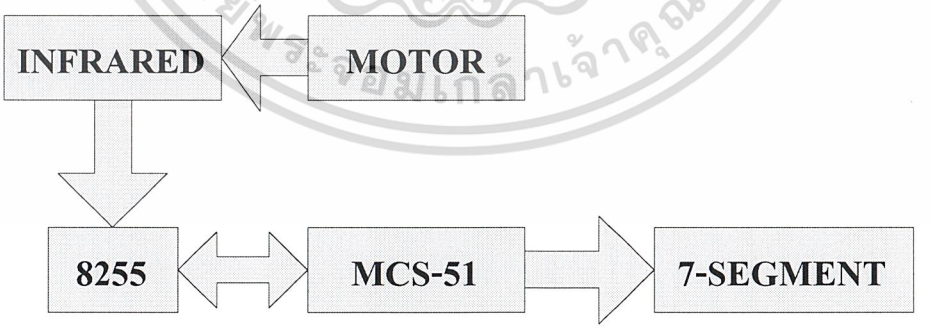
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข. 61 วงจรตรวจจับความเร็ว (Speed Sensor)

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 1/6, SW 1/7 และ SW 2/1 มาที่ตำแหน่ง “ON”
5. ปรับสวิตช์ SW 3 มาที่ตำแหน่ง “ADC”



รูปที่ ข. 62 แผนผังการทำงานการตรวจจับความเร็วมอเตอร์ (Speed Sensor)

6. จ่ายไฟเลี้ยงให้กับ Single Board และรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ป้อนโปรแกรมที่ 18.1 แล้วสั่งโปรแกรมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 18.1 โปรแกรมวัดความเร็วรอบของมอเตอร์ (Speed Sensor)

```
                PORT_A  EQU    0E020H
                PORT_B  EQU    PORT_A+1
                PORT_C  EQU    PORT_A+2
                C_PORT  EQU    PORT_A+3
                PORT_A2 EQU    0E000H
                PORT_B2 EQU    PORT_A2+1
                PORT_C2 EQU    PORT_A2+2
                C_PORT2 EQU    PORT_A2+3
                        ORG    8000H
8000 : 90E023   INIT:   MOV    DPTR,#C_PORT
8003 : 748A             MOV    A,#08AH
8005 : F0               MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255#1
8006 : 90E003           MOV    DPTR,#C_PORT2
8009 : 7488             MOV    A,#88H
800B : F0               MOVX   @DPTR,A          ; กำหนด Command Word ให้ 8255#2

800C : 90E020   START:  MOV    DPTR,#PORT_A
800F : 7480             MOV    A,#80H
8011 : F0               MOVX   @DPTR,A          ; สั่งให้มอเตอร์หมุน
8012 : 7800     LOOP:   MOV    R0,#00H
8014 : 7900             MOV    R1,#00H
8016 : 90E022           MOV    DPTR,#PORT_C
8019 : 1127             ACALL  CHKL
801B : 113A             ACALL  CAL
801D : 1145             ACALL  HTOD
801F : 7D00             MOV    R5,#00H
8021 : 1164     DISPP:  ACALL  DISP
8023 : DDFC             DJNZ   R5,DISPP
8025 : 80E5             SJMP   START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8027 : E0       CHKL:  MOVX   A,@DPTR
8028 : 30E7FC          JNB    ACC.7,CHKL
802B : E0       CHKH:  MOVX   A,@DPTR
802C : 20E7FC          JB     ACC.7,CHKH

        ;******* COUNT SPEED *******
802F : E0       COUNTL: MOVX  A,@DPTR        ; 2T โปรแกรมนับเวลาที่ใช้ใน 1 รอบ
8030 : 08               INC    R0            ; 1T
8031 : 30E7FB           JNB    ACC.7,COUNTL  ; 2T
8034 : E0       COUNTH: MOVX  A,@DPTR        ; 2T
8035 : 08               INC    R0            ; 1T
8036 : 20E7FB           JB     ACC.7,COUNTH  ; 2T
8039 : 22               RET

        ;******* CALULATE SPEED *******
803A : 740A     CAL:   MOV    A,#0AH         ; โปรแกรมคำนวณเวลาทีใช้ใน 1 รอบ
803C : 88F0            MOV    B,R0
803E : A4              MUL    AB
803F : F582            MOV    DPL,A
8041 : 85F083          MOV    DPH,B
8044 : 7405            MOV    A,#05H
8046 : 89F0            MOV    B,R1
8048 : A4              MUL    AB
8049 : 2582            ADD    A,DPL
804B : F582            MOV    DPL,A
804D : E5F0            MOV    A,B
804F : 2583            ADD    A,DPH
8051 : F583            MOV    DPH,A
8053 : 22              RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;******* HEX TO DEC *******
8054 : E4       HTOD:  CLR   A              ; โปรแกรมแปลงฐาน 16 เป็นฐาน 10
8055 : F9              MOV   R1,A
8056 : FA              MOV   R2,A
8057 : FB              MOV   R3,A
8058 : 7C10            MOV   R4,#16
805A : E582     HTOD1: MOV   A,DPL
805C : 33              RLC   A
805D : F582            MOV   DPL,A
805F : E583            MOV   A,DPH
8061 : 33              RLC   A
8062 : F583            MOV   DPH,A
8064 : 7D03            MOV   R5,#3
8066 : 7803            MOV   R0,#3
8068 : E6       HTOD2: MOV   A,@R0
8069 : 35E0            ADDC  A,ACC
806B : D4              DA    A
806C : F6              MOV   @R0,A
806D : 18              DEC   R0
806E : DDF8            DJNZ  R5,HTOD2
8070 : DCE8            DJNZ  R4,HTOD1
8072 : 22              RET

        ;******* DISPLAY  TO 7-SEGMENT *******
8073 : 7805     DISP:  MOV   R0,#05H        ; หาค่าหลักที่ 1
8075 : EB              MOV   A,R3
8076 : 540F            ANL   A,#0FH
8078 : 1197            ACALL D_OUT0
807A : EB              MOV   A,R3           ; หาค่าหลักที่ 2
807B : 54F0            ANL   A,#0F0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
807D : 03          RR      A
807E : 03          RR      A
807F : 03          RR      A
8080 : 03          RR      A
8081 : 1197        ACALL   D_OUT0
8083 : EA          MOV     A,R2          ; หาค่าหลักที่ 3
8084 : 540F        ANL     A,#0FH
8086 : 1197        ACALL   D_OUT0
8088 : EA          MOV     A,R2          ; หาค่าหลักที่ 4
8089 : 54F0        ANL     A,#0F0H
808B : 03          RR      A
808C : 03          RR      A
808D : 03          RR      A
808E : 03          RR      A
808F : 1197        ACALL   D_OUT0
8091 : E9          MOV     A,R1          ; หาค่าหลักที่ 5
8092 : 540F        ANL     A,#0FH
8094 : 1197        ACALL   D_OUT0
8096 : 22          RET

;******* CHANGE DATA TO SEGMENT1 *******
8097 : 908100  D_OUT0: MOV   DPTR,#TABLE   ; โปรแกรมเปลี่ยน BCD เป็น
809A : F582            MOV   DPL,A         ; รหัส 7-Segment แล้วส่งออก
809C : E0              MOVX  A,@DPTR
809D : 90E001          MOV   DPTR,#PORT_B2
80A0 : F0              MOVX  @DPTR,A
80A1 : 90E002          MOV   DPTR,#PORT_C2
80A4 : E8              MOV   A,R0
80A5 : F0              MOVX  @DPTR,A
80A6 : 11AA            ACALL DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
80A8 : 18              DEC     R0
80A9 : 22              RET

80AA : 7E05    DELAY:  MOV     R6,#05H     ; โปรแกรมย่อยหน่วงเวลา
80AC : 7F00    DELAY1: MOV     R7,#00H
80AE : DFFE    DELAY2: DJNZ    R7,DELAY2
80B0 : DEFA            DJNZ    R6,DELAY1
80B2 : 22              RET

                       ORG     8100H
8100 : 3F065B4F TABLE: DB      3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
8104 : 666D7D07
8108 : 7F6F
                       END
```

8. ปรับสวิตช์ SW 4/3 มาที่ตำแหน่ง “ON” แล้วสังเกตตัวเลขที่ปรากฏที่ 7-Segment และบันทึกผลแต่ละครั้ง (รอประมาณ 4 วินาทีก่อนบันทึกครั้งแรก)

ครั้งที่	ค่าของตัวเลข
1	
2	
3	
4	
5	
ค่าเฉลี่ย	

9. นำค่าที่ได้มาคำนวณหาค่าเวลาที่ใช้ใน 1 รอบจากสมการ

$$\text{ค่าเวลาที่ใช้ใน 1 รอบ} = \text{ค่าเฉลี่ยที่ได้} \times \frac{12}{11.0592 \times 10^{6}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ค่าที่ได้จากการคำนวณเท่ากับ _____มิลลิวินาที
11. นำค่าที่ได้มาคำนวณหาค่าความเร็วจากสมการ

$$\text{ค่าความเร็วที่ได้} = \frac{1{,}000 \times 10^{-3}}{\text{ค่าเวลาที่ใช้ใน 1 รอบ}}$$

12. ค่าที่ได้จากการคำนวณเท่ากับ _____รอบต่อวินาที
13. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

14. จงอธิบายการทำงานของโปรแกรมที่ 18.1 มาโดยละเอียด
15. ในปัจจุบันการวัดความเร็วรอบของมอเตอร์นำไปใช้กับอุปกรณ์ชนิดใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 19

การส่งชุดคำสั่งผ่านช่องสื่อสารอนุกรม RS-232

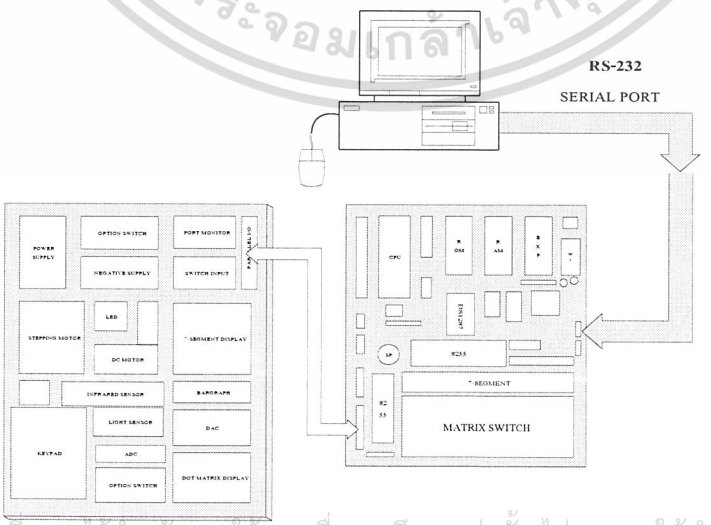
วัตถุประสงค์

1. เพื่อศึกษาการใช้โปรแกรม SXA51 ในการแปลงโปรแกรมแอสเซมบลีให้อยู่ในรูปแบบของ Intel Hex File
2. เพื่อศึกษารูปแบบของ Intel Hex File
3. เพื่อศึกษาการส่งชุดคำสั่งผ่านช่องสื่อสารอนุกรม RS-232
4. เพื่อสามารถเขียนโปรแกรมเป็นภาษาแอสเซมบลีของ MCS-51 และทำการแปลงโปรแกรมให้อยู่ในรูปแบบ Intel Hex File ได้
5. เพื่อสามารถส่งชุดคำสั่งจากเครื่องคอมพิวเตอร์ไปยังซิงเกิ้ลบอร์ดผ่านทางช่องสื่อสารอนุกรม RS-232 ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 19 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข. 63 ลักษณะการเชื่อมต่อชุดทดลอง,ซิงเกิ้ลบอร์ดและเครื่องคอมพิวเตอร์

ทฤษฎีเบื้องต้น

โปรแกรม SXA51

โปรแกรม SXA51 เป็นโปรแกรม 8051/52 ครอสแอสเซมเบลอร์ เวอร์ชัน 1.3 ของบริษัท Binary Technology ใช้สำหรับการแปลง Source Program ซึ่งเขียนเป็นภาษาแอสเซมบลีของ MCS-51 (FILE.ASM) และได้เอาต์พุตเป็น Intel Hex File (FILE.HEX) การเขียน Source Program สามารถกระทำจากโปรแกรม Editor ใดๆ ก็ได้ แต่ต้องทำการบันทึกเป็น FILE.ASM เท่านั้น จากนั้นนำมาแปลงด้วย SXA51 ซึ่งทำได้โดยการเรียกใช้โปรแกรม SXA51.EXE และตามด้วยชื่อของ Source Program ที่เขียนขึ้น การแปลงของ SXA51 จะกระทำในแบบ 2 Pass Assembler ซึ่งเมื่อมีข้อผิดพลาดใดๆ เกิดขึ้น SXA51 ก็จะแสดงให้ทราบในขณะที่แปลง ลักษณะของ Source Program จะเป็นไปตามมาตรฐานของการเขียนภาษา Assembler ทั่วๆ ไป

การใช้งานโปรแกรมแอสเซมเบลอร์ SXA51.EXE โดยแอสเซมเบลอร์จะทำการแปลง Source File ซึ่งถูกกำหนดไว้เป็น FILE.ASM ซึ่งเมื่อทำการแปลงแล้ว จะได้ไฟล์เอาต์พุตเป็น Intel Hex File ซึ่งถูกกำหนดไว้เป็น FILE.HEX โดย FILE.HEX ผู้ใช้สามารถนำไปโหลดเข้า Single Board ต่างๆ หรือโปรแกรมเข้าตัว EPROM เพื่อนำไปทดสอบโปรแกรมตามที่เขียนกับระบบได้ นอกจากนี้ SXA51 ยังสามารถสร้าง Listing File ต่างๆ ได้ตามรูปแบบของ Option ดังนี้

C:\>SXA51 <-OPTION> FILE.ASM

โดย

FILE.ASM คือชื่อไฟล์ของโปรแกรม กรณีที่ไม่ใส่นามสกุลของไฟล์ ตัว SXA51 จะกำหนดให้เป็น .ASM โดยอัตโนมัติ

<-OPTION> เป็นส่วนของการกำหนด Options ต่างๆ ประกอบด้วยเครื่องหมาย "-" และตามด้วยอักษรเพียงตัวเดียว การกำหนด Option นี้สามารถทำได้มากกว่า 1 ตัวในครั้งเดียว รายละเอียดของ Option จะมีดังนี้

–L หมายถึงให้สร้างไฟล์แสดงรายละเอียด (Listing) โดยไฟล์เอาต์พุตที่ได้คือ FILE.LST

–N หมายถึงให้คอมไพล์ โดยไม่สร้างไฟล์เอาต์พุตใดๆ ทั้งนี้เพื่อการตรวจสอบความถูกต้องของโปรแกรมอย่างรวดเร็ว

-C หมายถึงให้สร้างไฟล์แสดงรายละเอียด เช่นเดียวกับ –L โดยจะเพิ่ม Symbol cross-reference ไว้ที่ท้ายของไฟล์ ทั้งนี้เพื่อความสะดวกในการค้นหาตำแหน่งต่างๆ ตามชื่อ LABEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

–D ให้แสดง Process ขณะที่กำลังทำการ Assembler โดยโปรแกรมจะ แสดงเลขบรรทัดที่กำลังแปลงอยู่

ตัวอย่างการใช้งาน SXA51

C:\>SXA51 PROJECT

หมายถึง SOURCE: PROJECT.ASM
LISTING: ไม่สร้าง
HEX: สร้าง HEX FILE ชื่อ PROJECT.HEX

C:\>SXA51 -L PROJECT.ASM

หมายถึง SOURCE: PROJECT.ASM
LISTING: สร้าง LISTING FILE ชื่อ PROJECT.LST
HEX: สร้าง HEX FILE ชื่อ PROJECT.HEX

ข้อกำหนดและรูปแบบของ Source File

1. ในแต่ละบรรทัดต้องมีคำสั่งในภาษาแอสเซมบลี หรือ Assembler Directive เพียงหนึ่ง คำสั่งเท่านั้น
2. Label Declaration ทุกเลเบลต้องปิดท้ายด้วยเครื่องหมาย " :" และไม่ซ้ำกัน
3. โปรแกรม SXA51 จะมองตัวอักษรตัวเล็กหรือใหญ่เหมือนกัน เช่น hh: หรือ HH:
4. บรรทัดว่างในโปรแกรมต้นแบบจะยังคงปรากฏในไฟล์แสดงรายละเอียด (FILE.LST) แต่ในไฟล์แสดงภาษาเครื่อง (HEX.FILE) จะถูกตัดทิ้งไป
5. Source File ในแต่ละบรรทัดต้องประกอบด้วยตัวอักษรไม่เกิน 128 ตัวอักษร ซึ่งในแต่ละบรรทัดต้องมีรูปแบบดังนี้

[LABEL:] **[OPERATION [OPERAND1][, OPERAND2]]** **[; COMMENT]**

LABEL เป็นกลุ่มตัวอักษรและตัวเลขที่จะใช้เป็นตัวแปร หรือตำแหน่งหน่วยความจำสำรอง สำหรับอ้างอิงในโปรแกรม LABEL ต้องขึ้นต้นด้วยตัวอักษร A ถึง Z หรือ "_", ".", "?" และต้องปิดท้ายด้วยเครื่องหมาย ":" เสมอ

OPERATION ส่วนนี้จะเป็นคำสั่งมาตรฐานของ 8051 เช่นคำสั่ง MOV, INC, DEC, ADD

OPERAND เป็นกลุ่มของตัวอักษรและตัวเลขซึ่งอาจจะอยู่ในรูปของการกระทำทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

คณิตศาสตร์หรือการกระทำทางด้าน Logic หรืออื่นๆ

COMMENT หมายถึงประโยคหรือข้อความที่มีอยู่ข้างหลังเครื่องหมาย ";" โดยทั่วๆ ไปมักจะใช้อธิบายการทำงานโปรแกรมนั้นแบบสั้นๆ เพื่อให้ทำความเข้าใจได้ง่าย เครื่องหมาย ";" จะอยู่ที่ใดก็ได้ข้อความที่ตามหลังเครื่องหมาย ";" จะไม่มีผลต่อการแปลของ Assembler

Assembler Directives

DB expression [, expression….]

DB (Define Byte) คำสั่งชี้ใช้กำหนดข้อมูลเป็นไบต์ลงในหน่วยความจำโปรแกรม สามารถใส่ข้อความรหัสแอสกี้ (ASCII String) ลงตำแหน่งในหน่วยความจำได้ โดยข้อความนั้นต้องอยู่ในเครื่องหมาย Quotes (" ") และยังกำหนดค่าข้อมูลที่เป็นตัวเลขและตัวอักษรในบรรทัดเดียวได้โดยคั่นด้วยเครื่องหมาย " , " เช่น

```
        POINTER:    DB "THIS IS A STRING"
Or      DATA:       DB 00H, 01011101B, "A", 79
Or      TABLE:      DB 00H, 01H, 02H, 03H
```

DS expression

DS (Define Space) ใช้สำหรับที่ว่างเป็นจำนวนไบต์เพื่อเก็บตำแหน่งค่าตัวแปร หรือเพื่อเป็น BUFFER หรือส่วนของ Working ต่างๆ

DW expression [, expression…]

DW (Define Word) ใช้สำหรับเก็บข้อมูลที่เป็นแบบ 16 บิต ของหน่วยความจำโปรแกรม โดยการกำหนดค่าของข้อมูลไบต์ต่ำก่อนแล้วตามด้วยข้อมูลไบต์สูง โดยข้อมูลแต่ละ Word ต้องคั่นด้วยเครื่องหมาย "," เช่นเดียวกับคำสั่ง DB

ORG expression

ORG (Origin) ใช้สำหรับกำหนดค่าเริ่มต้นของโปรแกรม (Origin Program Location) โดยกำหนดค่าแอดเดรสของหน่วยความจำ

END

END คำสั่งนี้มีไว้เพื่อบอกให้ Assembler ทราบถึงจุดสิ้นสุดของโปรแกรมว่าอยู่ที่ตำแหน่งที่เท่าใด ดังนั้นคำสั่งนี้จำเป็นต้องใส่เอาไว้ในส่วนท้ายของโปรแกรมด้วย เพราะถ้าไม่ใส่อาจจะทำให้ผิดพลาดในขั้นตอนการแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol_Name EQU expression

Symbol_Name BIT expression

สำหรับการกำหนดค่าให้กับชื่อ Symbol_Name (หรือที่เรียกกันว่า LABEL) โดยสามารถใช้คำสั่ง EQU หรือคำสั่ง BIT ก็ได้ ซึ่งจะทำงานเหมือนกันทุกประการ

Symbol_Name SET expression

คำสั่ง SET จะทำงานเหมือนกับ EQU ทุกประการ แต่จะสามารถกำหนดให้ Symbol_Name มีค่าหลายจุดตามต้องการ

Expressions

กำหนดค่าใน Expressions ของแต่ละคำสั่งในภาษา Assembly นั้น สามารถใส่ตัว Operator ต่างๆ เพื่อความสะดวกในการใช้งานได้ รวมทั้งยังกำหนดเป็นเลขฐานได้หลายแบบด้วย โดยสรุปได้ ดังนี้

1. โดยปกติตัวเลขถือว่าเป็นเลขฐานสิบเสมอ (Decimal) แต่ถ้าต้องการกำหนดเป็นเลขฐานสิบหก (Hexadecimal) จะต้องเขียนนำหน้าด้วยตัวเลขที่ลงท้ายด้วยตัวอักษร H เสมอ เช่น 12H, 0F7H และถ้าต้องการกำหนดเป็นเลขฐานแปด (Octal) ให้ลงท้ายด้วยตัวอักษร 0 หรือ Q และถ้าเป็นเลขฐานสอง (Binary) ให้ลงท้ายด้วยอักษร B
2. สัญลักษณ์ $ จะหมายถึงการอ้างอิงตำแหน่งของ Address ในบรรทัดนั้นๆ ซึ่งจะเป็นประโยชน์ต่อการเขียนโปรแกรมอย่างมาก และช่วยลดการใช้ LABEL ไม่ให้มากเกินความจำเป็นด้วย เช่น

```
CJNE   A, #11, $+6     ; 3 ไบต์ ถ้า A ไม่เท่ากับ 11 ฐานสิบ ให้ JUMP ไปอีก 6 ไบต์
LJMP   MAIN            ; 3 ไบต์ นั่นคือคำสั่ง CJNE ถัดไป
CJNE                   ; 3 ไบต์
```

3. สามารถใช้คำว่า HIGH หรือ LOW เพื่อกำหนดค่าเฉพาะส่วนสูงหรือต่ำได้ เช่น ถ้ามีการกำหนดค่าให้ชื่อ TEST EQU 1234H จะสามารถเรียกค่าส่วนสูงได้คือ

```
MOV R0, #HIGH TEST
```

ซึ่งจะทำให้ได้ค่า 12H ที่ R0 ตามที่ต้องการ

4. สามารถกำหนด Operator ในการคำนวณได้คือ +, -, *, /
5. สามารถกระทำทางด้าน Logic ได้คือ MOV, SHR, SHL, AND, OR, XOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT Operations

SXA51 มีความหมายในการอ้างค่าตัวแปรระดับ BIT ได้ โดยใช้หลักการ N.M ซึ่ง N คือค่า Byte Address และ M คือ หมายเลข BIT ดังนั้นจะต้องใช้กับค่าที่กำหนดเป็น Bit Address เท่านั้น เช่น 22H.3 จะมีค่าเท่ากับ 13H รวมทั้งจะใช้มาตรฐานได้ด้วยเช่น ACC.7, P1.2

ข้อควรระวังของการใช้งาน SXA51

ในกรณีที่ผู้ใช้ใช้คำสั่ง MOV BUFFER, #35H โดยชื่อ BUFFER คือส่วนหนึ่งของโปรแกรมที่ได้ทำงานกำหนดเอาไว้ด้วยคำสั่งเทียม DS หรือ EQU ในกรณีการกำหนดชื่ออยู่ก่อนการใช้อ้างอิงก็จะไม่มีปัญหาอะไร แต่ถ้าอยู่หลังจากการอ้างอิงจะเกิดปัญหาการแปลงไม่ถูกต้องและที่สำคัญคือ ไม่มีการแจ้ง Error ให้ผู้ใช้ทราบเมื่อ Assembler ดังนั้นขอให้จำหลักการง่ายๆ ดังนี้คือ การกำหนดค่าให้กับ LABEL ไม่ว่าจะจากคำสั่งเทียม DS หรือ EQU ก็ตาม ให้กระทำที่ต้นโปรแกรมเสมอ

รูปแบบไฟล์แบบ Intel Hex File

Intel Hex File เป็น FILE มาตรฐานหนึ่งที่นิยมใช้กันมาก ซึ่งสังเกตได้จากโปรแกรม Compiler หรือ Assembler ต่างๆ มักจะกำหนดให้เลือก Output ที่ต้องการเป็นแบบ Intel Hex File ได้ ทั้งนี้เนื่องจาก Intel Hex File มีรูปแบบที่เหมาะสมหลายประการ กล่าวคือ มีระบบ Checksum ซึ่งสามารถตรวจสอบความถูกต้องของข้อมูลได้ในแต่ละบรรทัดรวมทั้งมีการกำหนด Address ของข้อมูลได้ และที่สำคัญคือเป็นไฟล์แบบ ASCII คือ สามารถใช้ในการส่งออกทางพอร์ตสื่อสารต่างๆ ได้ และยังสามารถใช้กับโปรแกรม Editor ทั่วๆ ไป เพื่อการแก้ไขได้ (FILE ที่จะส่งออกทางพอร์ตสื่อสารจะต้องเป็นไฟล์แบบ ASCII เท่านั้น ทั้งนี้เนื่องจากการสื่อสาร จำเป็นต้องมีรหัสเพื่อการสื่อสารเองอยู่แล้ว จึงไม่สามารถใช้ไฟล์แบบ Object ได้เพราะข้อมูลจะไปซ้ำกับรหัสของการสื่อสารได้) รายละเอียดของ Intel Hex File ในแต่ละบรรทัดเป็นดังนี้

: BCAAAATTHH.....HHCC

: คือ Start Character (Colon) จำนวน HH

BC คือ จำนวน Byte ของข้อมูลในบรรทัดที่ค่าเป็นเลขฐาน 16 (Hex)

ถ้า BC = 0 จะเป็น End-of-file Record

AAAA คือ ADDRESS ของข้อมูลใน Byte แรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TT คือ ชนิดของข้อมูลในบรรทัดนั้นๆ

ถ้า TT = 0 เป็น Data Record

ถ้า TT = 1 เป็น End-of-file Record

HH คือ ข้อมูลแต่ละ Byte

CC คือ ค่า Checksum ของบรรทัดนั้นๆ โดยจะเป็นค่า Two's Complement ของผลบวกของข้อมูลทุกๆ Byte ในบรรทัดซึ่งรวมทั้ง BC, AAAA และ TT ด้วย

การรับข้อมูลจากเครื่องคอมพิวเตอร์

สำหรับการรับข้อมูลจากเครื่องคอมพิวเตอร์ PC โดยรับข้อมูลที่มีลักษณะมาตรฐานแบบ Intel Hex File เท่านั้น โปรแกรมในรูปแบบนี้สามารถโหลดลงบนบอร์ด ET-8032 หรือ Single Board อื่นๆ เพื่อทำการทดลอง วิธีการเรียกใช้ก็สามารถทำได้โดยการเรียกใช้ฟังก์ชันของโปรแกรม Monitor ดังนี้

1. กดปุ่มฟังก์ชัน 2 ครั้ง จะปรากฏข้อความบน 7- Segment ดังนี้ " Func2 "
2. กดปุ่ม DWLD (ปุ่ม B) 1 ครั้งจะปรากฏข้อความ "0000" บน 7- Segment
3. จากนั้นให้กด YES (ปุ่ม INC) บน 7-Segment จะว่างเปล่า แสดงว่าเครื่องพร้อมในการโหลดข้อมูล

การโหลดข้อมูลจากเครื่องคอมพิวเตอร์ PC นั้นที่เครื่อง PC จะต้องมีการใช้คำสั่งต่างๆ เพื่อการส่งไฟล์ออกมาทางพอร์ตอนุกรมด้วยเช่นกัน โดยในขั้นตอนต้นต้องทำการกำหนดคุณสมบัติของพอร์ตอนุกรมบนเครื่อง PC ก่อน ด้วยคำสั่ง MODE ดังนี้

C:\>MODE COMA: 96, N, 8, 1, P

โดย **A** คือหมายเลขพอร์ตอนุกรม 1 หรือ 2

96 คือความเร็วในการสื่อสารที่ 9600 ซึ่งเป็นค่า Default ของ Single Board ปกติธรรมดาสามารถเปลี่ยนแปลงได้แต่ต้องเท่ากับทั้ง Single Board และเครื่องคอมพิวเตอร์ PC

ส่วนอื่นๆ ที่เหลือคือการกำหนดจำนวนบิตและบิต Stop ตามแบบที่ Single Board ต้องการ

การกำหนดคุณสมบัติด้วยคำสั่ง MODE นี้จะคงอยู่ตลอดไปจนกว่าจะปิดเครื่อง PC เพราะฉะนั้นให้ทำเพียงครั้งเดียวเมื่อเริ่มต้น PC ก็พอ การส่งๆ ไฟล์จะทำได้ดังนี้

C:\>COPY FILE.HEX COMA แล้วกด ENTER

โดย **A** คือหมายเลขพอร์ตอนุกรม 1หรือ 2 ที่ใช้งาน

เมื่อทำคำสั่งเสร็จแล้ว เครื่อง PC จะส่ง FILE.HEX ออกมาทางพอร์ตอนุกรม โดยขณะที่ Single Board ก็จะรับข้อมูลทันที และแสดงข้อมูลที่รับเข้ามาทาง LED แสดงค่ารีจิสเตอร์ PSW ที่อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางขวาของ 7-Segment ตามลักษณะข้อมูลที่ได้รับ หากไฟล์ไม่เป็นไปตามรูปแบบที่กำหนด หรือมีความผิดพลาดจากการสื่อสาร เครื่องจะส่งเสียงเตือนให้ทราบ

ใน Intel Hex File จะมีการกำหนดค่า Address อยู่ในตัว ซึ่งจะให้ข้อมูลที่โหลดนั้นสามารถเข้าไปในตำแหน่ง Address ที่ถูกต้องได้ แต่บางครั้งผู้ใช้อาจจะต้องการโยกย้ายตำแหน่ง Address ให้เป็นไปตามที่ต้องการ ซึ่งสามารถทำได้ด้วยการเพิ่มคำสั่งพิเศษแทรกเข้าไปเพียงบรรทัดแรกของ Intel Hex File การแทรกคำสั่งเข้าไปนี้ ผู้ใช้สามารถทำได้ที่ โปรแกรม Editor ทั้งๆ ไป โดยเปิดไปที่มีสกุลเป็น *. HEX ขึ้นมาและเพิ่มคำสั่ง S หรือคำสั่ง O เข้าไปดังนี้

S2100 ◄—— (บรรทัดที่แทรกเข้าไป)

:100000007A……..

:100001010AB……..

:0000001FF……..

ตัวอย่างนี้ใน Intel Hex File จะกำหนด Address ไว้ที่ 0000H แต่ด้วยคำสั่งที่กำหนด Address ไว้ที่ 2100H จะทำให้การโหลดข้อมูลลงที่ ADDR 2100H ตามที่ต้องการส่วนคำสั่ง O ก็สามารถทำได้ในทำนองเดียวกัน เพียงแต่จะต้องใช้ Address ที่กำหนดบวกเข้ากับ Address ที่มีอยู่ใน Intel Hex File อีกที แล้วจึงจะโหลดข้อมูลลงในตำแหน่งของหน่วยความจำนั้นๆ

คำสั่ง	ความหมาย	Address ที่จะโหลดข้อมูล
S	Steam	Steam Address
O	Offset	Offset Address + File Address

การใช้คำสั่งที่สองนี้ต้องเป็นไปตามรูปแบบที่กำหนดเท่านั้น คือขึ้นด้วยตัวอักษรแรกตามคำสั่ง (ตัวเล็กหรือตัวใหญ่ก็ได้) ตามด้วยตัวเลข Address อักษร 4 หลัก (Hex) และต้องกำหนดไว้เพียงบรรทัดแรกเสมอในบางครั้ง Intel Hex File อาจจะมีช่วงข้อมูลอยู่หลายๆ ช่วง กล่าวคือมักกำหนด Address ที่ไม่ต่อเนื่องกันตลอด ซึ่งกรณีนี้ เมื่อใช้คำสั่งพิเศษ S และ O แทรกเข้าไปแล้ว ลักษณะของช่วงที่กำหนดไว้ จะยังคงเป็นไปตามที่กำหนดอยู่ใน Intel Hex File เหมือนเดิม เพียงแต่ย้ายไปตาม Address ที่กำหนดให้ใหม่เท่านั้น

ข้อควรระวัง ในการใช้คำสั่งพิเศษที่เพิ่มเข้าไปนี้ จะทำให้การทำงานโปรแกรมเกิดการผิดพลาด เพราะเมื่อโปรแกรมใน Intel Hex File มีจุดเริ่มต้นใหม่ที่ 8100H และมีการอ้างอิงหรือ JUMP แบบ ADDR16 (ใช้คำสั่ง LJMP, LCALL) เมื่อเพิ่มคำสั่ง S หรือ O เข้าไป ข้อมูลจะไม่อยู่ในตำแหน่งของหน่วยความจำที่ 8100H ฉะนั้นหาก RUN โปรแกรมที่จุดเริ่มต้นของโปรแกรม (ขึ้นอยู่กับการใช้คำสั่ง Sหรือ O) และมีการกระทำคำสั่ง LJMP หรือ LCALL จะทำให้กระโดดไปทำงานแอดเดรสที่ไม่ถูกต้อง (ถูกต้องในส่วนของโปรแกรม แต่ไม่ถูกต้องในส่วนของข้อมูลในหน่วยความจำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. ปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “OFF”
2. ปรับสวิตช์ SW 4, SW 2 และ SW 1 ทุกตัวมาที่ตำแหน่ง “OFF” ยกเว้น SW 1/2 “ON”
3. ปรับ Potentiometer (POT) ให้อยู่ที่ตำแหน่ง 0 โวลต์
4. ปรับสวิตช์ SW 4/1 มาที่ตำแหน่ง “ON”
5. ต่อสาย RS-232 จากเครื่องคอมพิวเตอร์ (เป็นแบบ DB-9 หรือ DB-25 ก็ได้) มายัง Single Board
6. จ่ายไฟเลี้ยงให้กับ Single Board และปรับสวิตช์ SW 7 (Power) มาที่ตำแหน่ง “ON”
7. ทำเครื่องคอมพิวเตอร์อยู่ที่ Dos Prompt และอยู่ใน Directory ที่มีไฟล์ดังนี้
   7.1 SXA51.EXE
   7.2 MODE.COM
   7.3 EDIT.COM
8. พิมพ์คำสั่งที่ Dos Prompt ดังนี้
   C:\TEST>EDIT LAB19_1.ASM แล้วกด ENTER
9. เมื่อเข้าสู่โปรแกรม EDIT แล้วให้พิมพ์โปรแกรมดังนี้

```
PORT_A  EQU   0E020H
PORT_B  EQU   PORT_A+1
PORT_C  EQU   PORT_A+2
C_PORT  EQU   PORT_A+3

        ORG   8000H
INIT:   MOV   DPTR,#C_PORT
        MOV   A,#08AH
        MOVX  @DPTR,A
START:  MOV   DPTR,#PORT_A
        MOV   R2,#08H
        MOV   A,#11H
LOOP1:  MOVX  @DPTR,A
        ACALL DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RL      A
        DJNZ    R2,LOOP1
        ACALL   CLEAR

        MOV     R2,#02H
LOOP2:  MOV     A,#0AAH
        MOVX    @DPTR,A
        ACALL   DELAY
        ACALL   CLEAR

        MOV     A,#55H
        MOVX    @DPTR,A
        ACALL   DELAY
        ACALL   CLEAR
        DJNZ    R2,LOOP2
        SJMP    START

CLEAR:  CLR     A
        MOVX    @DPTR,A
        ACALL   DELAY
        RET

DELAY:  MOV     R6,#00H     ;DELAY
DELAY1: MOV     R7,#00H
DELAY2: NOP
        NOP
        DJNZ    R7,DELAY2
        DJNZ    R6,DELAY1
        RET
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ทำการบันทึกโปรแกรมให้ชื่อว่า LAB19_1.ASM

11. พิมพ์คำสั่งที่ Dos Prompt ดังนี้

C:\TEST>SXA51 -L LAB19_1.ASM แล้วกด ENTER

จะปรากฏข้อความดังนี้

8051 Cross-Assembler (1.3) Copyright (c) 1987, 1989

Binary Technology, Inc. Meriden, NH

No errors detected

Object file size: 57 bytes

Program entry address: 0000 (Hex)

ถ้าหากปรากฏข้อความว่า Error ให้กลับไปดูที่ Source File ว่าพิมพ์ถูกหรือไม่ หรือดูที่ FILE.LST จะมีข้อความบ่งชี้ว่าบรรทัดใดผิด (C:\>EDIT LAB19_1.LST) ทำการแก้ไขให้ถูกต้องแล้วทำซ้ำในข้อ 4

8051 Cross-Assembler (1.3) Copyright (c) 1987, 1989

Binary Technology, Inc. Meriden, NH

3 errors

Object file size: 56 bytes

Program entry address: 0000 (Hex)

ดูที่ FILE.LST จะมีข้อความบ่งชี้ว่าบรรทัดใดผิด (LAB19_1.LST)

```
******* Assembly error *******
S,7 8119        21          MOV   @DPTR,A       ; บรรทัดที่ผิด
******* Assembly error *******
E,5 8119 1100   22          ACALL  DEALY        ; บรรทัดที่ผิด
  811B 3128     23          ACALL  CLEAR
******* Assembly error *******
E,5 8124 DA00   28          DJNZ   R2,LOOP3     ; บรรทัดที่ผิด
  8126 80DE     29          SJMP   START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. หากไม่มีข้อผิดพลาดให้กดปุ่ม FUNC 2 ครั้ง ปุ่ม DWLD 1 ครั้งและปุ่ม YES 1 ครั้งที่ Single Board เพื่อเข้าสู่โปรแกรมรับข้อมูลทางพอร์ตอนุกรม RS-232
13. พิมพ์คำสั่งที่ Dos Prompt ดังนี้

    C:\TEST>MODE COM1: 96, N, 8, 1, P แล้วกด ENTER

หาก COM1 ถ้าไม่ว่างใช้ COM2 ก็ได้ และพิมพ์คำสั่ง

C:\TEST>COPY LAB19_1.HEX COM1 แล้วกด ENTER

14. โปรแกรมจะถูกโหลดลงสู่หน่วยความจำหลักตำแหน่งที่กำหนดใน Intel Hex File
15. สั่งให้โปรแกรมทำงานและบันทึกผลการทดลอง
16. เปลี่ยนโปรแกรมเป็นโปรแกรมที่ 3.2 แล้วทำเหมือนเดิมตั้งแต่ข้อ 7 ถึงข้อ 15 โดยให้เปลี่ยนชื่อไฟล์ที่จะบันทึกเป็น LAB19_2.ASM
17. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

18. เลือกการทดลองตั้งแต่การทดลองที่ 3-18 มาหนึ่งการทดลองแล้วทำตามข้อ 6 ถึงข้อ 15
19. เปรียบเทียบลักษณะการทดลองแบบเดิมกับการทดลองแบบใหม่นี้ว่ามีข้อดีและข้อเสียอย่างไรบ้าง
20. จงเขียนผังงานของขั้นตอนการส่งข้อมูลมาที่ Single Board ตั้งแต่การเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 20

การพัฒนาโปรแกรมด้วย EPROM Emulator

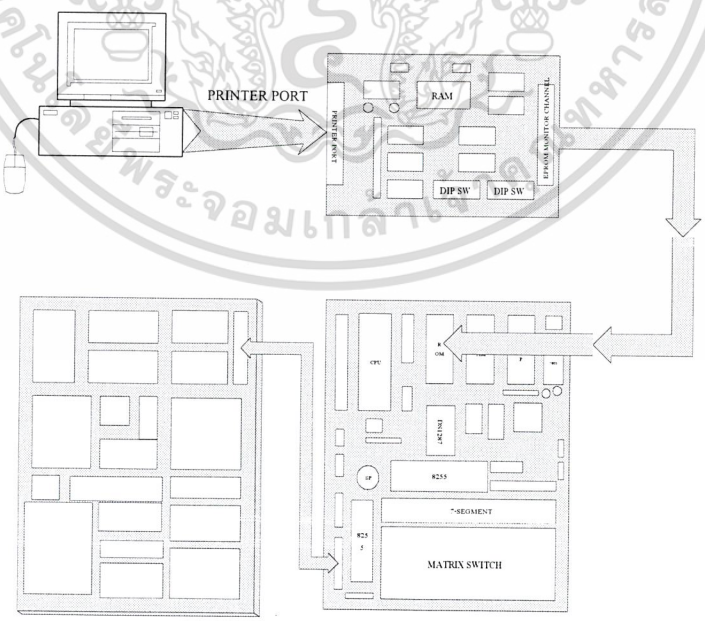
วัตถุประสงค์

1. เพื่อศึกษาการพัฒนาโปรแกรมควบคุมระบบต่างๆ
2. เพื่อสามารถพัฒนาโปรแกรมควบคุมระบบต่างๆ ได้
3. เพื่อศึกษาการใช้งานของ EPROM Emulator
4. เพื่อสามารถใช้ EPROM Emulator ในการพัฒนาโปรแกรมควบคุมระบบได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 20 1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0 1 ชุด
4. EPROM Emulator รุ่น ET-EM8/EM32 Plus 1 ชุด
5. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80) 1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 64 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

การพัฒนาโปรแกรมทางด้านไมโครคอนโทรลเลอร์ ถือว่าสำคัญมากเพราะระบบแต่ละระบบมีการทำงานที่ไม่เหมือนกัน หรือแม้กระทั้งการเขียนโปรแกรม Monitor ก็เช่นกัน หากเราจะเขียนโปรแกรมผ่านเครื่องคอมพิวเตอร์ PC และต้องการทดลองว่าโปรแกรมที่เขียนถูกต้องหรือสมบูรณ์เพียงใด การโปรแกรมลง EPROM ก็เป็นวิธีหนึ่งที่สามารถทราบได้ แต่หากโปรแกรมนั้นถูกต้องและสมบูรณ์ก็ดี (คงเป็นได้น้อยมาก) แต่หากต้องนำมาแก้ไขและทดลองใหม่คงไม่ดีแน่เพราะจะทำให้เสียเวลามาก

ดังนั้นจึงมีการพัฒนาเครื่องมือขึ้นมาชิ้นหนึ่งเรียกว่า EPROM Emulator ที่ให้ความสะดวกกับผู้ใช้ในการพัฒนาโปรแกรมต่างๆ ได้ โดยตัว EPROM Emulator จะทำหน้าที่เสมือนเป็นตัว EPROM ตัวหนึ่งซึ่งการโปรแกรมข้อมูลบนตัว EPROM นี้ ไม่จำเป็นต้องใช้ขบวนการโปรแกรมในลักษณะของ EPROM จริงๆ แต่ใช้การโหลดข้อมูลผ่านทางพอร์ตคอมพิวเตอร์แทน ซึ่งจะสามารถโหลดข้อมูลได้จากเครื่องคอมพิวเตอร์ PC ได้โดยตรงทำให้ลดขั้นตอนการพัฒนาโปรแกรมไปได้มาก ไม่ต้องมีการถอดอุปกรณ์ไปมา (มีแต่น้อย) มีความสะดวกและรวดเร็วในการโหลดข้อมูลสูงๆ และยังให้ความสะดวกในการแก้ไขหรือการเปลี่ยนแปลงข้อมูลด้วย ผลที่ได้รับจากการทำงานของ EPROM Emulator จะเหมือนกับการโปรแกรมลงบน EPROM จริงทุกประการ

ในปัจจุบัน EPROM Emulator มีอยู่มากมายหลายบริษัท หลากหลายคุณสมบัติ และการติดต่อผ่านเครื่องคอมพิวเตอร์ PC การเลือกใช้ขึ้นอยู่กับผู้ใช้และลักษณะของงาน แต่โดยรวมจะมีคุณสมบัติใกล้เคียงกัน ในการทดลองนี้ได้ใช้ EPROM Emulator ของ บริษัท ETT เป็น ET-EM8/32 PLUS ซึ่งคุณสมบัติและวิธีการใช้งานสามารถศึกษาได้จากคู่มือที่ให้มาพร้อมกับตัว EPROM Emulator

รูปแบบของไฟล์ที่จะทำการโหลดลงบน EPROM Emulator นี้จะเหมือนกับการ Down Load ข้อมูลลงบน Single Board ทุกประการ คือรูปแบบ Intel Hex File นั่นเอง การเขียนโปรแกรมต้องทำการกำหนด Origin ที่ 0000H เท่านั้น เพราะเมื่อเรานำ EPROM Emulator ไปเสียบแทน EPROM ที่อยู่บน Single Board โดยที่ CPU จะทำการ Fetch ข้อมูลที่ ADDR 0000H ก่อนทุกครั้งเมื่อจ่ายไฟให้แก่ Single Board หรือทำการ Reset ในการเขียนโปรแกรมที่ไม่มีอะไรพิเศษเป็นการเขียนที่ปกติในรูปแบบที่ Assembler รู้จัก และทำการแปลงเป็น Intel Hex File และส่งข้อมูลไปที่ EPROM Emulator การส่งข้อมูลสามรถทำได้โดยผ่านทางช่องสื่อสารแบบขนาน LPT1 และ LPT2 หรือขึ้นอยู่กับการออกแบบของ EPROM Emulator รุ่นที่ใช้นี้เป็นแบบ LPT1 หรือ LPT2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน EPROM Emulator

การใช้งาน EPROM Emulator สามารถศึกษารายละเอียดได้จากคู่มือ แต่จะขอกล่าวอย่างสั้นๆ ถึงการใช้งาน โดยการนำเอา EPROM Emulator ต่อกับพอร์ตขนาน LPT1 หรือ LPT2 ของเครื่องคอมพิวเตอร์ และนำเอาสายแพที่มี Socket ขนาด 28 Pin มาเสียบบนแทน EPROM ของ Single Board

EPROM Emulator ต้องจำลองขนาดหน่วยความจำเท่ากับที่ Single Board โดยการปรับเปลี่ยนขนาดของ EPROM Emulator จาก Dip-SW หรือ Jumper ที่ปรากฏอยู่บน Single Board ตามที่คู่มือกำหนดมา จากนั้นต่อสาย Reset จาก EPROM Emulator ไปยัง Reset ของ Single Board เพื่อให้หลังจากการโหลดข้อมูลเสร็จแล้ว EPROM Emulator จะทำการ Reset CPU ให้ทำงานทันที เพื่อความสะดวกในการทดลอง (ต้องเลือกลักษณะการทำงาน Reset ให้ถูกต้องกับความต้องการของ CPU)

ลำดับขั้นการทดลอง

1. ให้ Dos Prompt อยู่ที่ Directory ที่มีโปรแกรมดังนี้ก่อน
   1.1 SXA51.EXE
   1.2 MODE.COM
   1.3 EDIT.COM
   1.4 EM2.EXE ของ บริษัท ETT
2. ทำการสร้าง Intel Hex File ตามแบบการทดลองที่ 19 โดยกำหนดให้ Origin ที่ 0000H ตามโปรแกรมดังนี้
3. ทำการสร้าง Intel Hex File ตามแบบการทดลองที่ 19 โดยกำหนดให้ Origin ที่ 0000H ตามโปรแกรมดังนี้

```
                PORT_A  EQU     0E020H
                PORT_B  EQU     PORT_A+1
                PORT_C  EQU     PORT_A+2
                C_PORT  EQU     PORT_A+3
                        ORG     0000H
0000 : 7E00     INIT:   MOV     R6,#00H
0002 : 112F             ACALL   DELAY1          ; หน่วงเวลาสำหรับ 8255
0004 : 90E023           MOV     DPTR,#C_PORT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0007 : 748A              MOV     A,#08AH
0009 : F0                MOVX    @DPTR,A          ; กำหนด Command Word ให้ 8255
000A : 7A00    START:    MOV     R2,#00H
000C : 7BFF              MOV     R3,#0FFH
000E : 7C00              MOV     R4,#00H
0010 : 900100  LOOP1:    MOV     DPTR,#TABLE
0013 : 8A82              MOV     DPL,R2
0015 : E4                CLR     A
0016 : 93                MOVC    A,@A+DPTR
0017 : 0A                INC     R2
0018 : 0B                INC     R3
0019 : 90E020            MOV     DPTR,#PORT_A
001C : 2B                ADD     A,R3
001D : F0                MOVX    @DPTR,A
001E : 112D              ACALL   DELAY
0020 : BB07ED            CJNE    R3,#07H,LOOP1
0023 : 0C                INC     R4
0024 : 7BFF              MOV     R3,#0FFH
0026 : AA04              MOV     R2,04
0028 : BC2BE5            CJNE    R4,#2BH,LOOP1
002B : 80DD              SJMP    START

002D : 7E19    DELAY:    MOV     R6,#25           ; โปรแกรมหน่วงเวลา
002F : 7FC8    DELAY1:   MOV     R7,#200
0031 : DFFE    DELAY2:   DJNZ    R7,DELAY2
0033 : DEFA              DJNZ    R6,DELAY1
0035 : 22                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ORG   0100H
0100 : 00000000TABLE:DB    00H,00H,00H,00H,00H,00H,00H          ; ไม่มีการแสดงผล
0104 : 000000
0107 : F8102040      DB    0F8H,10H,20H,40H,0F8H,00H            ; N
010B : F800
010D : 70888870      DB    70H,88H,88H,70H,00H                  ; O
0111 : 00
0112 : F8282810      DB    0F8H,28H,28H,10H,00H                 ; P
0116 : 00
0117 : F8282810      DB    0F8H,28H,28H,10H,00H                 ; P
011B : 00
011C : 70888870      DB    70H,88H,88H,70H,00H                  ; O
0120 : 00
0121 : F828E810      DB    0F8H,28H,0E8H,10H,00H                ; R
0125 : 00
0126 : F8102040      DB    0F8H,10H,20H,40H,0F8H,00H            ; N
012A : F800
012C : 00000000      DB    00H,00H,00H,00H,00H,00H              ; ไม่มีการแสดงผล
0130 : 0000
                    END
```

4. ปรับเปลี่ยนสวิตช์ที่ชุดทดลองให้วงจร DOT Matrix Display ทำงาน
5. ทำการส่งโหลดข้อมูล EPROM Emulator โดยพิมพ์คำสั่งดังนี้

   C:\TEST>EM2 FILE.HEX H แล้วกด ENTER

จะปรากฏข้อความดังนี้

```
**********************************
* ET-EM              Version 1.1  *
* Copyright (C) 1993  ETT CO., LTD.*
*                                 *
* Printer Adapter 1               *
**********************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET-EM HEX FILE DOWLOAD = PROG20_1.HEX

OFFSET ADDRESS = 0000H

END OF INTEL FILE

0039H BYTE

6. เมื่อทำการโหลดข้อมูลลงบน EPROM Emulator แล้ว EPROM Emulator จะทำการ Reset CPU ทันทีสังเกตผลการทดลองและบันทึกผล
7. สรุปและวิจารณ์ผลการทดลอง

แบบฝึกหัด

8. จงเปรียบเทียบการพัฒนาโปรแกรมด้วย EPROM Emulator กับการพัฒนาโปรแกรมแบบโปรแกรมลงบนตัว EPROM จริงๆ ว่ามีข้อดีและข้อเสียอย่างไร
9. จงเขียนโปรแกรมควบคุมระบบโดยให้รับข้อมูลจาก Switch Input ตำแหน่งที่ 0 ถึง 3 มาควบคุมรอบโปรแกรมไฟวิ่งของ LED บน Port Monitor (แบบใดก็ได้) โดยเมื่อครบรอบแล้วให้ LED ติดหมดเป็นเวลา 3 วินาทีแล้วเริ่มต้นใหม่ Switch Input ตำแหน่งที่ 4 ถึง 7 ใช้ในการควบคุมความเร็วในการวิ่งของ LED
10. จงเขียนโปรแกรมควบคุมความเร็วของ Stepping Motor โดยเริ่มให้ Stepping Motor หมุนตามเข็มนาฬิกาจากหมุนช้าๆ จนเร็วสุด จากนั้นให้หยุดหมุน และให้ Stepping Motor หมุนทวนเข็มนาฬิกาจากหมุนช้าๆ จนเร็วสุด จากนั้นให้หยุดหมุน แล้วเริ่มต้นใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 21

การโปรแกรม EPROM

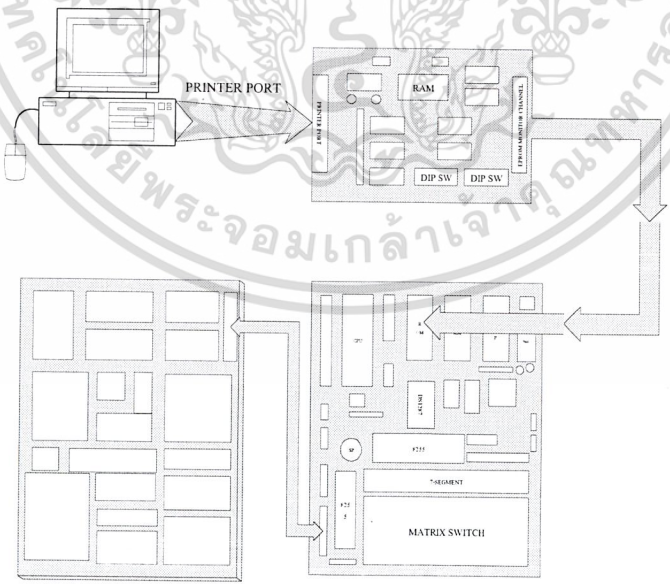
วัตถุประสงค์

1. เพื่อศึกษาลำดับขั้นตอนการเขียนโปรแกรมจนถึงการโปรแกรม EPROM
2. เพื่อสามารถนำโปรแกรมที่เขียนขึ้นไปโปรแกรมลงบนตัว EPROM ได้

เครื่องมือและอุปกรณ์ที่ใช้ในการทดลอง

1. ใบงานการทดลองที่ 1	1 ชุด
2. ชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051	1 ชุด
3. ซิงเกิ้ลบอร์ดของ บริษัท ETT รุ่น ET-8032 V2.0	1 ชุด
4. EPROM EMULATOR รุ่น ET-EM8/EM32 Plus	1 ชุด
5. เครื่องโปรแกรม EPROM	1 เครื่อง
6. กล่องแปลงพอร์ตสำหรับซิงเกิ้ลบอร์ด ที่ไม่ใช้ของบริษัท ETT (72IOZ80)	1 ชุด

ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0



รูปที่ ข. 65 ลักษณะการเชื่อมต่อชุดทดลองและเครื่องมือกับ ET-8032 V2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีเบื้องต้น

เมื่อเราต้องการเขียนโปรแกรมหรือพัฒนาโปรแกรมควบคุมระบบต่างๆ หรือแม้กระทั่งการเขียนโปรแกรม Monitor และให้ทำการพัฒนาจนเสร็จสมบูรณ์ สามารถเขียนและทดลองโดยผ่าน EPROM Emulator ได้ แต่ถ้าเราจะนำไปใช้งานจริงคงจะไม่ดีแน่ เพราะเราจะทำให้ชุดควบคุมใหญ่มากและหากเกิดไฟดับหรือแหล่งจ่ายไฟเลี้ยงเกิดขัดข้อง โดยที่ตัวโปรแกรมยังอยู่ที่ตัวของ EPROM Emulator จะทำให้สูญหายได้หากไม่มีแบตเตอรี่สำรอง ดังนั้นในทางปฏิบัติคือ นำโปรแกรมที่สมบูรณ์ในรูปแบบ Intel Hex File มาโปรแกรมลงบนตัว EPROM Emulator เพราะด้วยคุณสมบัติของ ROM ทำให้ไม่ต้องการไฟเลี้ยงในการเก็บรักษาข้อมูล

การโปรแกรมลงบนตัว EPROM นั้นสามารถทำได้เองโดยใช้เครื่องโปรแกรม EPROM ที่มีขายตามบริษัทที่ทำงานเกี่ยวกับไมโครโพรเซสเซอร์ หรือตามสถานศึกษาต่างๆ โดยอ่านวิธีการใช้จากคู่มือที่ให้มาพร้อมกับการซื้อเครื่องโปรแกรมหรือนำ FILE.HEX ไปให้ทางร้านหรือทางบริษัททำการโปรแกรมให้ก็ได้ ในอัตราค่าใช้จ่ายประมาน 30 บาท (เฉพาะค่าโปรแกรม) โดยตัว EPROM ต้องมีขนาดเท่ากับค่าที่ตัวบน EPROM Emulator ด้วย

เมื่อได้ EPROM ที่บรรจุโปรแกรมที่เราต้องการแล้ว ให้นำมาเสียบแทนที่ EPROM Emulator ได้เลยและทำการทดสอบโปรแกรม Reset และทดสอบตามลักษณะของการเขียนโปรแกรม

ลำดับขั้นการทดลอง

1. เลือกการทดลองหรือโปรแกรมใดก็ได้มา 1 โปรแกรม หรืออาจจะเขียนโปรแกรมนอกเหนือจากนี้ก็ได้
2. ทดลองและทดสอบโปรแกรมให้สมบูรณ์และถูกต้อง ด้วย EPROM Emulator ตามการทดลองที่ 20
3. นำ FILE.HEX ไปโปรแกรมลงบน EPROM ที่มีขนาดเท่ากับขนาดของ EPROM Emulator ตามที่ตั้งเอาไว้
4. ทำ EPROM ที่ได้มาเสียบแทน EPROM Emulator
5. ทดสอบโปรแกรมและระบบการควบคุมตามลักษณะของโปรแกรมที่เขียนเอาไว้
6. สรุปและวิจารณ์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฝึกหัด

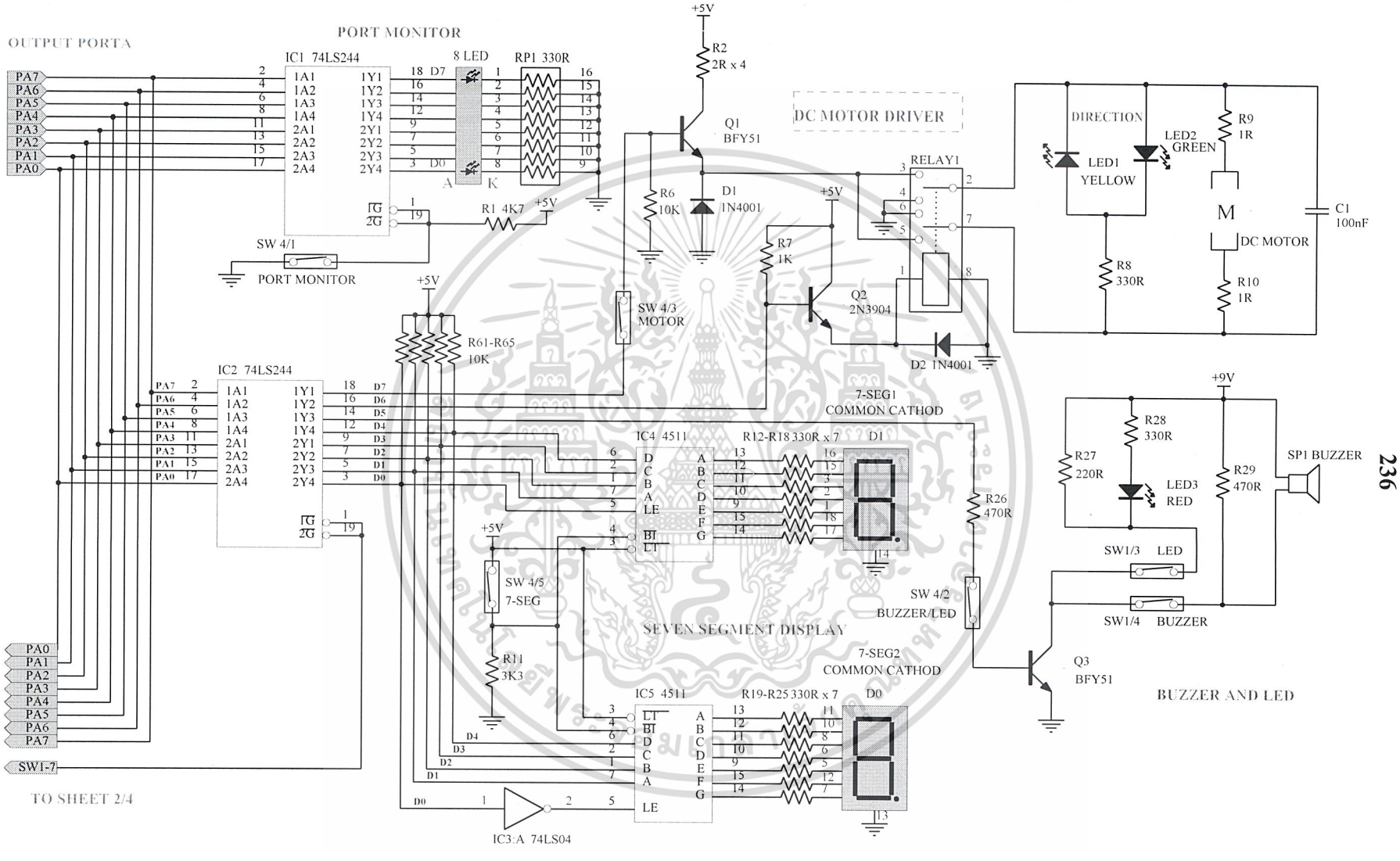
7. ทดลองเปลี่ยนโปรแกรมหรือพัฒนาโปรแกรมให้มีความซับซ้อนมากขึ้น
8. จงเขียนผังงานของลำดับการโปรแกรม EPROM ตั้งแต่การเขียนโปรแกรมจนถึงผลของกระบวนการจนได้ EPROM ที่บรรจุข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

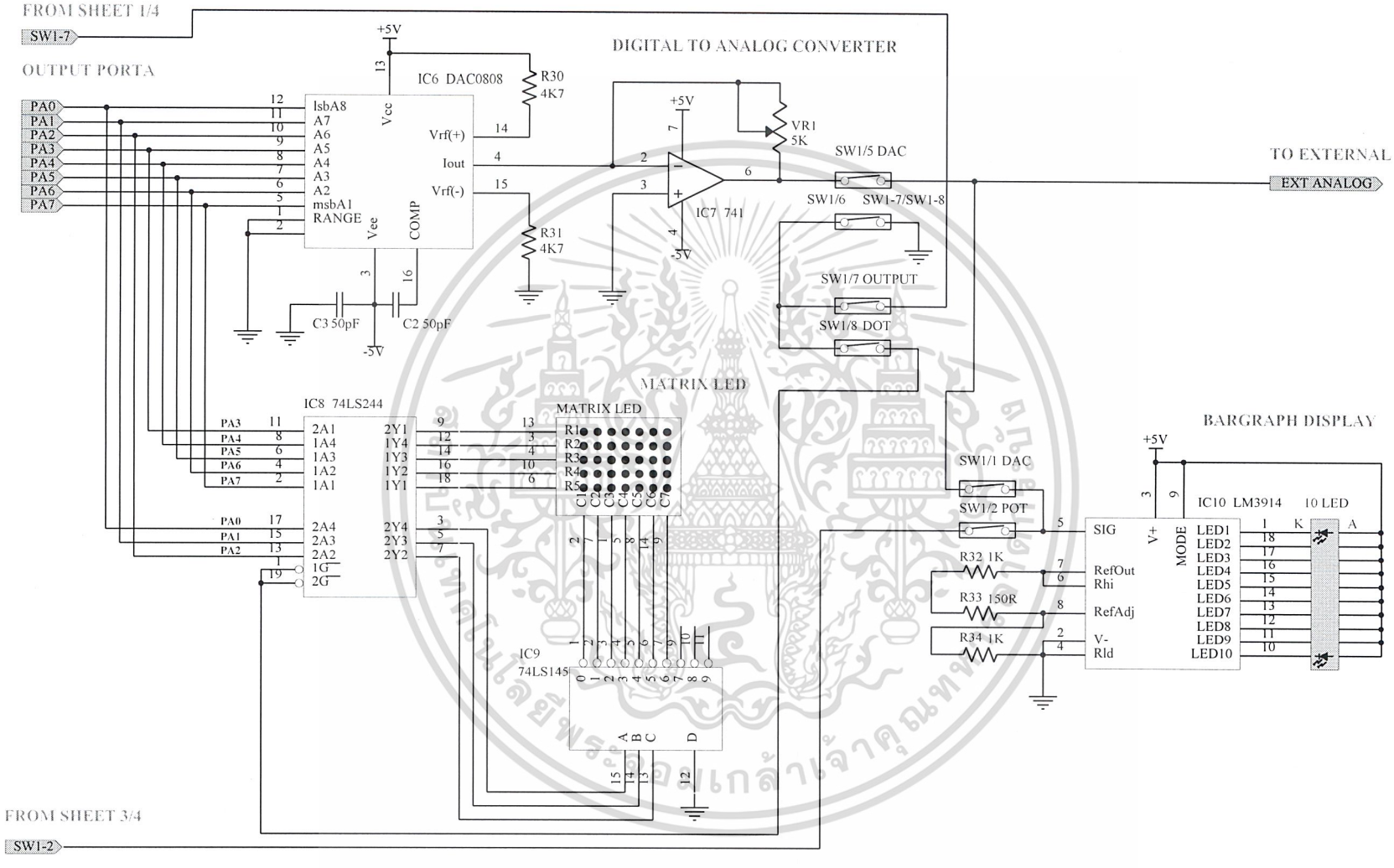
ภาคผนวก ค

วงจรและลายวงจรพิมพ์

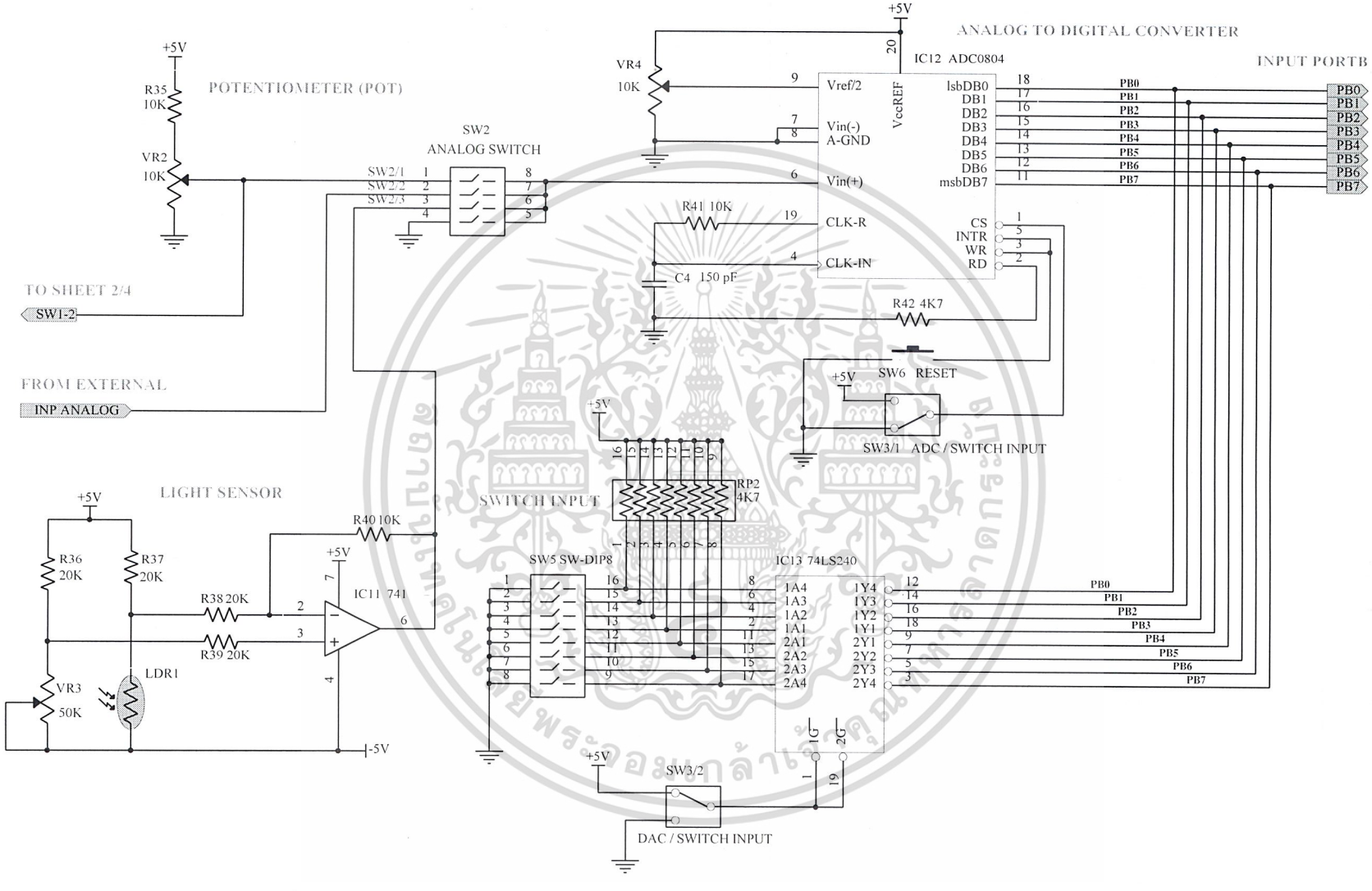
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



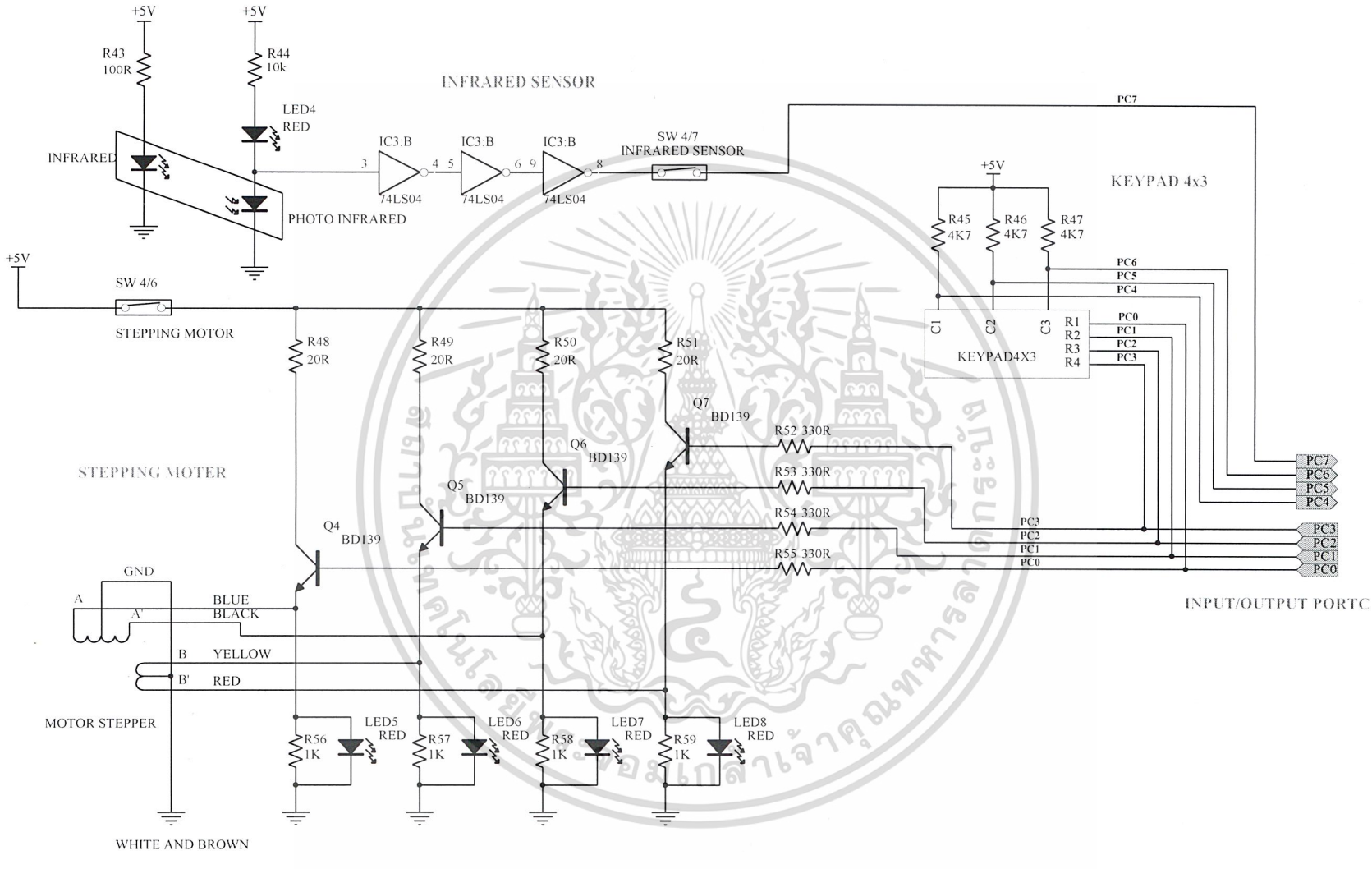
รูปที่ ค.1 ลายวงจรชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ส่วนที่ 1



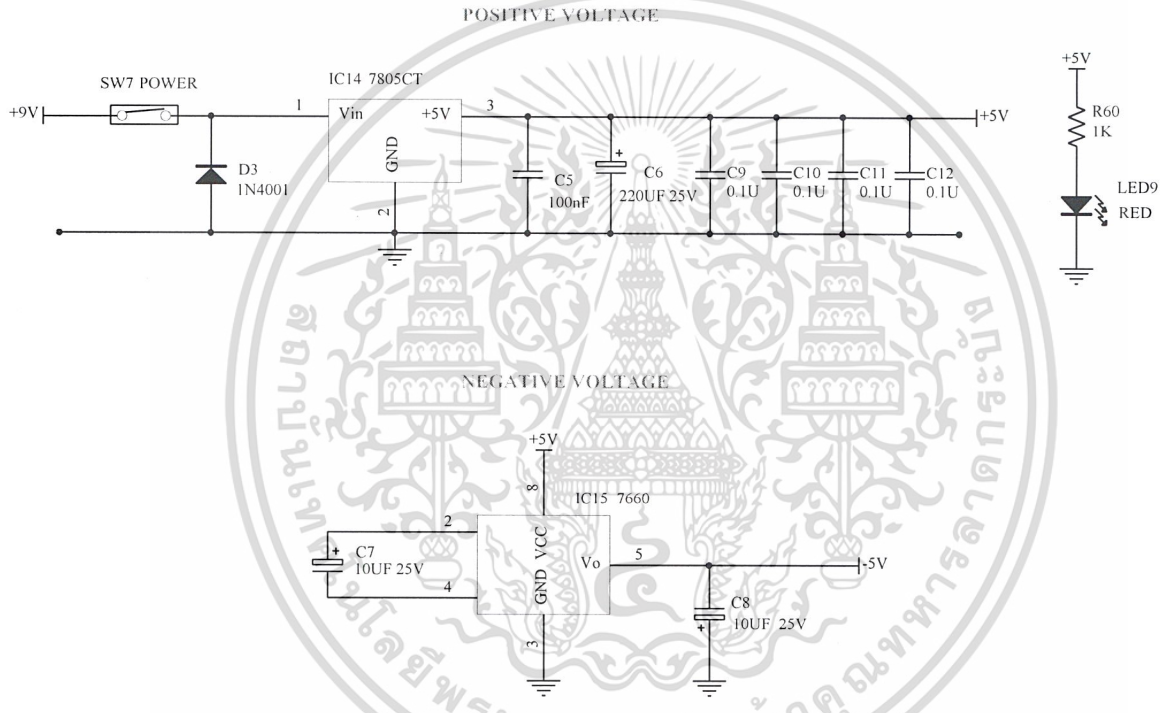
รูปที่ ค.2 ลายวงจรชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ส่วนที่ 2



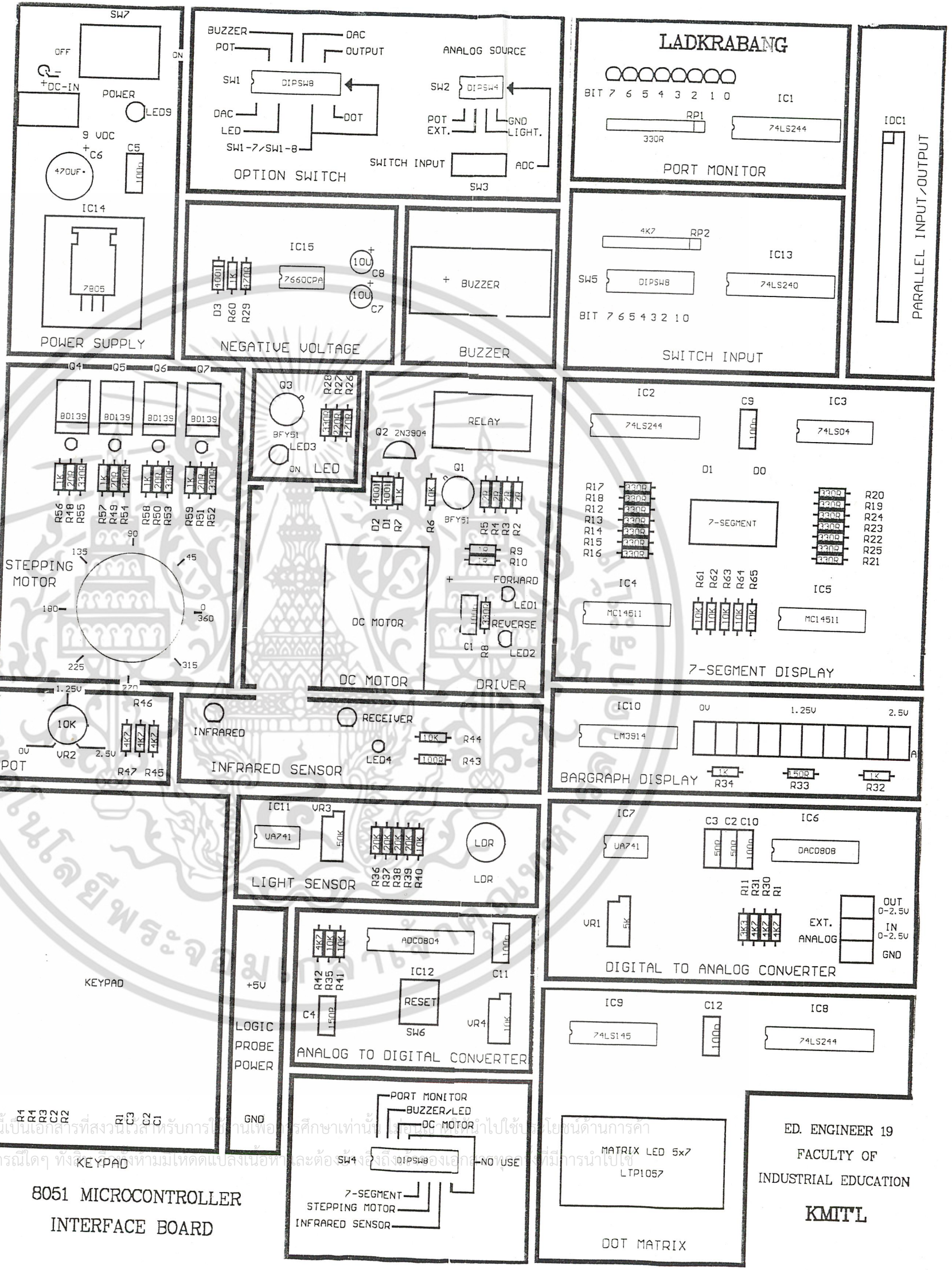
รูปที่ ค.3 ลายวงจรชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ส่วนที่ 3



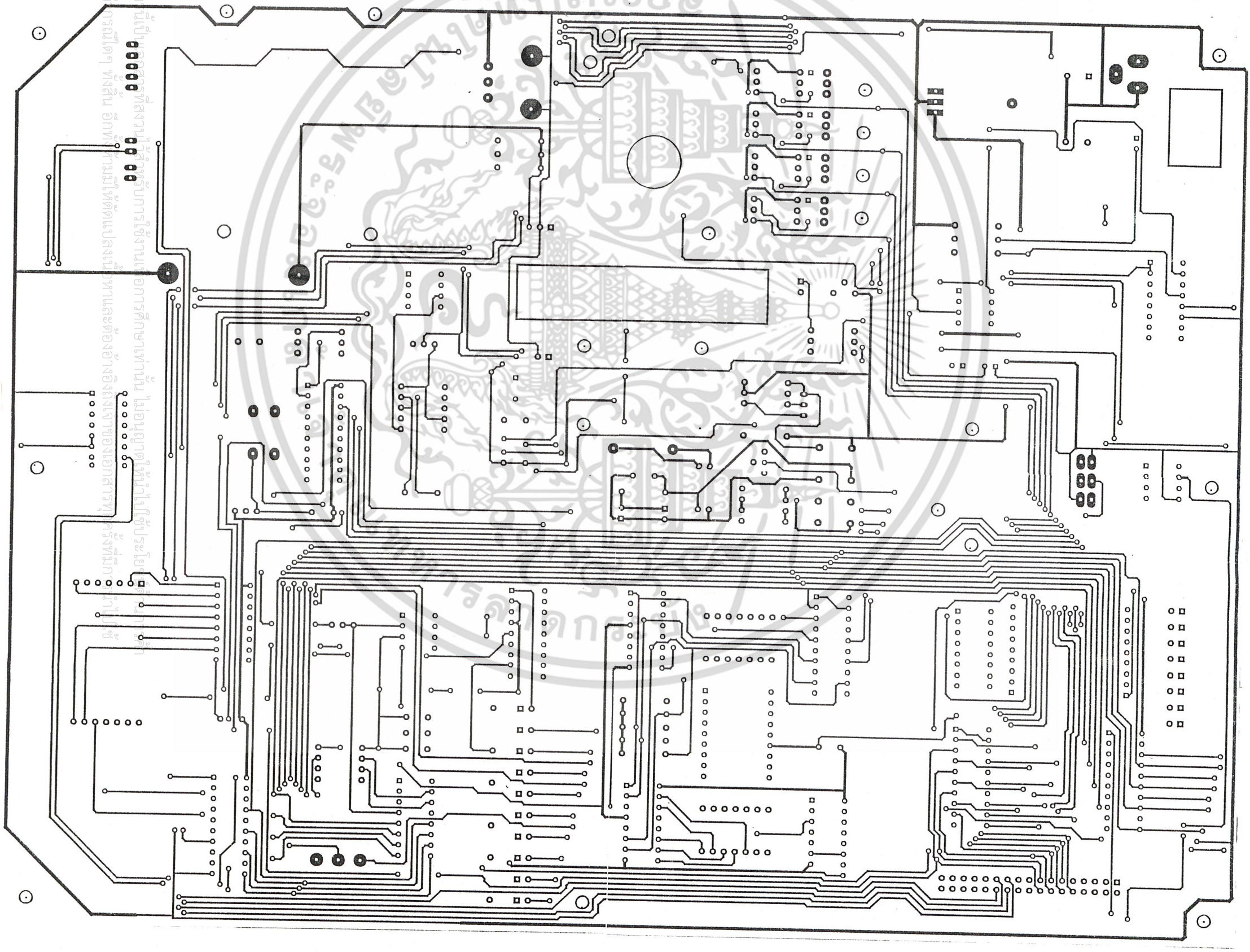
รูปที่ ค.4 ลายวงจรชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ส่วนที่ 4

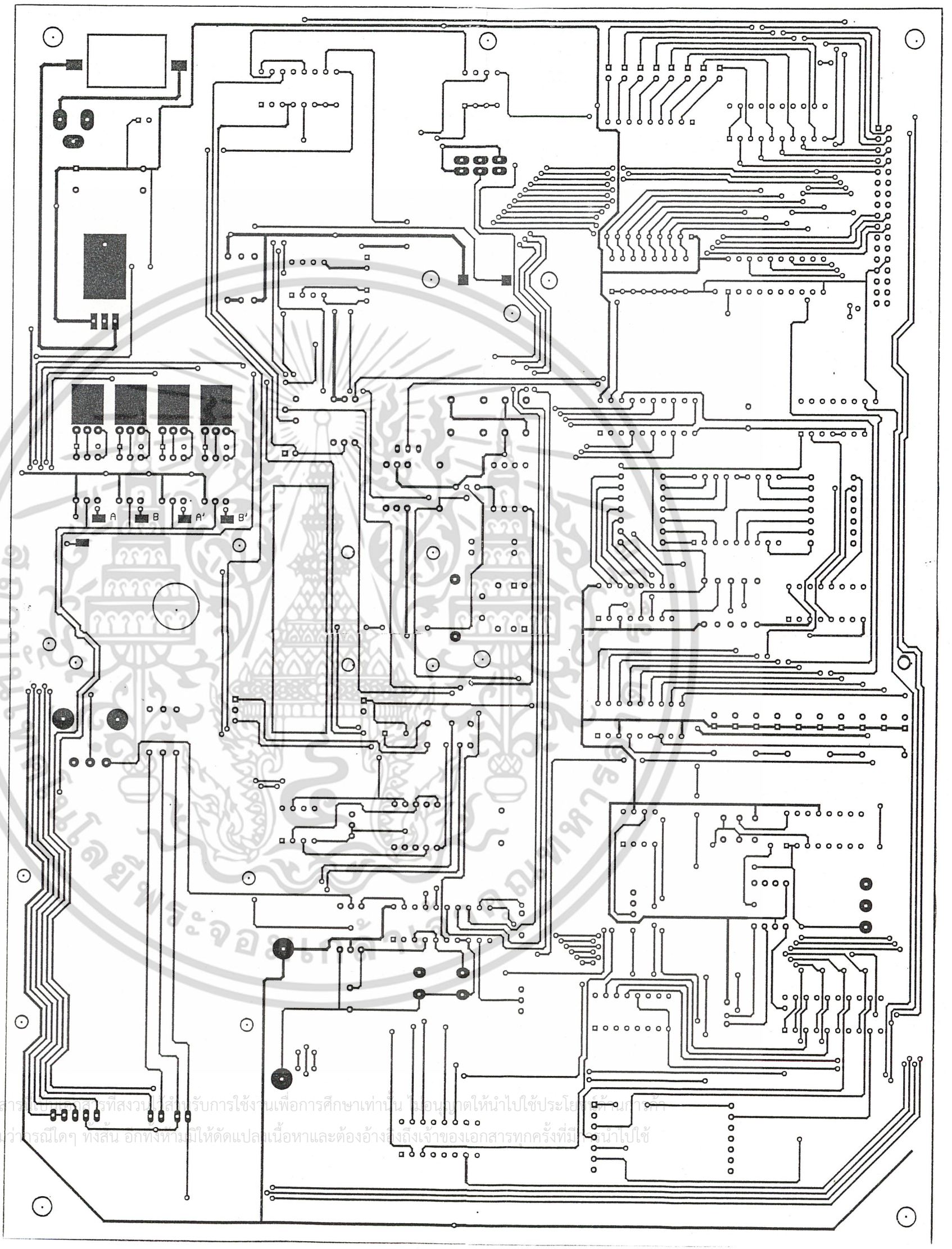


รูปที่ ค.5 ลายวงจรชุดทดลองการเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 ส่วนที่ 5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการวัสดุและอุปกรณ์ที่ใช้ในการทำชุดทดลอง

ไอซี (IC)

ลำดับ	รายการ	จำนวน
1.	SN74LS244N	3 ตัว
2.	HD74LS04P	1 ตัว
3.	MC14511BCP	2 ตัว
4.	DAC0808LCN	1 ตัว
5.	UA741CN	2 ตัว
6.	SN74LS145N	1 ตัว
7.	LM3914N	1 ตัว
8.	ADC0804LCN	1 ตัว
9.	SN74LS240N	1 ตัว
10.	7805	1 ตัว
11.	7660S CPA	1 ตัว

ตัวเก็บประจุ (Capacitor)

ลำดับ	รายการ	จำนวน
1.	C_2 - 50 pF	2 ตัว
2.	C_1 - 100 nF	6 ตัว
3.	C_4 - 150 pF	1 ตัว
4.	C_7 - 10 uF 16 V	2 ตัว
5.	C_6 - 470 uF 16 V	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดโอด (Diode)

ลำดับ	รายการ	จำนวน
1.	D_1-D_3 เบอร์ 1N4001	3 ตัว

ไดโอดเปล่งแสง (LED)

ลำดับ	รายการ	จำนวน
1.	LED, กลมขนาด 3 mm (แดง)	7 ดวง
2.	LED, กลมขนาด 3 mm (เหลือง)	1 ดวง
3.	LED, กลมขนาด 3 mm (เขียว)	2 ดวง
4.	LED, กลมขนาด 5 mm (แดง)	1 ดวง
5.	LED, กลมขนาด 2 mm (แดง)	5 ดวง
6.	LED, เหลี่ยมขนาด 3×6 mm (แดง)	2 ดวง
7.	LED, เหลี่ยมขนาด 3×6 mm (เหลือง)	2 ดวง
8.	LED, เหลี่ยมขนาด 3×6 mm (เขียว)	6 ดวง
9.	7-Segment Common Cathode 2 หลักคู่ เบอร์ LTP1507	1 ตัว
10.	Dot Matrix โดย เบอร์ LTP1507 Anode –Row, Cathode-Column	1 ตัว

ตัวต้านทาน (Resistor)

ลำดับ	รายการ	จำนวน
1.	1 โอห์ม	2 ตัว
2.	2 โอห์ม	4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวต้านทาน (Resistor)

ลำดับ	รายการ	จำนวน
3.	20 โอห์ม	4 ตัว
4.	100 โอห์ม	1 ตัว
5.	150 โอห์ม	1 ตัว
6.	220 โอห์ม	1 ตัว
7.	330 โอห์ม	27 ตัว
8.	470 โอห์ม	2 ตัว
9.	1 กิโลโอห์ม	8 ตัว
10.	3.3 กิโลโอห์ม	1 ตัว
11.	4.7 กิโลโอห์ม	5 ตัว
12.	10 กิโลโอห์ม	6 ตัว
13.	20 กิโลโอห์ม	4 ตัว
14.	ตัวต้านทานแบบแถว 9 ขา 330 กิโลโอห์ม	1 ตัว
15.	ตัวต้านทานแบบแถว 9 ขา 4.7 กิโลโอห์ม	1 ตัว
16.	ทริมพ็อต 5 กิโลโอห์ม	1 ตัว
17.	ทริมพ็อต 10 กิโลโอห์ม	1 ตัว
18.	ทริมพ็อต 50 กิโลโอห์ม	1 ตัว
19.	ตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม	1 ตัว

สวิตช์ (Switch)

ลำดับ	รายการ	จำนวน
1.	SW_1 DIP Switch 8 ตำแหน่ง	3 ตัว
2.	SW_2 DIP Switch 4 ตำแหน่ง	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตช์ (Switch)

ลำดับ	รายการ	จำนวน
3.	SW_3 Switch เลือก 2 ทาง 6 ขา	1 ตัว
4.	SW_6 Switch กดติดปล่อยดับ	1 ตัว
5.	SW_7 Switch กดติดกดดับ	1 ตัว

ทรานซิสเตอร์ (Transistor)

ลำดับ	รายการ	จำนวน
1.	Q_1 เบอร์ BFY51	2 ตัว
2.	Q_2 เบอร์ 2N3904	1 ตัว
3.	$Q_4 - Q_7$ เบอร์ BD139	4 ตัว

รายการอื่นๆ

ลำดับ	รายการ	จำนวน
1.	ชุดอินฟราเรด รับ-ส่ง	1 ชุด
3.	รีเลย์ 2 ขั้ว 2 ทาง ขนาด 5 V	1 ตัว
4.	มอเตอร์กระแสตรง 5 V	1 ตัว
5.	Stepping motor 5V แบบ 4 เฟส 6 สาย	1 ตัว
6.	แผ่นระบายความร้อน	1 ตัว
7.	ขั้ว DC IN	1 ตัว
8.	จุดเชื่อมต่อขนาด 3 จุด	1 ตัว
9.	ตัวต้านทานปรับค่าตามแสง	1 ตัว
10.	Keypad (คีย์โทรศัพท์)	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอื่นๆ

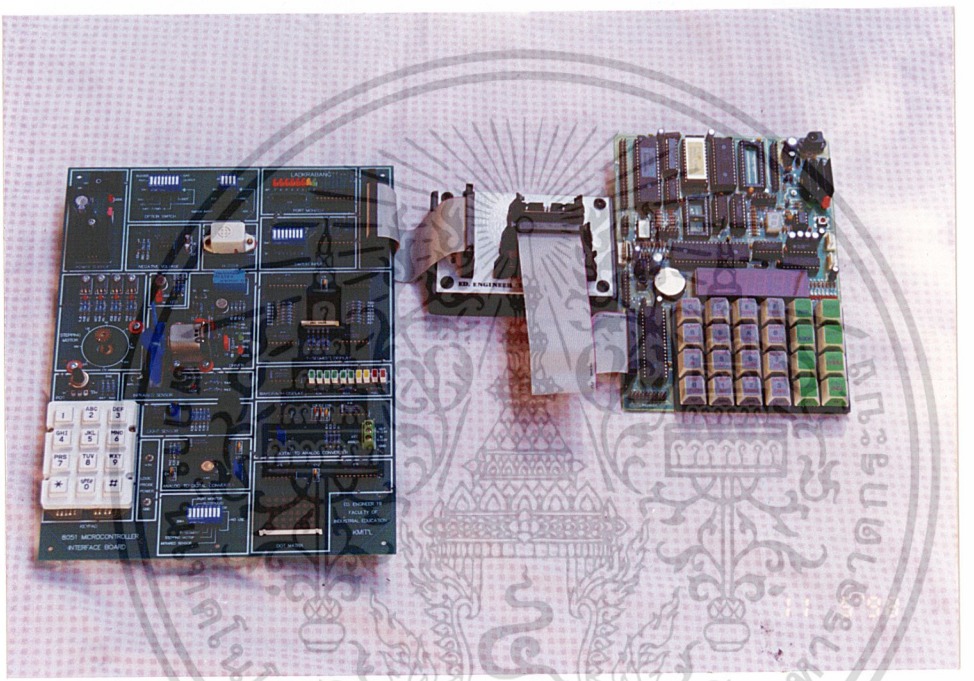
ลำดับ	รายการ	จำนวน
11.	บัซเซอร์ขนาด 9 V	1 ตัว
12.	คอนเน็กเตอร์ IDC ขนาด 34 pin	1 ตัว
13.	ตาไก่ขนาดเล็ก	2 ตัว
14.	แผ่นรองน๊อต	37 ชิ้น
15.	น๊อตขนาด 2 mm. 3 mm.	4 ชุด 20 ชุด
16.	ใบพัดสำหรับมอเตอร์ขนาดเล็ก	1 ใบ
17.	คอนเน็กเตอร์ IDE แถวเดี่ยว 180 องศา	9 ตัว
18.	สายไฟอ่อน 12 cm	-
19.	เสารองแผ่น PCB พร้อมน๊อต	18 ชุด
20.	หัวต่อสายแพ 34 pin	2 ตัว
21.	สายแพ 40 pin	3 ฟุต
22.	หม้อแปลง input 220 Vac 50 Hz output 9-12 Vdc 800 mA	1 ตัว
23.	ขั้วต่อคอนเน็กเตอร์ ตัวผู้ ขนาด 40 pin ขนาด 34 pin ขนาด 26 pin ตัวเมีย ขนาด 40 pin ขนาด 34 pin ขนาด 26 pin	 1 ตัว 2 ตัว 1 ตัว 2 ตัว 4 ตัว 2 ตัว
24.	กล่องเอนกประสงค์	1 กล่อง
25.	กล่องใส่ชุดทดลอง	1 กล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

รูปต้นแบบของชุดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.1 ต้นแบบของชุดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ปรเมษฐ์ ประนยานันทน์ และ ปิยพงศ์ เผ่าวณิช. **คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51.** กรุงเทพ: บริษัท เอช. เอ็น. กรุ๊ป จำกัด, 2536

ธานินทร์ ถาวรศาสนวงศ์. **การอินเทอร์เฟส.** กรุงเทพ: ห้างหุ้นส่วนจำกัด สำนักพิมพ์ฟิสิกเซ็นเตอร์, 2536

BARRY B. BREY. **THE INTEL MICROPROCESSORS** : PRENTICE HALL INTERNATIONAL, PRENTICE HALL ENGLEWOOD CLIFFS, 1995

VITRO P.NELSON H. TROY NAGLE BILL D. CARROLL and J.DAVID IRWIN. **DIGITAL CIRCUIT ANALYSIS and DESIGN :** ENGLEWOOD CLIFFS, N.J PRENTICE HALL, 1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายนพพร วัฒนสิทธิ์
วันเดือนปีเกิด	22 มิถุนายน 2520
สถานที่เกิด	จังหวัดจันทบุรี
ภูมิลำเนาเดิม	16 หมู่ 4 ตำบลสีพยา อำเภอท่าใหม่ จังหวัดจันทบุรี 22120
ที่อยู่ปัจจุบัน	384 หมู่1 ซอยจินดานิเวศน์ แขวงหัวตะเข้ เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	02-7391224, 142-4742054

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอำนวยวิทย์
มัธยมศึกษาตอนต้น	โรงเรียนเบญจมราชูทิศจังหวัดจันทบุรี
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคจันทบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคจันทบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	นักเรียนดีเด่นประจำปี 2539
ทุนการศึกษา	-
คติพจน์	ไม่มีใครทำอะไรให้ใครเป็นอะไรไปได้นอกจากตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นางสาวรจนา อัชชาศัย
วันเดือนปีเกิด	22 สิงหาคม 2519
สถานที่เกิด	จังหวัดระยอง
ภูมิลำเนาเดิม	จังหวัดระยอง
ที่อยู่ปัจจุบัน	53 หมู่ 6 ตำบลบ้านแลง อำเภอเมือง จังหวัดระยอง 21000
โทรศัพท์	01-6631404 , 152-688122

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านตะเกราทอง
มัธยมศึกษาตอนต้น	โรงเรียนวัดป่าประดู่
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคระยอง
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ตนเป็นที่พึ่งแห่งตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นางสาวศุภรัตน์ โพธิ์สิทธิ์
วันเดือนปีเกิด	6 กรกฎาคม 2519
สถานที่เกิด	จังหวัดนนทบุรี
ภูมิลำเนาเดิม	จังหวัดนนทบุรี
ที่อยู่ปัจจุบัน	7/10 หมู่2 ซอยนนทบุรี 7 ตำบลสวนใหญ่ อำเภอเมือง จังหวัดนนทบุรี 11000
โทรศัพท์	02-5273441

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดแจ้งศรีสัมพันธ์
มัธยมศึกษาตอนต้น	โรงเรียนศรีบุณยานนท์
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีวิทยาเขตนนทบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ทำวันนี้ให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นาย เอกรินทร์ บุญยเนตร
วันเดือนปีเกิด	21 ธันวาคม 2519
สถานที่เกิด	จังหวัดสระบุรี
ภูมิลำเนาเดิม	จังหวัดลพบุรี
ที่อยู่ปัจจุบัน	11 หมู่1 ตำบลช่องสาริกา อำเภอพัฒนานิคม
	จังหวัดลพบุรี 15220
โทรศัพท์	036-638045, 162-041983

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนพรหมรังษี
มัธยมศึกษาตอนต้น	โรงเรียนพุแควิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคสระบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคสระบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ตัวเองดีแค่ไหน ถึงจะว่าคนอื่นไม่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้