สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวัดอุณหภูมิในตู้อบเด็ก
(TEMPERATURE SENSOR IN INFANT INCUBATOR)

T 055481

โดย

นาย ณัฐพล มโนปิยอนันต์

นาย ณัฐวุฒิ แสงอินทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....
เลขทะเบียน 55481
วัน,เดือน,ปี 10 พ.ค. 2548

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

เครื่องวัดอุณหภูมิในตู้อบเด็ก
(TEMPERATURE SENSOR IN INFANT INCUBATOR)

โดย

นาย ณัฐพล มโนปิยอนันต์ 43010132

นาย ณัฐวุฒิ แสงอินทร์ 43010135

อาจารย์ที่ปรึกษา

ดร.ชูชาติ บิณฑวิรุจน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดอุณหภูมิในตู้อบเด็ก

TEMPERATURE SENSOR IN INFANT INCUBATOR

ผู้จัดทำ

นาย ณัฐพล มโนปิยอนันต์ 43010132

นาย ณัฐวุฒิ แสงอินทร์ 43010135

.....(อาจารย์ที่ปรึกษา)

(ดร.ชูชาติ ปิณฑวิรุฬห์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดอุณหภูมิในตู้อบเด็ก
(TEMPERATURE SENSOR IN INFANT INCUBATOR)

นาย ณัฐพล มโนปิยอนันต์
นาย ณัฐวุฒิ แสงอินทร์
ดร.ชูชาติ ปิณฑวิรุจน์(อาจารย์ที่ปรึกษา)
ปีการศึกษา 2546

บทคัดย่อ

โครงงานนี้มีที่มาจากการต้องการวัดการกระจายอุณหภูมิที่เกิดขึ้นในระบบปิดที่มีการถ่ายเทการไหลของอากาศร้อนอยู่ตลอดเวลา ซึ่งในที่นี้ระบบปิดที่ได้ให้ความสำคัญในการทดลองคือตู้อบเด็กทารกแรกเกิดที่ใช้ในทางการแพทย์ เนื่องจากตู้ที่ใช้ในการอบเด็กทารกนี้มีลักษณะเป็นตู้ปิดทรงเหลี่ยมที่มีการให้อากาศร้อนไหลเวียนเข้าไปภายในตู้และมีการกำหนดอุณหภูมิที่จะใช้ในการอบอย่างชัดเจน จากการไหลเวียนของอากาศร้อนนี้ทำให้เราอยากทราบว่าตู้อบเด็กทารกที่ใช้นี้สามารถให้อุณหภูมิภายในตู้กระจายได้ถูกต้องเพียงใด ณ ตำแหน่งต่างๆภายในตู้ที่มีการไหลเวียนของอากาศอยู่ตลอดเวลามีการกระจายอุณหภูมิอย่างไร โดยได้ทำการวัดค่าอุณหภูมิที่ตำแหน่งต่างๆจากตัว Thermister รหัส DS 1820 ของ Dallas ที่มีการแปลงค่าอุณหภูมิที่วัดได้มาเป็นค่าข้อมูลทาง digital มาเก็บและทำการแสดงผลเป็นภาพ 3 มิติ ผ่านทางหน้าจอคอมพิวเตอร์ ซึ่งจากภาพที่ได้จะทำให้เราสามารถเห็นถึงการกระจายอุณหภูมิที่เกิดขึ้นในระบบปิดที่เป็นตู้อบเด็กทารกแรกเกิดได้อย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(TEMPERATURE SENSOR IN INFANT INCUBATOR)

Mr. Nuttapon Manopiya-anan

Mr. Nattawut Sang-in

Dr. Chuchat Pintavirooj (Advisor)

Education year 2003

Abstract

This project is brought up from measuring the temperature that occurs in closed system to ventilate from hot temperature. The main part of closed system in this experiment is infant incubator in medical due to characterize of infant incubator is closed square that can ventilate hot temperature inside it and set temperature to use obviously. From flowing of hot temperature, it makes us to curious that how infant incubator can give exactly temperature at the various positions inside it by measuring the temperature at various positions from Thermister code DS 1820 of Dallas that convert the value of measured temperature to the digital data, it will be collected and represented to graphic 3D (three dimensions) through screen of computer. Consequently, we can see the graphic that give result of spreading temperature in closed system in infant incubator clearly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของโครงงาน	1
1.3 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีพื้นฐานและหลักการทำงานของ DS1820	3
2.1 ลักษณะของไทม์สล็อต	4
2.2 ไทม์สล็อตการรีเซ็ตและไทม์สล็อตการตอบสนอง	5
2.3 ไทม์สล็อตการอ่านข้อมูลของอุปกรณ์มาสเตอร์ และการเขียนข้อมูลของอุปกรณ์สเลฟ	6
2.4 ไทม์สล็อตการเขียนข้อมูลของอุปกรณ์มาสเตอร์	7
2.5 การติดต่อสื่อสารกับ DS 1820	7
บทที่ 3 ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์	13
3.1 หลักการทำงานของสเตปมอเตอร์	13
3.2 ชนิดของสเตปมอเตอร์	16
3.3 วงจรขับสเตปมอเตอร์	22
บทที่ 4 โครงสร้างและหลักการทำงานของไมโครคอนโทรลเลอร์	37
4.1 ไมโครคอนโทรลเลอร์ MCS-51 (AT89C51)	37
4.2 ขาพอร์ตของไมโครคอนโทรลเลอร์	38
4.3 โครงสร้างและการทงานของพอร์ต	40
4.4 หน่วยความจำโปรแกรม (Program memory)	41
4.5 หน่วยความจำข้อมูล (Data memory)	43
บทที่ 5 Visual Basic และ MATLAB	53
5.1 ฟังก์ชั่น Visual Basic 6.0 ที่ใช้กับ Serial Port (RS-232)	53
5.2 การเข้าถึงไฟล์ข้อมูล ใน Visual Basic 6.0	54
5.3 MATLAB การประมวลผลข้อมูลด้วยโปรแกรม MATLAB	57
5.4 คำสั่ง(ฟังก์ชั่น) Function ใน MATLAB	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 ขั้นตอนการออกแบบ การทดลองและผลการทดลอง 65
6.1 การหาอุปกรณ์ที่จะนำมาใช้ 65
6.2 การออกแบบวงจร 66
6.3 การทำงานของวงจร 66
6.4 การเขียนโปรแกรมแอพพลิเคชั่นเพื่อทำการเก็บข้อมูลและ
ประมวลผลข้อมูลที่เก็บได้ 66
6.5 การประมวลผลภาพโดยใช้ MATLAB 68
6.6 ผลการทดลอง 68
6.5 สรุปผลการทดลอง 75
6.6 แนวทางการพัฒนา 76
กิตติกรรมประกาศ 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1

บทนำ(Introduction)

ในปัจจุบันนี้อุปกรณ์ทางการแพทย์ที่ใช้กันส่วนมากเราต้องนำเข้าจากต่างประเทศซึ่งมักจะมีราคาสูง ถ้าเราสามารถประดิษฐ์ขึ้นใช้เองโดยที่คุณภาพทัดเทียมกับสินค้านำเข้าก็จะเป็นการช่วยประหยัดเงินตราของประเทศเราได้เป็นจำนวนมาก ด้วยเหตุนี้เองจึงเป็นแรงบันดาลใจให้เกิดโครงงานนี้ขึ้น เด็กทารกที่เกิดมาในปัจจุบันส่วนใหญ่จะมีสุขภาพสมบูรณ์แข็งแรงแต่ก็จะมีบางส่วนที่สุขภาพไม่แข็งแรง เนื่องจากเหตุผลหลายประการ อาทิเช่น คลอดก่อนกำหนด ฯลฯ ซึ่งเราก็ต้องนำเด็กเข้าไปอยู่ในตู้อบเด็กเพื่อควบคุมอุณหภูมิและความชื้น ให้อยู่ในสภาพที่เหมาะสม เพื่อทำให้เด็กนั้นมีสุขภาพที่แข็งแรง ดังนั้น อุณหภูมิในตู้จึงมีความสำคัญกับการเจริญเติบโตของทารกเป็นอย่างมาก เนื่องจากตู้แต่ละตู้นั้นมีคุณภาพไม่เหมือนกัน ซึ่งถ้าเราใช้เครื่องวัดอุณหภูมิเราก็จะรู้และสามารถบอกได้ว่าตู้อบเด็กตู้ไหนมีคุณภาพดีกว่ากัน โดยดูจากการกระจายตัวของอุณหภูมิที่สม่ำเสมอ

โครงงานนี้ได้ใช้อุปกรณ์ที่หาซื้อได้ตามท้องตลาดมาประกอบกัน โดยใช้ไมโครคอนโทรลเลอร์เบอร์ 89C51 เป็นตัวควบคุมการทำงานและใช้สเต็ปมอเตอร์เป็นตัวขับเคลื่อนแผงอ่านอุณหภูมิ และใช้เทอร์มิสเตอร์เบอร์ DS1820 เป็นตัวอ่านอุณหภูมิ แล้วแสดงผลการอ่านบนหน้าจอคอมพิวเตอร์ ซึ่งภาษาที่ใช้ในการเขียนโปรแกรมในการคอนโทรลคือภาษาแอสเซมบลี โดยได้ใช้โปรแกรมแอพพลิเคชั่นในการช่วยเก็บข้อมูลอุณหภูมิลงในเครื่องและประมวลผลข้อมูลออกมาให้เห็นเป็นภาพที่ จุดเด่นของโครงงานนี้คือการใช้การติดต่อระหว่างเทอร์มิสเตอร์ 1820 ที่ใช้สายเพียงเส้นเดียวในการควบคุมการอ่านอุณหภูมิของเทอร์มิสเตอร์ ซึ่งควบคุมโดยไมโครคอนโทรลเลอร์

1.1 วัตถุประสงค์ของโครงงาน

- เพื่อศึกษาการใช้งานไมโครคอนโทรลเลอร์เบอร์ 89C51
- เพื่อศึกษาการใช้งานเทอร์มิสเตอร์เบอร์ DS1820
- เพื่อศึกษาการใช้แอพพลิเคชั่นโปรแกรมในการติดต่อระหว่างฮาร์ดแวร์และการประมวลผล
- เพื่อประยุกต์ใช้งานระหว่างโปรแกรมแอพพลิเคชั่นและฮาร์ดแวร์ภายนอก
- เพื่อประดิษฐ์อุปกรณ์ตรวจวัดทางการแพทย์ที่มีคุณภาพดีแต่ราคาถูก
- เพื่อให้สามารถนำความรู้ที่เรียนมาใช้ให้เกิดเป็นผลสำเร็จ

1.2 ขอบเขตของโครงงาน

- ใช้ไมโครคอนโทรลเลอร์เบอร์ 89C51 ควบคุมการทำงานของวงจรส่วน ต่างๆ ได้
- ใช้ภาษาแอสเซมบลีในการควบคุมการทำงานของไมโครคอนโทรลเลอร์เบอร์ 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถอ่านอุณหภูมิ ภายในตู้ออกมาแสดงผลที่จอคอมพิวเตอร์ได้ตามตำแหน่งที่เราต้องการ
- สามารถแสดงการกระจายอุณหภูมิภายในตู้อบเด็กทารกได้

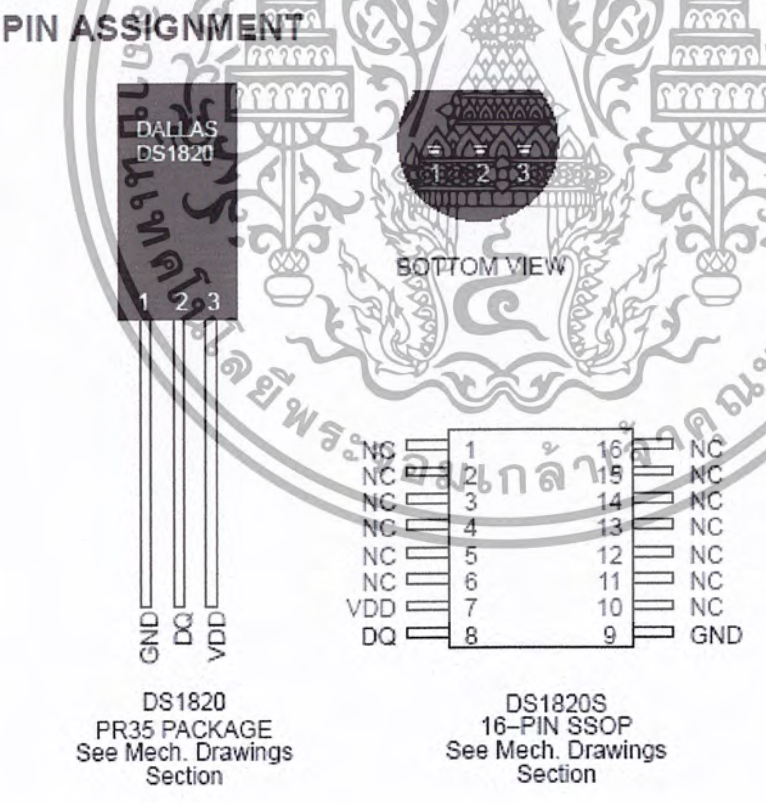
1.3 วิธีการดำเนินงาน

- ศึกษาการเขียนโปรแกรมภาษาแอสเซมบลี
- ศึกษาการทำงานและการควบคุมไมโครคอนโทรลเลอร์เบอร์ 89C51
- ศึกษาการทำงานและการควบคุมเทอร์มิสเตอร์เบอร์ DS1820
- ศึกษาการสื่อสารผ่านพอร์ตอนุกรม
- ศึกษาการออกแบบและสร้างวงจรอ่านอุณหภูมิ
- ศึกษาการรับข้อมูลจากพอร์ตอนุกรมเพื่อเก็บข้อมูลที่วัดได้ และประมวลผลภาพ
- ทดสอบการทำงานและปรับปรุงแก้ไข ผลที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีพื้นฐานและหลักการทำงานของ DS1820

ระบบการสื่อสารข้อมูลแบบหนึ่งสายนั้นเป็นการสื่อสารที่ใช้สายเพียงเส้นเดียว โดยไม่ต้องมีสายสัญญาณนาฬิกามาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น ๆ เนื่องจากสายข้อมูลนั้นจะทำหน้าที่เสมือนหนึ่งเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาหรือต่อไปนี้จะขอเรียกว่า ไทม์สล็อต(time-slot) โดยคาบเวลาต่ำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อต มีการกำหนดขอบเขตได้อย่างชัดเจนการถ่ายทอดข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายทอดข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ สายสัญญาณบนระบบบัสแบบหนึ่งสายนี้จะเป็นสายสัญญาณแบบสองทิศทาง แต่ข้อมูลจะสามารถดินทางได้ในทิศทางเดียวภายในช่วงเวลาหนึ่ง ๆ นั่นคือ มีลักษณะคล้ายกับระบบสื่อสารแบบฮาล์ฟดูเพล็กซ์(half-duplex)



รูปที่ 2.1 ลักษณะของ DS1820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นในระบบการสื่อสารแบบนี้จะต้องมีการระบุให้ชัดเจนว่าอุปกรณ์ตัวใดเป็นตัวมาสเตอร์ อุปกรณ์ตัวใดเป็นตัวสเลฟ ระหว่างการทำงานอุปกรณ์มาสเตอร์และสเลฟสามารถเป็นได้ทั้งตัวรับและตัวส่ง นอยู่กับเงื่อนไขการทำงานในขณะนั้น สายสัญญาณของระบบบัสนี้ต้องกำหนดสภาวะปกติไว้ที่ลอจิกสูง สามารถทำได้โดยการต่อตัวต้านทานค่าประมาณ 4.7 kΩ พูลอัปกับไฟเลี้ยง +5v

2.1 ลักษณะของไทม์สล็อต

อุปกรณ์มาสเตอร์จะเป็นอุปกรณ์เพียงตัวเดียวบนระบบบัสหนึ่งสายนี้ที่สามารถทำการอินนิเชียลสายสัญญาณได้ โดยอุปกรณ์มาสเตอร์จะกำเนิดจุดเริ่มต้นของไทม์สล็อตด้วยการทำให้สายสัญญาณเป็นลอจิกต่ำในช่วงในช่วงเวลาหนึ่ง จากนั้นก็จะทำให้กลับมาเป็นลอจิกสูง ถ้าหากอุปกรณ์สเลฟต้องการส่งข้อมูลมายังอุปกรณ์มาสเตอร์ อุปกรณ์สเลฟจะเป็นตัวควบคุมสภาวะของสายสัญญาณต่อไป จนเสร็จสิ้นกระบวนการ แต่ถ้าหากอุปกรณ์มาสเตอร์ต้องการส่งข้อมูลก็จะสามารดำเนินการต่อไปได้เลย

DETAILED PIN DESCRIPTION

PIN 16–PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	Ground.
8	2	DQ	Data Input/Output pin. For 1–Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V_{DD}	Optional V_{DD} pin. See "Parasite Power" section for details of connection.

DS1820S (16–pin SSOP): All pins not specified in this table are not to be connected.

รูปที่ 2.2 ตารางแสดงการทำงานของขาต่าง ๆ

ฟังก์ชั่นของไทม์สล็อตที่กำหนดโดยอุปกรณ์มาสเตอร์มีด้วยกัน 4 ฟังก์ชั่นคือ

- ไทม์สล็อตของการรีเซ็ต (reset)
- ไทม์สล็อตการอ่านข้อมูล(read data)
- ไทม์สล็อตการเขียนข้อมูล “1”(write one)
- ไทม์สล็อตการเขียนข้อมูล “0” (write zero)

ส่วนอุปกรณ์สเลฟมีฟังก์ชั่นอยู่ด้วยกัน 3 ฟังก์ชั่นคือ

- ไทม์สล็อตการตอบสนอง(presence)
- ไทม์สล็อตการเขียนข้อมูล “1” (write one)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

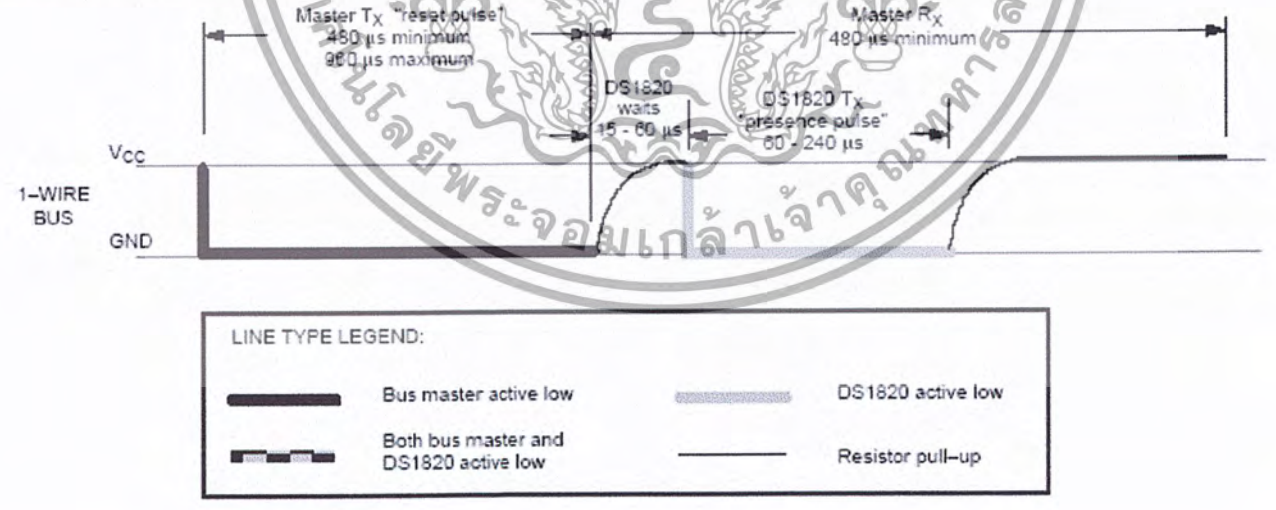
- ไทม์สล็อตการเขียนข้อมูล “0” (write zero)

การแยกแยะฟังก์ชั่นของแต่ละไทม์สล็อตจะใช้ความยาวของคาบเวลาและลักษณะของรูปสัญญาณเป็นตัวกำหนด และทุกครั้งที่มีการเปลี่ยนแปลงฟังก์ชั่นต้องทำให้สายสัญญาณอยู่ในสภาวะว่างเสมอ

2.2 ไทม์สล็อตการรีเซ็ตและไทม์สล็อตการตอบสนอง

อุปกรณ์มาสเตอร์จะทำให้เกิดการรีเซ็ตบนสายสัญญาณเพื่อแจ้งแก่อุปกรณ์สเลฟทราบว่าจะเริ่มต้นการติดต่อ โดยการทำให้สายสัญญาณเป็นลอจิกต่ำต่อเนื่องนานอย่างน้อย 480 ไมโครวินาที และจะต้องทำให้สายสัญญาณกลับมาเป็นลอจิกสูงภายใน 480 ไมโครวินาทีหลังจากนั้น ถ้าหากมีอุปกรณ์สเลฟอยู่บนสายสัญญาณ จะมีการตอบสนองสัญญาณรีเซ็ตโดยการส่งสัญญาณตอบสนอง ทำโดยการทำให้สายสัญญาณเป็นลอจิกต่ำนานประมาณ 60-240 ไมโครวินาที หลังจาสัญญาณรีเซ็ตปรากฏประมาณ 15-60 ไมโครวินาที

INITIALIZATION PROCEDURE “RESET AND PRESENCE PULSES” Figure 11



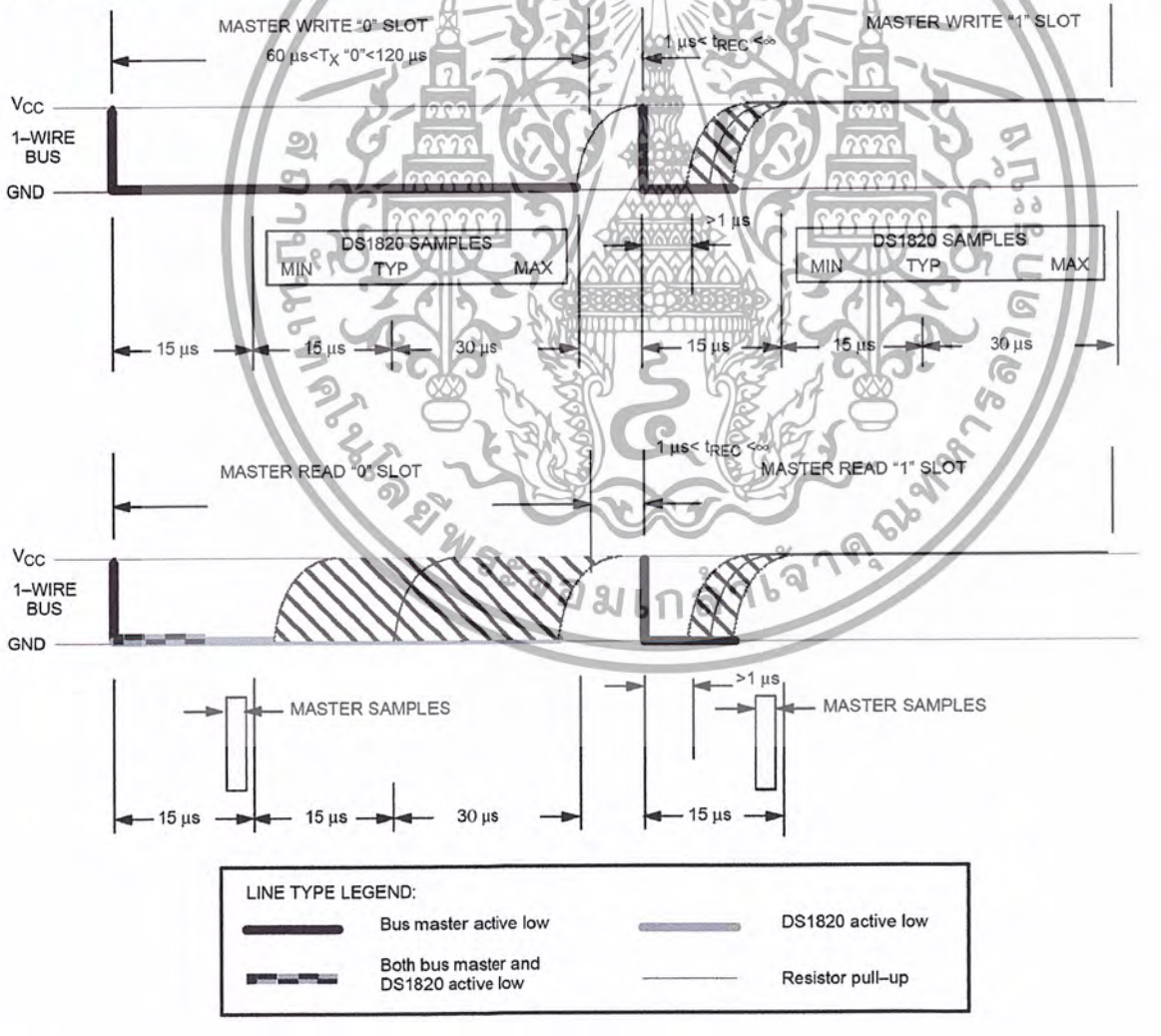
รูปที่ 2.3 แสดงการรีเซ็ตและตอบสนอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไทม์สล็อตการอ่านข้อมูลของอุปกรณ์มาสเตอร์และการเขียนข้อมูลของอุปกรณ์สเลฟ

เมื่อต้องการอ่านข้อมูล จากอุปกรณ์สเลฟ อุปกรณ์มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกต่ำนานประมาณ 1-15 ไมโครวินาที จากนั้นต้องทำให้สถานะของสายกลับมาเป็นลอจิกสูง อุปกรณ์สเลฟจะส่งข้อมูลมาให้อุปกรณ์มาสเตอร์โดยถ้าเป็นข้อมูล “0” อุปกรณ์สเลฟจะทำให้สายสัญญาณเป็นลอจิกต่ำนานประมาณ 45 ไมโครวินาที แล้วทำให้สายสัญญาณกลับมาเป็นลอจิกสูงอีกครั้ง แต่ถ้าเป็นข้อมูล “1” อุปกรณ์สเลฟจะทำให้สายสัญญาณเป็นลอจิกสูงต่อเนื่องไปอีก45 ไมโครวินาที รวมเวลาทั้งหมด 60-120 ไมโครวินาที ก็คือในไทม์สล็อตนี้จะต้องใช้เวลารวมไม่เกิน 120 ไมโครวินาที ในขณะที่อุปกรณ์มาสเตอร์จะใช้เวลาในการอ่านข้อมูลอยู่ระหว่าง 15
และ 60 ไมโครวินาทีหลังจากเริ่มต้นไทม์สล็อตนี้ ซึ่งก็คือถ้าอุปกรณ์มาสเตอร์อ่าน อุปกรณ์สเลฟก็ต้องทำการเขียน

READ/WRITE TIMING DIAGRAM Figure 12



รูปที่ 2.4 แสดงการอ่านและเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

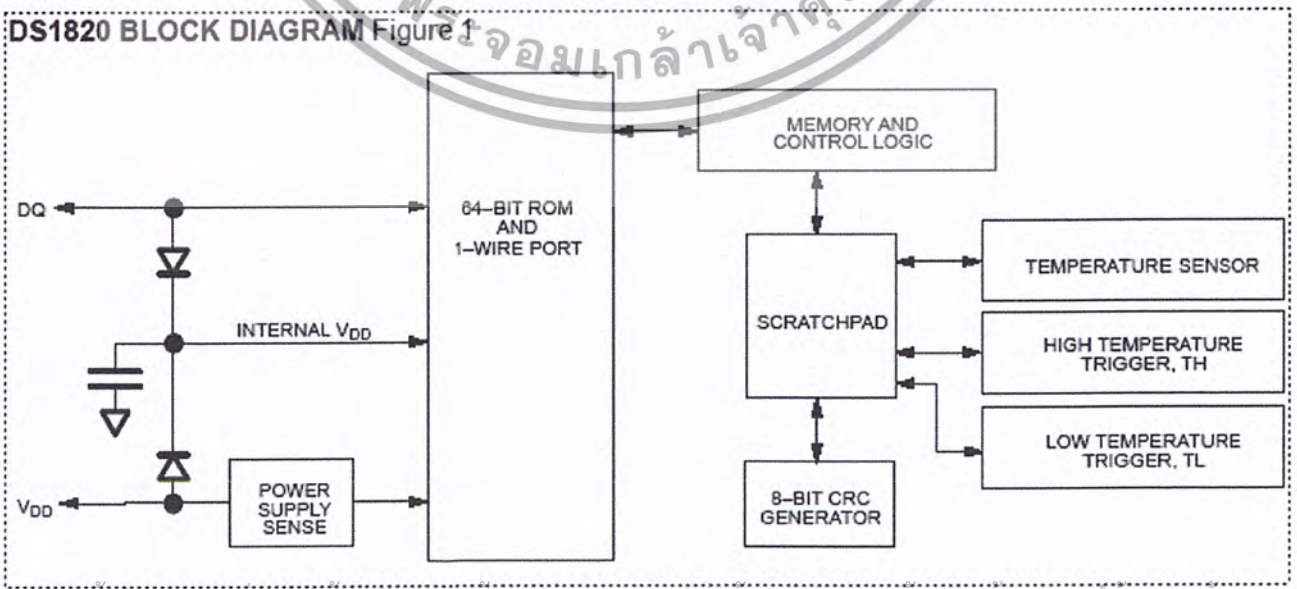
2.4 ไทม์สล็อตการเขียนข้อมูลของอุปกรณ์มาสเตอร์

เมื่ออุปกรณ์มาสเตอร์ต้องการเขียนข้อมูล อุปกรณ์มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกต่ำประมาณ 1-15 ไมโครวินาที จากนั้นต้องทำให้สถานะของสายเป็นลอจิกสูง แล้วดำเนินการเขียนข้อมูลได้ในทันที ถ้าข้อมูลที่ต้องการเขียนไปยังอุปกรณ์สเลฟเป็น "0" อุปกรณ์มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกต่ำนานประมาณ 45 ไมโครวินาที แล้วทำให้สายสัญญาณ กลับมาเป็นลอจิกสูงอีกครั้ง แต่ถ้าต้องการเขียนข้อมูล "1" อุปกรณ์มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกสูง ต่อเนื่องไปอีก 45 ไมโครวินาที รวมเวลาทั้งหมดในไทม์สล็อตนี้ประมาณ 60-120 ไมโครวินาที

2.5 การติดต่อสื่อสารกับ DS 1820

ในการติดต่อนั้นอุปกรณ์มาสเตอร์จะทำการติดต่อกับอุปกรณ์สเลฟได้ครั้งละ 1 ตัวเท่านั้น ดังนั้นอุปกรณ์สเลฟแต่ละตัวจะต้องมีข้อมูลกำหนดแอดเดรสเฉพาะตัว โดยเก็บไว้ในหน่วยความจำรอมภายในซึ่งรอมภายในตัว DS1820 นั้นจะประกอบด้วย

1. รหัสของตระกูล จำนวน 8 บิต
2. เลขหมายประจำตัว(serial number) จำนวน 48 บิต
3. รหัสตรวจสอบความผิดพลาด(crc) จำนวน 8 บิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 แสดงโครงสร้างภายในของ DS1820

การติดต่อนั้นจะเริ่มต้นด้วยการที่ตัวมาสเตอร์ส่งสัญญาณรีเซ็ตไปยังตัว DS1820 จากนั้นตัว DS1820 ก็จะส่งสัญญาณตอบสนองมายังตัวมาสเตอร์(ไมโครคอนโทรลเลอร์) หลังจากนั้นตัวมาสเตอร์ก็จะส่งคำสั่งเลือก Rom function ไปยังตัว DS1820 เพื่อเลือกว่าจะทำงานในลักษณะใด ซึ่ง Rom function จะประกอบไปด้วย

1. Read rom
2. Match rom
3. Search rom
4. Alarm search
5. Skip rom

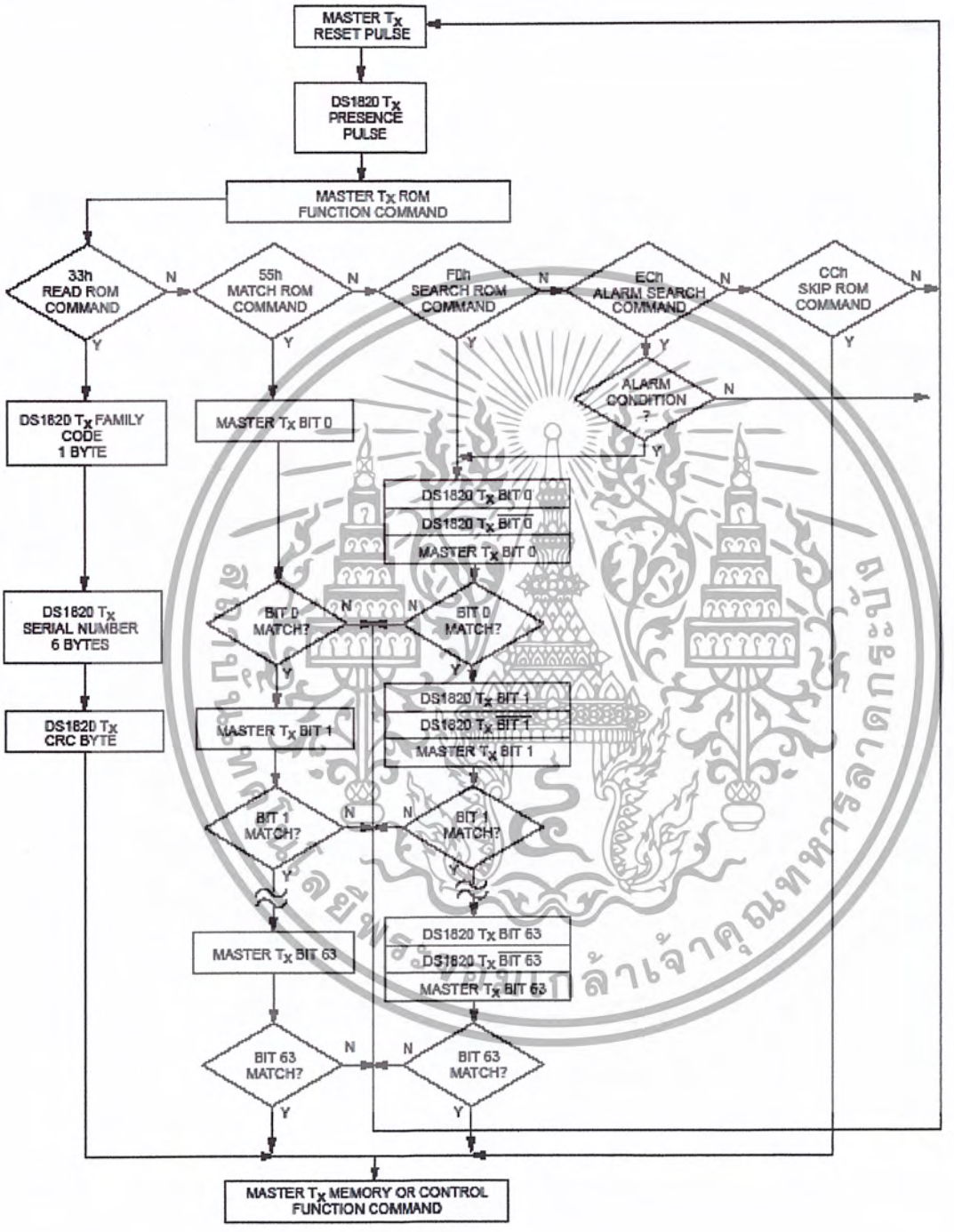
หลังจากที่ทำงานในคำสั่งของ Rom function เสร็จแล้ว ตัวมาสเตอร์ก็จะส่งคำสั่งไปให้ตัว DS1820 ว่าจะทำงานใน Memory function ไหน โดยที่ Memory function ประกอบไปด้วย

1. Write scratchpad
2. Read scratchpad
3. Copy scratchpad
4. Convert temperature
5. Recall E^2
6. Read power supply

โดยที่ทุกครั้งของการติดต่อจะต้องเริ่มต้นด้วยการรีเซ็ตจากตัวมาสเตอร์ซึ่งก็คือตัว 89C51 นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

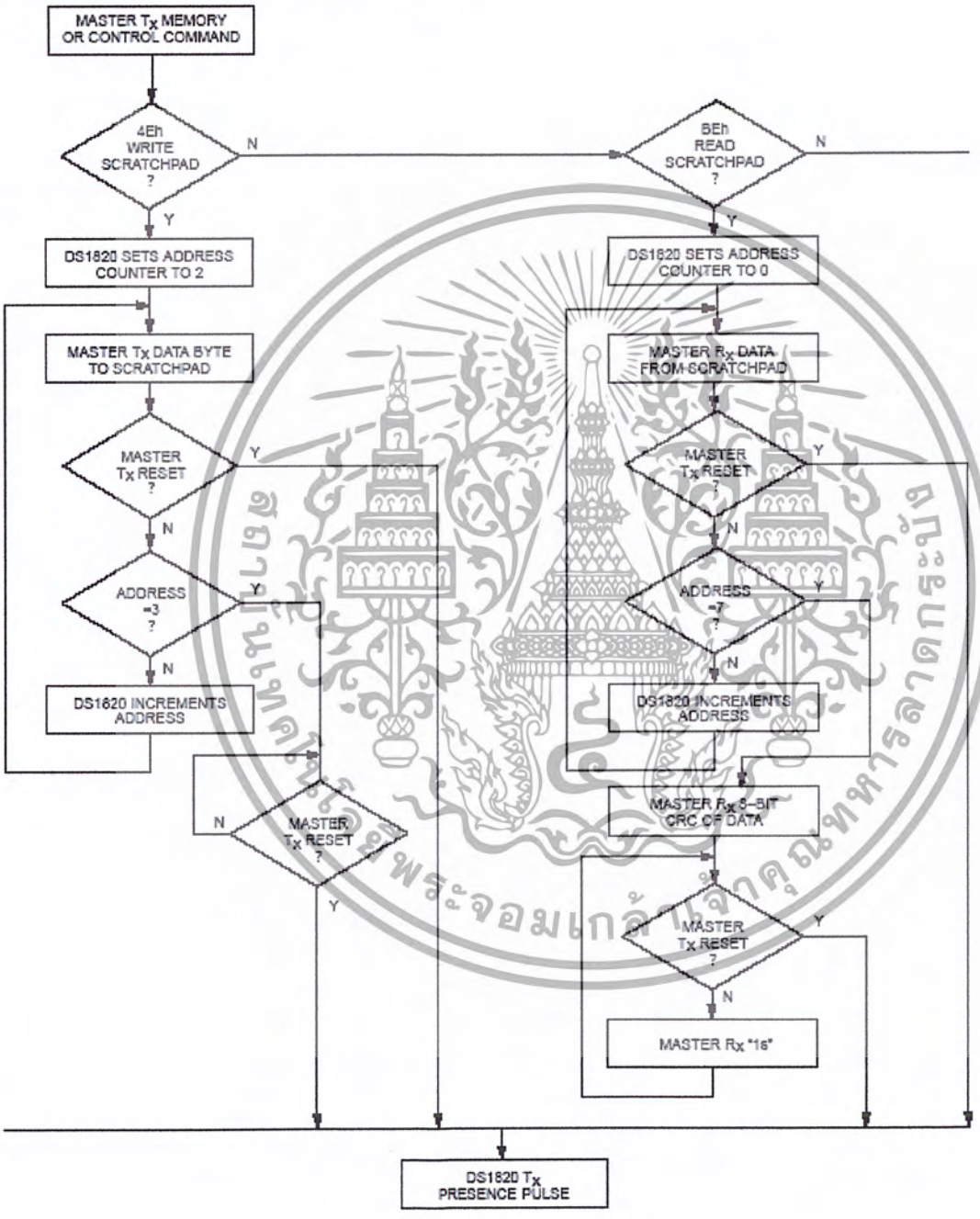
ROM FUNCTIONS FLOW CHART Figure 6



รูปที่ 2.6 แสดงโครงสร้างของ Rom function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

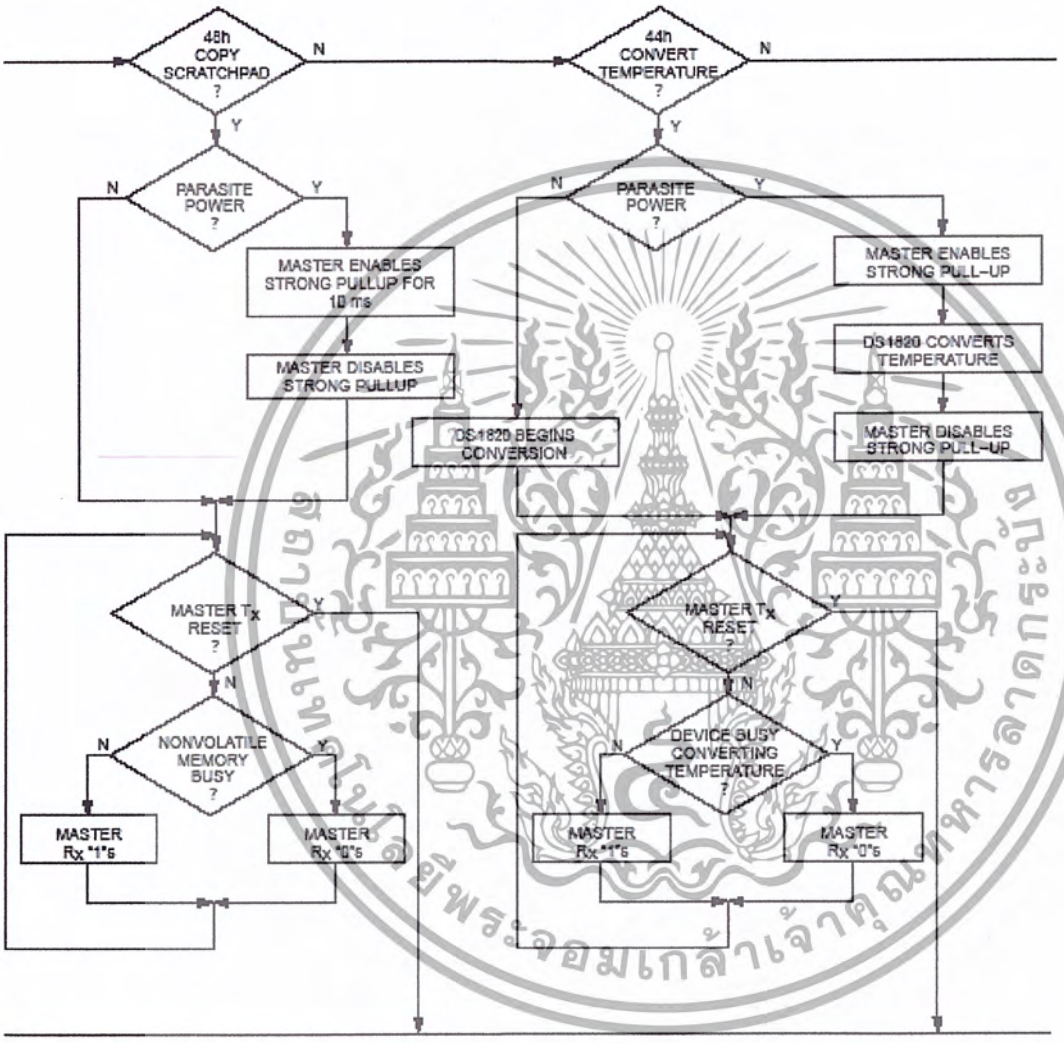
MEMORY FUNCTIONS FLOW CHART Figure 10



รูปที่ 2.7 แสดง โครงสร้างของMemory function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

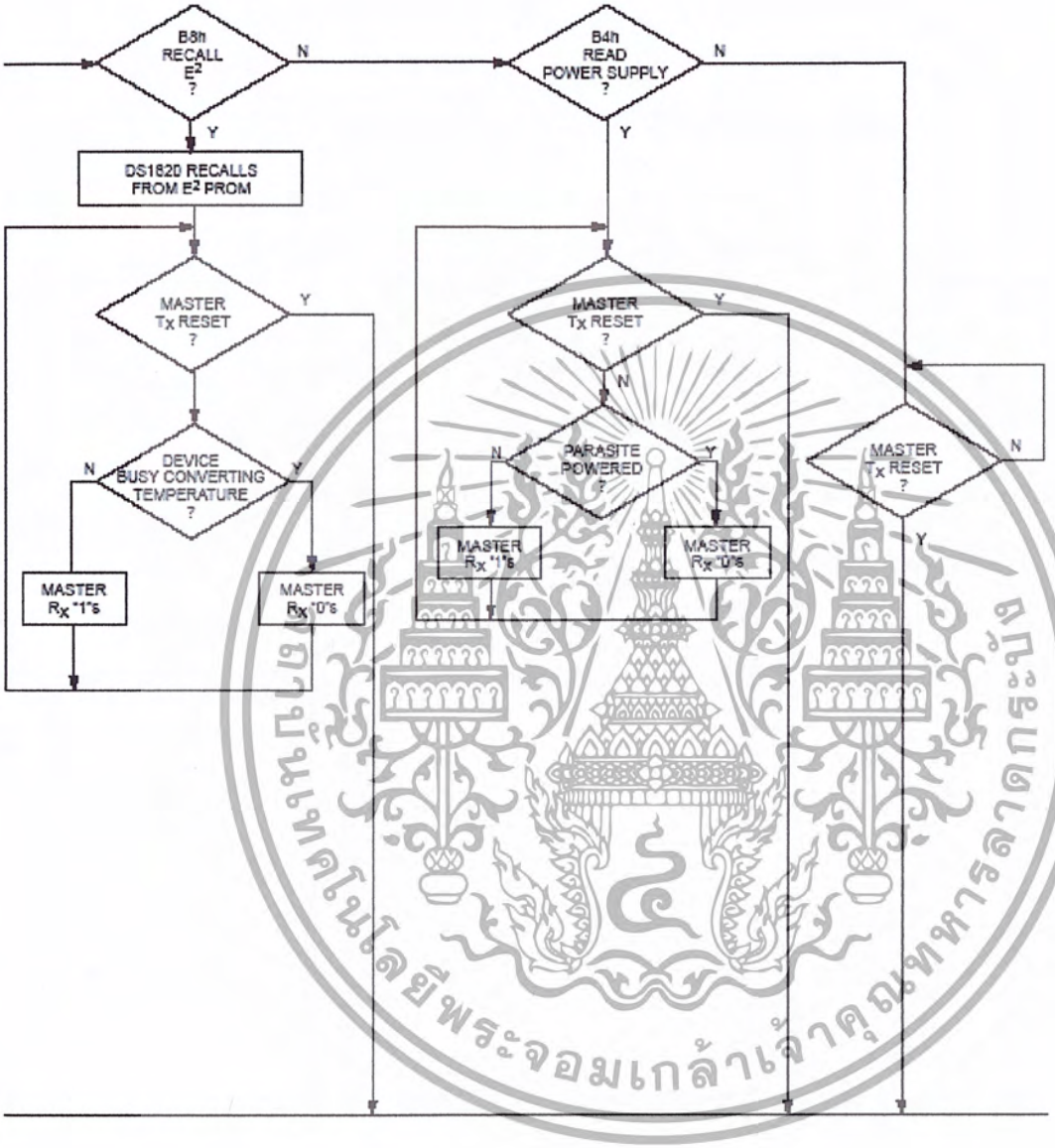
MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



รูปที่ 2.8 แสดงโครงสร้างของMemory function(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMORY FUNCTIONS FLOW CHART Figure 10 (cont'd)



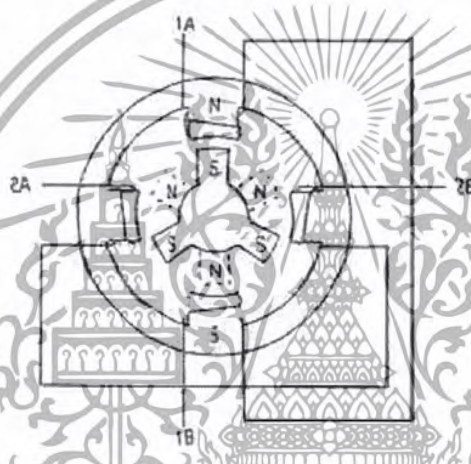
รูปที่ 2.9 แสดงโครงสร้างของMemory function(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์

3.1 หลักการทำงานของสเตปมอเตอร์

สเตปมอเตอร์สามารถแบ่งโครงสร้างทางกายภาพออกได้เป็น 2 ส่วน คือ สเตเตอร์ (STATOR) และโรเตอร์ (ROTOR) ตัวสเตเตอร์เป็นส่วนที่อยู่กับที่ ประกอบด้วยขดลวดทองแดงซึ่งพันอยู่รอบแกนเหล็ก เพื่อสร้างสนามแม่เหล็กเมื่อมีการจ่ายกระแสผ่านขดลวด ส่วนโรเตอร์เป็นส่วนที่เคลื่อนที่ มีลักษณะเป็นแท่งเหล็กทรงกลม และที่ผิวรอบนอกมีลักษณะเป็นซีกฟันซึ่งทำจากแม่เหล็กถาวร ดังรูปที่ 3.1

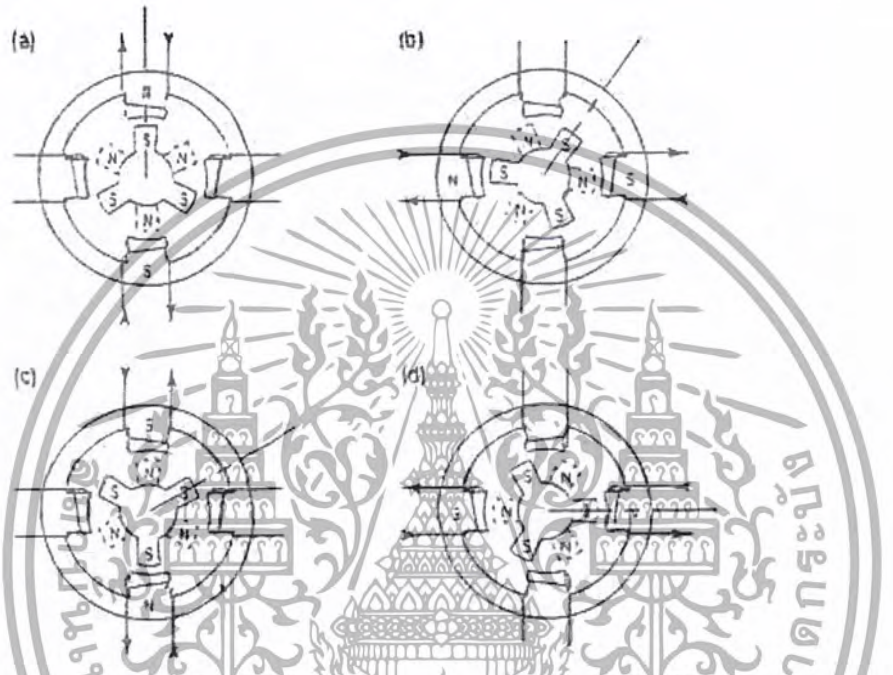


รูปที่ 3.1 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์ที่มีจำนวนสเตปมอเตอร์ต่อรอบเท่ากัน 12

เมื่อยังไม่มีการจ่ายกระแสให้กับขดลวดของมอเตอร์ ซีกฟันอันใดอันหนึ่งของโรเตอร์จะอยู่ในตำแหน่งที่ตรงกันกับซีกฟันอันใดอันหนึ่งของสเตเตอร์ ทั้งนี้เป็นเพราะแม่เหล็กถาวรที่ตัวของโรเตอร์พยายามที่จะทำให้ค่าต้านทานทางแม่เหล็กไฟฟ้า(RELUCTANCE) มีค่าน้อยที่สุด ซึ่ง ณ.ที่จุดที่ซีกฟันของตัวโรเตอร์และสเตเตอร์ตรงกันนั้นมีค่าความต้านทานทางแม่เหล็กไฟฟ้าน้อยที่สุดทำให้เกิดเส้นแรงแม่เหล็กไฟฟ้ามากที่สุด และจากรูปที่ 3.1 เส้นแรงแม่เหล็กไฟฟ้าจะทำให้เกิดขั้วแม่เหล็กเหนือและใต้ขึ้นมา 2 คู่ ทั้งที่ตัวสเตเตอร์และตัวโรเตอร์ดังรูป ค่าทอร์ค (TORQUE) ที่ทำให้ตัวโรเตอร์สามารถยึดอยู่ในตำแหน่งดังกล่าวนี้เรียกว่า ดีเท็นท์ทอร์ค (DETENTTORQUE) (หมายความว่า การที่จะทำให้มอเตอร์เคลื่อนที่ในขณะที่ไม่ได้จ่ายกระแสให้กับขดลวดของมอเตอร์จะต้องออกแรงมากกว่าค่าของดีเท็นท์ทอร์ค จึงจะทำให้โรเตอร์เริ่มเคลื่อนที่ได้) รูปที่ 3.1 นั้นมี 12 ตำแหน่งที่สามารถเกิดดีเท็นท์ทอร์คได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อจ่ายกระแสให้กับขดลวดที่อยู่ในสเตเตอร์คู่ใดคู่หนึ่ง ดังรูปที่ 3.2 (a) จะทำให้เกิดขั้วแม่เหล็กเหนือและใต้ที่ซีกฟันของตัวสเตเตอร์ ซึ่งจะดึงดูดซีกฟันของตัวโรเตอร์ที่มีขั้วแม่เหล็กที่มีศักย์ต่างกันที่อยู่ใกล้ที่สุดเข้าไว้ ตำแหน่งนี้เรียกว่า สเตเบิ้ล โพสซิชั่น (STABLE POSITION) ของโรเตอร์ จะมีจำนวนตำแหน่งเท่ากับจำนวนซีกฟันของโรเตอร์ และแรงที่จะทำให้โรเตอร์เปลี่ยนตำแหน่งไปจากตำแหน่งสเตเบิ้ล โพสซิชั่นได้นี้เรียกว่า โฮลดิ้ง ทอร์ค (HOLDING TORQUE)

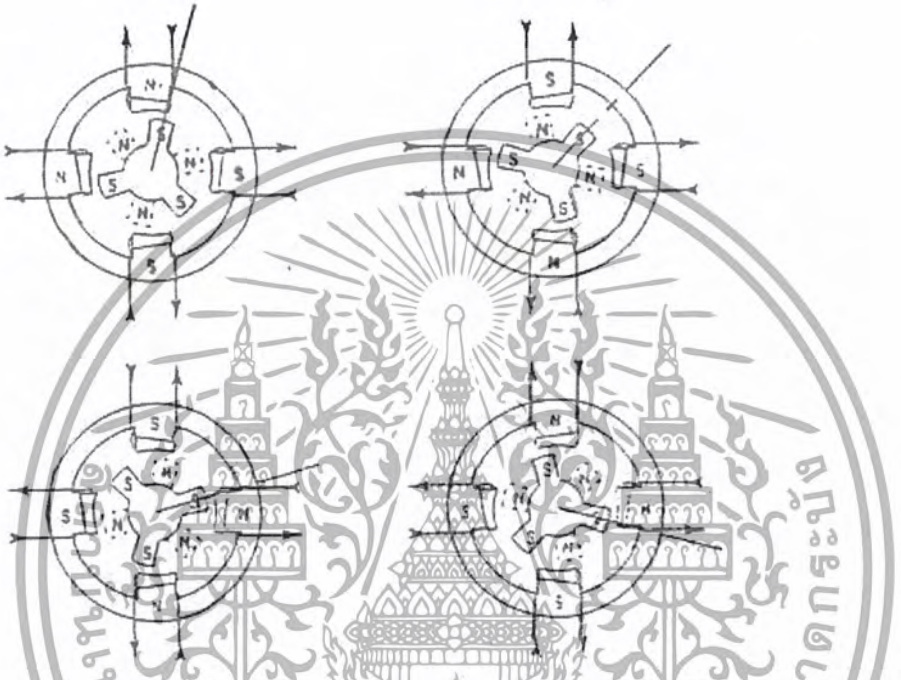


รูปที่ 3.2 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบ เต็มสเตปหนึ่งเฟส

เมื่อสับเปลี่ยนการจ่ายกระแสให้แก่ขดลวด จากขดหนึ่งไปยังอีกขดหนึ่ง เนื่องจากขดลวดวางอยู่ในตำแหน่งที่ต่างกัน 90° ก็จะทำให้ตัวสเตเตอร์ดึงดูดซีกฟันของตัวโรเตอร์อีกซีกหนึ่งที่ใกล้ที่สุดเข้าไว้ ซึ่งจะทำให้ตัวโรเตอร์เคลื่อนที่ไป 1 สเตปหรือ 30° ดังรูปที่ 3.2 (b) จากนั้นก็เปลี่ยนไปจ่ายกระแสให้กับขดลวดชุดแรกโดยในคราวนี้เปลี่ยนทิศทางการไหลของกระแสให้ตรงกันข้ามกับครั้งแรก ซึ่งจะทำให้ตัวโรเตอร์เคลื่อนที่ไปอีก 1 สเตป(เคลื่อนที่ไป 30°) ดังรูปที่ 3.2 (c) หลังจากนั้นก็ไปจ่ายกระแสให้กับขดลวดที่สองโดยกลับทิศ-ทางของกระแสที่ป้อนให้อีกเช่นกันทำให้โรเตอร์หมุนไป 90° ดังรูป 3.2 (d) และถ้าหากเราป้อนกระแสให้กับมอเตอร์เหมือนที่เราป้อนในครั้งแรกแล้ว ซีกฟันถัดไปของตัวโรเตอร์จะอยู่ในตำแหน่งที่เหมือนกับใน รูปที่ 3.2 (a) อีกครั้งหนึ่ง ถ้าหากเราต้องการเคลื่อนที่หนึ่งรอบ เราต้องทำการกระตุ้นให้มอเตอร์เคลื่อนที่ไปจนครบ 12 สเตป และถ้าต้องการให้โรเตอร์หมุนไปอีกทิศทางหนึ่ง ก็ทำการสลับลำดับในการจ่ายกระแส จากรูปที่ 3.2 (a) , (b), (c) และ(d) ไปเป็น (a), (d), (c) และ(b) ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

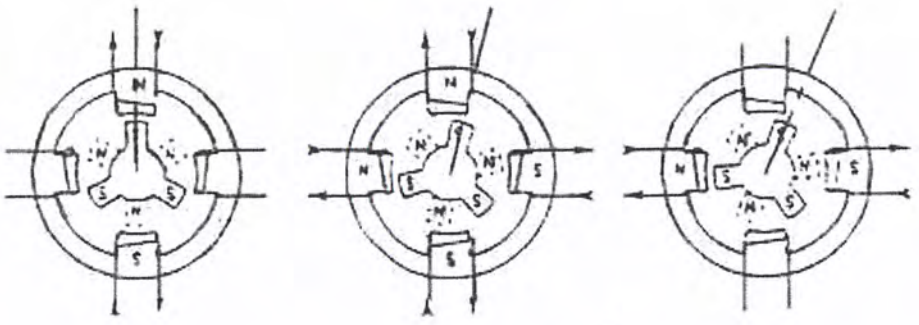
ถ้าจ่ายกระแสให้แก่ขดลวดทั้งสองขดพร้อม ๆ กัน ซีกฟันของโรเตอร์จะอยู่ ณ. ตำแหน่งระหว่างซีกฟันของสเตเตอร์ เพราะฉะนั้นการจ่ายกระแสให้แก่มอเตอร์แบบนี้จะให้ทอร์คมากกว่าแบบที่จ่ายกระแสในเวลาขณะใดขณะหนึ่งเพียงขดเดียว ดังรูปที่ 3.3 ซึ่งแสดงถึงการป้อนกระแสให้กับขดลวดแต่ละขด เพื่อให้สเตปมอเตอร์เคลื่อนที่แต่ถ้าในตอนแรกจ่ายกระแสให้กับขดลวดเพียงขดเดียวแล้วสเตปต่อไปจ่ายกระแสให้กับขดลวด พร้อม ๆ กันสองขด



รูปที่ 3.3 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบ เต็มสเตปสองเฟส

ทำอย่างนี้สลับกันไปจะทำให้โรเตอร์เคลื่อนที่ไปสเตปละ 15° ดังรูปที่ 3.4 และในการขับเคลื่อนแบบนี้จะทำให้สเตปต่อรอบ (STEP/REV) เพิ่มขึ้นเป็นเท่าตัว การขับสเตปมอเตอร์แบบนี้เรียกว่า การขับแบบครึ่งสเตป(HALF STEPPING)ซึ่งนิยมใช้กันในงานอุตสาหกรรมเป็นอย่างมาก แม้ว่าในบางครั้งจะให้ทอร์คน้อยกว่า แต่มีข้อดีกว่าการขับแบบเต็มสเตป (FULL STEPPING) ก็คือ การเคลื่อนที่เป็นไปอย่างราบเรียบ(ที่ความเร็วไม่สูงมากนัก)อีกทั้งมีความแม่นยำสูงและการสั่นสะเทือนน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



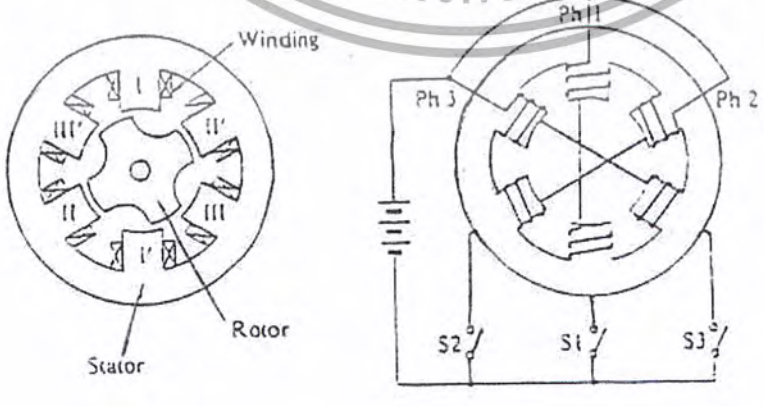
รูปที่ 3.4 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบ ครึ่งสเตป

3.2 ชนิดของสเตปมอเตอร์

สเตปมอเตอร์ สามารถแบ่งออกได้หลายชนิดตามลักษณะโครงสร้างและการใช้งานดังต่อไปนี้

- 3.2.1 สเตปมอเตอร์ชนิดปรับค่ารีลัคแตนซ์ได้(VARIABLE RELUCTANCE STEPPING MOTOR)

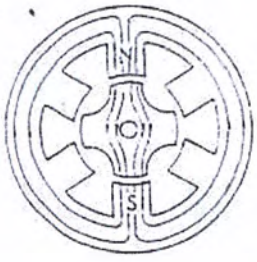
สเตปมอเตอร์ชนิดนี้สามารถปรับค่ารีลัคแตนซ์ได้ ซึ่งรูปที่ 3.5 แสดงภาพตัดขวางของสเตปมอเตอร์แบบ 3 เฟส โดยที่สเตเตอร์มีฟันทั้งหมด 6 ซี่ ซี่ที่อยู่ตรงข้ามกันหรือทำมุม 180° ซึ่งกันและกันจะเป็นเฟสเดียวกัน ขดลวดที่พันอยู่ที่ฟันของสเตเตอร์ในแต่ละเฟสจะต่ออนุกรมหรือขนานก็ได้ จากรูปที่ 3.5 เป็นการต่อแบบอนุกรม ส่วนโรเตอร์นั้นมีฟัน 4 ซี่ ทั้งโรเตอร์และสเตเตอร์ทำมาจากโลหะซิลคอน ซึ่งมีสภาพซึมซับทางแม่เหล็กสูงและยอมให้สนามแม่เหล็กจำนวนมากไหลผ่านได้ ฟันของสเตเตอร์ในเฟสเดียวกันจะมีขั้วต่างกัน โดยซี่ฟันโดยซี่ I, I, III เป็นขั้วเหนือและซี่ I, II, III เป็นขั้วใต้หลังจากถูกกระตุ้น



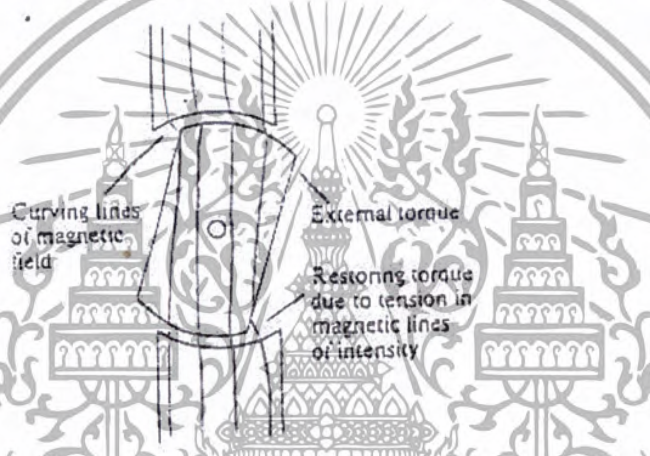
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

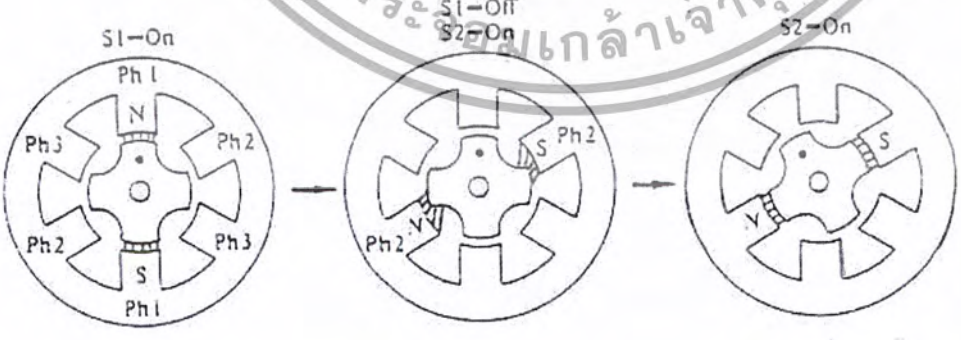
รูปที่ 3.5 แสดงภาพตัดขวางของสเตปมอเตอร์แบบ 3 เฟส



รูปที่ 3.6 แสดงตำแหน่งสมดุล เมื่อเฟสใดเฟสหนึ่งของสเตปมอเตอร์ถูกกระตุ้น



รูปที่ 3.7 แสดงแรงภายนอกที่มีผลต่อเส้นแรงแม่เหล็ก



รูปที่ 3.8 แสดงขั้นตอนการเคลื่อนที่ของโรเตอร์เมื่อสเตปมอเตอร์ถูกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

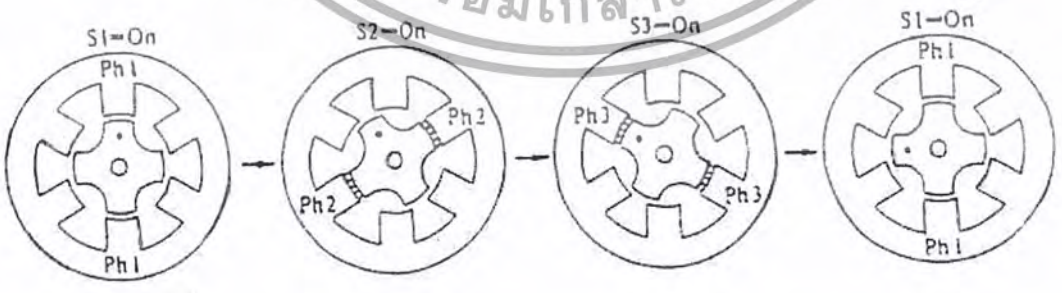
55481

กระแสที่ไหลในแต่ละเฟสถูกควบคุมโดยสวิทช์ ปิด/เปิด ถ้าเฟส I ถูกกระตุ้นจะมีกระแสไหลและเกิด ฟลักซ์แม่เหล็กดังแสดงในรูปที่ 3.6 แกนโรเตอร์จะอยู่ตำแหน่งเดียวกับซี่ I และ I' ทำให้ทั้งโรเตอร์และสเตเตอร์อยู่ในแนวเดียวกันกรณีนี้จะทำให้ค่ารีลัคแตนซ์มีค่าน้อยที่สุดซึ่งเป็นตำแหน่งที่สมดุล ถ้าโรเตอร์ถูกกระทำจากแรงภายนอกจะทำให้เปลี่ยนตำแหน่ง ดังรูปที่ 3.7 แรงบิดกระทำกับโรเตอร์ในทิศตามเข็มนาฬิกาทำให้ตำแหน่งเปลี่ยนไป มีผลทำให้เส้นแรงแม่เหล็กเคลื่อนที่จากซี่ของโรเตอร์และสเตเตอร์ เมื่อโรเตอร์และสเตเตอร์ไม่ได้อยู่ในแนวเดียวกันแล้วค่ารีลัคแตนซ์จะมีค่ามาก จากนั้นสเตปมอเตอร์จะทำตัวให้มีค่ารีลัคแตนซ์น้อยที่สุด พอเฟส II ถูกกระตุ้น ดังรูปที่ 3.8 โรเตอร์ถูกแรงภายนอกกระทำให้เคลื่อนไป 30° ในทิศทางทวนเข็มนาฬิกา จากนั้นก็จะย้ายจากมุมที่เกิดการกระตุ้นกลับไปยังตำแหน่งที่ค่ารีลัคแตนซ์น้อยที่สุด การย้ายจากมุมที่เกิดการกระตุ้นแต่ละครั้งให้กลับไปตำแหน่งเดิมเรียกว่า สเตป

คุณสมบัติพื้นฐานของสเตปมอเตอร์แบบปรับค่ารีลัคแตนซ์

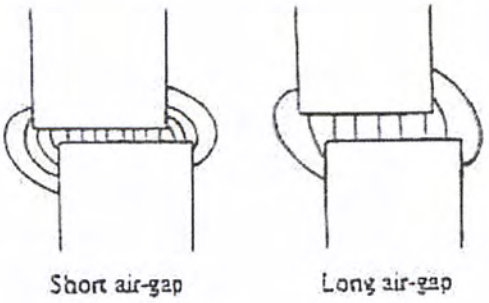
1. ช่องว่าง (AIR GAP) ต้องมีขนาดเล็กที่สุด

ช่องว่างระหว่างฟันของโรเตอร์และสเตเตอร์ต้องเล็กที่สุดเท่าที่จะเป็นไปได้ เพื่อให้ทอร์คที่เกิดขึ้นมีค่ามากและมีความแม่นยำทางตำแหน่งมากขึ้น รูปที่ 3.10 แสดงการเปรียบเทียบระหว่างช่องว่างที่กว้างและแคบในขณะที่แหล่งกำเนิดสนามแม่เหล็กแหล่งเดียวกันหรือมีระดับความแรงเท่ากันช่องอากาศที่เล็กจะให้ทอร์คมากกว่าและทำให้เคลื่อนที่จากจุดสมดุลน้อยกว่าช่องว่างขนาดใหญ่เมื่อมีแรงบิดจากภายนอกมากระทำต่อโรเตอร์



รูปที่ 3.9 แสดงขั้นตอนการเคลื่อนที่ของสเตปมอเตอร์

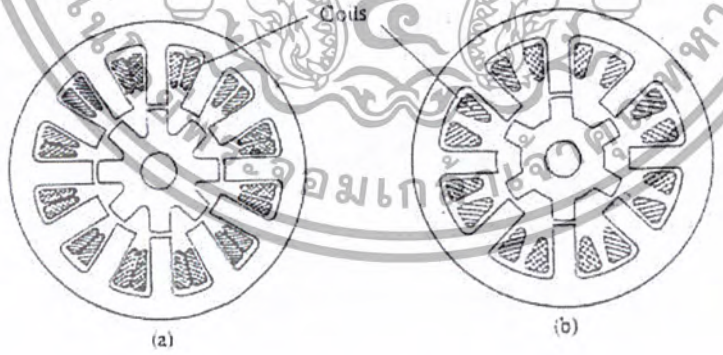
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงการเปรียบเทียบเส้นแรงแม่เหล็กระหว่างช่องว่างที่กว้างและแคบ

2. มุมของการสเตปแคบ

คุณสมบัติอีกประการของสเตอร์คือจะต้องมีมุมของการสเตปเล็กที่สุดเท่าที่จะเป็นไปได้ มุมที่แสดงในรูปที่ 3.6 ยังไม่ถือว่าเป็นมุมที่เล็ก แต่รูป 3.11 (a) แสดงมอเตอร์ 3 เฟสซึ่งมีจำนวนฟันของโรเตอร์และสเตเตอร์เป็น 2 เท่าของรูปที่ 3.6 ส่วน รูปที่ 3.11 (b) แสดงมอเตอร์ 4 เฟส มุมของการสเตปของทั้ง 2 โครงสร้างนี้เท่ากับ 15° นอกจากนี้ยังมีบางชนิดที่มีมุมของการสเตป 7.5° โดยฟันของโรเจอร์และสเตเตอร์มี 12 และ 16 ซี่ตามลำดับ



รูปที่ 3.11 (a) แสดงมอเตอร์ 3 เฟส รูปที่ 3.11 (b) แสดงมอเตอร์ 4 เฟส

ความสัมพันธ์ของมุมของการสเตป θ s, มุมเฟส m, จำนวนซี่ฟันของโรเตอร์ Nr, จำนวนสเตป S แสดงดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S = 360/\theta s = mNr$$

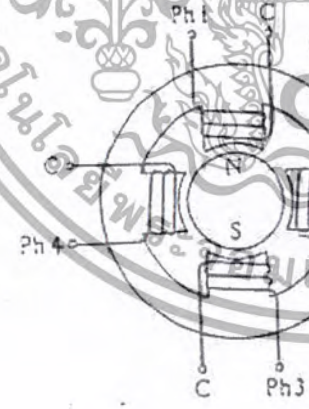
เพื่อที่จะลดขนาดของมุมสเตปลงต้องเพิ่มจำนวนซี่ฟันของโรเตอร์โดยที่โครงสร้างของแต่ละขั้วของเฟสใดๆ สเตเตอร์จะมีหลายซี่ฟัน แต่ก็ไม่ใช่องค์ประกอบโดยตรงที่จะกำหนดมุมของสเตปมอเตอร์

3. การสร้างสเตปมอเตอร์ให้มีโครงสร้างหลาสแตกเพื่อเพิ่มประสิทธิภาพ

โครงสร้างของสเตปมอเตอร์แบบนี้จะมี 1 เฟส โดยที่โรเตอร์และสเตเตอร์มีซี่ฟันเหมือนกันซึ่งจะเป็นการเพิ่มประสิทธิภาพในด้านทอร์คต่อหน่วยปริมาตรของโรเตอร์

- 3.2.2 สเตปมอเตอร์แบบแม่เหล็กถาวร

สเตปมอเตอร์ชนิดนี้ใช้แม่เหล็กถาวรเป็นโรเตอร์ และมีซี่ฟันของสเตเตอร์ล้อมรอบ ซี่ฟันของสเตเตอร์ถูกพันด้วยขดลวดสำหรับสร้างสนามแม่เหล็ก เมื่อต้องการให้สเตปมอเตอร์แบบแม่เหล็กถาวรมีขนาดมุมของสเตปเล็กลงจะต้องเพิ่มขั้วแม่เหล็กของโรเตอร์และจำนวนซี่ฟันของสเตเตอร์ แต่ก็มีขีดจำกัดในการเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์ เนื่องจากการสร้างแม่เหล็กถาวรสร้างให้มีโครงสร้างโดยมีขั้วแม่เหล็กหลายขั้วทำได้ยาก

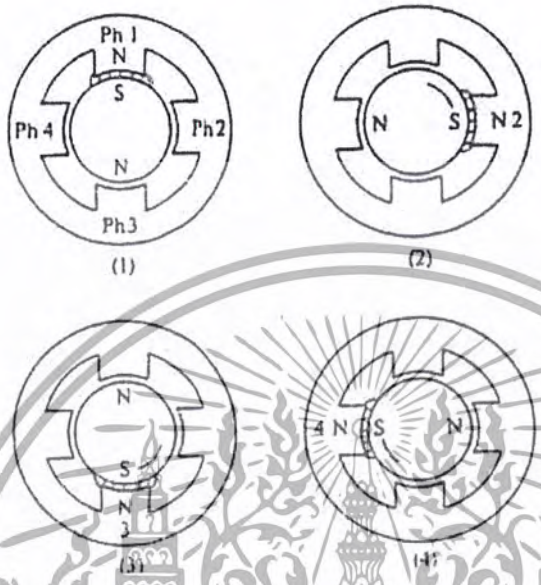


รูปที่ 3.12 แสดงโครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร

ตัวอย่างการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวร สมมติว่าสเตปมอเตอร์แบบแม่เหล็กถาวร ขนาด 4 เฟส มีโรเตอร์เป็นแม่เหล็กถาวรทรงกระบอกและสเตเตอร์มี 4 ซี่ฟันซึ่งรอบ ๆ พันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขดลวด มีรูปแบบพื้นฐานของการทำงานคือ เมื่อสร้างสัญญาณกระตุ้นตามลำดับเฟส โรเตอร์จะหมุนไปตามทิศทางของการกระตุ้น ดังแสดงในรูปที่ 3.13



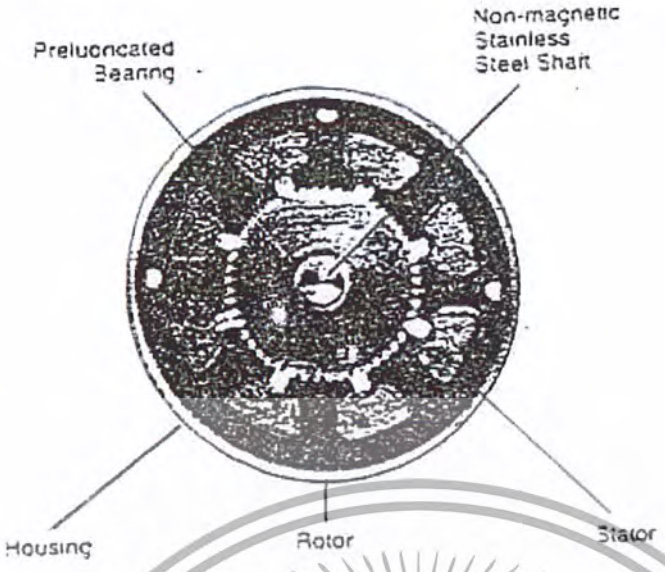
รูปที่ 3.13 แสดงการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวรขนาด 4 เฟส

ข้อเสียของสเตปมอเตอร์แบบแม่เหล็กถาวรคือ มีขนาดมุมสเตปใหญ่ทำให้ความละเอียดของสเตปต่อรอบน้อยเนื่องจากโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวร การสร้างแม่เหล็กถาวรให้มีขั้วหลายขั้วทำได้ยากทำให้ไม่สามารถสร้างสเตปขนาดเล็กได้ สเตปมอเตอร์แบบแม่เหล็กถาวรส่วนใหญ่จะมีโครงสร้างขนาดเล็ก ทำให้ค่าทอร์คที่ได้ต่อหน่วยต่อปริมาตรต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพในเรื่องของทอร์ค แม่เหล็กถาวรที่ใช้ต้องทำมาจากสารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูง

- 3.2.3 สเตปมอเตอร์แบบไฮบริดจ์

สเตปมอเตอร์ชนิดนี้มีแกนโรเตอร์เป็นแม่เหล็กถาวร โดยมีการทำงานร่วมกันของมอเตอร์แบบแม่เหล็กถาวรและมอเตอร์แบบปรับค่ารีลัคแตนซ์ได้ ไฮบริดจ์ สเตปมอเตอร์นี้มีโครงสร้างของสเตเตอร์คล้ายกับโครงสร้างของสเตปมอเตอร์แบบปรับค่ารีลัคแตนซ์ได้ แต่ต่างกันที่การต่อขดลวดโดยที่แต่ละเฟสของสเตปมอเตอร์ขดลวดทั้งสองจะพันอยู่ที่ขั้วเดียวกันเรียกว่าไบไฟลาร์ (BIFILAR) ซึ่งในการกระตุ้นแต่ละครั้งจะให้ขั้วที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์

คุณสมบัติที่สำคัญของไฮบริดจ์สเตปมอเตอร์

โครงสร้างของมอเตอร์จะมีแม่เหล็กถาวรอยู่ตรงกลางระหว่างเฟสทั้งสอง การเหนี่ยวนำสนามแม่เหล็กทำได้โดยใช้สนามแม่เหล็กซึ่งสร้างจากสเตเตอร์ซึ่งเป็นสนามแม่เหล็กแบบเฮทเทอโรโพลาร์ (HETEROPOLAR FIELD) ดังนั้นทอร์คเกิดจากการทำงานร่วมกันของสนามแม่เหล็ก 2 ชนิดคือ สนามแม่เหล็กถาวรและสนามแม่เหล็กเหนี่ยวนำที่เกิดจากการกระตุ้นของขดลวดแต่ละขด โครงสร้างของซี่ฟันของสเตเตอร์จะใหญ่กว่าซี่ฟันของโรเตอร์เล็กน้อยเพื่อเพิ่มความถูกต้องแม่นยำทางตำแหน่งของการเคลื่อนที่

หลักการทำงานของไฮบริดจ์สเตปมอเตอร์ที่แตกต่างจากสเตปมอเตอร์แบบแปรค่ารีลัคแตนซ์ได้ คือแรงบิดที่เกิดขึ้นจากสนามแม่เหล็กจะไม่ขึ้นอยู่กับกระแสที่ไหลผ่านขดลวดเพียงอย่างเดียวแต่ขึ้นอยู่กับโครงสร้างของซี่ฟันด้วย ซึ่งซี่ฟันถูกออกแบบเพื่อให้ได้โครงสร้างขนาดเล็ก และใช้แม่เหล็กถาวรเป็นแกนกลางเพื่อลดผลของการออสซิลเลททางแมคคานิคส์

ข้อดีของไฮบริดจ์สเตปมอเตอร์ทั้ง 3 ชนิดที่กล่าวมาแล้วยังมีสเตปมอเตอร์ชนิดอื่นๆ ที่ไม่ได้กล่าวถึงอีกเช่น ลิเนียร์สเตปมอเตอร์ ซึ่งเป็นมอเตอร์ที่ได้รับการออกแบบให้มีการเคลื่อนที่แบบเป็นเชิงเส้น อิเล็กโตรดไฮดรอลิคสเตปมอเตอร์ซึ่งเป็นสเตปมอเตอร์กำลังสูงที่ใช้ในงานอุตสาหกรรม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรขับสเตปมอเตอร์

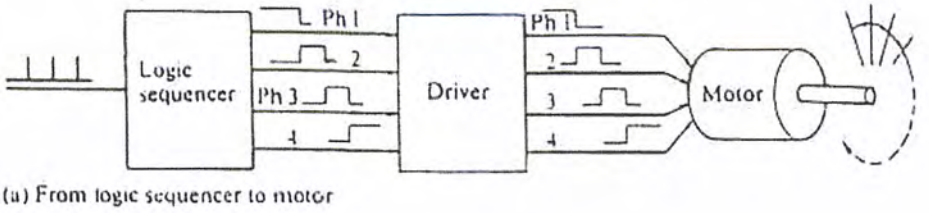
วงจรขับสเตปมอเตอร์ (STEPPER DRIVE) เป็นวงจรที่แปลงสัญญาณที่มีขนาดเล็กๆ (SMALL SIGNAL) จากระบบควบคุม (CONTROL SYSTEM) ให้เป็นสัญญาณไฟฟ้าที่มีกำลังมากขึ้น เพื่อส่งมอบให้กับมอเตอร์ในการทำให้สเตปมอเตอร์เคลื่อนที่มอเตอร์เป็นอุปกรณ์ที่ให้ผลิตผลเป็นทอร์คนี้ได้ถูกสร้างโดยสนามแม่เหล็ก (MAGNETIC FIELD) ซึ่งเกิดขึ้นเมื่อมีการจ่ายกระแสให้กับขดลวดของ สเตปมอเตอร์ ทำให้เกิดแรงที่เรียกว่า แรงขับเคลื่อนแม่เหล็ก (MAGNETIC-MOTIVE FORCE หรือ M.M.F.)

วงจรขับสเตปมอเตอร์จะรับสัญญาณอินพุทจากคอนโทรลเลอร์ ประกอบด้วย สัญญาณสเตปพัลส์ (STEP PULSE) และสัญญาณกำหนดทิศทางการเคลื่อนที่ (DIRECTION SIGNAL) ส่วนมากแล้วการใช้งานสเตปมอเตอร์ในการใช้งานอุตสาหกรรมจะใช้การขับแบบครึ่งสเตป (HALF-STEP MODE) ทำให้สเตปมอเตอร์มีจำนวนสเตปต่อรอบเพิ่มขึ้นจาก 200 สเตปต่อรอบไปเป็น 400 สเตปต่อรอบ

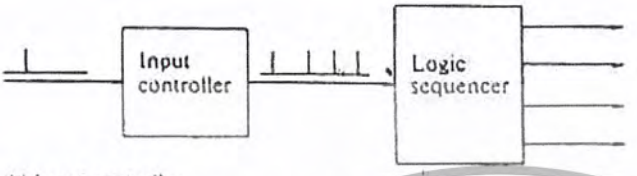
- 3.3.1 ระบบขับ (DRIVE SYSTEM)

ในรูปที่ 3.15 ได้แสดงถึงพื้นฐานของระบบขับของสเตปมอเตอร์ ในรูปที่ 3.15 (a) แสดงถึงส่วนของลอจิกซีเควนเซอร์ (LOGIC SEQUENCER), ไดรเวอร์ (DRIVER) และสเตปมอเตอร์ชนิด 4 เฟส (FOUR-PHASE STEPMOTOR) เมื่อมีสัญญาณสเตปพัลส์เข้ามาทางลอจิกซีเควนเซอร์ จะทำให้เกิดสัญญาณควบคุมออกจากลอจิกซีเควนเซอร์ ในลักษณะต่าง ๆ กันดังรูป เพื่อทำให้ไดรเวอร์ทำการขยายสัญญาณและส่งไปควบคุมให้สเตปมอเตอร์หมุนไปตามทิศทางที่ต้องการ ทิศทางของการหมุนถูกกำหนดโดยสถานะของสัญญาณกำหนดทิศทาง (DIRECTION SIGNAL) เช่นถ้ามีสถานะที่เข้ามาเป็นสถานะสูง (HIGH LEVEL) ก็ให้หมุนไปตามทิศทางตามเข็มนาฬิกาเป็นต้น ในบางครั้งเมื่อมอเตอร์ต้องการจำนวนสเตปหลาย ๆ สเตปในการเคลื่อนที่ จึงจำเป็นต้องมีอินพุทคอนโทรลเลอร์ (INPUT CONTROLLOR) ในการจัดเตรียมสัญญาณที่เข้ามาให้เป็นสัญญาณที่ลอจิกซีเควนเซอร์ สามารถทำให้มอเตอร์เคลื่อนที่ได้ ดังรูปที่ 3.15 (b) แสดงให้เห็นว่าเมื่อมีสัญญาณอินพุทเข้ามา 1 อินพุทหน่วยอินพุทคอนโทรลเลอร์จะกำเนิดสัญญาณพัลส์จำนวน 4 พัลส์ให้กับหน่วยลอจิกซีเควนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) From logic sequencer to motor



(b) Input controller

รูปที่ 3.15 แสดงบล็อกไดอะแกรมของระบบขับสเตปมอเตอร์

- 3.3.2 ลอจิกซีเควนเซอร์ (LOGIC SEQUENCERS)

ลอจิกซีเควนเซอร์ คือวงจรลอจิกที่ควบคุมการกระตุ้นให้เกิดอำนาจแม่เหล็กจากการจ่ายกระแสไฟฟ้าให้กับขดลวดของสเตปมอเตอร์ โดยทั่วไปแล้วลอจิกซีเควนเซอร์ ประกอบด้วยชิฟรีจิสเตอร์ (SHIFT REGISTER) และลอจิกเกท (LOGIC GATE) ตารางที่ 3.1 แสดงถึงคุณสมบัติของลอจิกเกทต่าง ๆ

- 3.3.2.1 การกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟส (TWO-PHASE ON EXITATION FOR A FOUR-PHASE MOTOR)

เป็นวงจรลอจิกซีเควนเซอร์ อีกแบบหนึ่งที่สามารถสร้างได้อย่างง่ายๆด้วย JK FFS เพียง 2 ตัว ดังรูปที่ 3.16 จากรูปเป็นแบบทิศทางเดียว โดยที่

Q1 = PHASE 1

$\overline{Q}$ = PHASE 2

Q2 = PHASE 3

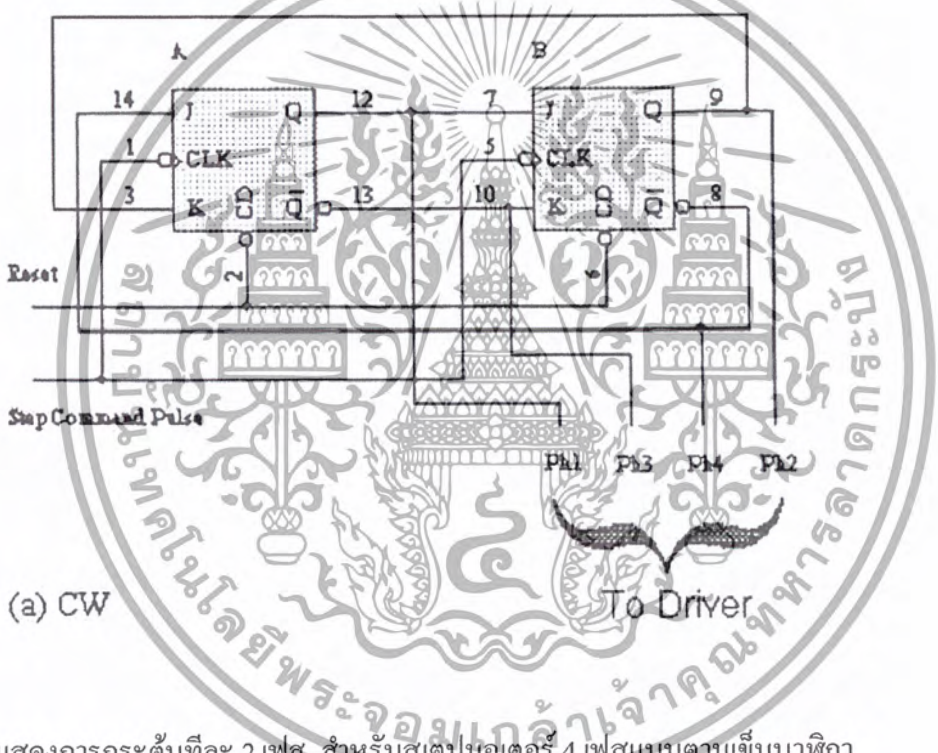
$\overline{Q2}$ = PHASE 4

เมื่อ Q1 มีสภาวะลอจิกเป็นสูง ขดลวดของ PHASE 1 จะถูกกระตุ้น และถ้าในทางกลับกัน PHASE 1 จะไม่ถูกกระตุ้น เมื่อเปรียบเทียบตารางระหว่างรูป 3.16 (a) และ (b) ปรากฏว่ามีความแตกต่างกันในลำดับของการกระตุ้น ในรูป 3.16 (a) นั้นจะทำให้สเตปมอเตอร์หมุนตามเข็มนาฬิกา ส่วนรูป 3.16 (a) นั้นหมุนทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CW	R	1	2	3	4	5	6	...
Ph1,Q1	0	1	1	0	0	1	1	...
Ph2,Q2	0	0	1	1	0	0	1	...
$\overline{\text{Ph3,Q1}}$	1	0	0	1	1	0	0	...
$\overline{\text{Ph4,Q2}}$	1	1	0	0	1	1	0	...

ตารางแสดงการกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟสแบบตามเข็มนาฬิกา

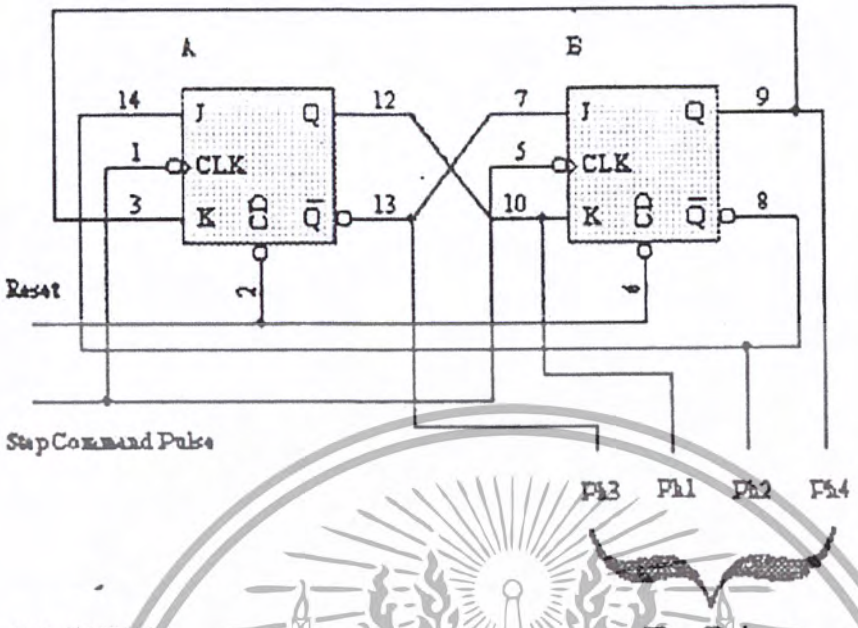


รูปแสดงการกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟสแบบตามเข็มนาฬิกา

CCW	R	1	2	3	4	5	6	...
Ph1,Q1	0	0	1	1	0	0	1	...
Ph2,Q2	0	1	1	0	0	1	1	...
$\overline{\text{Ph3,Q1}}$	1	1	0	0	1	1	0	...
$\overline{\text{Ph4,Q2}}$	1	0	0	1	1	0	0	...

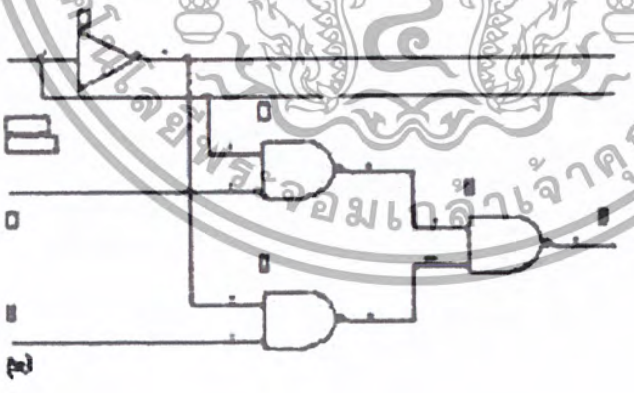
ตารางแสดงการกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟสแบบทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟสแบบทวนเข็มนาฬิกา

รูปที่ 3.16 แสดงการกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟส

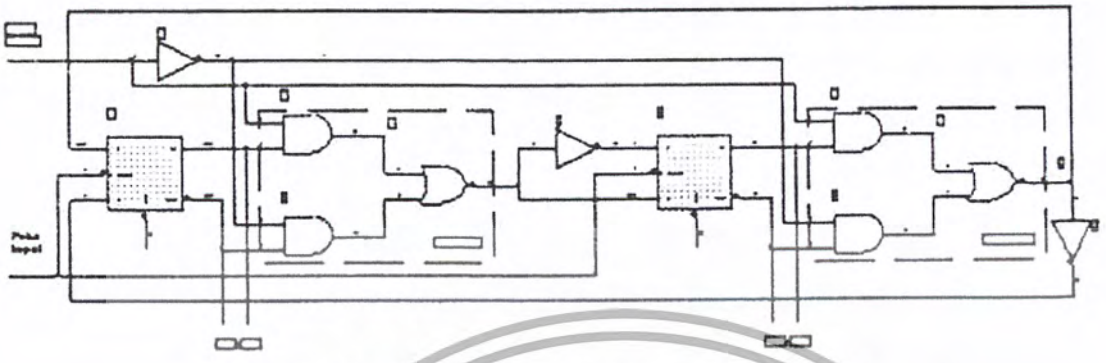


รูปที่ 3.17 แสดงลอจิกต่าง ๆ ที่ทำหน้าที่ควบคุมทิศทางการหมุน

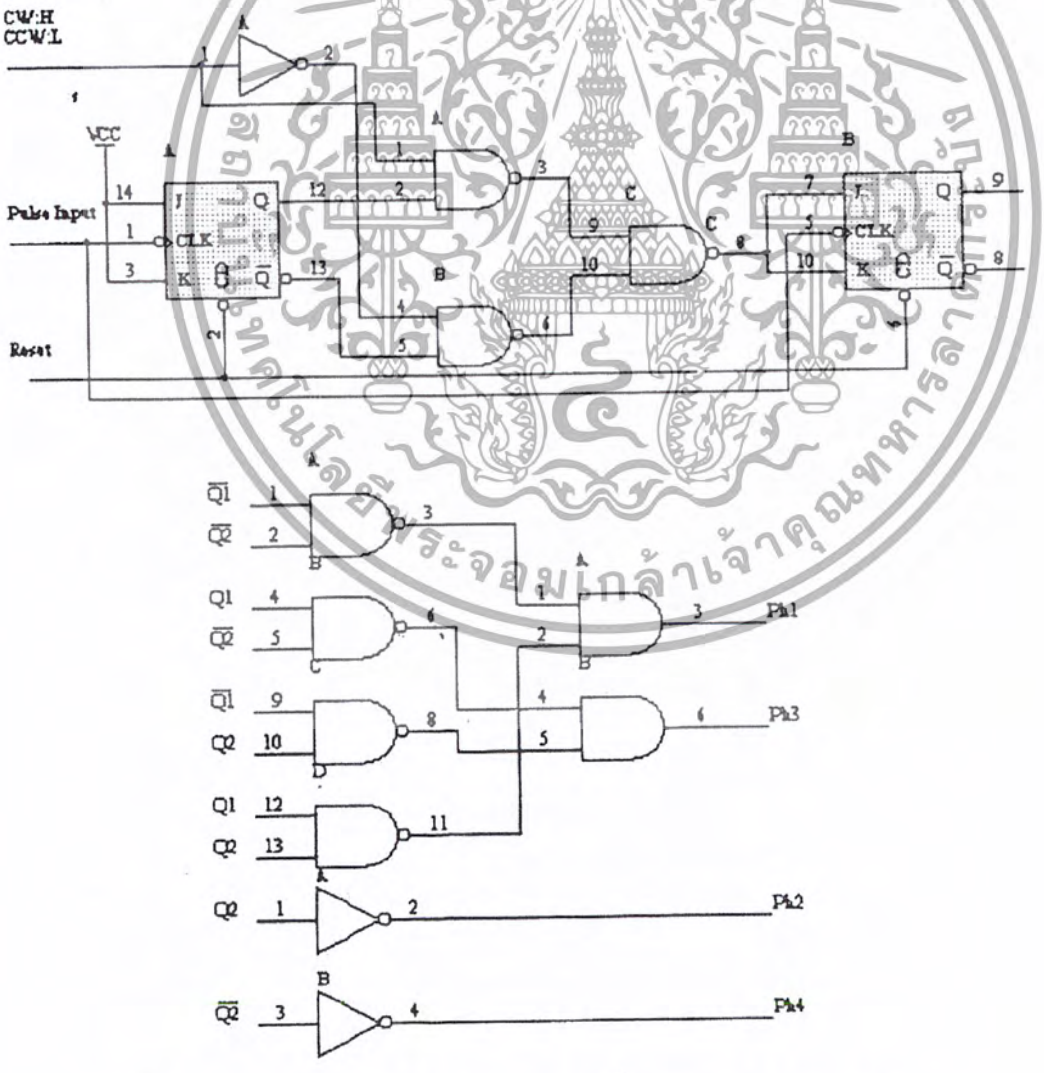
ในรูปที่ 3.17 แสดงถึงวงจรลอจิกที่ทำหน้าที่สับเปลี่ยนทิศทางการหมุนของสเตปมอเตอร์ซึ่งประกอบขึ้นด้วยลอจิกต่าง ๆ มารวมกัน ในวงจร 3.17 (a) ถ้าสมมติว่าสัญญาณควบคุมทิศทางมีสภาวะเป็นสูงจะทำให้เอาท์พุท C มีสภาวะที่เข้ามาทางอินพุท A ในทางกลับกันถ้าสัญญาณควบคุมทิศทางมีสภาวะเป็นต่ำจะทำให้เอาท์พุท $\overline{C}$ มีสภาวะเดียวกันกับสภาวะที่เข้ามาทางอินพุท B ในวงจร 3.17 (b) เอาท์พุท $\overline{C}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเท่ากับ A เมื่อสัญญาณควบคุมทิศทางมีสภาวะเป็นสูง ในทางกลับกันเอาท์พุท C จะเท่ากับ B ส่วนรูปที่ 3.18 แสดงถึงวงจรซีเควนเซอร์ 2 ทิศทางแบบกระตุ้นทีละ 2 เฟส สำหรับสเตปมอเตอร์ 4 เฟส อีกตัวอย่างหนึ่งของการกระตุ้นแบบนี้แสดงในรูปที่ 3.19



รูปที่ 3.18 แสดงวงจรซีเควนเซอร์ 2 ทิศทาง



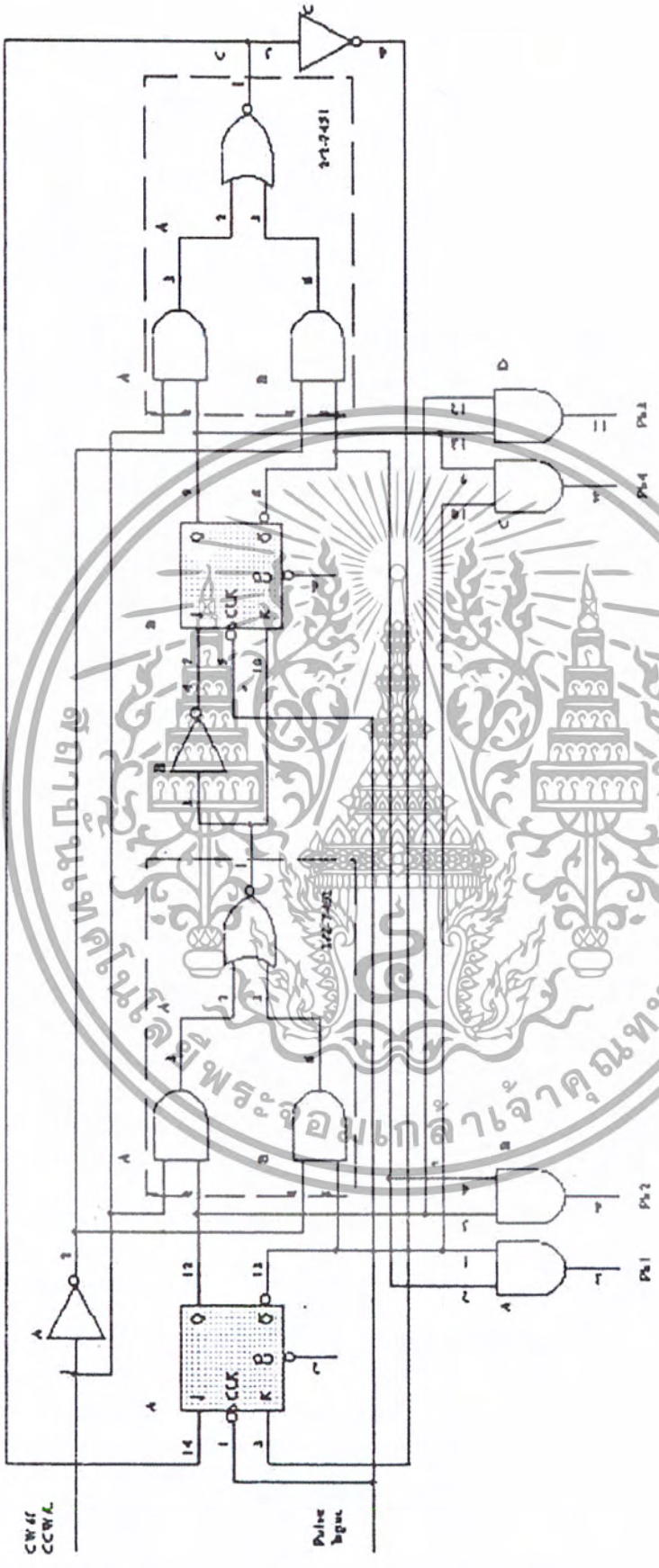
รูปที่ 3.19 แสดงก้ารกระตุ้นอีกรูปแบบหนึ่งของการกระตุ้นทีละ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.3.2.2 วงจรซีเควนเซอร์เฟสเดียวสำหรับมอเตอร์ 4 เฟส (SINGLE PHASE ON SEQUENCER FOR FOUR-PHASE MOTOR)

การกระตุ้นแบบเฟสเดียวสามารถทำได้โดยเพิ่ม AND เกท 4 ตัวเข้าที่เอาท์พุทของซีเควนเซอร์แบบกระตุ้นทีละ 2 เฟส ดังที่ได้แสดงในรูปที่ 3.20 และ รูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แสดงลอจิกซีเคานเซอร์เฟสเดียว 2 ทิศทางสำหรับสเตปมอเตอร์ 4 เฟส

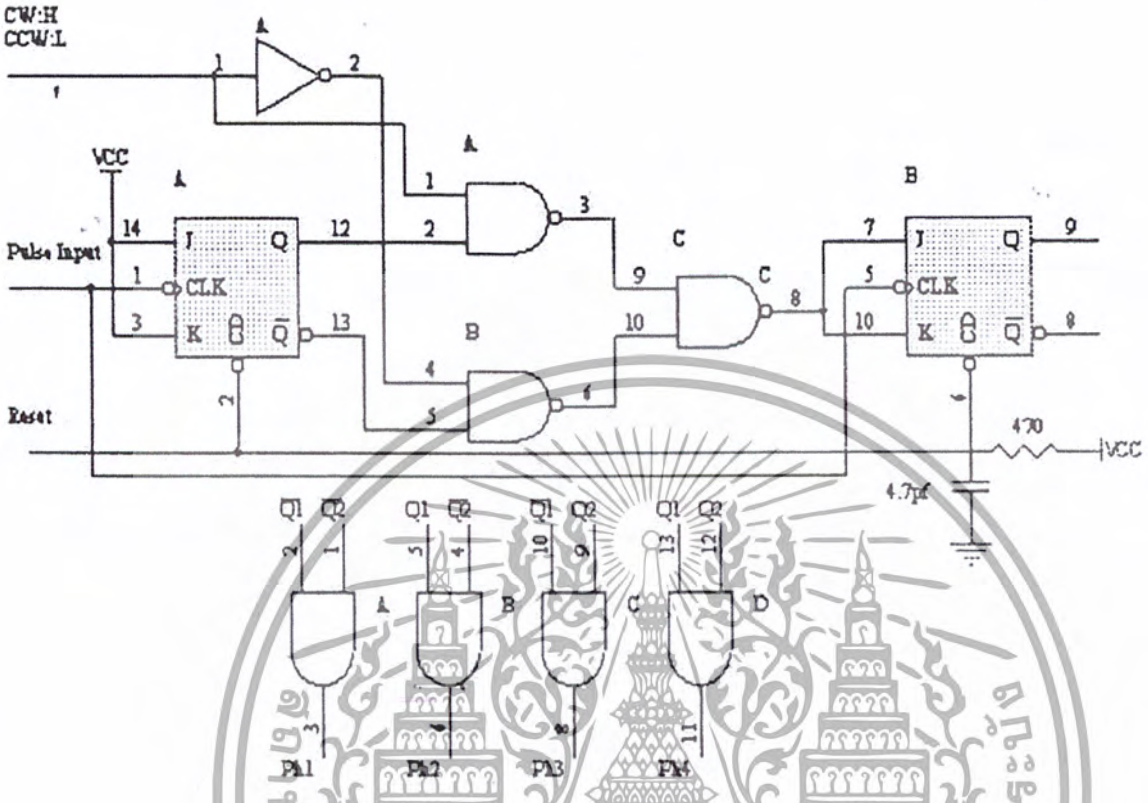
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	CW	R	1	2	3	4	5	6	...
	Q1	0	1	0	1	0	1	0	...
	$\overline{Q1}$	1	0	1	0	1	0	1	...
	Q2	0	0	1	1	0	0	1	...
	$\overline{Q2}$	1	1	0	0	1	1	0	...
Ph1	Q1.$\overline{Q2}$	1	0	0	0	1	0	0	...
Ph2	Q1.$\overline{Q2}$	0	1	0	0	0	1	0	...
Ph3	$\overline{Q1}$.Q2	0	0	1	0	0	0	1	...
Ph4	Q1.Q2	0	0	0	1	0	0	0	...

	CCW	R	1	2	3	4	5	6	...
	Q1	0	1	0	1	0	1	0	...
	$\overline{Q1}$	1	0	1	0	1	0	1	...
	Q2	0	1	1	0	0	1	1	...
	$\overline{Q2}$	1	0	0	1	1	0	0	...
Ph1	Q1.$\overline{Q2}$	1	0	0	0	1	0	0	...
Ph2	Q1.$\overline{Q2}$	0	0	0	1	0	0	0	...
Ph3	$\overline{Q1}$.Q2	0	0	1	0	0	0	0	...
Ph4	Q1.Q2	0	1	0	1	0	1	0	...

ตารางแสดงลอจิกซีเควนเซอร์เฟสเดียว 2 ทิศทางสำหรับสเตปมอเตอร์ 4 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

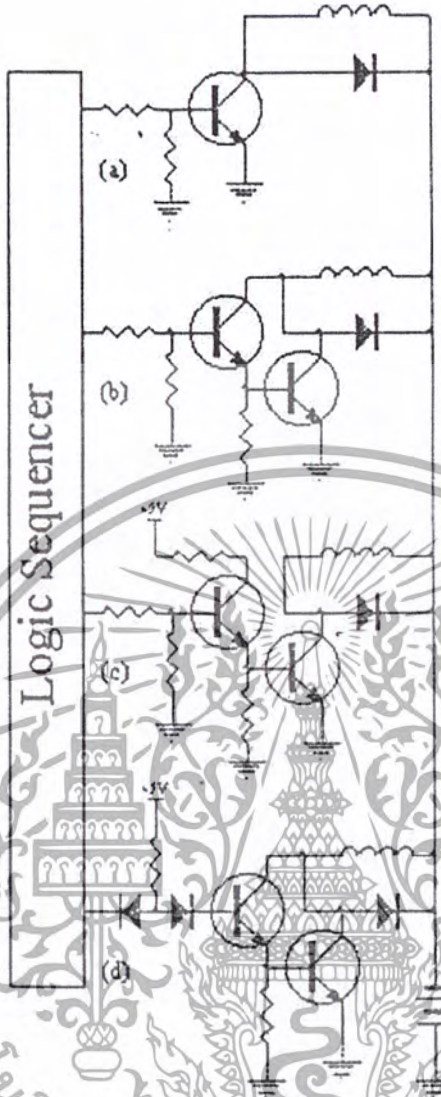


รูปที่ 3.21 แสดงลอจิกซีเควนเซอร์เฟสเดียวสำหรับสเตปมอเตอร์ 4 เฟส

- 3.3.3 การเชื่อมต่อระหว่างวงจรซีเควนเซอร์กับไดรเวอร์ (CONNECTION OF SEQUENCER AND DRIVER)

สัญญาณเอาท์พุทของวงจรลอจิกซีเควนเซอร์ถูกส่งไปยังอินพุทเทอร์มินอลของวงจรขับกำลัง (POWER DRIVER) เพื่อที่จะควบคุมการจ่ายกระแสให้กับขดลวดของสเตปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



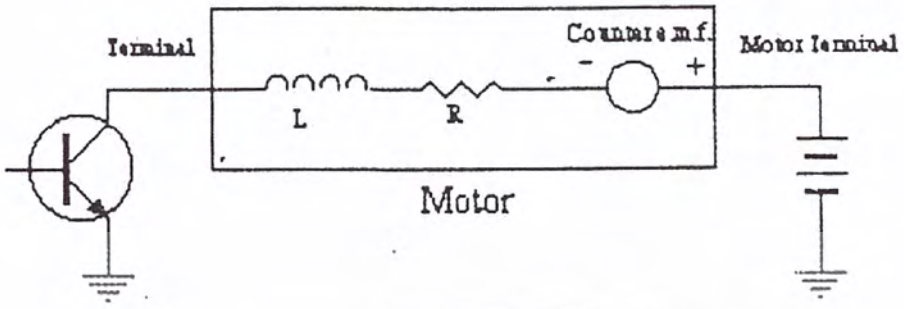
รูปที่ 3.22 แสดงตัวอย่างของการต่อระหว่างซีเควนเซอร์และไดรเวอร์

- 3.3.4 ปัญหาของไดรเวอร์ (PROBLEMS WITH DRIVER)

ขดลวดของสเตปมอเตอร์คือวงจรเหนี่ยวนำตัวหนึ่ง ซึ่งสามารถแสดงได้เป็นการต่อแบบอนุกรมระหว่างความเหนี่ยวนำกับความต้านทานและจะเกิดส่วนของเคาท์เตอร์ อี.เอ็ม.เอฟ (COUNTER E.M.F.) เมื่อ สเตปมอเตอร์มีการหมุน วงจรสมมูลย์ (EQUIVALENT CIRCUIT) ของขดลวดของสเตปมอเตอร์แสดงในรูปที่ 3.23 ในการออกแบบเพาเวอร์ไดรเวอร์ (POWER DRIVER) ต้องคำนึงถึงปัจจัยต่าง ๆ ที่มีผลต่อวงจรและพฤติกรรมของวงจรอย่างน้อยที่สุด ปัจจัยต่าง ๆ ของสเตปมอเตอร์ เพาเวอร์ ทรานซิสเตอร์ และแรงดันที่จ่ายให้กับมอเตอร์ ต้องถูกนำมาพิจารณาในการออกแบบด้วย พารามิเตอร์ต่างๆ ของสเตปมอเตอร์จะแปรผันไปตามความทนนานในการใช้งานและสภาวะแวดล้อมในการใช้งาน เมื่อสเตปมอเตอร์ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบให้ส่งผ่านกำลังสูง ๆ ทั้ง ๆ ที่มีขนาดเล็กจะทำให้อุณหภูมิสูงถึง 100° และค่าความต้านทานของขดลวดจะเพิ่มขึ้น 20-25%



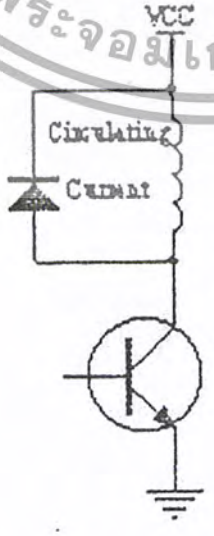
รูปที่ 3.23 แสดงวงจรสมมูลย์ของขดลวดของสเตปมอเตอร์

- 3.3.5 ซับเพรสเซอร์ (SUPPRESSOR)

เมื่อทรานซิสเตอร์ในรูปที่ 3.23 หยุดนำกระแสจะเกิดแรงดันสูง (HIGH VOLT) ที่สร้างโดย $L\left(\frac{di}{dt}\right)$ และแรงดันจำนวนนี้อาจจะทำให้ทรานซิสเตอร์เสียหายได้ เรามีหลายวิธีที่จะกำจัด (SUPPRESSING) แรงดันสไปด์ (SPIKE VOLTAGE) และป้องกันทรานซิสเตอร์ ดังต่อไปนี้

- 3.3.5.1 ไดโอดสัพเพรสเซอร์ (DIODE SUPPRESSOR)

เมื่อใส่ไดโอดขนานเข้าระหว่างขั้วทั้งสองของขดลวด ดังรูปที่ 3.24 เมื่อ ทรานซิสเตอร์หยุดนำกระแสไหลวนระหว่างขดลวดกับไดโอด ทำให้แรงดันสไปด์มีขนาดเล็กลงไปเรื่อยๆ การต่อแบบนี้มีข้อดีคือสามารถสร้างได้ง่ายแต่ก็มีข้อเสียคือระยะเวลาในการเกิดกระแสวนมีระยะเวลานานและก่อให้เกิดเบรกกิ้ง ทอร์ค (BRAKING TORQUE)



รูปที่ 3.24 แสดงไดโอดสัพเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.3.5.2 ไดโอด/รีจิสเตอร์ ซัพเพรสเซอร์(DIODE/REGISTER SUPPRESSOR)

ความต้านทานจะถูกต่ออนุกรมเข้ากับไดโอด ดังรูปที่ 3.25 เพื่อให้กระแสวนหมดไปอย่างรวดเร็ว วิธีนี้จะทำให้แรงดันที่ V_{CE} มีมีค่าดังสมการต่อไปนี้

$$V_{CE} = E + IR_S + V_{DF}$$

เมื่อ E = แรงดันของแหล่งจ่ายไฟ

I = กระแสที่ไหลก่อนการหยุดนำกระแส

RS = ค่าความต้านทานของตัวต้านทานซัพเพรสเซอร์

VDF = แรงดันไบอัสตรงของไดโอด

รูปที่ 3.25 แสดงไดโอด/รีจิสเตอร์ ซัพเพรสเซอร์

เมื่อความต้านทานยิ่งมากก็จะยิ่งทำให้กระแสวนลดลงอย่างรวดเร็วตามไปด้วยแต่ก็จะทำให้แรงดันที่ป้อนให้กับ คอลเลคเตอร์สูงขึ้นตามไปด้วย

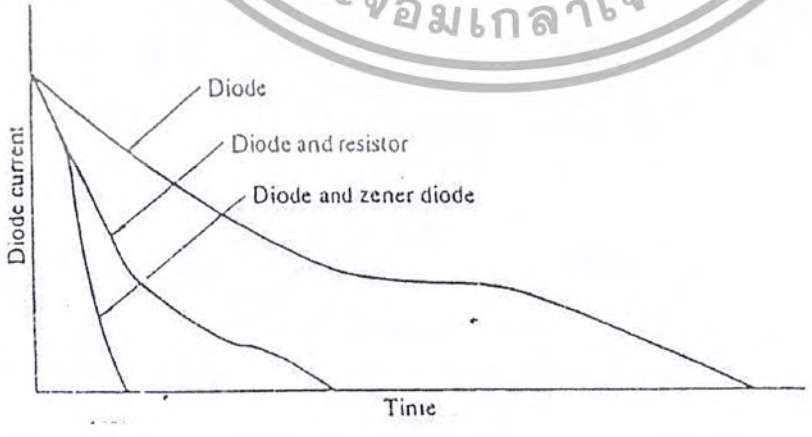
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซีเนอร์ไดโอดซัพเพรสเซอร์ (ZENOR DIODE SUPPRESSOR)

ซีเนอร์ไดโอดจะถูกต่ออนุกรมเข้ากับไดโอดธรรมดา ดังรูปที่ 3.26 เมื่อเปรียบเทียบกับ 2 วิธีที่กล่าวมาแล้วนั้น วิธีนี้จะสามารถกำจัดกระแสวนได้รวดเร็วกว่า ดังรูปที่ 3.27 และวิธีนี้ยังมีข้อดีอีกคือเมื่อทรานซิสเตอร์หยุดนำกระแส แรงดันที่จ่ายให้กับคอลเลคเตอร์จะเท่ากับแรงดันจากแหล่งจ่ายบวก

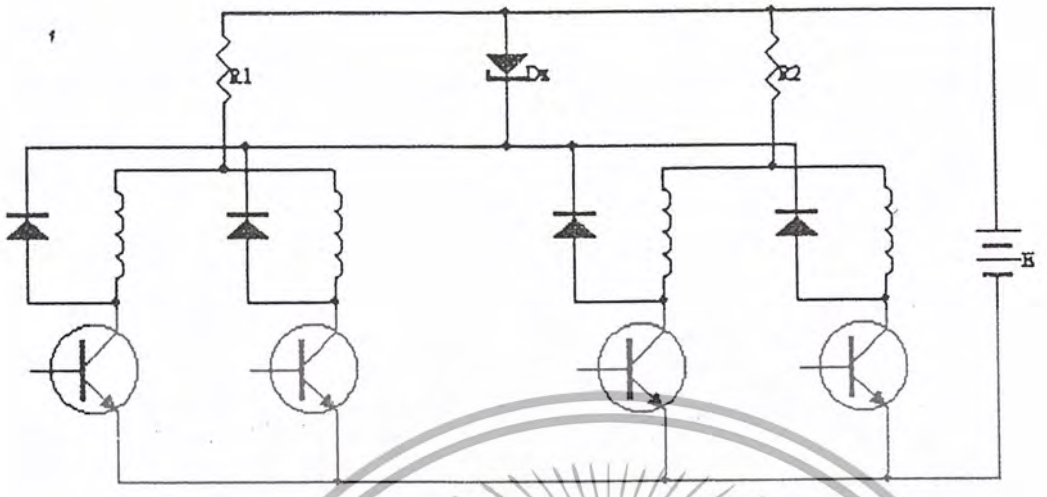
กับแรงดันที่ตกคร่อมซีเนอร์ไดโอด โดยจะไม่ขึ้นกับกระแสที่ไหลผ่านขดลวดก่นที่จะหยุดนำกระแส ทำให้ง่ายต่อการกำหนดแรงดันสูงสุดที่ตกคร่อมคอลเลคเตอร์

รูปที่ 3.26 แสดงซีเนอร์ไดโอดซัพเพรสเซอร์



รูปที่ 3.27 แสดงการเปรียบเทียบไดโอดชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 แสดงวงจรไดรเวอร์ที่ใช้ซีเนอร์ไดโอดซัพเพรสชั่น

รูปที่ 3.28 แสดงถึงวงจรไดรเวอร์ที่ใช้ซีเนอร์ไดโอด ซัพเพรสชั่น (ZENOR DIODE SUPPRESS) สำหรับสเตปมอเตอร์ 4 เฟส ความต้านทาน R1 และ R2 ใช้สำหรับช่วยสร้างกระแสกระตุ้นให้เกิดเร็วขึ้น

- 3.3.6 การปรับปรุงการสร้างกระแส (IMPROVEMENT OF CURENT BUILT-UP)

เมื่อทรานซิสเตอร์นำกระแสตามการกระตุ้นของสัญญาณอินพุท แหล่งจ่ายไฟก็กลายเป็นสิ่งที่มีผลกระทบต่อค่าความต้านทานเหนี่ยวนำ ก่อนที่มันจะทำการขับเคลื่อนให้มอเตอร์หมุนต่อไปเนื่องจากค่าความต้านทานเหนี่ยวนำมีผลตรงข้ามกับกระแสที่ไหลเข้ามาและเมื่อความถี่ในการสวิทช์เพิ่มขึ้น บิวด์อัพไทม์(BUILD-UP TIME) ก็จะเพิ่มขึ้น จะทำให้ทอร์คลดลงและทำให้เกิดการตอบสนองช้าลงด้วย เรามีหลายวิธีที่จะทำให้เวลาในการบิวด์อัพไทม์สั้นลงและค่าทอร์คที่ความเร็วสูงมีค่าสูงขึ้นดังต่อไปนี้

- 3.3.6.1 แบบความต้านทานอนุกรม (SERIES RESISTANCE)

เป็นวิธีที่ประหยัดที่สุดและง่ายที่สุด สามารถทำได้โดยเพิ่มความต้านทานต่ออนุกรมเข้ากับขดลวดของสเตปมอเตอร์ดังรูปที่ 3.29 ความต่างศักย์ E จะถูกจ่ายให้กับขดลวดภายใต้สภาวะคงที่(STEADY STATE CONDITION) ค่าเวลาคงตัว (TIME CONSTANT) ของวงจรจะลดลงจาก $\frac{L}{R_w}$ กลายเป็น $\frac{L}{R_C + R_W}$ กระแสจะไหลผ่านขดลวดแต่ละเฟสเมื่อเมื่อสวิทช์ซึ่งทรานซิสเตอร์อยู่ในสภาวะอิ่มตัว เนื่องจากกระแสที่ไบอัสทางด้านเบส ในสภาวะเช่นนี้แรงดัน DC จากแหล่งจ่ายไฟจะไหลผ่านตัวต้านทาน

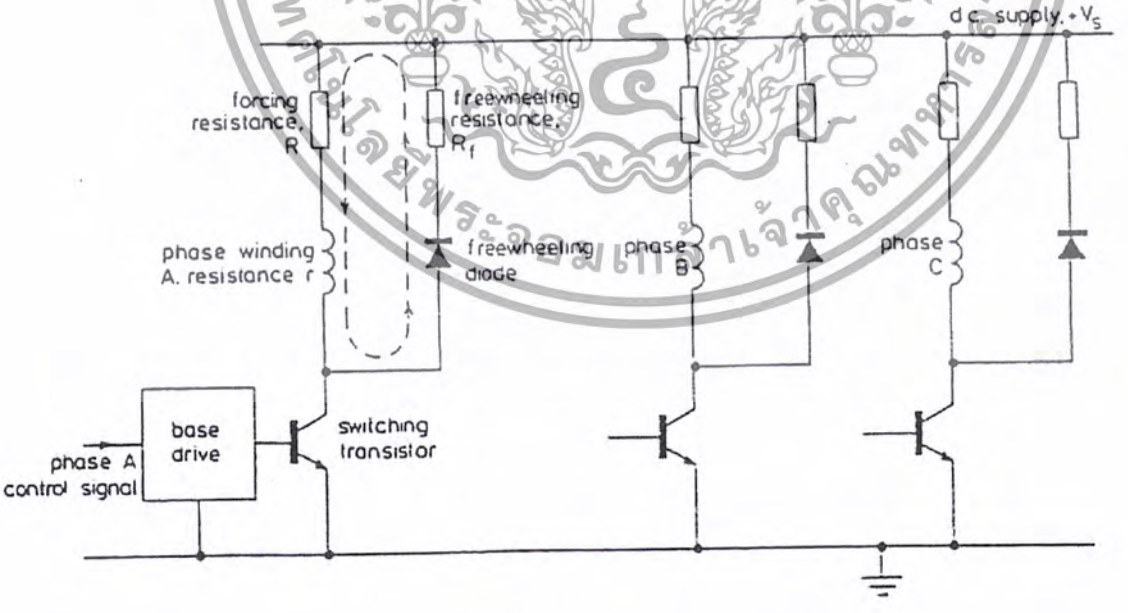
ลิขสิทธิ์เป็นของมหาวิทยาลัยเชียงใหม่ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านขดลวดของสเตปมอเตอร์และไหลผ่านทรานซิสเตอร์ เนื่องจากแรงดันตกคร่อมทรานซิสเตอร์ในสภาวะอิ่มตัวมีค่าน้อย(ประมาณ 0.1 โวลต์) แรงดันจากแหล่งจ่ายไฟทั้งหมดจะทำให้เกิดกระแสไหลผ่านขดลวด โดยมีความสัมพันธ์ กับผลรวมของความต้านทานของขดลวด(r) ดังนี้

$$V_s = I(r + R)$$

ความเหนี่ยวนำของขดลวดยังทำให้กระแสไม่หยุดไหลในทันทีทันใดที่ทรานซิสเตอร์อยู่ในสถานะปิด ทำให้เกิดแรงดันเหนี่ยวนำตกคร่อมคอลเลคเตอร์และอิมิตเตอร์ซึ่งเป็นสาเหตุทำให้วงจรขับเสียหาย ปัญหานี้แก้ไขได้โดยการต่อ ฟรีวิลลิ่งไดโอด(FREEWHEELING DIODE) และ ฟรีวิลลิ่งรีซิสเตอร์(FREEHEELING RESISTANCE) เพื่อเป็นทางผ่านของกระแสแทน

แม้ว่าการใช้ความต้านทานต่ออนุกรมกับขดลวดจะเป็นวิธีที่ง่ายที่สุดก็จาม แต่ก็มีข้อเสียคือ มีการสูญเสียกำลังงานในตัวต้านทานสูง ถ้าความต้านทานของขดลวดของสเตปมอเตอร์ชนิด 4 เฟส มีค่าเท่ากับ 1.5 โอห์ม และกระแสที่ไหลผ่านมีค่าเท่ากับ 4 แอมป์ และใช้แรงดันมีค่าเท่ากับ 24 โวลท์ เมื่อเพิ่มความต้านทานที่มีค่าเท่ากับ 4.5 โอห์ม เข้าที่แต่ละเฟสของสเตปมอเตอร์ กำลังสูญเสียโดยประมาณจะมีค่าเท่ากับ $4.5 \times 4^{2} = 72$ W ในการขับแบบกระตุ้นเฟสเดียว และจะเพิ่มขึ้นเป็นสองเท่าเมื่อเป็นการขับแบบกระตุ้นทีละสองเฟส

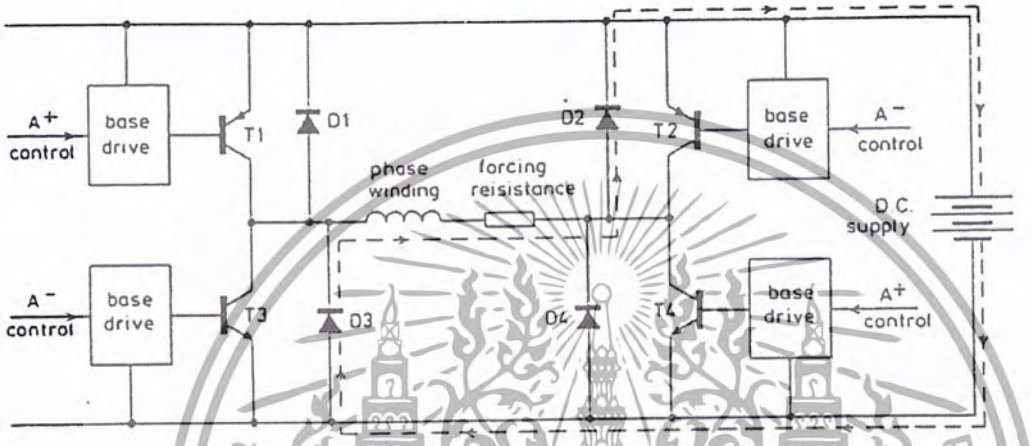


รูปที่ 3.29 แสดงการปรับปรุงการสร้างกระแสแบบความต้านทานอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.3.6.2 แบบแหล่งจ่ายคู่ (DUAL VOLTAGE)

เพื่อที่จะลดกำลังสูญเสียในการขับและเพิ่มประสิทธิภาพของสเตปมอเตอร์ จึงนำวิธีการใช้แหล่งจ่ายคู่มาประยุกต์ใช้งาน โดยที่วงจรนี้จะต่อทรานซิสเตอร์เป็นแบบบริดจ์ วงจรบริดจ์หนึ่งชุดจะขับมอเตอร์ได้หนึ่งเฟส เหมาะที่ใช้กับสเตปมอเตอร์แบบแม่เหล็กถาวร หรือแบบไฮบริดจ์ ดังแสดงในรูปที่ 3.30



รูปที่ 3.30 แสดงการปรับปรุงการสร้างกระแสแบบแหล่งจ่ายคู่

ทรานซิสเตอร์จะผลัดกันทำงานทีละคู่ตามทิศทางกระแสที่ต้องการ สำหรับการกระตุ้นขดลวดในทิศบวก ทรานซิสเตอร์ T1 และ T4 จะทำงาน ทำให้เกิดทางเดินของกระแสจากแหล่งจ่ายไฟ ไหลผ่านทรานซิสเตอร์ T1 จากนั้นไหลผ่านขดลวดของมอเตอร์ผ่านตัวต้านทานแล้วจึงไหลเข้าทรานซิสเตอร์ตัว T4 กลับเข้าสู่แหล่งจ่ายไฟ ในทางกลับกันในกรณีการจ่ายกระแสในทิศลบ(ทิศตรงกันข้าม) ทรานซิสเตอร์ T2 และ T3 จะทำงาน เพื่อให้กระแสไหลผ่านขดลวดในทิศทางตรงกันข้าม

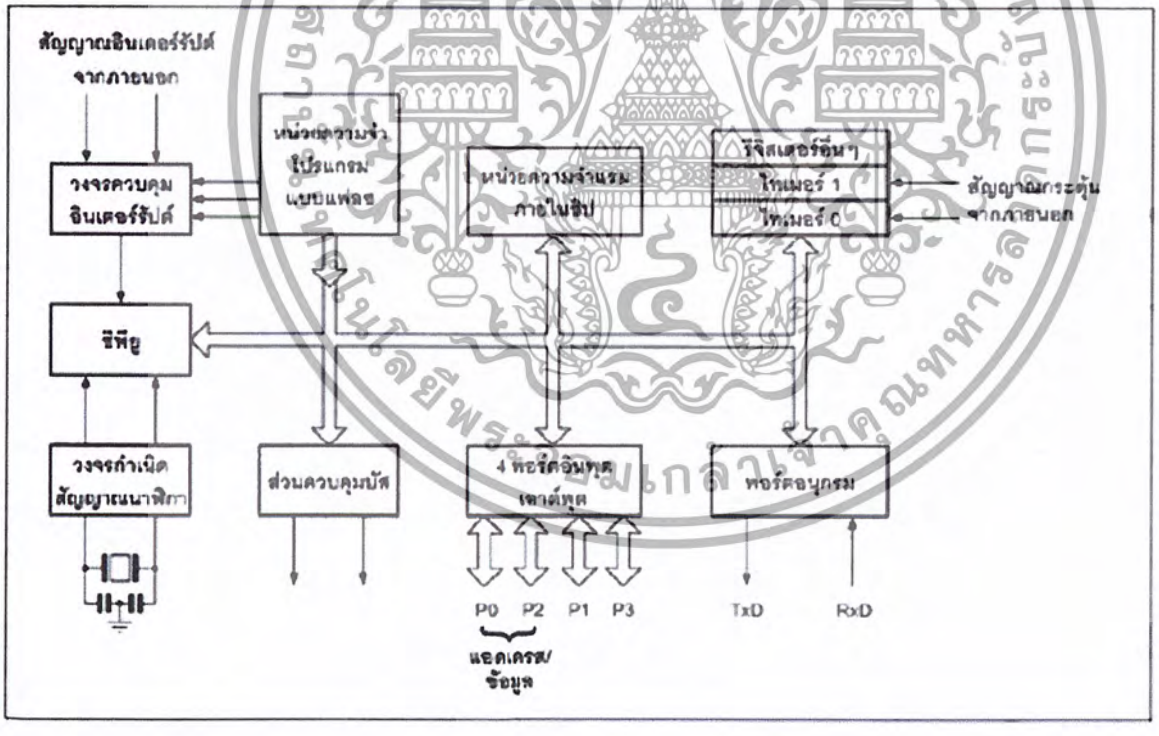
จากรูปจะเห็นว่ามีไดโอด 4 ตัว ต่อขนานกับทรานซิสเตอร์แต่ละตัว จุประสงค์เพื่อให้เกิดทางไหลของกระแสฟรีวิลลิ่ง(FREEWHEELING CURRENT) บริเวณไดโอด D2 และ D3 จะเป็นทางผ่านของกระแสหลังจากที่ทรานซิสเตอร์ T1 และ T4 หยุดทำงาน ส่วน D1 และ D4 จะมีกระแสไหลผ่านขณะที่ทรานซิสเตอร์ T2 และ T3 หยุดทำงาน กระแสฟรีวิลลิ่งในวงจรขับแบบนี้จะสิ้นสุดลงเร็วกว่าแบบความต้านทานอนุกรม จึงไม่จำเป็นต้องต่อฟรีวิลลิ่งรีซิสแตนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โครงสร้างและหลักการทำงานของไมโครคอนโทรลเลอร์

4.1 ไมโครคอนโทรลเลอร์ MCS-51 (AT89C51)

ไมโครคอนโทรลเลอร์ MCS-51 มีคุณสมบัติเป็นไมโครคอนโทรลเลอร์ที่ใช้ CPU ขนาด 8 บิต มีหน่วยความจำส่วนที่โปรแกรมได้ที่สามารถเขียนและลบได้พันครั้ง มีหน่วยความจำพื้นฐานแบบแรม ขาพอร์ตที่ใช้ในการติดต่อเป็นแบบ bidirectional ที่สามารถใช้งานได้ทั้งอินพุตและเอาท์พุท มีส่วนวงจรสื่อสารที่ใช้อนุกรมแบบฟูลดูเพล็กซ์ มีส่วนไทเมอร์เคาท์เตอร์ 16 บิต สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงถึง 64 กิโลไบต์ มีวงจรกำเนิดสัญญาณนาฬิกา มีวงจรสื่อสารอนุกรมแบบ SPI และมีวอตช์ดีอกไทเมอร์อยู่ภายในตัว



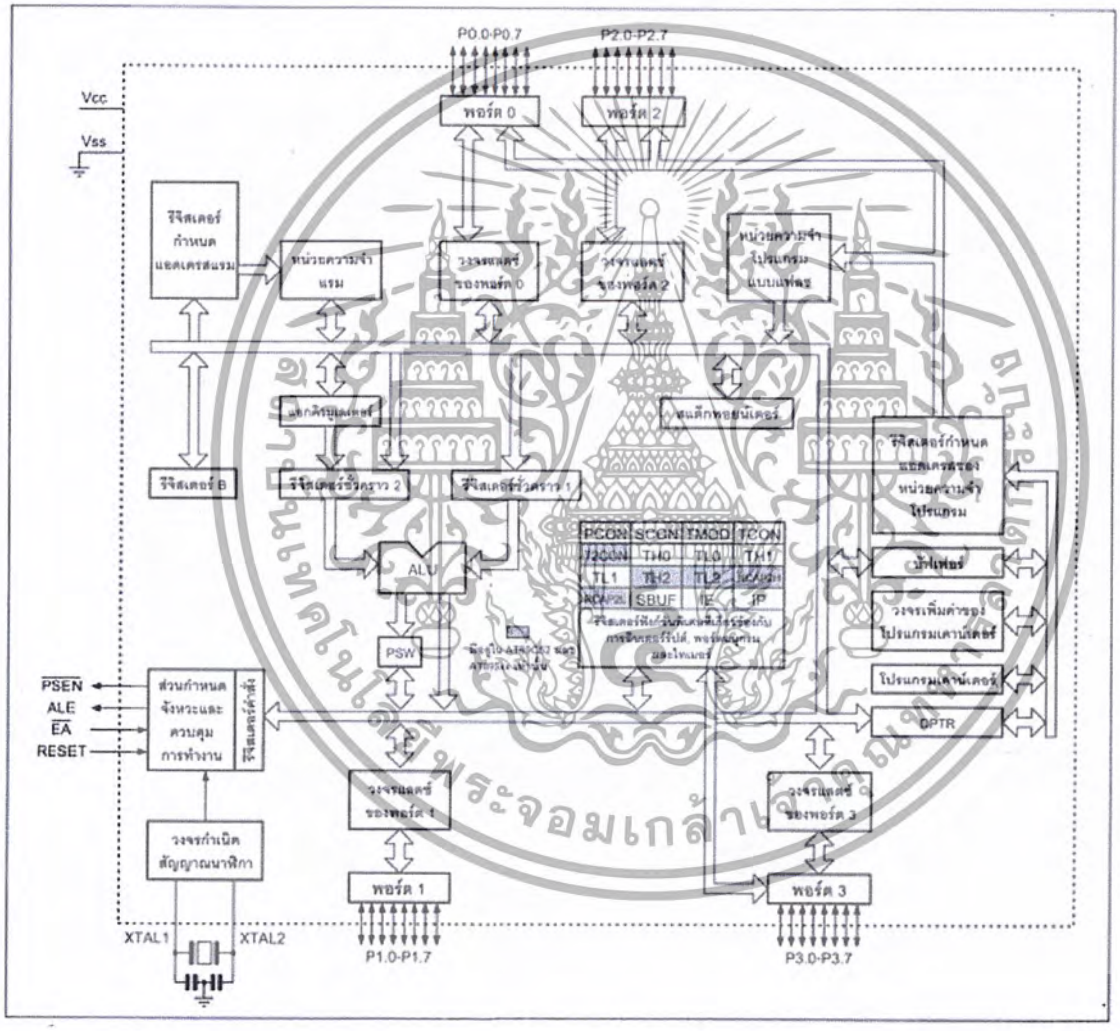
รูปที่ 4-1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 89C51

4.2 ขาพอร์ตของไมโครคอนโทรลเลอร์

- ขา Vcc ใช้สำหรับต่อไฟเลี้ยง + 5 v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ
- ขาพอร์ต 0 (P0.0-P0.7) มี8ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาท์พุต โดยถ้าต้องการให้ขาใดเป็นอินพุตจะทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะ float มีอินพุตอิมพิแดนซ์สูงสามารถใช้งานเป็นขาอินพุตได้ ขาพอร์ตนี้ยังสามรถนำไปใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล(D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

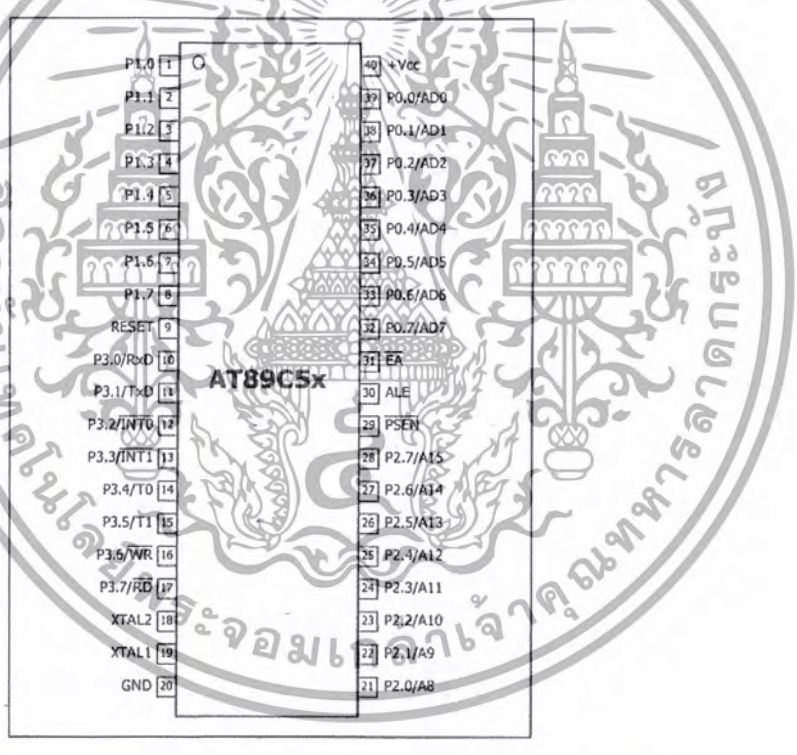


รูปที่ 4-2 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ 89C51

- ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา สามรถใช้งานเป็นพอร์ตอินพุต/เอาท์พุตเหมือนกับพอร์ต P0
- ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา สามรถใช้งานเป็นพอร์ตอินพุต/เอาท์พุตเหมือนกับพอร์ต P0 และยังสามารถใช้เป็นพอร์ตติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก(A8-A15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา สามรถใช้งานเป็นพอร์ตอินพุต/เอาท์พุตเหมือนกับพอร์ต P0 และยังมีขาพิเศษดังนี้
  - P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
  - P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
  - P3.2 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{INT0}$
  - P3.3 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1หรือขา $\overline{INT1}$
  - P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0
  - P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1
  - P3.6 ใช้เป็นขาสัญญาณ $\overline{WR}$
  - P3.7 ใช้เป็นขาสัญญาณ $\overline{RD}$



รูปที่ 4-3 แสดงรูปไมโครคอนโทรลเลอร์และตำแหน่งขามาตรฐาน

- ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยจะต้องป้อนสัญญาณอย่างน้อย 2 แมชชีนไซเคิลในการรีเซต

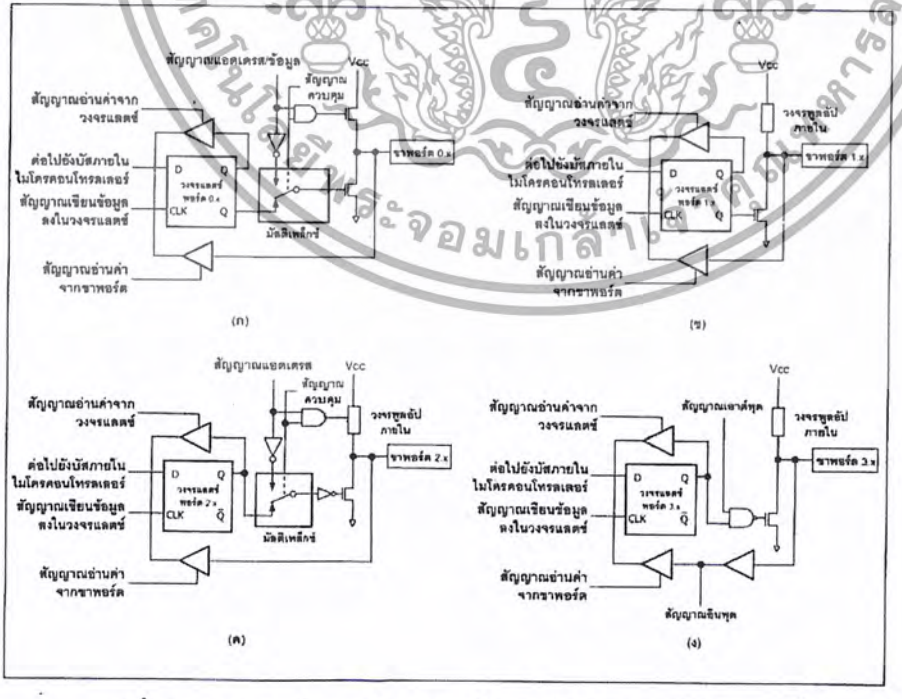
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา ALE/PROG (Address Latch Enable/Program Pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก
- ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อติดต่อกับหน่วยความจำโปรแกรมภายนอก
- ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ โดยถ้าขานี้เป็น “0” เป็นการเลือกให้ติดต่อกับหน่วยความจำภายนอก และถ้าให้ขานี้เป็น “1” เป็นการเลือกให้ตอต่อกับหน่วยความจำภายใน
- ขา XTAL1 และXTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

4.3 โครงสร้างและการทงานของพอร์ต

ไมโครคอนโทรลเลอร์ 89C51 มีพอร์ตทั้งหมด 4 พอร์ต แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ bidirectional ที่สามารถเป็นได้ทั้งอินพุตและเอาท์พุต โดยที่ทุกพอร์ตมีวงจรแลตซ์และวงจรขับตลอดจนบัฟเฟอร์อินพุท

ที่พอร์ต 0 และ 2 จะใช้งานเป็นพอร์ตอินพุทและเอาท์พุตสำหรับทั่วไปและใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3ทั้งหมดและพอร์ต 1 บางขานอกจากจะเป็นขาพอร์ตอินพุท/เอาท์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษด้วย



รูปที่ 4-4 วงจรภายในพอร์ตทุกพอร์ตของไมโครคอนโทรลเลอร์

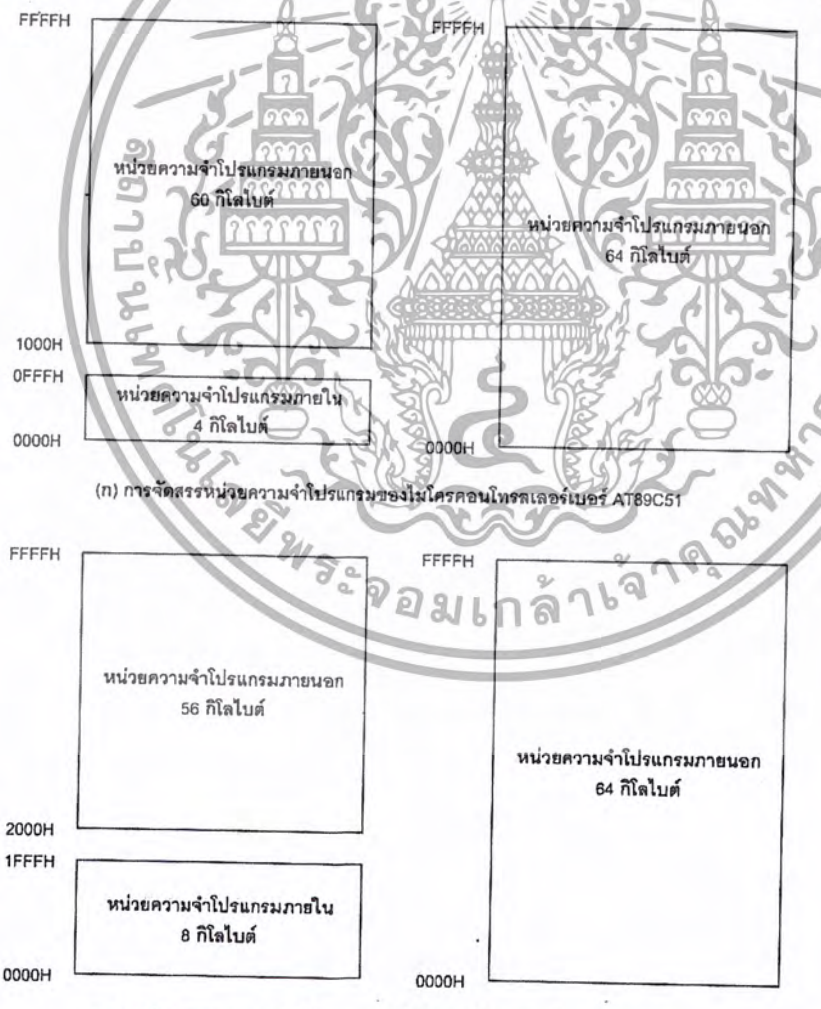
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 หน่วยความจำโปรแกรม (Program memory)

ในรูปที่ 4-5 แสดงการจัดการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ 89C51 ซึ่งสามรถติดต่อหน่วยความจำโปรแกรมได้สูงถึง 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียว โดย 89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ และจะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์

หน่วยความจำโปรแกรมใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ หรือที่เรียกว่า monitor program หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิด EPROM ซึ่งสามารถอ่านได้เพียงอย่างเดียว

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มที่ 0000H เมื่อซีพียูได้รับการรีเซต ให้เริ่มต้นทำงานจะต้องมาเริ่มที่แอดเดรส 0000H เสมอ โดยจะมีพื้นที่ส่วนที่ใช้ในการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย



รูปที่ 4-5 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พื้นที่สำหรับการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H
- พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH
- พื้นที่สำหรับการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H
- พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH
- พื้นที่สำหรับการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H
- พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

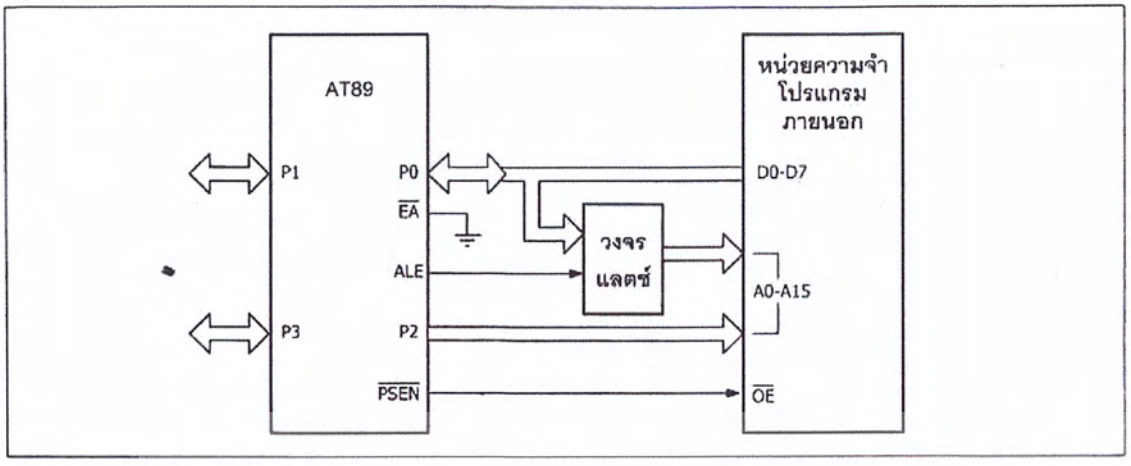
กรณีที่มีการต่อกับหน่วยความจำภายนอกด้วยจะกำหนดให้แอดเดรสของหน่วยความจำภายนอกต่อจากแอดเดรสสุดท้ายของหน่วยความจำภายใน โดยที่หน่วยความจำภายในมีแอดเดรสอยู่ที่ 0000H-0FFFH เมื่อมีการต่อกับหน่วยความจำภายนอกก็จะมีแอดเดรสอยู่ที่ 1000H-FFFFH

การต่อหน่วยความจำภายนอกแสดงดังรูปที่ 2-6 ขาพอร์ต P0.0-P0.7 ใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตช์(74HC573) และใช้สัญญาณ ALE และ PS EN ในการเลือกใช้งานขา P0.0-P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรส ในขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตเพียง 16 บิต คือ ที่ขาพอร์ต P1.0-P1.7 และ P3.0-P3.7

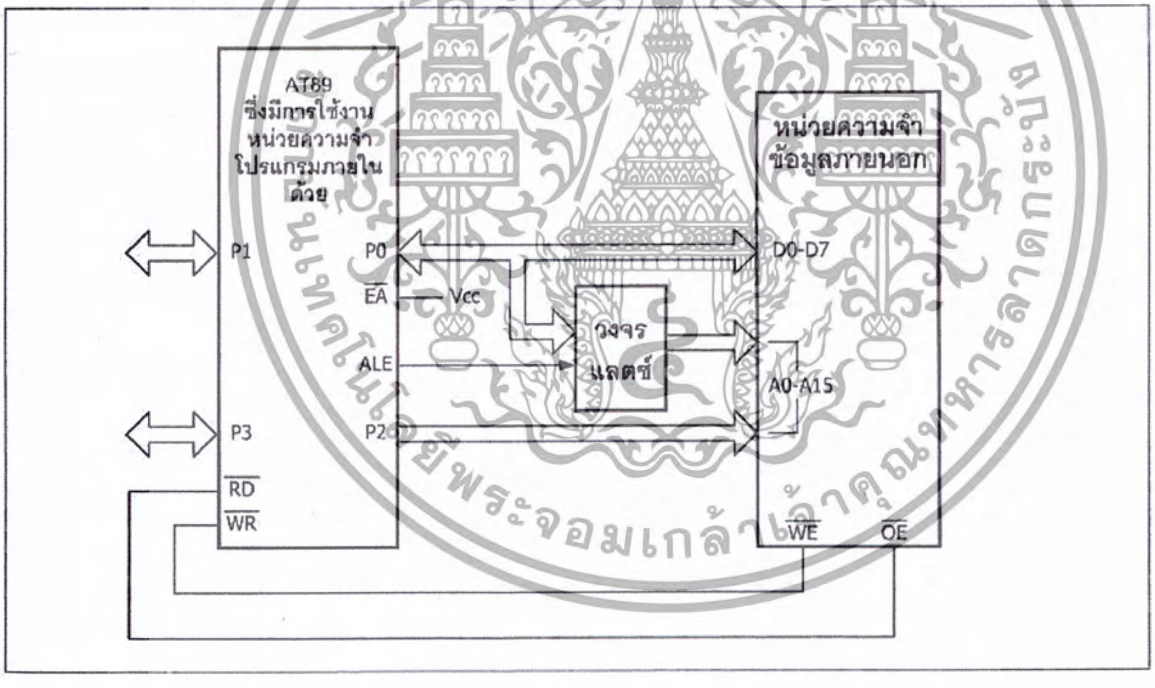
4.5 หน่วยความจำข้อมูล (Data memory)

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดยไมโครคอนโทรลเลอร์สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์แสดงดังรูปที่ 2-7 ซึ่งจะเห็นได้ว่ามีลักษณะคล้ายกับหน่วยความจำโปรแกรมภายนอกแต่จะแตกต่างกันตรงที่มีสัญญาณสำหรับการอ่านและเขียนหน่วยความจำภายนอกที่ขา $\overline{RD}$ และ $\overline{WR}$ สำหรับ 89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ สำหรับการแบ่งหน่วยความจำภายในเป็น 3 ส่วน คือ หน่วยความจำข้อมูลส่วนล่าง(lower) ส่วนบน(upper) และรีจิสเตอร์ฟังก์ชั่นพิเศษ(SFR:Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์ แสดงการแบ่งหน่วยความจำดังรูปที่ 2-8 โดยในหน่วยความจำส่วนล่างมีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้ทั้งโดยตรงและโดยอ้อม ส่วนบนมีแอดเดรสอยู่ที่ 80H-FFH ซึ่งเข้าถึงได้เพียงทางอ้อมเท่านั้น ส่วนSFRมีแอดเดรสอยู่ที่ 80H-FFH ใช้การเข้าถึงโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

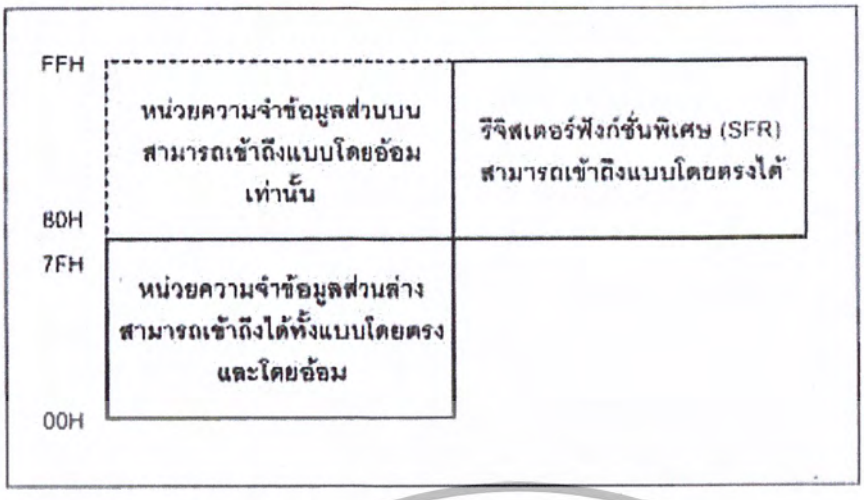


รูปที่ 4-6 แสดงการเชื่อมไมโครคอนโทรลเลอร์ต่อกับหน่วยความจำโปรแกรมภายนอก

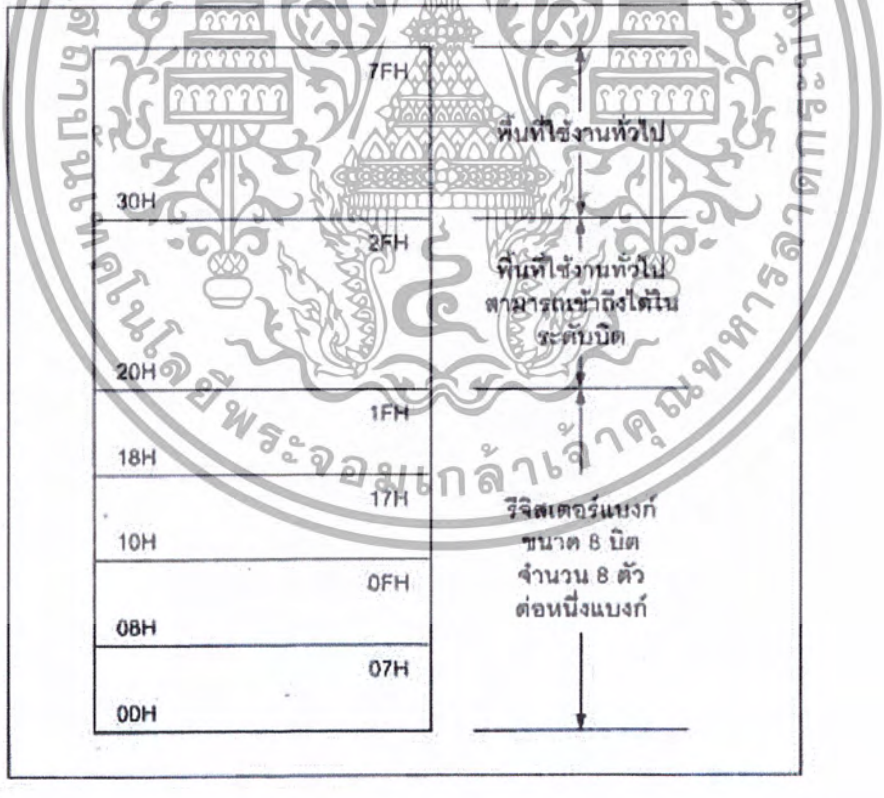


รูปที่ 4-7 แสดงการเชื่อมไมโครคอนโทรลเลอร์ต่อกับหน่วยความจำโปรแกรมภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์

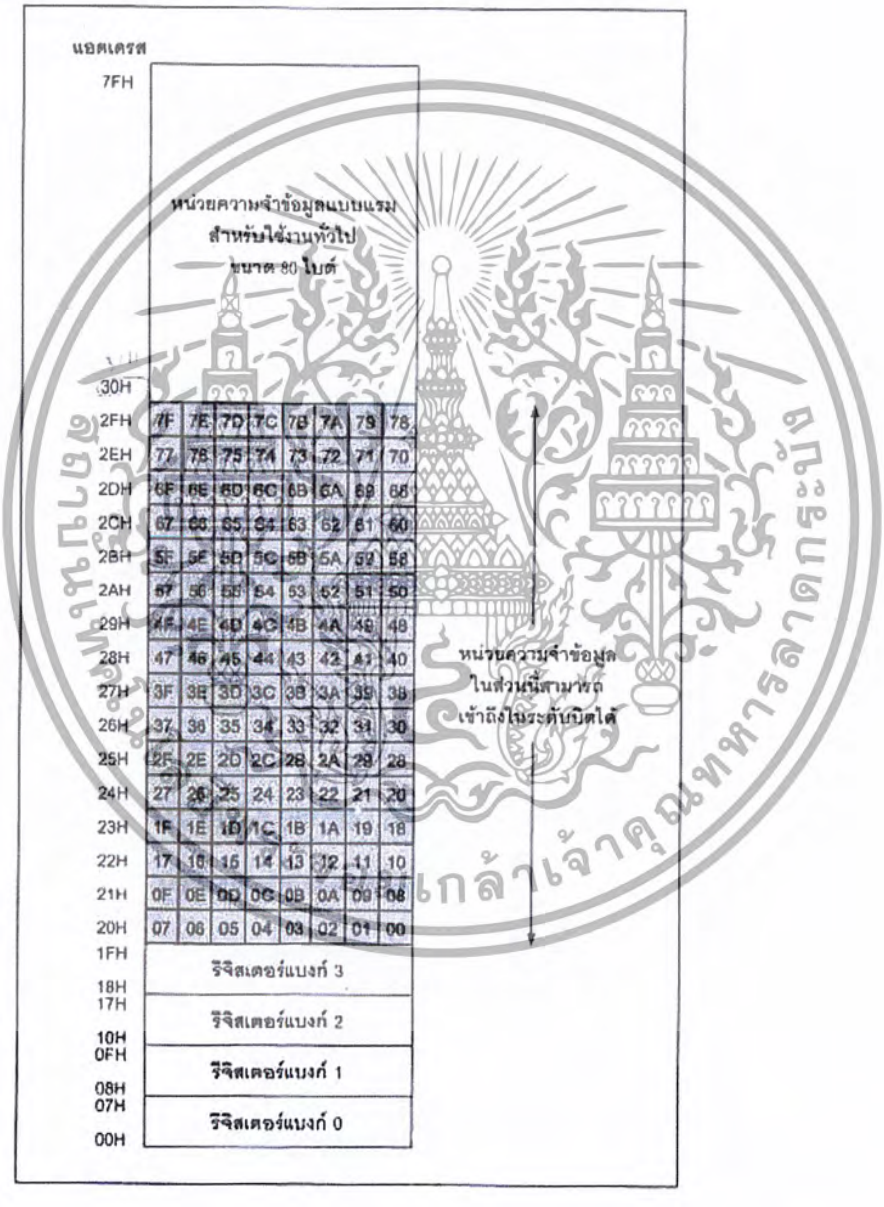


รูปที่ 4-9 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-9 แสดงการแบ่งหน่วยความจำข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์สุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่าแบงค์(bank) แต่ละแบงค์มีรีจิสเตอร์ 8 ตัว คือ R0-R7 การติดต่อกับหน่วยความจำในแบงค์ใดให้กำหนดที่รีจิสเตอร์ PSW(Program Status Word register)

หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไป สามารถเข้าถึงได้ในระดับบิต(Bit addressable) และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก(stack:ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำส่วนนี้ต้องใช้การเข้าถึงระดับไบต์



รูปที่ 4-10 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนบนของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4-10 แสดงโครงสร้างของหน่วยความจำส่วนบน ซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำข้อมูลส่วนล่าง หากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สำหรับ stack และต้องใช้การเข้าถึงโดยอ้อมเท่านั้น

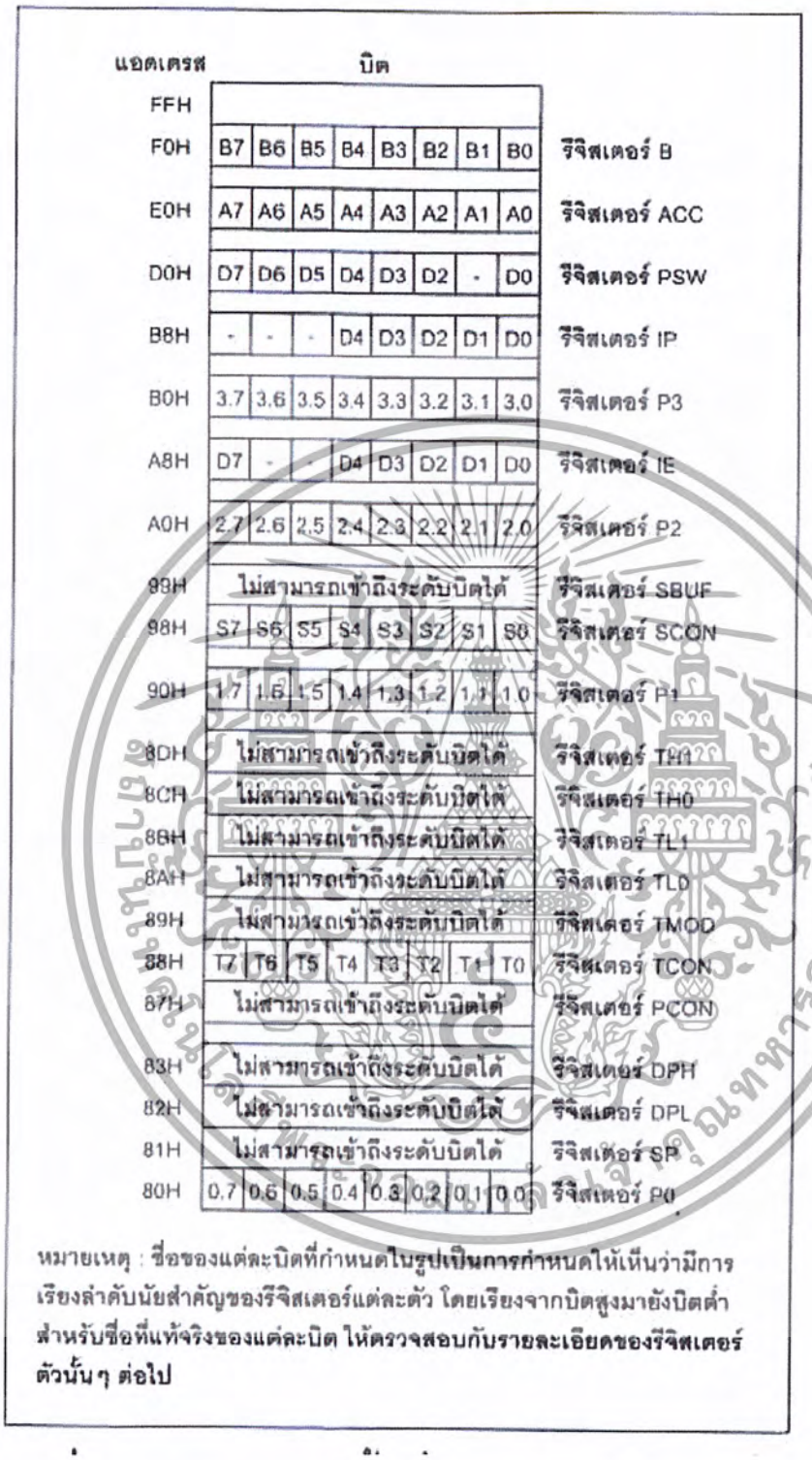
- รีจิสเตอร์ฟังก์ชั่นพิเศษ(Special Function Register:SFR)

เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ มี 22 ตัวสำหรับ 89C51 รีจิสเตอร์ SFRมีแอดเดรสอยู่ที่ 80H-FFH

- รีจิสเตอร์แสดงสถานะของโปรแกรม(Program Status Word:PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ D0H ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกเก็บสถานะต่างๆของโปรแกรมเรียกว่า แฟลก(Flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วจะเกิดการเปลี่ยนแปลงสถานะขึ้น ผลการเปลี่ยนแปลงจะมาปรากฏที่บิตต่างๆของรีจิสเตอร์ PSW รายละเอียดของแต่ละบิตในรีจิสเตอร์ PSW แสดงดังรูปที่ 4-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แอดเดรส	บิต								
FFH									
F0H	B7	B6	B5	B4	B3	B2	B1	B0	รีจิสเตอร์ B
E0H	A7	A6	A5	A4	A3	A2	A1	A0	รีจิสเตอร์ ACC
D0H	D7	D6	D5	D4	D3	D2	-	D0	รีจิสเตอร์ PSW
B8H	-	-	-	D4	D3	D2	D1	D0	รีจิสเตอร์ IP
B0H	3.7	3.6	3.5	3.4	3.3	3.2	3.1	3.0	รีจิสเตอร์ P3
A8H	D7	-	-	D4	D3	D2	D1	D0	รีจิสเตอร์ IE
A0H	2.7	2.6	2.5	2.4	2.3	2.2	2.1	2.0	รีจิสเตอร์ P2
99H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ SBUF
98H	S7	S6	S5	S4	S3	S2	S1	S0	รีจิสเตอร์ SCON
90H	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0	รีจิสเตอร์ P1
8DH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TH1
8CH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TH0
8BH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TL1
8AH	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TL0
89H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ TMOD
88H	T7	T6	T5	T4	T3	T2	T1	T0	รีจิสเตอร์ TCON
87H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ PCON
83H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ DPH
82H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ DPL
81H	ไม่สามารถเข้าถึงระดับบิตได้								รีจิสเตอร์ SP
80H	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	รีจิสเตอร์ P0

หมายเหตุ : ชื่อของแต่ละบิตที่กำหนดในรูปเป็นการกำหนดให้เห็นว่ามีการเรียงลำดับนัยสำคัญของรีจิสเตอร์แต่ละตัว โดยเรียงจากบิตสูงมายังบิตต่ำ สำหรับชื่อที่แท้จริงของแต่ละบิต ให้ตรวจสอบกับรายละเอียดของรีจิสเตอร์ตัวนั้น ๆ ต่อไป

รูปที่ 4-11 การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษSFR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CY	AC	F0	RS1	RS0	OV	-	P

CY : แฟลกทด (Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิก แล้วค่าของแอกคิวมูลเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC : แฟลกทดเสริม (Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4

F0 : แฟลกใช้งานทั่วไป เมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใด ๆ ที่บิตนี้จะไม่มีการเปลี่ยนแปลง

RS1 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

RS0 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

OV : บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ทำให้เกิดการทดข้ามจากบิต 6 มายังบิต 7 ของแอกคิวมูเลเตอร์ หรือแอกคิวมูเลเตอร์มีค่าเกิน 127 (ฐานสิบ) นอกจากนั้นยังใช้เป็นการแสดงค่าลบด้วย

\- : บิตนี้ผู้ใช้งานสามารถกำหนดใช้งานได้อย่างอิสระ

P : บิตพาริตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแอกคิวมูเลเตอร์ ถ้าหากในแอกคิวมูเลเตอร์มีจำนวนบิตที่เป็น "1" รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" ถ้ารวมกันเป็นเลขคี่ บิตนี้จะเป็น "1"

รูปที่ 4-12 แสดงรายละเอียดของรีจิสเตอร์แสดงสถานะของโปรแกรม(Program Status Word:PSW)

จะเห็นว่านอกจากรีจิสเตอร์ PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้ว ที่บิต RS0 และ RS1 ยังใช้ในการเลือกแบงค์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0-R7 ด้วย

RS1	RS0	แบงก์ของรีจิสเตอร์	ช่วงแอดเดรส
0	0	แบงก์ 0	00H-07H
0	1	แบงก์ 1	08H-0FH
1	0	แบงก์ 2	10H-17H
1	1	แบงก์ 3	18H-1FH

รูปที่ 4-13 แสดงการเลือกแบงค์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงค์ R0-R7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แอกคิวมูเลเตอร์(Accumulator:ACC) หรือ register A

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะทำการส่งข้อมูลหรือผลลัพธ์ที่ได้ให้กับซีพียูเพื่อทำการประมวลต่อไป

- รีจิสเตอร์ B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษคือหากต้องการคูณหรือหารทางคณิตศาสตร์ต้องนำข้อมูลที่ต้องการมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งกับรีจิสเตอร์ A

- โปรแกรมเคาน์เตอร์(Program Counter:PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมอยู่ใน SFR การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำโปรแกรมที่กำหนด PC มีความสำคัญในการตรวจสอบการทำงานของโปรแกรมว่าดำเนินไปถูกต้องหรือไม่

- สแต็กพอยเตอร์(Stack Pointer:SP)

รีจิสเตอร์ตัวชี้สแต็กมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81H ใช้ในการเก็บค่าตำแหน่งของตัวชี้สแต็กซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อยหรือกระโดดจากโปรแกรมย่อยกลับมายังโปรแกรมหลักเมื่อมีการรีเซตเกิดขึ้น

- รีจิสเตอร์ชี้ข้อมูล(Data Pointer:DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง(DPH) และรีจิสเตอร์ไบต์ต่ำ(DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H และ 83H ตามลำดับ สำหรับ DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ต้องการติดต่อด้วย

- รีจิสเตอร์พอร์ต(Port Register)

เป็นรีจิสเตอร์ขนาด 8 บิตที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ มี 4 ตัวคือ P0,P1,P2 และ P3 โดยมีแอดเดรสอยู่ที่ 80H 90H A0H B0H ตามลำดับรีจิสเตอร์ทุกตัวสามารถกระทำได้ในระดับบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม(Serial Data Buffer:SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้า ผ่านทาง TxD และ RxD ที่ขา P3.1 และ P3.0 ตามลำดับ

- รีจิสเตอร์ไทม์เมอร์(Timer register)

เป็นรีจิสเตอร์ขนาด 16 บิต แบ่งเป็นไบต์สูงและไบต์ต่ำ ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (counter)ภายในไมโครคอนโทรลเลอร์ เพื่อใช้สร้างฐานเวลา,จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน โดยรีจิสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH รีจิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH

- รีจิสเตอร์ควบคุม(Control register)

  - รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรม
  - รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของการสื่อสารแบบอนุกรม
  - รีจิสเตอร์ TCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ภายในตัวไมโครคอนโทรลเลอร์
  - รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทมเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์
  - รีจิสเตอร์ IE และ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ โดย IE เป็นรีจิสเตอร์สำหรับ enable กำหนดลักษณะการตอบสนองการอินเตอร์รัปต์ IP เป็นรีจิสเตอร์สำหรับการกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่าจะทำอันใดก่อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

Visual Basic และ MATLAB

Visual Basic

การควบคุมและติดต่อผ่านทางพอร์ตอนุกรม(Serial) ด้วย Com Port สามารถทำได้ 2 วิธี

1. การติดต่อแบบอินเตอร์รัพต์

โดยที่โปรแกรมจะอินเตอร์รัพต์ เมื่อมีข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnCommEvent

2. การติดต่อแบบโพลลิ่ง

การโพลใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด วิธีการนี้จะตรวจสอบคีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลาโดยใช้ Control Timer

ฟังก์ชั่น Visual Basic 6.0 ที่ใช้กับ Serial Port (RS-232)

ในโครงงานนี้ได้ใช้การติดต่อแบบอินเตอร์รัพต์

- ตัวคอนโทรล MSComm

  มี Event เดียวที่ใช้ก็คือ OnComm ซึ่งการติดต่อแบบอินเตอร์รัพต์โดยติดต่อกับ Serial Port แบบธรรมดาจะใช้ comEvent จะมีเพียงแค่ comEvReceive กับ comEvSend

  ในการใช้ MSComm จะต้องตั้งค่าเพื่อที่จะติดต่อกับพอร์ต

  - ComPort คือ การกำหนดหมายเลข Port ที่ต้องการติดต่อ (Com1, Com2)
  - Setting คือ การกำหนดอัตรา Baud (อัตราการส่งข้อมูล), Parity (ถ้าไม่ใช้ให้ใส่ N), Data(จำนวนบิต), Stop(บิตสต๊อบ)

- การใช้ Buffer ในการรับส่งข้อมูล
  - InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลที่เข้ามา
  - OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลที่ออกไป
  - RThreshold คือ การทำให้การเกิดกระตุ้นด้วย Event-driven เมื่อมีBuffer ทำการรับข้อมูลเข้ามาจาก ComPort มันจะทำให้เกิด CommEvent ใน OnComm Event

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Inputlen คือ กำหนดจำนวนของข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดใน Buffer ที่รับข้อมูล
- EOFEnable คือ การบอกว่าสิ้นสุดของไฟล์ (End Of File); EOF
- PortOpen คือ จะให้เปิดพอร์ตในการงานหรือไม่ ถ้าเปิด = True ถ้าปิด = False

ตัวอย่างเช่น

```
MSComm1.CommPort = 1
MSComm1.Settings = "9600,N,8,1”
MSComm1.InputLen = 48
MSComm1.PortOpen = True
MSComm1.RThreshold = 48

Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent
        Case comEvReceive
    Dim X As String
    X = MSComm1.Input
End Sub
```

การกำหนดชนิดของตัวแปร เพื่อเป็นส่วนของการรับข้อมูล

ใน Sub MSComm1_OnComm() ใช้เงื่อนไข Select Case เข้ามาเพื่อใช้ตรวจสอบเมื่อเกิด Events ของการรับข้อมูล

การเข้าถึงไฟล์ข้อมูล ใน Visual Basic 6.0 สามารถเข้าถึงไฟล์ข้อมูลได้ 3 แบบ

1. Sequential File หรือ ไฟล์ข้อมูลแบบเรียงลำดับต่อเนื่อง

มีการบันทึก และอ่านไฟล์ข้อมูล ที่ได้ป้อนลงไปในเครื่องคอมพิวเตอร์เป็นลำดับ ตั้งแต่ต้นไฟล์ถึงท้ายไฟล์ (First In First Out) โดยข้อมูลที่นำไปเก็บจะต่อเนื่องกัน โดยไม่มีช่องว่าง ทั้งนี้แม้ว่า แต่ละ เรกคอร์ด (Record) จะมีความยาวข้อมูลไม่เท่ากัน

ข้อเสีย ของการบันทึกข้อมูลของไฟล์ประเภทนี้คือเมื่อจะใช้ข้อมูลทุกครั้งจากไฟล์จะต้องเริ่มใช้ตัวแรก เรียงลำดับกันไป หากจะเรียกเจาะจงข้อมูลไม่สามารถทำได้เป็นผลให้ทำงานช้าจึงไม่เหมาะกับการเก็บ ไฟล์ข้อมูลจำนวนเยอะๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Random File หรือ ไฟล์แบบสุ่ม

วิธีการจัดการข้อมูลของ Random File นี้สามารถจะเข้าถึงข้อมูลได้รวดเร็วเพราะไม่ต้องมีการเรียงข้อมูลเหมือนอย่าง Sequential File ซึ่งเราสามารถเจาะจงใช้ข้อมูลได้โดยตรงยัง Record นั้นๆ แต่มีข้อดีก็ต้องข้อเสียเล็กน้อยคือการใช้งานจะซับซ้อนกว่า Sequential File

3. Binary File หรือ ไฟล์แบบเลขฐานสอง

ระบบการจัดการไฟล์แบบนี้ จะทำให้เราสามารถรับส่งข้อมูลเป็นอย่างสมบูรณ์ เพราะไฟล์ประเภทนี้จะสามารถแทนข้อมูลในลักษณะใดๆก็ได้ และยังสามารถกำหนดรูปแบบโครง สร้างข้อมูล ตัวแปรและความยาวในการบันทึกข้อมูล ทั้งนี้ขนาดความจุของไฟล์ก็มีขนาดเล็ก

เนื่องจากการส่งข้อมูลผ่าน Serial Port ที่มีบัฟเฟอร์ข้อมูลไม่มาก แต่ด้วยข้อมูลที่มีขนาดเล็ก โดยสามารถที่จะรับส่งไฟล์ ข้อมูลต่างๆได้ในระดับ Binary แล้ว เช่น ไฟล์รูปภาพ, ไฟล์เอกสาร, ไฟล์โปรแกรม เป็นต้น ซึ่งมีข้อดีที่แตกต่าง กว่าแบบ Sequential File และ Random File ที่จะมีข้อมูลในแต่ Record ขนาดใหญ่ จำกัดประเภท ไฟล์ที่รับส่งข้อมูล เข้าถึงข้อมูลได้ช้า

ฟังก์ชั่นในการเปิดไฟล์ของ Visual Basic 6.0

Open pathname For mode [Access access] [lock] As [#]Filenumber [Len=Reclength]

- Pathname คือ ชื่อ ไฟล์ที่จะใช้งาน
- Mode คือ คำสงวนที่ใช้ในเลือกโหมดของไฟล์มีคำว่า Append, Binary, Input, Output, or Random ถ้าไม่ได้กำหนด ค่าปกติจะตั้งไว้สำหรับ โหมด Random
- Access คือ คำสงวนใช้ในการกระทำเพื่อเข้าถึง ไฟล์ข้อมูลมีคำว่า Read, Write, or Read Write
- Lock คือ คำสงวนที่ใช้กำหนดคุณสมบัติของไฟล์ข้อมูล มีคำว่า Shared, Lock Read, Lock Write, and Lock Read Write
- Filenumber คือ หมายเลขของไฟล์ข้อมูลที่เราใช้ เปิด อยู่ในช่วง 1-511 ทั้งนี้ สามารถหาหมายเลขไฟล์ที่ว่างด้วยคำสั่ง FreeFile
- Reclength คือ ความยาวในการบันทึกข้อมูลสามารถกำหนดได้ถึง 32,767 bytes

ตัวอย่างเช่น

```
Open "C:\t1.dat" For Output As #1
Open "C:\t2.dat" For Output As #2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Open "C:\t3.dat" For Output As #3
Open "C:\t4.dat" For Output As #4
Open "C:\t5.dat" For Output As #5
```

ทุกครั้งที่เปิดด้วยคำสั่ง Open แล้วเมื่อไม่มีการใช้การอ่านและเขียนข้อมูลแล้ว
จะต้องปิดไฟล์ข้อมูลด้วยคำสั่ง Close #filenumber

```
Close #1
Close #2
Close #3
Close #4
Close #5
```

คำสั่งทั่วๆ ไป ของ Visual Basic 6.0 ที่ใช้ในโครงงานนี้
Mid ฟังก์ชั่น

```
Mid(String,start[,length])
```

String คือ ข้อมูลที่มีชนิดตัวแปรเป็น String
start คือ การกำหนดจุดเริ่มของ String ถ้ากำหนดจำนวนค่าเกินกว่าเกินจำนวนความยาวของ String ที่กำหนดไว้ ค่าปกติที่ให้ จะเป็น String ว่างๆ (“”)
length คือ การกำหนดจุดจบของ String ที่จะให้แบ่ง

ตัวอย่างเช่น

```
dim txt
txt = ”This is a beautiful day”
Mid(txt,1,12)
```

ผลที่จะได้คือ This is a be

Print ฟังก์ชั่น

เป็นการ print ไฟล์,

```
Print #1,
```

ใน โครงงานนี้ได้กำหนดค่าเริ่มต้นของคอมพอร์ต โดยเปิดการใช้รับข้อมูลจากพอร์ต RS-232
- ค่า Input คือ การรับข้อมูลมาจากพอร์ต ใน MSComm_OnComm ซึ่งจะอ่าน ข้อมูลมาจากทางพอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ค่า Output คือ ข้อมูลเป็นชุดๆ ที่ได้เซฟลงในไฟล์ที่ได้เปิดไว้

MATLAB

การประมวลผลข้อมูลด้วยโปรแกรม MATLAB

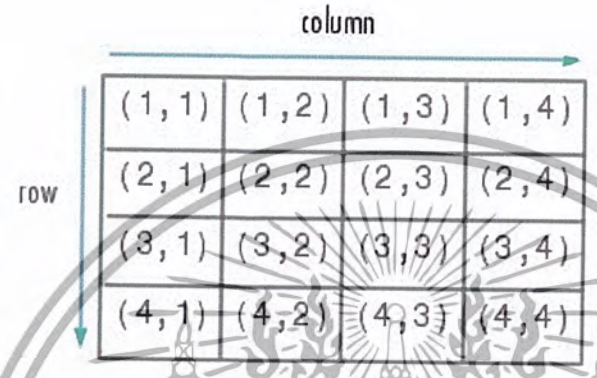
โปรแกรม MATLAB สามารถทำการนำเอาข้อมูลจากผลการทดลองมาทำการประมวลผลในลักษณะต่างๆได้ โดยจะมีการแสดงผลต่างๆออกมาในลักษณะกราฟ รูปภาพ และการแสดงผลภาพเคลื่อนไหว ในการทดลองได้นำเอาความสามารถในการประมวลภาพของโปรแกรมมาทำการประมวลผลข้อมูล โดยนำเอาข้อมูลอุณหภูมิที่ได้จากการเก็บค่าด้วย Visual Basic ที่กำหนดปลายทางในการเก็บค่าไว้ที่ C:\MATLAB6p5\work\ แล้วตามด้วยชื่อไฟล์ที่ต้องการ เนื่องจาก MATLAB ต้องใช้ข้อมูลที่อยู่ใน directory ของ MATLAB เองมาทำการประมวลผล ค่าที่ได้จากการเก็บจะนำมาทำการปรับขนาด matrix เสียใหม่ โดยจะปรับให้มีขนาด เป็น 4x4 เพื่อให้เป็นลักษณะเดียวกับโครงสร้าง Detector ที่มีการวางตำแหน่ง 16 ตัว ในลักษณะเป็นโครงร่างตาข่าย 4x4 จากนั้นจึงนำเอาอุณหภูมิที่ได้มาทำการประมวลผลโดยการประมาณทางคณิตศาสตร์โดยใช้สมการ Bilinear Interpolation ในการประมาณซึ่งในการทดลองจะใช้ DS1820 วัดอุณหภูมิเป็นจุดๆจึงทำให้ช่วงระหว่างจุดที่มีการวัดจริงไม่มีค่า การใช้สมการ Bilinear Interpolation จะทำให้สามารถประมาณค่าในระหว่างจุดที่มีค่าได้ทำให้สามารถเก็บรายละเอียดอุณหภูมิได้มากขึ้น จากการทำการประมาณค่านี้ทำให้ขนาด matrix ของข้อมูลมีขนาดใหญ่ขึ้นเนื่องจากมีค่าประมาณเพิ่มขึ้นมาในระหว่างค่าจาก detector 1820 ในขั้นตอนนี้สามารถทำให้เกิดระนาบอุณหภูมิที่ระเอียดขึ้น นอกจากนี้เรายังสามารถเก็บค่าในแบบปริมาตรได้โดยการเก็บอุณหภูมิที่แต่ละระนาบในระยะการเคลื่อนที่ต่างๆ ที่ให้ได้ข้อมูลในลักษณะ array 3 มิติ การเก็บแบบนี้จะทำให้สามารถเก็บค่าเป็นปริมาตรได้และเมื่อทำการประมาณค่าอุณหภูมิจากข้อมูลปริมาตรนี้ จะทำให้เราสามารถเก็บรายละเอียดในตำแหน่งต่างๆในปริมาตรได้ละเอียดขึ้นมาก ทำให้การประมวลผลภาพ 3 มิติ สามารถบอกได้ถึงการเปลี่ยนแปลงอุณหภูมิในปริมาตรอย่างต่อเนื่อง บอกได้ถึงลักษณะการกระจายอุณหภูมิภายในปริมาตรนั้นด้วย จากการทดลองจะสามารถเลือกตำแหน่งในการดูลักษณะอุณหภูมิได้โดยใช้การตัดแกนเพื่อเปิดระนาบดูรายละเอียดอุณหภูมิต่างๆในจุดต่างๆภายในปริมาตร ซึ่งจุดอุณหภูมิต่างๆจะสามารถมองเห็นความแตกต่างกันได้โดยการแสดงสีที่ต่างๆกัน โดยในการทดลองจะให้สีที่บอกถึงอุณหภูมิสูงเป็นสีแดงเข้ม สีที่บอกถึงอุณหภูมิต่ำเป็นสีน้ำเงิน ซึ่งอุณหภูมิในระหว่างนี้จะแสดงเป็นสีที่มีความเข้มหรืออ่อนไล่เลี่ยนกันไป ความแตกต่างของความเข้มสีจะบอกได้ถึงการกระจายอุณหภูมิในระนาบต่างๆได้อย่างดี สำหรับ functions ต่างๆที่เกี่ยวข้องในการประมวลผลภาพของ MATLAB มีดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง(ฟังก์ชั่น) Function ใน MATLAB

- ตัวแปรอะเรย์แบบหลายมิติ (Multidimensional Arrays)

การประกาศตัวแปรอะเรย์แบบหลายมิติใน MATLAB เป็นการเพิ่มของข้อมูลจากการประกาศตัวแปรอะเรย์แบบสองมิติ ซึ่งมีทางด้านแถว (Row) และหลัก (Column)



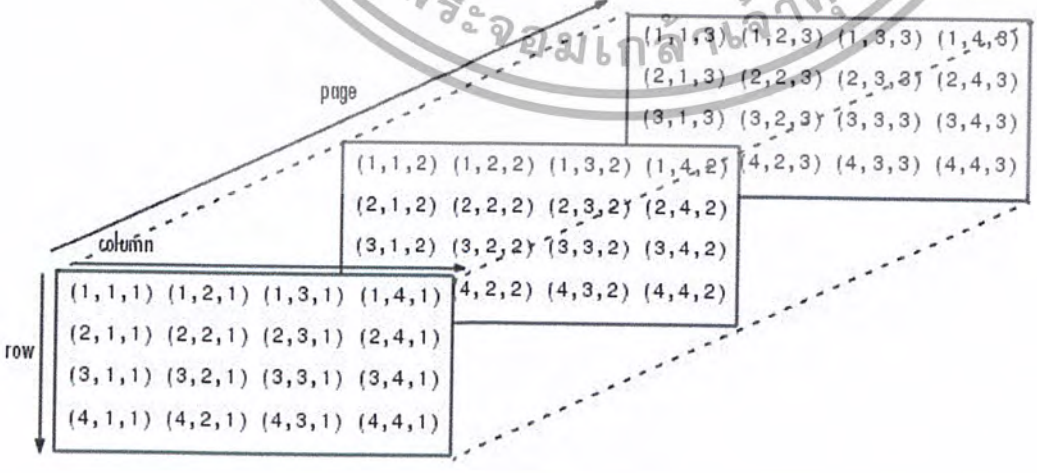
แสดงอะเรย์ 2 มิติ

ในการสร้างตัวแปรอะเรย์แบบหลายมิติ จำเป็นต้องอ้างถึงค่าในอะเรย์และจะใช้การอ้างลำดับของอะเรย์ หรือ (Indexing) โดยมีสมาชิกในอะเรย์ 3 ค่า คือ แถว, หลัก และหน้า ตามลำดับ

แถว (Row) คือ สมาชิกตัวแรกที่อ้างถึงมิติแรกของอะเรย์

หลัก (Column) คือ สมาชิกตัวที่สองที่อ้างถึงมิติที่สองของอะเรย์

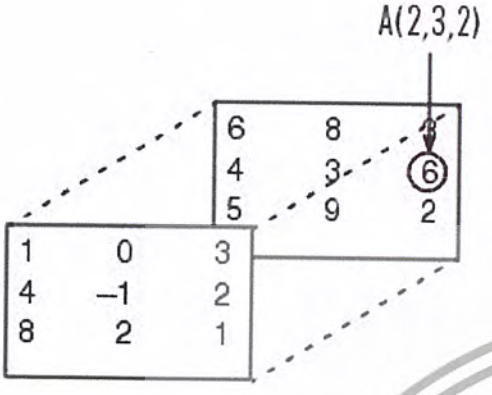
หน้า (Page) คือ สมาชิกตัวที่สามที่อ้างถึงมิติที่สามของอะเรย์



แสดงอะเรย์หลายมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างถึงค่าในอะเรย์แถวที่สอง หลักที่สาม หน้าที่สอง A(2,3,2)




```
A(:,:,1) =

     1     0     3
     4    -1     2
     8     2     1

A(:,:,2) =

     6     8     3
     4     3     6
     5     9     2
```

แสดงการอ้างถึงข้อมูลในอะเรย์

การใช้Colon (:) ในการอ้างถึงตัวแปรอะเรย์แบบหลายมิติ

ใน MATLAB การอ้างถึงตัวแปรอะเรย์แบบหลายมิติด้วย Colon หมายถึงการเข้าถึงสมาชิกอื่นๆ ของข้อมูล เช่น

A(:,3,2) เป็นการเข้าถึงหลักที่ 3 ทั้งหมด ในหน้าที่ 2 ของ A

A(2:3,2:3,1) เมตริกซ์นี้ประกอบด้วย ข้อมูลแถวที่ 2 และ 3, หลักที่ 2 และ 3 ของหน้าแรกในอะเรย์ A

ทั้งนี้ เครื่องหมาย Colon สามารถใช้ในการกำหนดค่าระหว่างอะเรย์นึงไปยังอีกอันหนึ่งได้

C = zeros(4,4) การสร้างอะเรย์ 4x4 โดยกำหนดค่าเริ่มต้นเป็น 0 หมด

C = (2:3,2:3) = A(2:3,1:2,2) การกำหนด 2x2 ในอะเรย์ A ไปยังสมาชิก 4 ตัวในตรงกลาง

การจัดการข้อมูลในอะเรย์แบบหลายมิติ

1. ถือว่าหน้าของข้อมูลอะเรย์แบบ 2 มิติเป็นเมตริกซ์
2. ข้อมูลในอะเรย์หลายมิติสามารถเป็นได้หลายแบบ เช่น อุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากสร้างอะเรย์แบบหลายมิติ สามารถใช้คำสั่งเหล่านี้ในการดูข้อมูลต่างๆ

ข้อมูลของอะเรย์	คำสั่ง	ตัวอย่าง
ขนาด	size	size(c) ans = 2 2 1 2 rows columns dim3 dim4
มิติ	ndims	ndims(c) ans = 4
ความจุและรูปแบบ	whos	whos Name Size Byte Class A 2 x 2 x 2 64 double array B 2 x 2 x 2 64 double array C 4-D 64 double array D 4-D 192 double array

- Meshgrid

การแสดงค่า x และ y เป็นโครงสร้างแท้ 3 มิติ

[x,y] = meshgrid(x,y) เป็นการแปลงโดเมนกำหนดโดย ค่าเวคเตอร์ x และ y ไปเป็นอะเรย์ x และ y ที่สามารถนำมาประเมินค่าของฟังก์ชั่นของตัวแปร 2 ตัว และ พื้นผิวแบบ 3 มิติ

```
[X,Y] = meshgrid(1:3,10:14)
X =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
Y =
    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
```

แสดงลักษณะอะเรยีของ Meshgrid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[x,y,z] = meshgrid(x,y,z) แสดงถึงอะเรย์ 3 มิติ โดยใช้ค่าตัวแปร 3 ค่า ในฟังก์ชั่นแล้วจุดเป็นโครงสร้าง 3 มิติ

- Slice

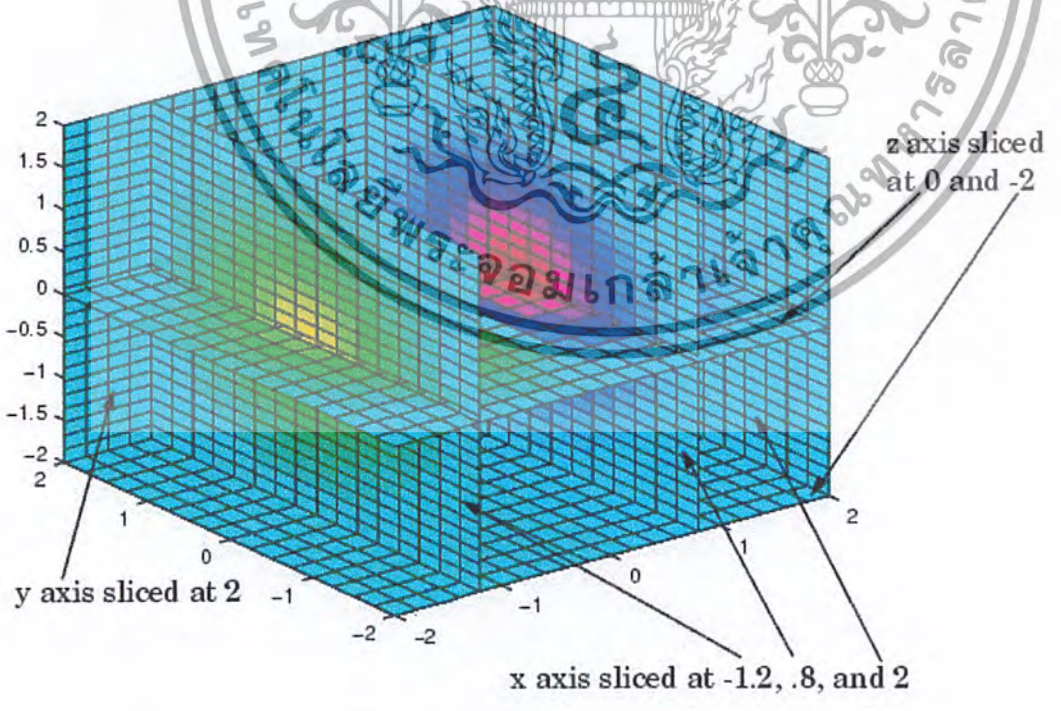
  คำสั่งในการแบ่งหน้าหรือแผ่นเป็นส่วนๆ เป็นไปตามปริมาตรของรูปสี่เหลี่ยมมุมฉาก

slice(V,XI,YI,ZI) เป็นการวาดข้อมูลจากการแบ่งเป็นส่วนๆ ในปริมาตรของ V ที่กำหนดโดย XI, VI และ ZI ขณะที่ XI, YI และ ZI เป็นเมตริกซ์ที่กำหนดพื้นผิว และปริมาตรจะถูกประเมินที่จุดของพื้นผิว โดยที่ XI, YI และ ZI ต้องมีขนาดที่เท่ากัน

slice(..., 'method') เป็นการเจาะจงเมถอดของการประมาณค่าของช่วงข้อมูล (Interpolation) เช่น'linear', 'cubic' หรือ 'nearest'

เมตตอทที่ใช้ในการประมาณค่า

- linear ระบุถึง การประมาณค่าของเส้นตรง
- cubic ระบุถึง การประมาณค่าในรูปแบบของลูกบาศก์
- nearest ระบุถึง การประมาณค่าบริเวณจุดที่ใกล้เคียง หรือรอบๆ



แสดงการตัดระนาบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการ slice ในฟังก์ชั่น $v = xe^{(-x^2-y^2-z^2)}$

โดยที่มี $-2 \leq x \leq 2$

$-2 \leq y \leq 2$

$-2 \leq z \leq 2$

- Interp2

  การประมาณค่าระหว่างข้อมูลแบบ 2 มิติ

ZI = interp2(X,Y,Z,XI,YI)

ZI = interp2(Z,XI,YI)

ZI = interp2(Z,ntimes)

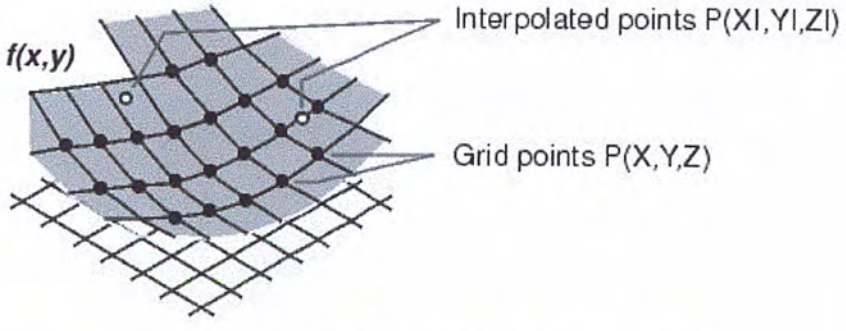
ZI = interp2(…,method)

ZI = interp2(X,Y,Z,XI,YI) จะส่งค่ากลับมาให้ ZI โดยที่เก็บค่าของ XI และ YI และกำหนดค่าประมาณภายในฟังก์ชั่น 2 มิติ ที่ระบุในเมตริกซ์ X, Y และ Z ค่าของ XI และ YI สามารถเป็นเมตริกซ์ ซึ่งคำสั่ง interp2 จะคืนค่า Z กลับมาตามที่จุดของ XI(i,j) และ YI(i,j) โดยที่สามารถส่งค่าไปในเว็คเตอร์ของแถวและหลักได้ XI และ YI ตามลำดับ

เมตอดที่ใช้ใน interp2

- linear ระบุถึง การประมาณค่าของเส้นตรง
- cubic ระบุถึง การประมาณค่าในรูปแบบของลูกบาศก์
- nearest ระบุถึง การประมาณค่าบริเวณจุดที่ใกล้เคียง หรือรอบๆ

ในคำสั่ง interp2 เป็นการประมาณค่าระหว่างจุด ซึ่งจะหาค่าได้จากฟังก์ชั่น 2 มิติของ f(x,y)



แสดงการประมาณค่าแบบ Bilinear Interpolation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Interp3

  การประมาณค่าระหว่างข้อมูลแบบ 3 มิติ

VI = interp3(X,Y,Z,V,XI,YI,ZI)

VI = interp3(V,XI,YI,ZI)

VI = interp3(V,ntimes)

VI = interp3(…,method)

VI = interp3(X,Y,Z,V,XI,YI,ZI) การประมาณค่าหา VI ซึ่งค่าจะได้จากการหาค่าฟังก์ชั่น 3 มิติ ของ V ที่ชี้ในอะเรย์ XI, YI และ ZI โดยที่ค่าเวคเตอร์ของ XI, YI และ ZI ต้องมีขนาดที่เท่ากัน ถ้าหากค่าในเวคเตอร์มีจำนวนขนาดที่ไม่เท่ากัน ทั้งแถวและหลัก จะส่งค่าไปให้ meshgrid เพื่อสร้างอะเรย์ Y1, Y2 และ Y3

VI = interp3(V,XI,YI,ZI) โดยที่ X = 1:N, Y = 1:M, Z = 1:P ขณะที่ M[N,M,P] = size(V)

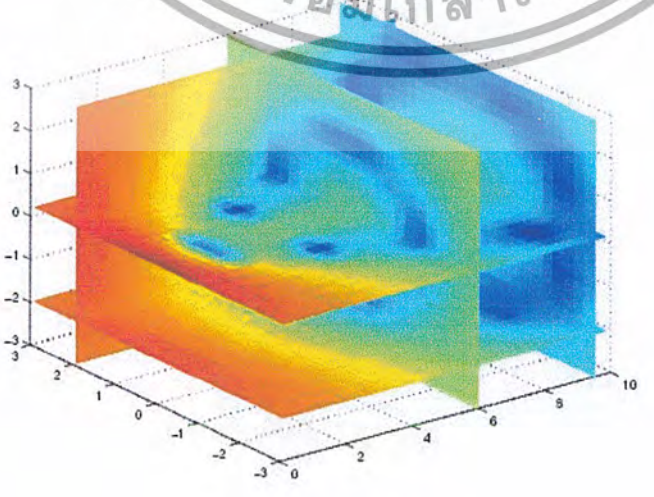
เมตตอทที่ใช้ใน interp3

- linear ระบุถึง การประมาณค่าของเส้นตรง
- cubic ระบุถึง การประมาณค่าในรูปแบบของลูกบาศก์
- nearest ระบุถึง การประมาณค่าบริเวณจุดที่ใกล้เคียง หรือรอบๆ

ตัวอย่างเช่น

vi = interp3(x,y,z,v,xi,yi,zi)

slice = (xi,yi,zi,vi[6 9.5],2,[-2 .2])



การ Interpolation3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- colormap

colormap(map)

คำสั่ง colormap เป็นเมตริกซ์ m x 3 โดยที่ค่าเป็นจำนวนจริงระหว่าง 0.0 และ1.0 แต่ละแถวเป็น RGB (Red Green Blue) ที่กำหนดต่อหนึ่งสี ซึ่งจำนวนแถวของ k^{th} ใน colormap จะเป็นตัวกำหนดสีให้มี จำนวน k^{th} สี ขณะที่ map(k,:) = [r(k) g(k) b(k)] แสดงถึงความเข้มของสีแดง สีเขียว และสีฟ้า

colormap(map) กำหนด colormap ให้กับเมตริกซ์เป็น map โดยให้ค่าใน map อยู่ช่วงระหว่าง[0,1] เท่านั้น

ในสีของ m-files สามารถกำหนดจำนวนของสีใน colormapได้ เช่น

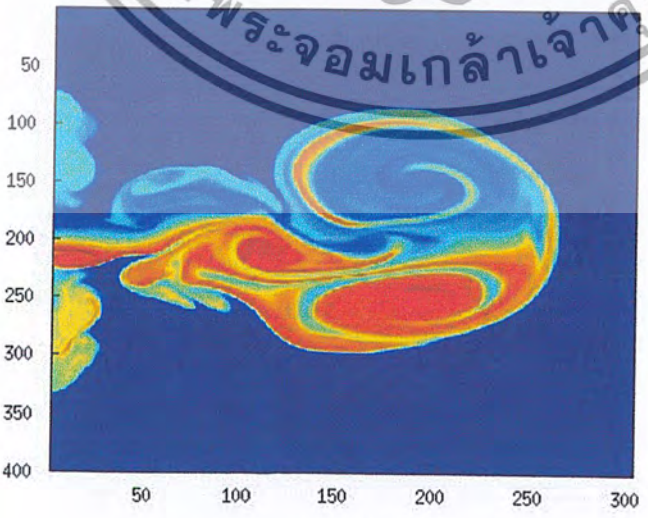
colormap(hsv(128))

MATLAB สามารถรองรับจำนวนสีของ colormap เช่น

- hsv(hue-saturation-value) โดยมีสี 128 สี แสดงถึง(สี-ความอิ่มตัวของสี-ค่าของสี) ซึ่งเริ่มจากสีแดงไล่ไปสีเหลือง เขียว cyan สีฟ้า magenta และกลับมาที่สีแดงอีกครั้ง

- jet มีระยะสีจากสีฟ้าถึงสีแดง และไล่สีจากสี cyan ไปสีเหลือง และสีส้ม ในคำสั่ง jet ของ colormap มีความสัมพันธ์กับ astrophysical fluid

colormap(jet)



แสดงภาพสีจากคำสั่ง Jet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ขั้นตอนการออกแบบ การทดลองและผลการทดลอง

6.1 การหาอุปกรณ์ที่จะนำมาใช้

ขั้นตอนแรกของการทดลองโครงงานก็คือการหาอุปกรณ์ที่จะนำมาใช้ในการทดลองและการออกแบบวงจร ในส่วนของการหาอุปกรณ์ที่จะนำมาใช้ในโครงงานนี้เราเริ่มต้นจากการค้นคว้าหาข้อมูลในอินเทอร์เน็ต ซึ่งในการค้นคว้านั้นเราได้ข้อมูลมาเป็นจำนวนมาก ในหัวข้อนี้จะแบ่งเนื้อหาเป็นส่วน ๆ ที่เกี่ยวกับอุปกรณ์แต่ละตัวที่สำคัญ

- DS 1820 เนื่องจากในโครงงานนี้เกี่ยวข้องกับการวัดอุณหภูมิเราจึงจำเป็นที่จะต้องใช้เทอร์มิสเตอร์ สาเหตุที่เราใช้เทอร์มิสเตอร์เบอร์นี้ก็เพราะ

1.เทอร์มิสเตอร์เบอร์นี้สามารถที่จะส่งข้อมูลออกมาเป็นแบบดิจิตอลซึ่งเราสามารถที่จะนำไปประมวลผลได้ทันที

2.เทอร์มิสเตอร์เบอร์นี้สามารถที่จะเปลี่ยนค่าอุณหภูมิเป็นตัวเลขดิจิตอลได้ภายใน 200 มิลลิวินาที

3.เทอร์มิสเตอร์เบอร์นี้ไม่ต้องการอุปกรณ์ภายนอกต่อเพิ่มเติมซึ่งก็สามารถที่จะทำงานได้ทันที

4.เทอร์มิสเตอร์เบอร์นี้สามารถใช้สายต่อเพียงเส้นเดียวได้กับเทอร์มิสเตอร์หลาย ๆ ตัวโดยใช้ตัวคอนโทลเลอร์เป็นตัวควบคุม

5.เทอร์มิสเตอร์เบอร์นี้สามารถวัดอุณหภูมิได้ตั้งแต่ -55°c ถึง 125°c ซึ่งถือได้ว่ามีช่วงของอุณหภูมิที่กว้างพอที่จะนำมาใช้ในโครงงานนี้ได้

ในการทดลองขั้นแรกนั้นเราใช้เทอร์มิสเตอร์ประมาณ 4-5 ตัวในการทดลอง เหตุผลที่เราใช้เทอร์มิสเตอร์ไม่มากนักเนื่องจาก เทอร์มิสเตอร์ตัวนี้มีราคาค่อนข้างแพง และในขั้นตอนต่อมาจึงได้ทำการเพิ่มอุปกรณ์เป็นโครงร่างแบบ matrix 4x4

- สเต็ปมอเตอร์ ในโครงงานนี้ต้องมีการขับเคลื่อนแผงอ่านอุณหภูมิเราจึงจำเป็นต้องใช้มอเตอร์ในการขับเคลื่อนแผงอ่านอุณหภูมิ เหตที่เราเลือกใช้สเต็ปมอเตอร์ก็เพราะว่า สเต็ปมอเตอร์นั้นมีความแม่นยำและเที่ยงตรงมากในการเคลื่อนที่ สต็ปมอเตอร์ที่มีขายอยู่ตามท้องตลาดนั้นมีอยู่ด้วยกันหลายชนิด แต่ที่เราเลือกใช้คือ สเต็ปมอเตอร์แบบ 4 เฟส 5สาย (สายไฟเลี้ยง 1 เส้น)

- ไมโครคอนโทลเลอร์ ในการทดลองโครงงานนี้เราเลือกใช้ไมโครคอนโทลเลอร์เบอร์ 89C51 เป็นตัวควบคุมการทำงานของวงจรทั้งหมด ซึ่งเนื่องจากในการทดลองนี้ไม่มีการเก็บข้อมูลมากนัก จึงไม่จำเป็นที่ต้องต่อแรมภายนอกเพิ่มเติม ใช้แค่หน่วยความจำภายในก็เพียงพอแล้ว

- ไอซีที่ใช้ในการขับสเต็ปมอเตอร์ เราใช้ไอซีเบอร์ ULN 2003 ซึ่งใช้ไฟเลี้ยงอยู่ที่ 12 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาเหตที่เราใช้ไอซีเบอร์นี้ก็เพราะว่าหาซื้อง่าย และราคาถูก

- ไอซีที่ใช้ติดต่อกับ serial port เราใช้ไอซีของ MAX เบอร์ 232 ซึ่งเป็นที่นิยมใช้กันอย่างแพร่หลาย หาซื้อง่ายและราคาไม่แพง

6.2 การออกแบบวงจร

ในขั้นตอนนี้เราจะต้องทำการออกแบบวงจรที่ใช้ในการทดลองซึ่งเราจะใช้โปรแกรม protel ในการออกแบบ ซึ่งวงจรที่เราได้ออกแบบไว้แล้วนั้นจะมีลักษณะดังรูปที่ 6.1

6.3 การทำงานของวงจร

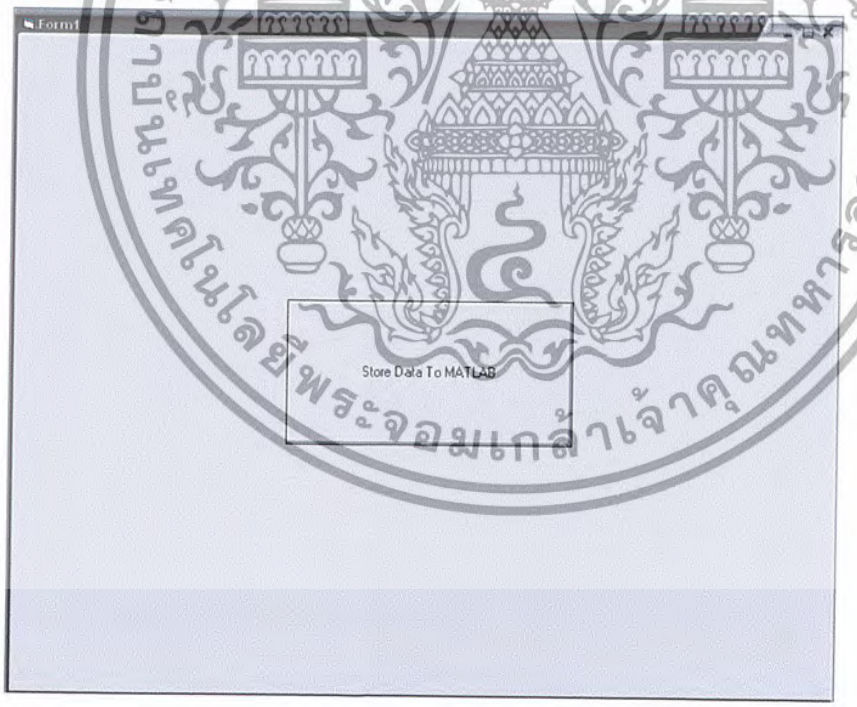
การทำงานของวงจรดังกล่าวเริ่มจากการที่เราสั่งงานให้ไมโครคอนโทลเลอร์เริ่มต้นการอ่านอุณหภูมิแล้วนำมาแสดงที่หน้าจอผ่านทางโปรแกรม Hyper Terminal ซึ่งในการอ่านอุณหภูมินั้นเราจะอ่านทีละตัว หลังจากที่อ่านอุณหภูมิเรียบร้อยแล้วไมโครคอนโทลเลอร์ก็จะไปสั่งให้สเต็ปมอเตอร์หมุนโดยการส่งค่าลอจิก 1 ไปที่พอร์ต 0.0 - 0.3 วนไปเรื่อย ๆ ทำให้แผงอ่านอุณหภูมิเคลื่อนที่ไปข้างหน้า 2 เซนติเมตร จากนั้นไมโครคอนโทรเลอร์ก็จะสั่งงานให้เทอร์มิสเตอร์อ่านอุณหภูมิแล้วส่งค่ากลับมาที่ตัวไมโครคอนโทลเลอร์อีกครั้งหนึ่ง โดยจะทำการอ่านทีละตัวเช่นเดิมและเมื่อทำการอ่านอุณหภูมิเสร็จเรียบร้อยแล้วไมโครคอนโทลเลอร์ก็จะไปสั่งงานให้สเต็ปมอเตอร์หมุนเพื่อที่จะขับเคลื่อนแผงอ่านอุณหภูมิไปอีก 2 เซนติเมตรเช่นเดิม ซึ่งก็จะทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะสิ้นสุดระยะทางตามที่เรากำหนดไว้ หลังจากที่ทำการอ่านอุณหภูมิจนถึงระยะทางที่เรากำหนดไว้แล้วไมโครคอนโทลเลอร์ก็จะขับสเต็ปมอเตอร์ผ่านทางไอซีเบอร์ ULN 2003 ให้เคลื่อนแผงอ่านอุณหภูมิถอยหลังกลับมาที่ตำแหน่งเริ่มต้น เพื่อรอรับคำสั่งที่จะให้มันทำงานต่อหรือไม่ โดยภาษาที่เราใช้ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทลเลอร์เบอร์ 89C51 ก็คือภาษาแอสเซมบลี ซึ่งในการทดลองต่อไปอาจจะเปลี่ยนมาใช้ภาษาซีในการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทลเลอร์เนื่องจากภาษาแอสเซมบลีนั้นค่อนข้างยุ่งยากในการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทลเลอร์แต่ภาษาซีจะมีการใช้งานที่ง่ายกว่า

6.4 การเขียนโปรแกรมแอพพลิเคชั่นเพื่อทำการเก็บข้อมูลและประมวลผลข้อมูลที่เก็บได้

ในโครงงานนี้ได้มุ่งเน้นไปที่การติดต่อสื่อสารเพื่อทำการแสดงการกระจายของข้อมูลอุณหภูมิ จึงมีการใช้โปรแกรมมาช่วยในการเก็บข้อมูลอุณหภูมิที่วัดได้ และประมวลผลข้อมูลออกมาเป็นภาพ ในโครงงานนี้จึงมีการใช้โปรแกรม Visual Basic เพื่อทำการสร้างลักษณะการติดต่อสื่อสารผ่านพอร์ตอนุกรมของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์กับอุปกรณ์ทางฮาร์ดแวร์ภายนอกที่ควบคุมด้วยไมโครคอนโทรลเลอร์ โดยได้มีการใช้คำสั่ง Open "C:\MATLAB6p5\work\d1.dat" For Output As #1 เพื่อเป็นการเปิดไฟล์ข้อมูลชื่อ d1.dat ในไฟล์ work ใน MATLAB เพื่อใช้ในการเก็บข้อมูลเนื่องจากมีการกำหนดให้เป็น Output และเป็นตำแหน่งหน่วยความจำที่ 1 จากการกำหนด As#1 รูปแบบในการติดต่อเป็นการอินเตอร์รัปต์คือเมื่อมีการส่งข้อมูลจากไมโครคอนโทรลเลอร์เข้ามาก็จะเกิดอินเตอร์รัปต์ขึ้นที่โปรแกรม Visual Basic ซึ่งเกิดเนื่องจากโครงสร้างโปรแกรมที่มีการกำหนดคำสั่ง MSComm1.CommEvent ซึ่งจะแสดงว่ามีอินเตอร์รัปต์ขึ้นในโปรแกรมเมื่อมีการส่งข้อมูลผ่านพอร์ตอนุกรมเข้ามา ซึ่งในที่นี้ได้กำหนดการเกิดอินเตอร์รัปต์ในการรับข้อมูลผ่านพอร์ตอนุกรมเข้ามาที่ 48 ไบต์ โดยคำสั่ง MSComm1.RThreshold = 48 ซึ่งจะก่อให้เกิดอินเตอร์รัปต์ว่ามีการรับข้อมูลเข้ามาแล้วเป็นจำนวน 48 ไบต์ จากนั้นหลังจากมีการรับชุดข้อมูลเข้ามาเป็นจำนวน 48 ไบต์แล้วจึงสั่งให้เกิดการตัดเอาข้อมูลจำนวน 3 ไบต์มาทำการเก็บเป็น 16 ตัวข้อมูล ซึ่งข้อมูลตัวละ 3 ไบต์นี้หมายถึงค่าอุณหภูมิในหลักสิบหลักหน่วยและหลักทศนิยม 1 ตำแหน่ง ในเกิดการเก็บข้อมูลเป็นชุดๆจะทำด้วยการใช้คำสั่ง Print #(ตามด้วยหมายเลขไฟล์ที่มีการเปิดเพื่อใช้ในการเก็บข้อมูล) ในที่นี้ได้กำหนดให้เกิดการเก็บข้อมูลระนาบใน 20 เวลาเป็นจำนวน 20 ครั้ง ดังนั้นจึงเกิดการวนการทำงานตามที่ได้กล่าวมาเป็นจำนวน 20 ครั้ง ทำให้เกิดการเก็บข้อมูลจำนวน 20 ชุดไฟล์ข้อมูล



รูปที่ 6.1 ภาพแสดงลักษณะหน้าต่างของโปรแกรม Visual Basic ที่ใช้ในการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

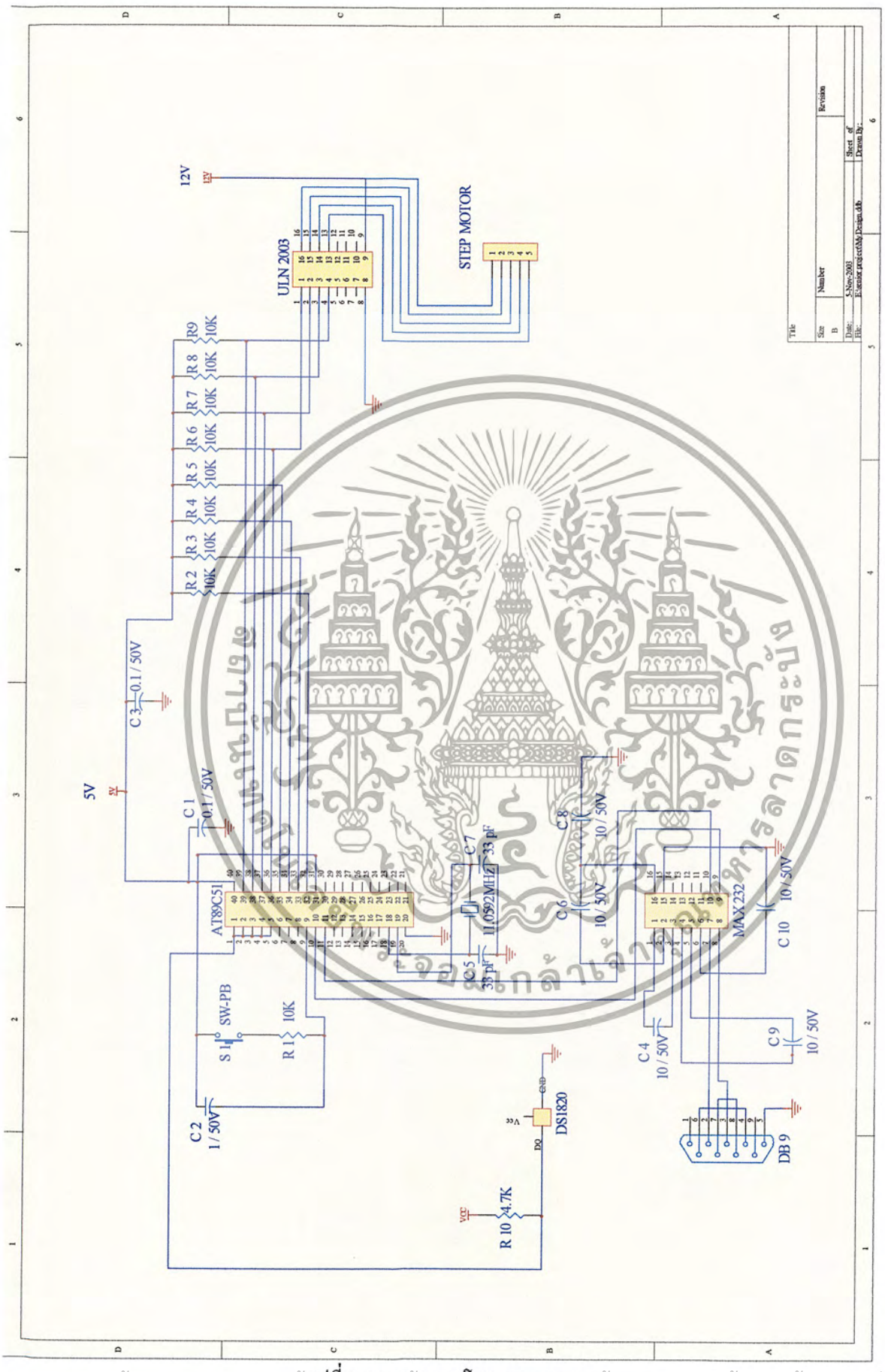
6.5 การประมวลผลภาพโดยใช้ MATLAB

หลังจากที่ได้มีการเก็บข้อมูลเข้ามาเป็นจำนวน 20 ไฟล์แล้ว จะใช้ MATLAB ในการประมวลผล ซึ่งการประมวลผลชุดข้อมูลต่างๆ MATLAB จะต้องทำการโหลดข้อมูลออกมาเป็นชุดๆ เพื่อใช้ในการประมวลผลตามขบวนการต่างๆ ดังนั้นจึงได้มีการทำการโหลดชุดข้อมูลออกมาโดยคำสั่ง Load ตามด้วยชื่อไฟล์ที่ต้องการโหลด จากนั้นจึงต้องใช้ตัวแปรมาทำการรับค่าชุดข้อมูลที่ทำการโหลด เพื่อที่จะนำเอาข้อมูลต่างๆในชุดข้อมูลนี้ใช้ในการคำนวณและทำการประมวลผลทางภาพได้ ในโครงงานนี้ได้มีการนำเอาข้อมูลมาทำการประมวลผลให้มีความละเอียดของข้อมูลมากขึ้น เนื่องจากชุดข้อมูลที่ได้นั้นมาจากการวัดค่าอุณหภูมิจริงที่ตำแหน่งตามลักษณะ matrix 4x4 ซึ่งการวัดจริงเราได้ใช้อุปกรณ์เทอร์มิสเตอร์เพียง 16 ตัว ใน 16 จุดเท่านั้น ทำให้ระหว่างตัวเทอร์มิสเตอร์ ไม่มีค่าข้อมูล จึงใช้การประมาณค่าโดยขบวนการใน MATLAB โดยการใช้ Bilinear Interpolation ให้เกิดค่าประมาณขึ้นระหว่างตัวเทอร์มิสเตอร์เสมือนว่ามีการวัดค่าในช่วงช่องว่างนั้นด้วย ดังนั้นการประมาณค่าที่ได้จะทำให้เกิดข้อมูลที่มีความละเอียดมากขึ้น จากนั้นจึงทำการนำเอาข้อมูลที่ได้ทำการประมาณค่าระหว่างช่วงแล้วไปทำการแสดงเป็นภาพสีที่มีการไล่ลำดับสีโดยถ้าอุณหภูมิที่มีค่าสูงๆก็จะมีสีแดงเข้มโดยสีจะไล่เลี่ยกันลงไปจนถึงสีน้ำเงินที่มีความหมายว่าเป็นอุณหภูมิที่ต่ำที่สุดในชุดข้อมูล จากการประมาณค่าในชุดข้อมูลและประมวลผลภาพสีจะทำให้ได้ภาพสีที่มีการประมาณค่าค่าของอุณหภูมิที่ใช้แสดงสีไล่เลี่ยกัน แสดงให้เห็นถึงลักษณะการกระจายของค่าอุณหภูมิ จากนั้นเมื่อเราทำการประมาณค่าอุณหภูมิและประมวลผลทั้ง 20 ชุดข้อมูล จะทำให้เห็นถึงการกระจายของอุณหภูมิที่เปลี่ยนแปลงไปที่เวลาต่างๆใน 20 ช่วงเวลา ซึ่งทำให้เราสามารถสังเกตถึงการกระจายได้อย่างดี

6.6 ผลการทดลอง

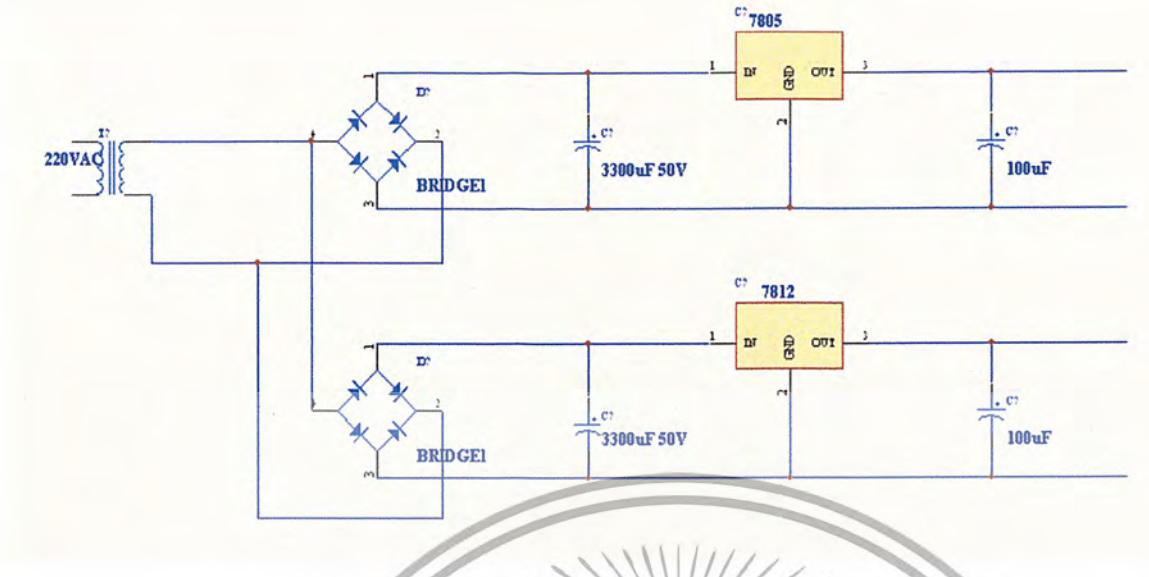
ในการทดลองโครงงานนี้ เราได้เริ่มต้นด้วยการนำโปรแกรมที่เราเขียนขึ้นมาใส่ในไมโครคอนโทลเลอร์จากนั้นก็เปิดโปรแกรม Hyper Terminal เสร็จแล้วก็จ่ายไฟให้กับวงจรโดยที่ในการจ่ายไฟนั้นเราต้องจ่ายไฟ +5 โวลท์ให้กับไมโครคอนโทลเลอร์และเทอร์มิสเตอร์ DS1820 และจ่ายไฟ +12 โวลท์ให้กับไอซี ULN 2003 และสเต็ปมอเตอร์ หลังจากเราจ่ายไฟแล้วเราก็เริ่มทำการทดลอง โดยเริ่มให้คอนโทรลเลอร์ทำการสั่งให้เทอร์มิสเตอร์ทั้ง 16 ตัวทำการส่งข้อมูลเข้ามาทางพอร์ตอนุกรม จากนั้น Visual Basic จะเริ่มทำการเก็บชุดข้อมูลแต่ละชุดลงในไฟล์ข้อมูลเพื่อจะนำไปทำการประมวลผลออกมาเป็นภาพลักษณะการกระจายอุณหภูมิทาง MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

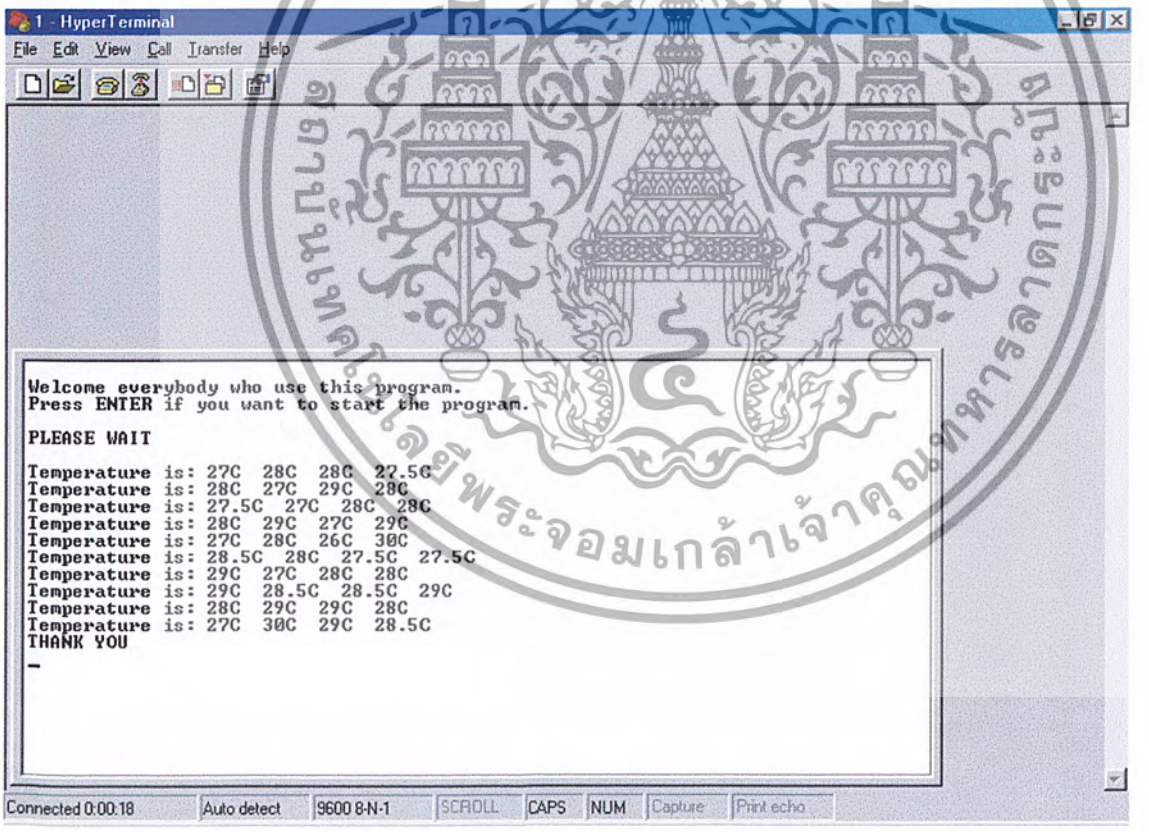


รูปที่ 6.2 วงจรคอนโทรลdetector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

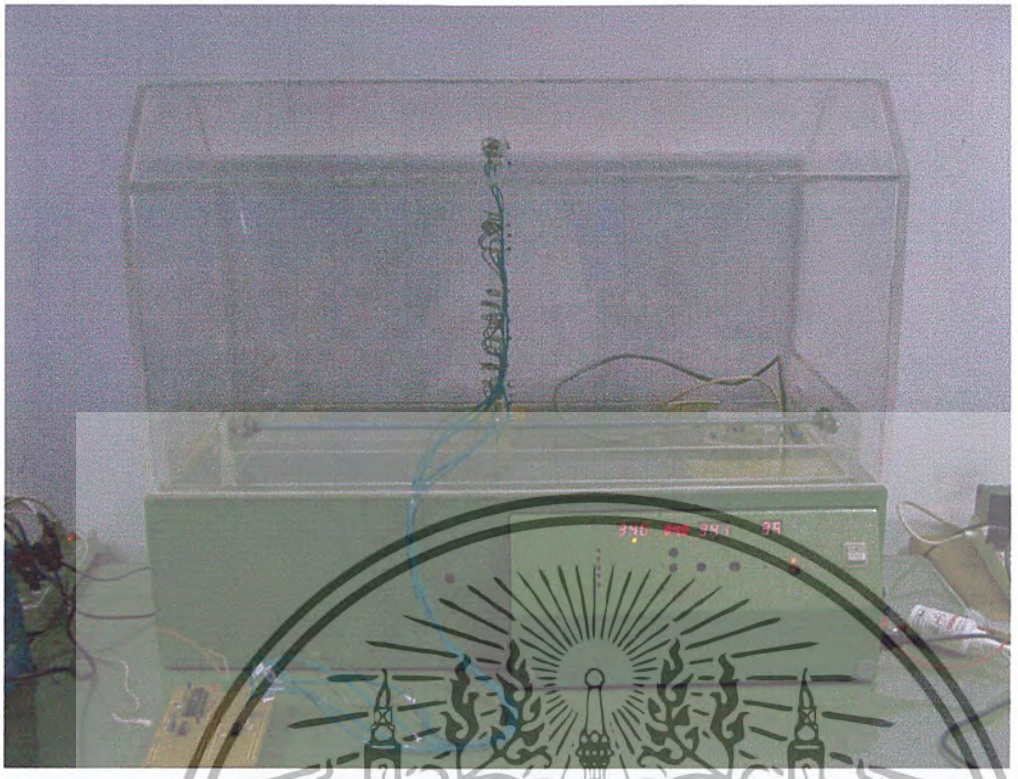


รูปที่ 6.3 วงจรจ่ายไฟเลี้ยงให้วงจรคอนโทรล



รูปที่ 6.4 ผลการทดลองโดยใช้โปรแกรม Hyper Terminal

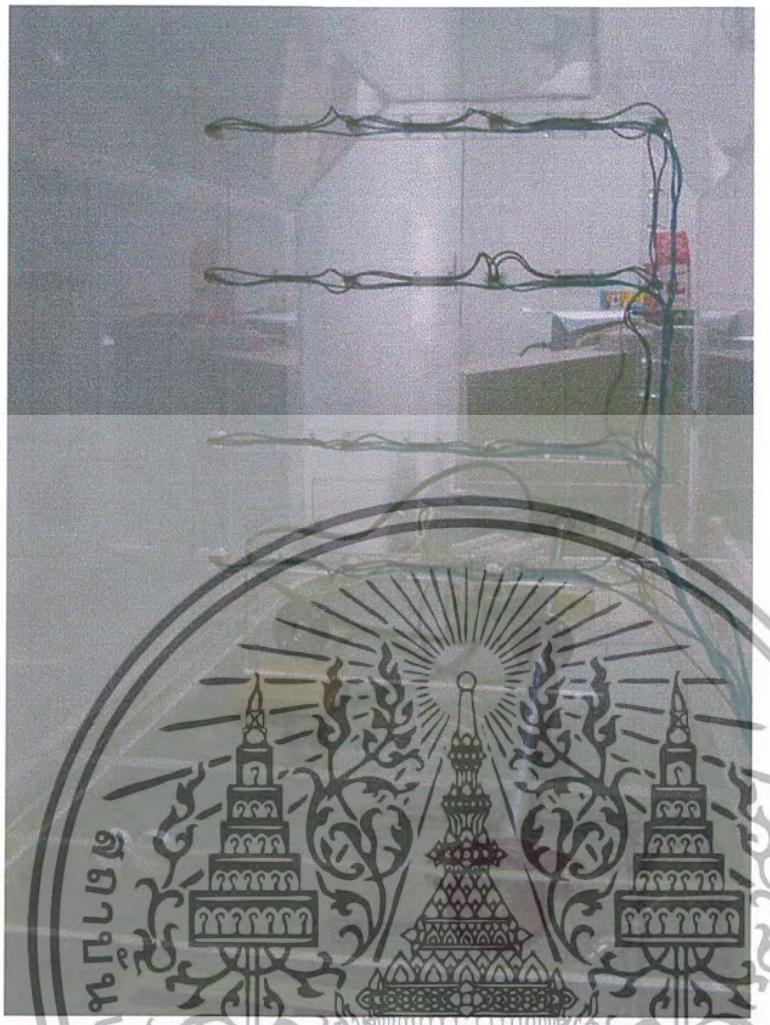
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 แสดงแผงวัดอุณหภูมิที่กำลังทำการวัดอุณหภูมิภายในตู้อบเค็กทารก

รูปที่ 6.6 แสดงแผงวัดอุณหภูมิที่กำลังทำการวัดอุณหภูมิภายในตู้อบเค็กทารก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

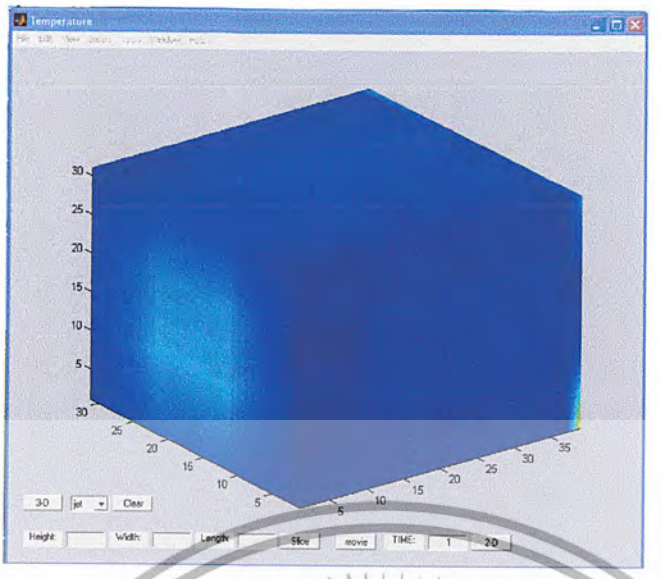


รูปที่ 6.7 แสดงแผงวัดอุณหภูมิที่กำลังทำการวัดอุณหภูมิภายในตู้อบเด็กทารก

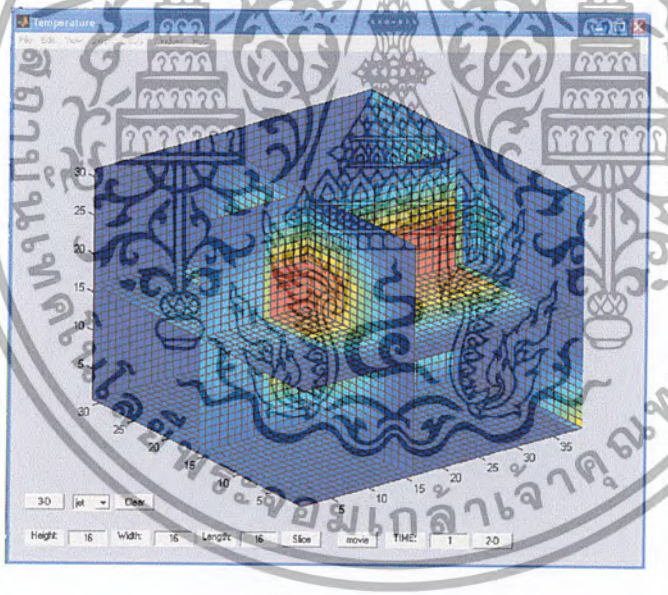
จากการทำการวัดอุณหภูมิในลักษณะต่างๆทั้งที่เป็นระบบปิดและระบบเปิดจะทำให้เราสามารถแสดงภาพจากข้อมูลต่างๆได้ดังนี้

1. การวัดอุณหภูมิโดยวางระนาบวัดอุณหภูมิในแนวระนาบพื้นและใช้หัวบัดกรีมาทำการจ่อตรงกลางระนาบเป็นมุมฉากกับระนาบวัดอุณหภูมิที่ระยะความห่างต่างๆ โดยไล่จากห่างมากเข้ามาจนใกล้ระนาบ แล้วทำการวัดอุณหภูมิ เก็บค่าอุณหภูมิที่วัดได้เป็นจำนวน 20 ครั้ง และทำการประมวลผลภาพออกมาทาง MATLAB จะแสดงภาพข้อมูลได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



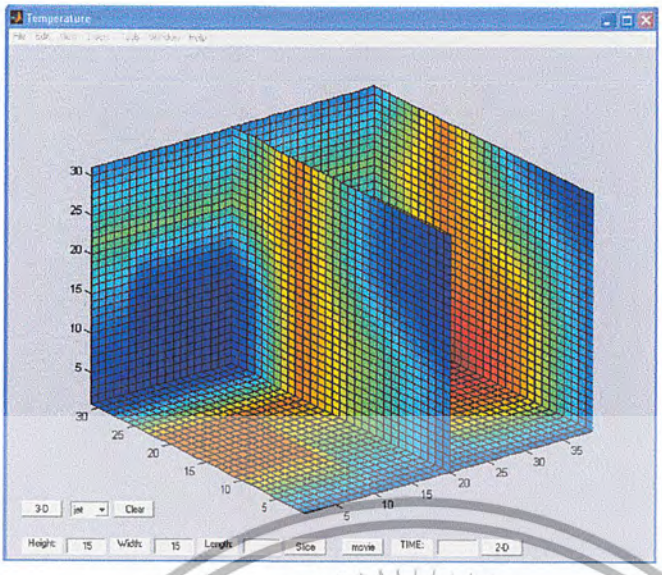
รูปที่ 6.8 แสดงภาพการกระจายอุณหภูมิที่ตำแหน่งต่างๆจำนวน 20 ตำแหน่ง โดยนำภาพทุกๆตำแหน่งมาซ้อนกันเป็นลำดับต่อๆกัน



รูปที่ 6.9 แสดงภาพตัดแกนการกระจายอุณหภูมิ

2. การวัดอุณหภูมิโดยวางระนาบวัดอุณหภูมิในแนวระนาบตั้งฉากกับพื้นและใช้หัวบัดกรีมาทำการวางด้านล่างสุดจ่อตรงเป็นมุมฉากกับระนาบวัดอุณหภูมิที่ระยะความห่างต่างๆ โดยไล่จากห่างมากเข้ามาจนใกล้ระนาบ แล้วทำการวัดอุณหภูมิ เก็บค่าอุณหภูมิที่วัดได้เป็นจำนวน 20 ครั้ง และทำการประมวลผลภาพออกมาทาง MATLAB จะแสดงภาพข้อมูลได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

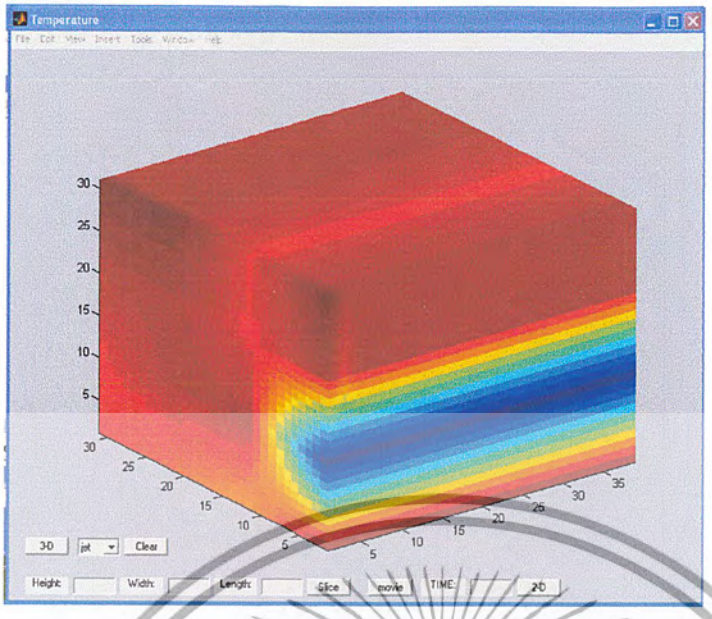


รูปที่ 6.10 แสดงภาพตัดแกนการกระจายอุณหภูมิที่ตำแหน่งต่างๆ

รูปที่ 6.11 แสดงภาพการกระจายอุณหภูมิในระนาบ

3. การวัดอุณหภูมิในตู้อบเด็กทารกที่มีการกำหนดอุณภูมิไว้คงที่ โดยที่จากภาพจะเห็นว่ามีส่วนที่เป็นสีน้ำเงินด้วยเนื่องมาจากตู้ไม่สนิท เพราะมีการใช้สายส่งข้อมูลจากเทอร์มิสเตอร์ทั้ง16ตัวเป็นจำนวน16เส้นเมื่อทำการปิดต์ทับสายจึงเกิดบริเวณที่อากาศเย็นเข้าไปในตำแหน่งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.12 แสดงภาพการกระจายอุณหภูมิใน 20 ตำแหน่งเวลา

รูปที่ 6.13 แสดงภาพการกระจายอุณหภูมิในระนาบการวัด

6.4 สรุปผลการทดลอง

จากผลการทดลองที่ได้มาแสดงให้เห็นได้ว่าโปรแกรมที่เราเขียนขึ้นมานั้นสามารถที่จะทำงานได้เป็นอย่างดี และความต้องการหลักในการที่จะวัดการกระจายอุณหภูมิภายในตู้อบเด็กทารกแสดงผลออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เห็นว่าภายในตู้มีการกระจายอุณหภูมิที่ค่อนข้างสม่ำเสมอกัน ซึ่งเป็นผลที่ดีในการที่จะวัดการทำงานที่มีประสิทธิภาพของตัวตู้อบทารก แต่เนื่องจากในชุดขับเคลื่อนแผงอ่านอุณหภูมิที่เราใช้ในการทดลองค่อนข้างจะมีปัญหาในการขับเคลื่อนจึงทำให้ไม่สามารถเคลื่อนที่ได้ดีเท่าที่ควร ซึ่งในบางครั้งก็ไม่สามารถเคลื่อนที่ได้เลย ทำให้การที่จะวัดการกระจายทั่วทั้งตู้นั้นยังไม่ประสบผลสำเร็จ และในการทดลองช่วงแรก ๆ มีปัญหาเกี่ยวกับการสื่อสารผ่านพอร์ตอนุกรมกับคอมพิวเตอร์ทำให้ต้องเสียเวลาในการแก้ปัญหาเป็นเวลานาน และในการทดลองช่วงหลัง ๆ ของการทดลองได้ประสบปัญหาเกี่ยวกับอุปกรณ์ต่าง ๆ เป็นอย่างมาก ทำให้ในบางครั้งไม่สามารถที่จะทำการทดลองให้ได้ผลตามที่ต้องการได้ทั้ง ๆ ที่โปรแกรมที่ใช้ในการทดลองก็เป็นโปรแกรมเดียวกับที่ใช้ในการทดลองตอนแรก ซึ่งข้อบกพร่องและข้อผิดพลาดต่าง ๆ ที่ผ่านมาในการทดลองจะถูกปรับปรุงและแก้ไขให้ดีขึ้นต่อไป

6.6 แนวทางการพัฒนา

- เปลี่ยนชุดขับเคลื่อนให้ดีขึ้นกว่าเดิม
- เปลี่ยนสเต็ปมอเตอร์ให้มีขนาดใหญ่ขึ้น
- เพิ่มจำนวนเทอร์มิสเตอร์ให้มากขึ้นเพื่อเพิ่มพื้นที่ในการอ่านอุณหภูมิให้มากขึ้น
- ปรับปรุงเปลี่ยนแปลงโปรแกรมให้ดีขึ้นกว่าเดิม
- เพิ่มความเร็วในการอ่านอุณหภูมิของ DS1820 ให้เร็วขึ้นโดยใช้ไมโครคอนโทรเลอร์เป็นตัวควบคุม
-ปรับปรุงแอพพลิเคชั่นโปรแกรมให้มีความยืดหยุ่นในการใช้งานให้มากขึ้น
-ปรับปรุงการแสดงภาพให้มีความละเอียดและบอกถึงค่าอุณหภูมิและตำแหน่งต่างๆให้ละเอียดยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จได้ด้วยดีเนื่องมาจากได้รับความอนุเคราะห์จากบุคคลหลายท่าน โดยเฉพาะท่านอาจารย์ที่ปรึกษา ดร.ชูชาติ ปิณฑวิรุจน์ ที่กรุณาให้คำแนะนำในการแก้ปัญหาต่างๆมาโดยตลอดจนกระทั่งโครงงานชิ้นนี้สำเร็จ ลุล่วงไปได้ด้วยดี รวมทั้งอาจารย์ท่านอื่น ๆ ที่ให้คำแนะนำต่างๆ ขอบพระคุณคุณพ่อ คุณแม่และครอบครัวที่คอยเป็นแรงผลักดันตลอดมาจนสำเร็จการศึกษาและคอยเป็นกำลังใจช่วยให้มีพลังใจในการทำโครงงานนี้ ขอบคุณเพื่อน ๆ ทุกคนที่ได้ช่วยเหลือและคอยเป็นกำลังใจตลอดมา และขอขอบคุณทุกท่านที่มีส่วนเกี่ยวข้องทุกๆท่านมา ณ โอกาสนี้ด้วย

นาย ณัฐพล มโนปิยอนันต์

นาย ณัฐวุฒิ แสงอินทร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้