

ระบบค้นหาทิศทางคลื่นวิทยุ
RADIO WAVE DIRECTION FINDER SYSTEM



โดย
นายคณิงพจน์ อู๋ปมัย 36014055
นายจิระศักดิ์ ทิพย์อุเทน 36014079
นายชัยสุรติ มหาวินสุ 36014106

อาจารย์ที่ปรึกษา
ผศ.นิภา ธีลารุจิ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

เลขหมู่.....
เลขทะเบียน.....27901
วัน, เดือน, ปี.....26 ส.ย. 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2539

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาทิศทางคลื่นวิทยุ

RADIO WAVE DIRECTION FINDER SYSTEM

ผู้จัดทำ

นายคณิงพจน์ อูปมัย 36014055

นายจิระศักดิ์ ทิพย์อุเทน 36014079

นายชัยสุรติ มหาวันสุ 36014106


.....อาจารย์ที่ปรึกษา
(ผศ. นิกา สิลารัจ)

ระบบค้นหาทิศทางคลื่นวิทยุ

Radio wave direction finder system

โดย

นายคณิงพจน์	อุปมัย	36014055
นายจิระศักดิ์	ทิพย์อุเทน	36014079
นายชัยสุรติ	มหาวันสุ	36014106

บทคัดย่อ

โครงการนี้เป็นการศึกษาทิศทางของสถานีส่งสัญญาณคลื่นวิทยุ FM ซึ่งมีย่านความถี่ 87.5 ถึง 108 MHz หลักการทำงานเราจะใช้สายอากาศแบบยาگیเป็นตัวรับสัญญาณและใช้สเตปป์มอเตอร์เป็นตัวขับสายอากาศให้หมุนไปรอบ ๆ หลังจากที่ได้รับสัญญาณได้จากแต่ละทิศทางก็จะนำมาแสดงบนมอนิเตอร์และนำมาเปรียบเทียบว่าสัญญาณไหนมีความแรงที่สุด สายอากาศก็จะหันไปทิศทางนั้นอีกครั้ง

การทำงานจะใช้ไมโครคอนโทรลเลอร์เบอร์ 8031 เป็นตัวควบคุมการทำงานซึ่งเขียนโดยภาษาแอสเซมบลี และซอฟต์แวร์ที่จะเป็นส่วนแสดงผลบนหน้าจอคอมพิวเตอร์เขียนด้วยภาษา C โดยใช้การควบคุมผ่านพอร์ทอนุกรม

Abstract

The objective of this project is to find the direction of FM radio signal transmitted by the station which have the bandwidth between 87.5 - 108 Mhz. The principle of this project is to use the directional antenna, such as YAGI antenna, and driven by the stepping motor in omni-direction. After the signal level which is received from the antenna in each direction, it will display on the monitor. Having received the signal, the monitor will compare each signal with the others for finding the direction of the highest signal level and the controlling will turn the antenna to that direction.

The controlling part is operated by microtroller NO.8031 which is operated on A5sembly program and the monitor is a personal computer. The moniter program is written in C language that is connected to the controlling part by serial port.

สารบัญ

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 หลักการหาทิศทางในการรับส่งสัญญาณที่ดีที่สุด	2
2.2 หลักการค้นหาตำแหน่งของเครื่องรับส่ง	3
2.3 ทฤษฎีสเตปปีงมอเตอร์	5
2.4 ทฤษฎีการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล	17
2.5 เครื่องรับส่ง	22
2.6 การสื่อสารข้อมูลแบบซิงโครนัส	26
2.7 การสื่อสารข้อมูลแบบอะซิงโครนัส	26
2.8 ระดับลอจิกของ RS-232	27
บทที่ 3 การคำนวณและการออกแบบ	28
3.1 หลักการทำงานของระบบ	28
3.2 ส่วนควบคุมการทำงานของระบบ	29
3.3 หลักการทำงานและการออกแบบบอร์ดไมโครคอนโทรลเลอร์	34
3.4 การรับส่งข้อมูลแบบอนุกรม	40
3.5 ชุดหมุนสายอากาศ	40
3.6 โครงสร้างสายอากาศ	41
บทที่ 4 ผลการทดลอง	45
4.1 การทดลองวงจรสเตปปีงมอเตอร์	45
4.2 การทดลองวงจรีเทคเตอร์และขยายสัญญาณ DC	45
4.3 การทดสอบเพื่อหาระดับสัญญาณของอินพุท	52
4.4 การทดสอบโปรแกรมและการหาทิศทางคลื่น	55
บทที่ 5 บทวิจารณ์และสรุป	59

สารบัญรูป

รูปที่ 2.1	การหาทิศทางเครื่องส่ง	2
รูปที่ 2.2	การหาทิศทางของเครื่องส่งจากกราฟความแรงของสัญญาณ	3
รูปที่ 2.3	การหาตำแหน่งของเครื่องส่งจากเครื่องรับวิทยุ 2 ชุด	4
รูปที่ 2.4	การหาตำแหน่งของเครื่องส่งจากเครื่องรับวิทยุ 3 ชุด	5
รูปที่ 2.5	วงจรกระตุ้นสเตปป์มอเตอร์ 3 เฟส	6
รูปที่ 2.6	ไดอะแกรมของระบบขับเคลื่อนสเตปป์มอเตอร์	6
รูปที่ 2.7	โครงสร้างวาริเอเบิลรีลัคแทนซ์มอเตอร์	7
รูปที่ 2.8	ฟลักซ์แม่เหล็กที่เกิดขึ้น	8
รูปที่ 2.9	โรเตอร์และสเตเตอร์	9
รูปที่ 2.10	หน้าตัดของสเตปป์มอเตอร์ 4 เฟส	9
รูปที่ 2.11	โครงสร้างมอเตอร์แบบไฮบริด	10
รูปที่ 2.12	โครงสร้างของสเตปป์มอเตอร์แบบไฮบริด	10
รูปที่ 2.13	การวางแม่เหล็กตามยาวเพื่อสร้างสนามขั้วเดียวกัน	11
รูปที่ 2.14	การทำงานพื้นฐานของสเตปป์มอเตอร์	12
รูปที่ 2.15	ชนิดของสเตปป์มอเตอร์	13
รูปที่ 2.16	การพันแบบไบฟีลา	13
รูปที่ 2.17	การพันแบบโมโนฟีลา	14
รูปที่ 2.18	วงจรสมมูลย์ของสเตปป์มอเตอร์	15
รูปที่ 2.19	วงจร Diode Suppressor	15
รูปที่ 2.20	วงจร Diode/Resister Suppressor	16
รูปที่ 2.21	วงจร Zener Diode Suppressor	16
รูปที่ 2.22	วงจร Condenser Suppressor	17
รูปที่ 2.23	ADC แบบ flash converters	18
รูปที่ 2.24	ADC แบบ Dual Slope	19
รูปที่ 2.25	ADC แบบ successive approximation	21
รูปที่ 2.26	แผนผังของเครื่องรับ FM	22
รูปที่ 2.27	วงจรขยาย RF และวงจร Mixer ที่ใช้ MOSFET	23
รูปที่ 2.28	วงจร Mixer แบบซิงเกิลบาลานซ์ที่ใช้ MOSFET	24
รูปที่ 2.29	วงจร Mixer แบบอินบาลานซ์ที่ใช้ MOSFET	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.30	หลักการเบื้องต้นของระบบดับเบิลซูเปอร์เฮด	25
รูปที่ 3.1	บล็อกไดอะแกรมของระบบการค้นหาทิศทางของสัญญาณคลื่น	28
รูปที่ 3.2	ส่วนประกอบของส่วนควบคุมการทำงานของระบบ	29
รูปที่ 3.3	วงจรขับสเตปป์มอเตอร์	30
รูปที่ 3.4	บล็อกไดอะแกรมของวงจรดีเทคเตอร์และขยายสัญญาณ DC	30
รูปที่ 3.5	แสดงเครื่องรับ FM แบบดับเบิลคอนเวอร์ชัน	31
รูปที่ 3.6	แสดงวงจรดีเทคเตอร์และวงจขยายสัญญาณ DC	31
รูปที่ 3.7	แสดงวงจรรวมของระบบ	32
รูปที่ 3.8	แสดงวงจรสมมูลย์ของทรานซิสเตอร์ #2N5401	33
รูปที่ 3.9	แสดงวงจรสัญญาณตามจุดต่าง ๆ ของวงจรดีเทคเตอร์	34
รูปที่ 3.10	แสดงการต่อวงจร Reset และวงจรสร้างสัญญาณนาฬิกาที่ 8031	35
รูปที่ 3.11	แสดงการติดต่อระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำสำหรับโปรแกรม	36
รูปที่ 3.12	แสดงการต่อไมโครคอนโทรลเลอร์กับหน่วยความจำสำหรับข้อมูล	37
รูปที่ 3.13	แสดงการต่อระหว่างไมโครคอนโทรลเลอร์กับพอร์ทภายนอกแบบขนาน	37
รูปที่ 3.14	แสดงการต่อใช้งานพอร์ทที่สื่อสารแบบอนุกรม	38
รูปที่ 3.15	แสดงวงจรของไมโครคอนโทรลเลอร์	39
รูปที่ 3.16	แสดงเฟืองชุดหมุนสายอากาศ	41
รูปที่ 3.17	แสดงโครงสร้างของสายอากาศ YAGI 3 Element	42
รูปที่ 4.1	สัญญาณอินพุท 200 mVpp	48
รูปที่ 4.2	สัญญาณความถี่ IF หลังจากผ่านการขยายโดยทรานซิสเตอร์	48
รูปที่ 4.3	สัญญาณ IF ที่ผ่านวงจรดีเทคเตอร์	49
รูปที่ 4.4	สัญญาณเอาต์พุทที่ผ่านการขยายและวงจรกรองทางด้านเอาต์พุทแล้ว	49
รูปที่ 4.5	สัญญาณอินพุท 1 Vpp	50
รูปที่ 4.6	สัญญาณความถี่ IF หลังจากผ่านการขยายโดยทรานซิสเตอร์	50
รูปที่ 4.7	สัญญาณความถี่ IF ที่ผ่านวงจรดีเทคเตอร์	51
รูปที่ 4.8	สัญญาณเอาต์พุทที่ผ่านการขยายและวงจรกรองทางด้านเอาต์พุทแล้ว	51
รูปที่ 4.9	วงจรดีเทคเตอร์และวงจขยายสัญญาณ DC	52
รูปที่ 4.10	กราฟแสดงความสัมพันธ์ระหว่างสัญญาณ IF อินพุท และ เอาต์พุท ของวงจร ADC ในรูปเลขฐานสิบ	54
รูปที่ 4.11	แสดงเมนูสำหรับเลือกการทำงานของโปรแกรม	55
รูปที่ 4.12	แสดงข้อผิดพลาดเมื่อเครื่องคอมพิวเตอร์ไม่สามารถติดต่อกับฮาร์ดแวร์ได้	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.13 แสดงภาพเมื่อเลือกกราฟเส้นตรงขณะกำลังรับสัญญาณ	56
รูปที่ 4.14 แสดงภาพเมื่อเลือกกราฟวงกลมขณะรับสัญญาณ	57
รูปที่ 4.15 แสดงการเลือกฟังก์ชัน Direction Set	57
รูปที่ 4.16 แสดงความผิดพลาดเมื่อป้อนค่ามุมที่ไม่ถูกต้อง	58
รูปที่ 4.17 แสดงการหมุนของสายอากาศไปยังตำแหน่งที่ตั้งค่าไว้	58



บทที่ 1

บทนำ

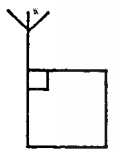
ปัจจุบันสถานีวิทยุตั้งขึ้นมากมาย ซึ่งจะมีผลโดยตรงกับความคมชัดของเสียงในการรับคลื่นของแต่ละสถานีที่ส่งมา บางครั้งเกิดการรบกวนกันของสัญญาณที่ส่งมา ทำให้เกิดปัญหาในการรับสัญญาณ การแก้ไขเบื้องต้นอาจทำได้โดยการหมุนสายอากาศไปตามมุมต่างๆ เพื่อให้รับได้ชัดเจนที่สุด ปัจจุบันเทคโนโลยีด้านคอมพิวเตอร์มีความเจริญก้าวหน้าเป็นอย่างมาก ดังนั้นเราจึงนำข้อดีตรงจุดนี้มาประยุกต์ใช้ในการแก้ปัญหาดังกล่าว โดยแทนที่เราจะไปหมุนสายอากาศเองก็ใช้คอมพิวเตอร์เป็นตัวสั่งการแทน เรา เก็บรวบรวมผลที่รับได้นำผลที่ได้มาเปรียบเทียบกัน เราก็จะได้ค่าความแรงของสัญญาณที่แน่นอน และสามารถบอกได้ว่าสถานีส่งมาจากทางทิศไหน เราควรจะต้องตั้งสายอากาศอย่างไร โดยโครงการนี้จะนำเสนอรูปแบบของการติดตามสัญญาณคลื่นวิทยุ. โดยบทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องที่เราควรจะต้องศึกษาก่อน เพื่อทำความเข้าใจในการปฏิบัติงานของโครงการนี้ บทที่ 3 จะกล่าวถึงการออกแบบและหลักการของโครงการนี้ ว่ามีส่วนต่างๆ อย่างไร บทที่ 4 จะนำเสนอวิธีการทดลองและ ผลการทดลองเพื่อศึกษาเปรียบเทียบกับทฤษฎีว่าสอดคล้องกันหรือไม่ บทที่ 5 กล่าวถึงการประยุกต์ใช้งานและแนวทางในการทำโครงการนี้ต่อไป

บทที่ 2 ทฤษฎีและหลักการ

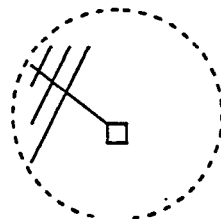
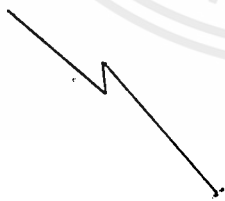
2.1 หลักการหาทิศทางในการรับส่งสัญญาณที่ดีที่สุด

การหาทิศทางในการรับส่งสัญญาณที่ดีที่สุดโดยใช้โครงงานนี้ ทำได้โดยการใช้สายอากาศที่มีทิศทางที่มีมุมในการรับสัญญาณที่แคบพอสมควรติดตั้งไว้บนชุดส่วนขับเคลื่อนสายอากาศเพื่อให้หมุนไปในทิศทางต่าง ๆ ได้ โดยตัวสายอากาศจะถูกควบคุมให้หมุนไปที่ละ 1 ตำแหน่ง ซึ่งก็คือการเปลี่ยนองศาหรือทิศทางกรับของสายอากาศไป 1 ตำแหน่ง พร้อมกันนั้นก็จะมีการรับสัญญาณความแรงจากเครื่องรับวิทยุเข้ามาผ่านการแปลงจากอนาลอกเป็นข้อมูลดิจิทัล แล้วนำไปเก็บไว้คู่กับตำแหน่งสายอากาศนั้น แล้วนำไปสั่งให้หมุนสายอากาศไปตำแหน่งต่อไปแล้วทำเช่นเดิมจนครบรอบหรือครบ 360 องศา แล้วนำข้อมูลความแรงที่เก็บได้ไปประมวลผลเพื่อหาทิศทางที่มีความแรงที่สุด

จากรูปที่ 2.1 จะเห็นว่าเมื่อเครื่องส่ง FM อยู่กับที่แล้ว ถ้าหมุนสายอากาศไปเรื่อย ๆ ทีละตำแหน่งจนครบรอบ (360 องศา) ความแรงของสัญญาณวิทยุที่สายอากาศที่มีทิศทางรับเข้ามาในแต่ละตำแหน่งของสายอากาศจะมีค่าไม่เท่ากัน โดยถ้าสายอากาศหันไปในทิศทางที่ตรงกับเครื่องส่ง ความแรงที่รับได้จะแรงที่สุด และเมื่อสายอากาศหมุนจากตำแหน่งนี้ไป ความแรงสัญญาณก็จะลดลงไป ถ้าเรานำข้อมูลที่ได้มาพล็อตเป็นกราฟความแรงของสัญญาณกับองศาแล้ว กราฟที่ได้สามารถบอกทิศทางของเครื่องส่งให้เราได้ คือ ทิศที่ความแรงของสัญญาณแรงที่สุดนั่นเอง ดังรูปที่ 2.2



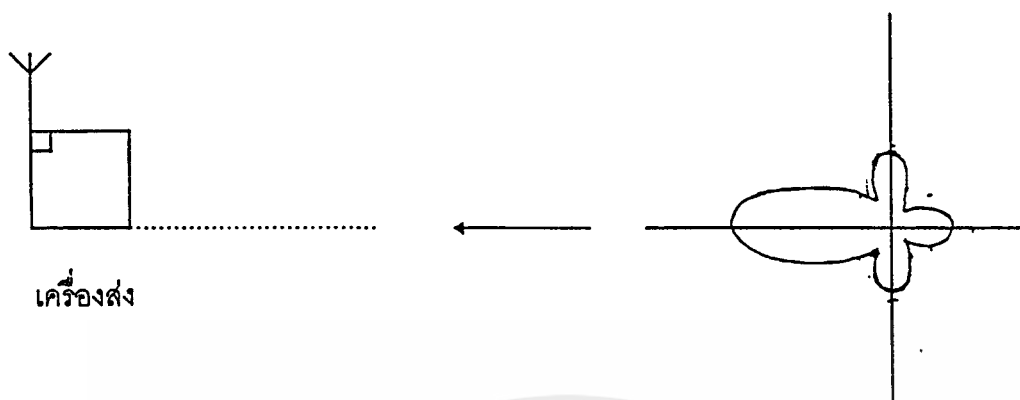
เครื่องส่ง



เครื่องรับ

รูปที่ 2.1 การหาทิศทางเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

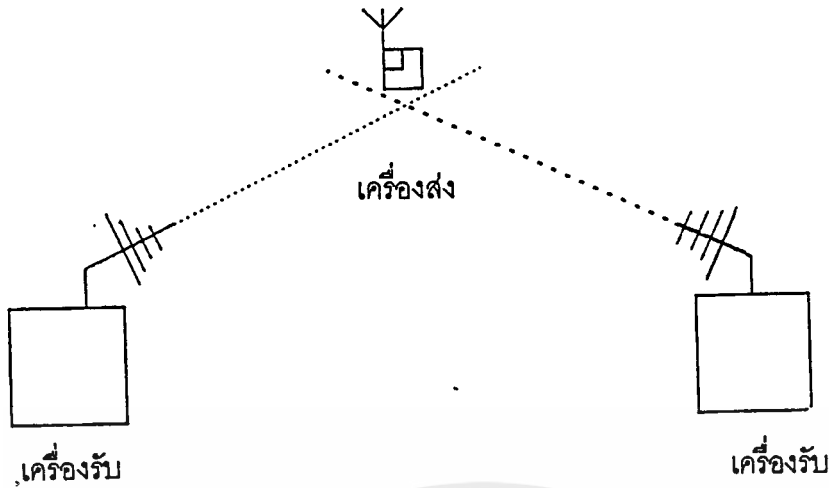


รูปที่ 2.2 การหาทิศทางของเครื่องส่งจากรูปภาพความแรงของสัญญาณ

จะเห็นได้ว่าการหมุนสายอากาศไปที่ละตำแหน่งนั้น ถ้าการหมุน 1 รอบมีจำนวนตำแหน่งมากหมายความว่า องศาที่เปลี่ยนไป 1 ตำแหน่งนั้นมีค่าน้อย ก็จะทำให้การหาทิศทางมีความละเอียดแม่นยำมากขึ้น แต่ก็ยังต้องคำนึงถึงมุมในการรับของสายอากาศด้วยว่าตำแหน่งในการรับสัญญาณมาน้อยเพียงใด ถ้าใช้สายอากาศที่มีบีมหลักที่กว้างการแบ่งจำนวนตำแหน่งที่ละเอียดมาก ๆ ก็จะไม่ค่อยมีผลต่อความแม่นยำในการหาทิศทางเท่าไรนัก เพราะค่าความแรงของสัญญาณที่เข้ามาในตำแหน่งใกล้เคียง ๆ กัน อาจจะมีค่าใกล้เคียงกันมากก็ได้ ทำให้แยกไม่ออกว่าทิศทางไหนมีความแรงกว่ากัน และสายอากาศทิศทางที่ใช้อยู่นั้นก็จะมีค่าความกว้างของบีมหลักอยู่ค่าหนึ่ง ด้วยเหตุนี้จำนวนตำแหน่งที่จะใช้หมุนใน 1 รอบนั้น ก็จะไม่แบ่งละเอียดมากเกินไป

2.2 หลักการการค้นหาค่าตำแหน่งของเครื่องส่งวิทยุ

เป็นการประยุกต์ใช้งานแบบการหาทิศทางของสัญญาณที่แรงที่สุด ดังได้กล่าวไปแล้ว โดยจะนำชุดเครื่องรับดังกล่าวมาจำนวน 2 หรือ 3 ชุด ไปวางในจุดตำแหน่งต่าง ๆ ที่ห่างออกไประยะหนึ่ง ชุดเครื่องรับแต่ละชุด ก็จะทำให้การค้นหาค่าทิศทางที่สัญญาณเข้ามาแรงที่สุด โดยวิธีการที่กล่าวไปแล้ว และนำข้อมูลความแรงของสัญญาณแต่ละชุดมารวมกัน ก็จะสามารถประมวลผลในการประมาณหาค่าตำแหน่งของเครื่องส่งได้ การใช้งานเป็นดังรูปที่ 2.3



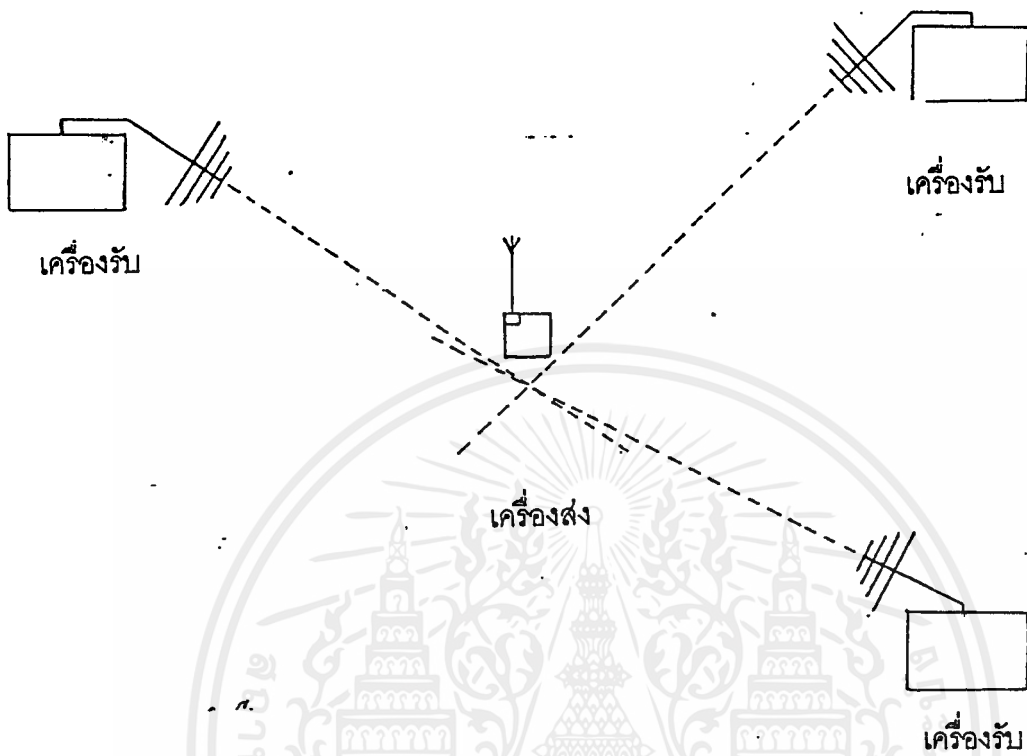
รูปที่ 2.3 การหาตำแหน่งของเครื่องส่งจากเครื่องรับวิทยุ 2 ชุด

จากรูป 2.3 เราใช้ชุดเครื่องรับวิทยุ 2 ชุด โดยแต่ละชุดจะวางอยู่กันคนละตำแหน่งโดยเราจะรู้ตำแหน่งพิกัดที่วางไว้ เครื่องรับแต่ละชุดก็จะทำการกวาดหาสัญญาณวิทยุโดยรอบตัวของมัน แล้วจะมีการส่งข้อมูลที่วัดได้จากเครื่องรับชุดหนึ่งไปยังเครื่องรับอีกชุดหนึ่งโดยในการส่งข้อมูลอาจจะส่งโดยใช้สาย หรืออาจจะใช้การส่งโดยคลื่นวิทยุอีกทีก็ได้หรือวิธีอื่นๆ ตัวประมวลผลจะนำข้อมูลจากชุดเครื่องรับวิทยุ 2 ชุดมาประมวลผลหาตำแหน่งเครื่องส่งต่อไป ถ้าดูจากรูปเราจะลากเส้นตรงจากเครื่องรับของแต่ละชุดไปในทิศทางที่รับสัญญาณที่แรงที่สุดของตัวเอง ก็จะได้เส้นตรง 2 เส้นลากตัดกันจะเกิดจุดตัดขึ้นจุดหนึ่งซึ่งจุด ๆ นั้นก็จะเป็นตำแหน่งที่ตั้งของเครื่องส่งนั่นเอง โดยที่เราสามารถคำนวณหาระยะห่างระหว่างชุดเครื่องรับกับเครื่องส่งได้จากจุดตำแหน่งพิกัดทั้งสอง โดยคอมพิวเตอร์เป็นตัวประมวลผลค่าต่าง ๆ เหล่านี้

เราอาจจะเพิ่มชุดเครื่องรับวิทยุที่จะใช้ในการค้นหาเป็น 3 ชุด ก็จะทำให้การประมวลผลหาตำแหน่งของเครื่องส่งได้ดี ถูกต้องแม่นยำยิ่งขึ้น โดยเครื่องรับแต่ละชุดก็จะวางอยู่ในตำแหน่งที่ห่างกันออกไประยะหนึ่ง โดยเราจะรู้พิกัดตำแหน่งของทั้งสามชุดนี้เช่นกัน ดังรูปที่ 2.4 โดยหลักการเช่นเดียวกับการใช้ชุดเครื่องรับวิทยุ 2 ชุด แต่ใช้เพิ่มเป็น 3 ชุดแทน ข้อมูลความแรงของสัญญาณวิทยุ อีก 2 ชุดจะถูกส่งมายังชุดเครื่องรับอีกชุดหนึ่ง ซึ่งจะนำข้อมูลทั้ง 3 ชุดที่ได้ ประมวลผลหาตำแหน่งเครื่องส่ง โดยการหาจุดตัดของเส้นตรง 3 เส้น เช่น เดิมเราก็จะรู้ตำแหน่งของเครื่องส่งได้

ปัญหาอย่างหนึ่งที่จะเกิดขึ้น อันเนื่องมาจากความกว้างของบีบหลักของสายอากาศและขนาดของเสาที่เปลี่ยนไปในการเคลื่อนที่ 1 ตำแหน่ง จะทำให้การประมวลผลหาตำแหน่งเครื่องส่งจากความแรงสัญญาณที่วัดได้แล้ว จะทำให้ได้ตำแหน่งเครื่องส่งที่ไม่ถูกต้องนัก คือจะออกมาเป็นบริเวณที่กว้างบริเวณหนึ่ง แทนที่จะเป็นจุดตำแหน่งที่แน่นอนไป ฉะนั้นจะเห็นว่า ถ้าบีบหลักของสายอากาศยิ่งแคบเท่าไร ก็จะทำให้การประมวลผลสามารถระบุตำแหน่งเครื่องส่งได้ดี ถูกต้องแม่นยำขึ้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

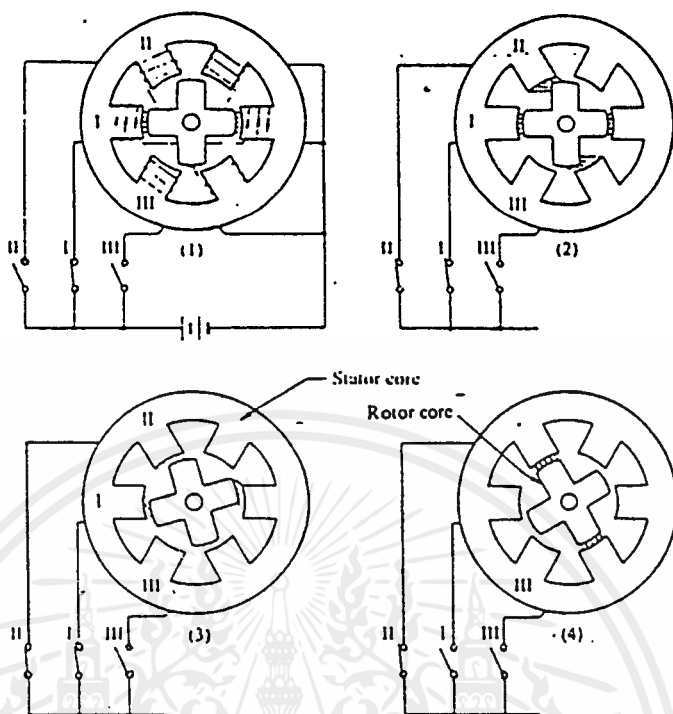


รูปที่ 2.4 การหาตำแหน่งของเครื่องรับส่งจากชุดเครื่องรับวิทยุ 3 ชุด

2.3 ทฤษฎีสเตปป์มอเตอร์

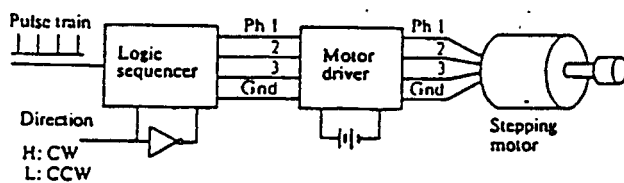
การทำงานของสเตปป์มอเตอร์ จากรูป 2.5 เป็นสเตปป์มอเตอร์แบบวาริเอเบิลรีลัคแตนซ์ (Variable Reluctance) แกนเหล็กสเตเตอร์ (Stator Core) มีซี่ฟัน (Teeth) 6 ซี่ ขณะที่โรเตอร์ (Rotor) มีฟัน 4 ซี่ ทั้งโรเตอร์และสเตเตอร์เป็นเหล็กอ่อน ขดลวด (Coil) 3 ชุด ถูกต่ออยู่ดังรูป แต่ละชุดขดลวดมี 2 ขดลวด ต่ออนุกรมกัน เรียกแต่ละชุดว่าเฟส (Phase) และผลจากการต่อแบบนี้เรียกว่า มอเตอร์ 3 เฟส (3 Phase Motor) กระแสถูกจ่ายไปยังขดลวด แต่ละขดลวดผ่านสวิตช์ I, II และ III ในสภาวะ (1) ขดลวดของเฟส 1 ได้รับกระแสโดยผ่านสวิตช์ I หรือเฟส I ถูกกระตุ้น เส้นแรงแม่เหล็กที่เกิดขึ้นในช่องอากาศ (Air Gap) เกิดขึ้นเนื่องจากการกระตุ้น ซึ่งแสดงด้วยลูกศร ในสภาวะนี้สเตเตอร์ 2 ขั้วของเฟส 1 จะอยู่ในแนวเดียวกับ 2 ซี่ที่อยู่ตรงข้ามกันของโรเตอร์ นี่เป็นสภาวะสมดุลย์ ซึ่งอยู่ในเทอมของไดนามิค (Dynamics) เมื่อสวิตช์ II ปิดเพื่อกระตุ้นเฟส 2 กับเฟส 1 เส้นแรงแม่เหล็กจะถูกสร้างขึ้นที่ขั้วของสเตเตอร์ของเฟส 2 ในลักษณะซึ่งแสดงในสภาวะ (2) ทอร์ก (Torque) ทิศทางทวนเข็มนาฬิกาจะถูกสร้างขึ้นมาจากความเครียด (Tension) ในฟลักซ์แม่เหล็กเอียงไปยังแกนมอเตอร์ที่อยู่ใกล้ หลังจากนั้นโรเตอร์จะมาอยู่ในสภาวะ (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 วงจรกระตุ้นสเต็ปिंगมอเตอร์ 3 เฟส

ดังนั้นโรเตอร์จะหมุนไปด้วยมุมการเปลี่ยนแปลงคงที่ ซึ่งเรียกว่ามุมสเต็ป (Step angle) ในที่นี้คือ 15 องศา ขณะที่สวิตช์จะมีการเปลี่ยนแปลงอีกครั้งหนึ่งคือ สวิตช์ I จะถูกเปิด เพื่อลดพลังงานในเฟส 1 โดยโรเตอร์จะหมุนไป 15 องศา มาอยู่ในสภาวะ (4) ตำแหน่งมุมของโรเตอร์จะถูกควบคุมโดยการเปิด-ปิดสวิตช์ ถ้าสวิตช์ถูกปิด-เปิดเป็นลำดับ โรเตอร์จะหมุนไปในลักษณะที่เป็นสเต็ป ความเร็วเฉลี่ยจะสามารถควบคุมได้ด้วยการเปิด-ปิดสวิตช์ดังแผนภาพรูปที่ 2.5



รูปที่ 2.6 โดอะแกรมของระบบขับสเต็ปึงมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวพอที่จะกล่าวถึงคุณสมบัติสแตปปิงมอเตอร์ได้ว่า

1. การหมุนของมอเตอร์จะเป็นสแตป (เป็นขั้นๆ) สแตปละกี่องศาขึ้นอยู่กับชนิดของมอเตอร์
2. ความเร็วในการหมุนขึ้นอยู่กับสัญญาณพัลส์ที่ให้เข้ามาทางอินพุทของมอเตอร์(ความถี่)
3. ความผิดพลาดในการหมุน 1 สแตปมีค่าน้อยมาก แต่ต้องจำกัดอยู่ในความเร็วที่พอดีด้วย
4. คุณสมบัติของการตอบสนองสัญญาณที่มอเตอร์เริ่มทำงานและหยุดทำงานดีมาก
5. เนื่องจากไม่มีแปรงถ่าน (Commutator) เหมือนมอเตอร์ในระบบดีซี ดังนั้นจึงมีความแน่นอนใน

การทำงานสูง

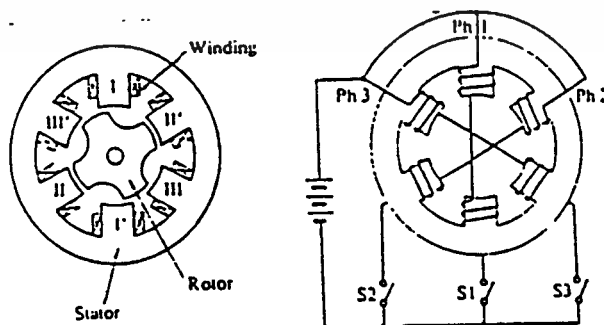
6. แหล่งจ่ายแรงดันไฟที่ใช้ในการขับมอเตอร์มีค่าไม่มาก
7. ไม่ทำให้เกิดเสียงรบกวนและการสั่นได้ง่าย
8. ทำงานแบบ โอเพ่นลูป (Open Loop)

สแตปปิงมอเตอร์ที่ใช้กันส่วนมากมี 3 ชนิด

1. วาไรเอเบิลรีลัคแทนซ์สแตปปิงมอเตอร์ (Variable Reluctance Stepping Motor)
2. สแตปปิงมอเตอร์แบบแม่เหล็กถาวร (Permanent magnetic Stepping Motor)
3. สแตปปิงมอเตอร์แบบไฮบริด (Hybrid Stepping Motor)

2.3.1 วาไรเอเบิลรีลัคแทนซ์สแตปปิงมอเตอร์

มอเตอร์แบบนี้โรเตอร์ทำด้วยเหล็กอ่อน ซึ่งค่าซึมซาบแม่เหล็ก (Permeability) สูงและสามารถให้ฟลักซ์แม่เหล็กผ่านได้มาก โดยโรเตอร์จะติดอยู่กับแกนมอเตอร์ และสเตเตอร์ติดอยู่กับโครงของตัวมอเตอร์ จากรูป 2.7 เป็นภาพตัดขวางของสแตปปิงมอเตอร์แบบนี้ ซึ่งเป็นมอเตอร์ 3 เฟส มีซี่ฟัน 6 ซี่ ฟันของสเตเตอร์จะอยู่ตรงข้ามจะต่อกันเป็นอนุกรม หรือขนานก็ได้ (ในที่นี้ต่อแบบอนุกรม)

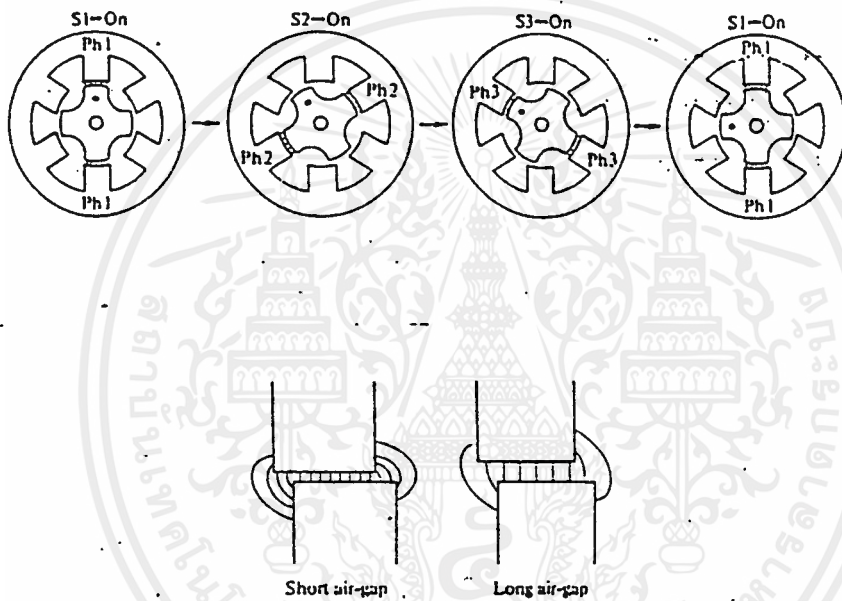


รูปที่ 2.7 โครงสร้างวาไรเอเบิลรีลัคแทนซ์มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะเห็นได้ว่าฟันของสเตเตอร์ 2 ที่มีเฟสเดียวกัน จะมีขั้วแม่เหล็กตรงข้ามกันและกันดังแสดงดังรูป สมมติว่าฟัน I, II และ III มีขั้วเป็นขั้วเหนือ ฟัน I, II และ III จะเป็นขั้วใต้เมื่อถูกกระตุ้น

กระแสแต่ละเฟสจะถูกกระตุ้น ฟลักซ์แม่เหล็กก็จะเกิดดังรูป 2.8 ฟันของโรเตอร์ก็จะมีตำแหน่งในแนวเดียวกันกับฟันของสเตเตอร์ ซึ่งจะมีผลให้แมกเนติกรีลัคแทนซ์ (Magnetic Reluctance) น้อยที่สุด สภาวะนี้คือตำแหน่งสมดุลย์



รูปที่ 2.8 ฟลักซ์แม่เหล็กที่เกิดขึ้น

โครงสร้างเบื้องต้น ของมอเตอร์แบบนี้ จะมีลักษณะดังนี้

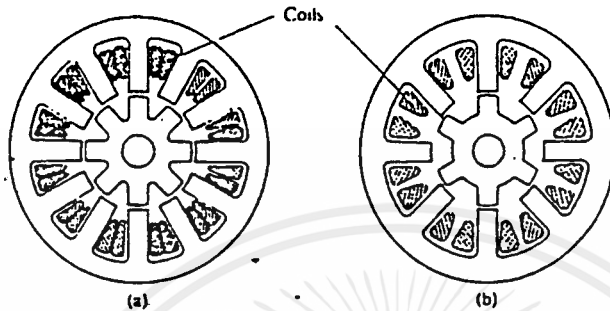
1. ช่องว่างอากาศควรจะมีค่าความห่างกันน้อยมาก เพื่อที่จะได้ทอร์คสูง และตำแหน่งที่แน่นอนขึ้น
2. สำหรับมอเตอร์เล็ก ๆ จากรูป 2.9a แสดง 3 เฟส มอเตอร์ที่สเตเตอร์มีฟัน 12 ซี่ และโรเตอร์มีฟัน 8 ซี่ รูป 2.9b เป็นรูป 4 เฟส มอเตอร์ที่สเตเตอร์มีฟัน 8 ซี่ และโรเตอร์มีฟัน 6 ซี่ ซึ่งทั้งสองรูปนี้มีมุมสเตปเท่ากับ 15 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่างมุมสเตป θ_s , จำนวนเฟส m , โดยจำนวนฟันของโรเตอร์ N_r และจำนวนสเตปใน 1 รอบ S จะหาได้โดย

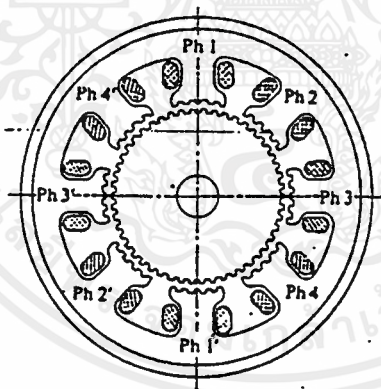
$$S = 360/\theta_s = m(N_r)$$

นั่นคือ เราสามารถลดสเตปลงได้โดยเพิ่มจำนวนฟันบนโรเตอร์ ดังแสดงในรูป 2.10



รูปที่ 2.9 โรเตอร์และสเตเตอร์

รูป 2.10 หน้าตัดของ 4 เฟสมอเตอร์ที่มีฟันโรเตอร์ 50 ซี่ มุมสเตป 1.8 องศา



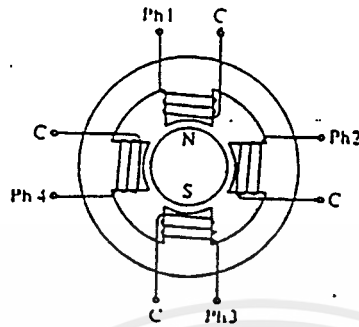
รูปที่ 2.10 หน้าตัดของสเตปปั๊มมอเตอร์ 4 เฟส

2.3.2 สเตปปั๊มมอเตอร์แบบแม่เหล็กถาวร

สเตปปั๊มมอเตอร์แบบนี้ใช้แม่เหล็กแบบถาวร รูปที่ 2.11 เป็นตัวอย่างของสเตปปั๊มมอเตอร์แบบถาวรแบบ 4 เฟส โรเตอร์เป็นทรงกระบอก สเตเตอร์มีฟัน 4 ซี่ โดยที่แต่ละซี่มีขดลวดพันรอบ ถ้าจำนวนซี่บนสเตเตอร์และขั้วแม่เหล็กบนโรเตอร์เพิ่มขึ้นเป็น 2 เท่า มุมแต่ละสเตปลดลงจากเดิมครึ่งหนึ่ง ดังนั้นเพื่อที่จะลดมุมสเตปลงไปอีก ในสเตปปั๊มมอเตอร์แบบแม่เหล็กถาวรจะต้องเพิ่มจำนวนแม่เหล็กและซี่ฟันอย่างไรก็ตาม ขั้วแม่เหล็กที่จะเพิ่มขึ้นได้นั้นมีจำนวนจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

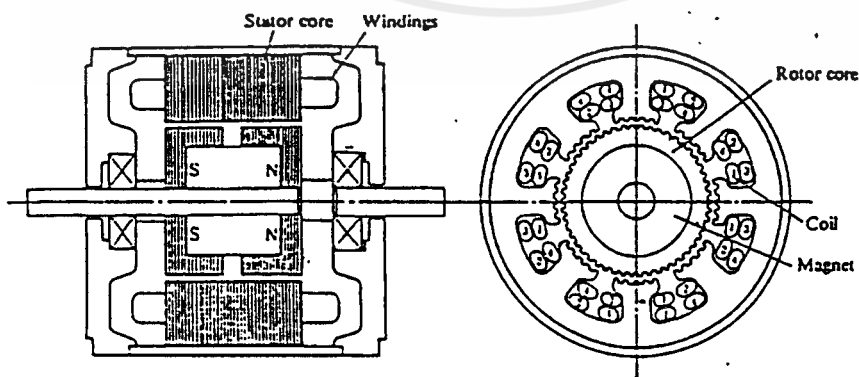
ลักษณะทั่วไปของมอเตอร์แบบนี้ก็คือ โรเตอร์จะถูกยึดอยู่กับที่ แม้ว่าไม่มีการกระตุ้นเฟส ลักษณะเช่นนี้เรียกว่า ดีเทนซ์ แมคคานิคซึม (DETENT MECHANISM)



รูปที่ 2.11 โครงสร้างมอเตอร์แบบไฮบริด

2.3.3 สเตปป์มอเตอร์แบบไฮบริด

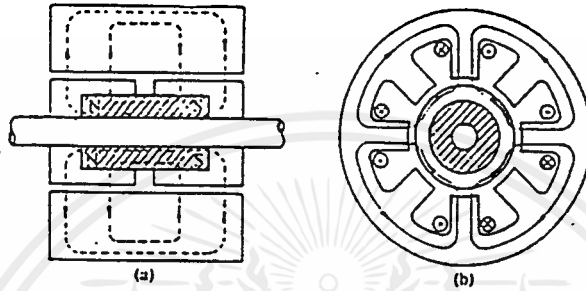
เป็นสเตปป์มอเตอร์แบบหนึ่ง ที่มีโรเตอร์เป็นแบบแม่เหล็กถาวร การใช้ชื่อไฮบริดได้มาจากการรวมหลักสำคัญของมอเตอร์แบบแม่เหล็กถาวรและแบบวาริเอเบิลรีลัคแทนซ์ โครงสร้างแกนของสเตเตอร์จะคล้ายกับแบบวาริเอเบิลรีลัคแทนซ์ แต่การพันและการต่อขดลวดจะต่างจากแบบวาริเอเบิลรีลัคแทนซ์ มอเตอร์ ซึ่งมีเพียง 1 ขดจาก 2 ขดที่ถูกพันเป็นขั้วเดียวในขณะที่ 4 เฟส ไฮบริดมอเตอร์ ขุดขดลวด 2 เฟสที่แตกต่างกันจะถูกพันบนขั้วเดียวกันดังรูป 2.12 เพราะฉะนั้น ขั้วหนึ่งจะไม่มีเพียงเฟสเดียว ขดลวด 2 ขด จะถูกพันเป็นขั้วเดียวกันแบบ ไบฟีลา (Bifilar Winding) ซึ่งจะทำให้ขั้วแม่เหล็กต่างกันขณะมีการกระตุ้น



รูปที่ 2.12 โครงสร้างของสเตปป์มอเตอร์แบบไฮบริด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

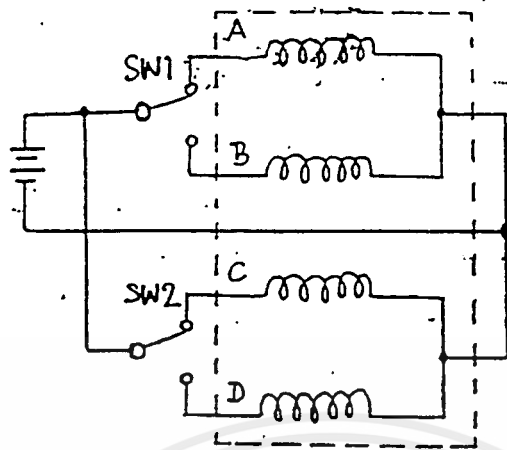
ลักษณะที่สำคัญอีกอย่างหนึ่งของไฮบริดมอเตอร์คือ โรเตอร์นั้นจะเป็นแม่เหล็กรูปร่างทรงกระบอกอยู่ในแกนเหล็กของโรเตอร์ มันถูกทำให้เป็นแม่เหล็กตามยาวเพื่อสร้างสนามขั้วเดียวดังรูปที่ 2.13 แต่ละขั้วของแม่เหล็กจะถูกล้อมรอบด้วยฟันเหล็กอ่อน ซึ่งฟันของโรเตอร์กับสเตเตอร์อยู่เหลื่อมกันอยู่ครึ่งช่วงฟัน



รูปที่ 2.13 การวางแม่เหล็กตามยาวเพื่อสร้างสนามขั้วเดียวกัน.

การทำงานพื้นฐาน

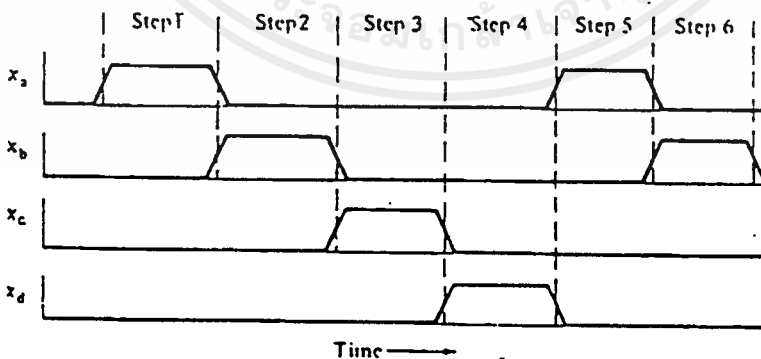
สเตปป์มอเตอร์เป็นมอเตอร์ที่ถูกใช้ทำงานโดยสัญญาณอินพุตที่เป็นพัลส์ ทุกๆ สัญญาณพัลส์ที่ให้เข้ามาจะทำให้การเปลี่ยนแปลงสถานะและสนามแม่เหล็กไฟฟ้าที่เกิดขึ้น และให้การหมุนของมอเตอร์เป็นมุมที่คงที่ ซึ่งจะแตกต่างกับการหมุนของมอเตอร์แบบธรรมดาในระบบควบคุมที่ใช้สัญญาณดิจิตอลนี้ เช่น ในการส่งข้อมูลการควบคุมข้อมูลโดยที่เป็นปริมาณค่าสัญญาณดิจิตอลทั้งหมด จึงทำให้การควบคุมการทำงานของสเตปป์มอเตอร์โดยตรงได้เป็นอย่างดี การทำงานพื้นฐานของสเตปป์มอเตอร์แสดงในรูปที่ 2.14 ซึ่งแสดงบล็อกของวงจรขับเคลื่อนมอเตอร์ ทำให้กระแสไฟ ดีซีเรียงลำดับเข้าไปตามเฟสต่างๆ ของมอเตอร์หมุนไปตามองศาที่กำหนดไว้ การที่จะทำให้กระแสไฟดีซีเรียงลำดับเข้าไปที่มอเตอร์ได้ ทำได้ด้วยการใช้งานของสวิตช์ และเพื่อให้มอเตอร์หมุนจำเป็นต้องมีวงจรขับเคลื่อนสเตปป์มอเตอร์ที่จะควบคุมการไหลของกระแสไฟ คือ สวิตช์ 1 และ 2 ถ้าให้การทำงานของสวิตช์ตามตารางการทำงาน ที่แสดงไว้ในรูปที่ 2.15 ถ้าให้การทำงานโดยใช้รีเลย์(Relay) หรือมือโยก สเตปป์มอเตอร์ก็จะหมุนเหมือนกันแต่หมุนช้า เนื่องจากรีเลย์หรือมือโยกให้การเปลี่ยนแปลงของสวิตช์ช้า ซึ่งจะได้เร็วเท่ากับความเร็วสูงสุดของสัญญาณพัลส์ที่มอเตอร์ทำงานสามารถทำงานได้ (เป็นจำนวนพัลส์ต่อวินาที)



รูปที่ 2.14 การทำงานพื้นฐานของสเต็ปมอเตอร์

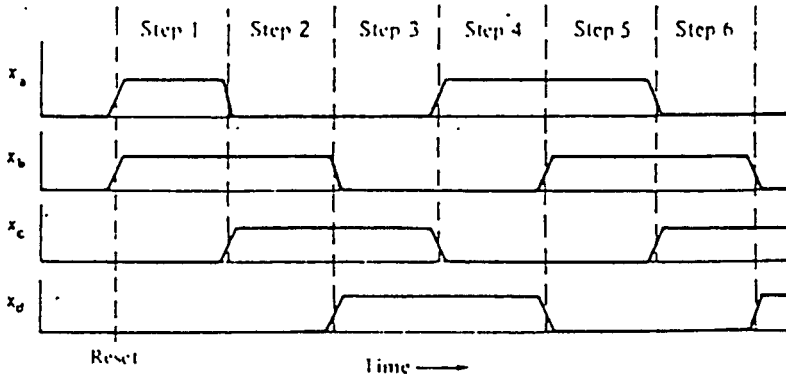
วงจรขับเคลื่อนมอเตอร์มีหลายชนิดขึ้นอยู่กับชนิดสเต็ปมอเตอร์ที่มีอยู่ 3 ระบบคือ

1. ระบบการกระตุ้นแม่เหล็กเฟสเดียว (Single Phase Excitation)
2. ระบบการกระตุ้นสนามแม่เหล็ก 2 เฟส (2-Phase Excitation)
3. ระบบการกระตุ้นสนามแม่เหล็ก 1-2 เฟส (Half-Step Excitation)

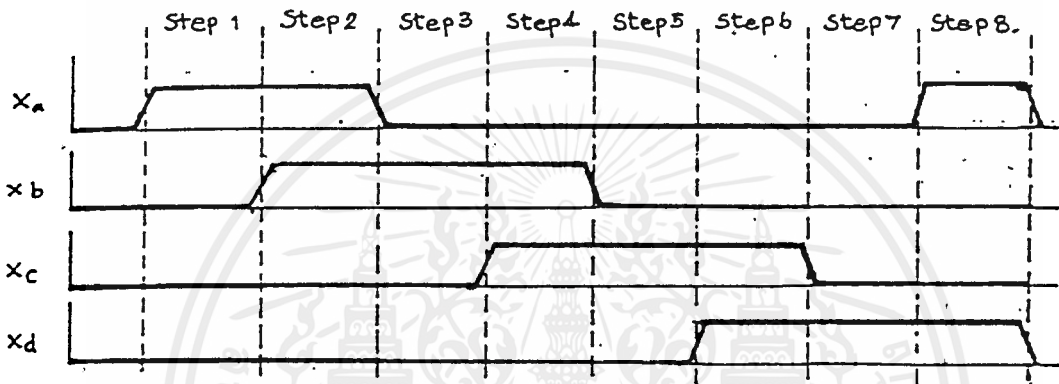


Single Phase Excitation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



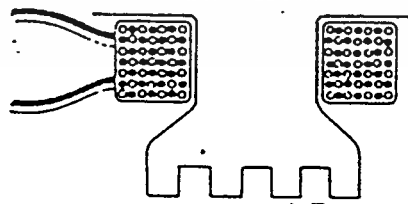
2-Phase Excitation



Half-Step Excitation

รูปที่ 2.15 ชนิดของการขับสเตปป์มอเตอร์

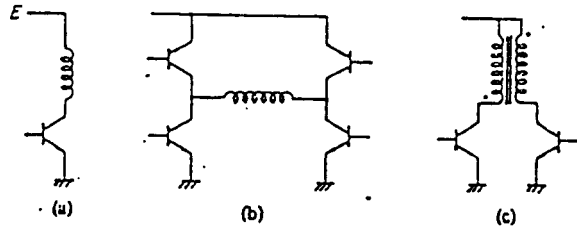
แบบโมโนฟีลา มีการพันลวดเส้นเดียว ส่วนแบบไบฟีลา เส้นลวดทั้ง 2 เส้นนี้จะถูกพันเหมือนกับเส้นเดียวกันชั่วคราวรูป 2.16 และขดลวดทั้งสองเส้นนี้จะแยกกันที่ปลายเป็นลักษณะ 2 เส้นแยกกัน ถ้าขดลวดเป็นของเฟส 1 อีกขดหนึ่งเป็นเฟส 3 และทำนองเดียวกัน ถ้าขดหนึ่งเป็นของเฟส 2 อีกขดหนึ่งเป็นของเฟส 4 (กรณีมอเตอร์ 4 เฟส)



รูปที่ 2.16 การพันแบบไบฟีลา

จุดประสงค์ของการพันแบบไบฟีลา ก็คือ เพื่อให้จะให้พลังงานกับขั้วแม่เหล็กสเตเตอร์โดยการสลับขั้วแม่เหล็ก การกระตุ้นแต่ละเฟสอาจเป็นแบบโดยแบบหนึ่งใน 3 แบบในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การพันแบบโมโนฟีลา

ในวงจรรูปที่ 2.17a เป็นโมโนฟีลา ขั้วแม่เหล็กจะถูกกระตุ้นเป็นขั้วเหนือขั้วใต้เสมอซึ่งแสดงว่าไม่สามารถกลับขั้วแม่เหล็กได้ การกระตุ้นแบบนี้เป็นการกระตุ้นแบบขั้วเดียว (Unipolar Exited)

ในวงจรรูปที่ 2.17b ทิศทางของกระแสในขดสามารถสลับได้เนื่องจากเป็นวงจรบริดจ์ (Bridge Exited)

อย่างไรก็ตามจะต้องใช้ทรานซิสเตอร์ถึง 4 ตัวต่อขดลวด 1 ขด แบบนี้จะทำให้ทอร์คที่ความเร็วต่ำมาก กว่าแบบไบฟีลา แต่ก็มีโอกาสที่ทรานซิสเตอร์จะพังได้เนื่องจากจัดเวลาผิดพลาด

ในวงจรที่ 2.17c เกี่ยวกับคู่สายไบฟีลา และทรานซิสเตอร์ 2 ตัว โดยทำให้สเตเตอร์ถูกกระตุ้นเป็นขั้วแบบไบฟีลา ก็เกิดสนามแม่เหล็กคัปปลิง (Coupling) เมื่อขั้วใดขั้วหนึ่งถูกกระตุ้น ถ้าแทนการพันแบบไบฟีลาด้วยเส้นลวด 2 เส้น ที่แยกจากกัน ความแตกต่างของอินดักแทนซ์ (Inductance) จะปรากฏระหว่าง ขด 2 ขด ทำให้ตำแหน่งผิดไป

โดยทั่วไปประสิทธิภาพของมอเตอร์แบบแม่เหล็กถาวรกับแบบไฮบริด ที่ใช้แบบไบฟีลาจะได้อะสิทธิภาพดีกว่าแบบ โมโนฟีลา

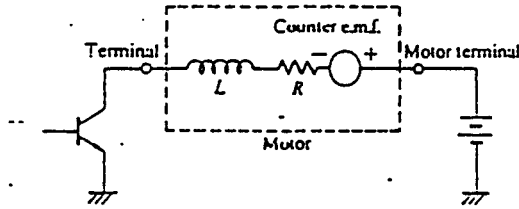
2.3.4 วงจรขับมอเตอร์ (MOTOR DRIVE CIRCUIT)

จุดประสงค์ของวงจรขับคือ เพื่อที่จะให้ได้โวลท์เตจ และกระแสที่ถูกต้องไปยังมอเตอร์ในช่วงเวลาที่สั้น และลักษณะที่มีประสิทธิภาพ ทิศทางของกระแสในขดลวดของมอเตอร์แบบแม่เหล็กถาวร และแบบไฮบริดมีความสำคัญ เพราะจะต้องใช้ในทิศทางที่เหมาะสมในเวลาต่าง ๆ สำหรับการขับมอเตอร์แบบวารีเอเบิลรีลัคแทนซ์ จะไม่ขึ้นกับทิศทางของกระแสที่ไหลในขดลวด เนื่องจากไม่มีแม่เหล็กอยู่ในตัวมันเลย

วงจรขับมอเตอร์จะรับสัญญาณควบคุมมาจากวงจรจัดลำดับลอจิก ในการสร้างวงจรขับมอเตอร์ จะมีปัญหาเกี่ยวกับ อินดักแทนซ์และรีซิสแทนซ์ (Resistance) อนุกรมกันในวงจรสมมูลย์ของมอเตอร์ ดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

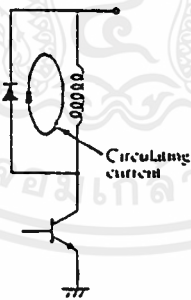
รูปที่ 2.18 อีกทั้งในการหมุนยังมีปัญหาเพิ่มขึ้นเกี่ยวกับแรงดันไฟกลับ(Back Electromotive Force : BACK EMF.) นอกจากนี้ยังต้องคำนึงถึงแหล่งจ่ายไฟ ดีซี และการใช้กับการป้องกัน Power ทรานซิสเตอร์ด้วย



รูปที่ 2.18 วงจรสมมุติของสเต็ปมิ่งมอเตอร์

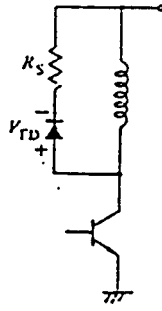
จากรูปที่ 2.18 เมื่อทรานซิสเตอร์หยุดทำงานจะเกิดสไปคโวลท์เตจ(Spike Voltage) เนื่องจากขดลวด (ตัวเหนี่ยวนำ) ซึ่งอาจทำให้ทรานซิสเตอร์พังได้ จึงต้องมีวงจรป้องกันและลดลักษณะแบบนี้อย่างรวดเร็ว โดยวิธีต่อไปนี้

1. ลดโดยใช้ไดโอด (Diode Suppressor) ใช้ไดโอดต่อคร่อมกับสนามกับขดลวดดังรูปที่ 2.19 กระแสจะไหลอยู่ในวงจรขดลวดกับไดโอด หลังจากทรานซิสเตอร์หยุดทำงาน ซึ่งกระแสนี้จะลดลงไปตามเวลา แต่ก็ใช้เวลานานมากจึงยังเกิดทอร์คียืดอยู่



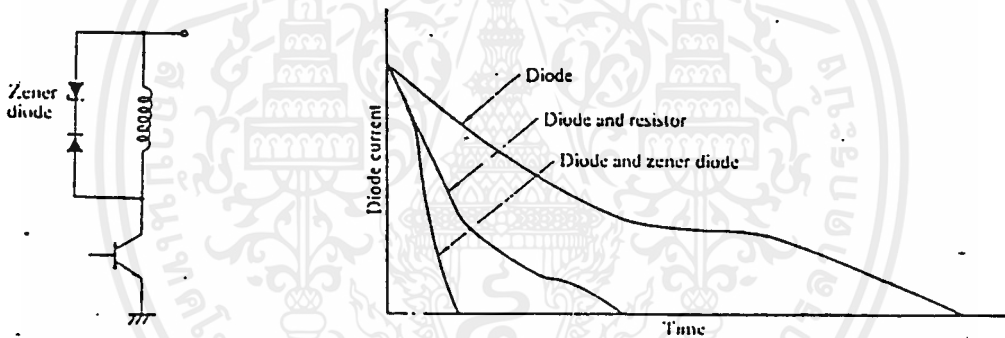
รูปที่ 2.19 วงจร Diode Suppressor

2. โดยใช้ไดโอดและตัวต้านทาน (Diode and Resistor Suppressor) โดยการต่อความต้านทานอนุกรมกับไดโอดแล้วต่อขานานคร่อมขดลวด ดังรูปที่ 2.20 วิธีนี้จะช่วยลดเวลาในการที่จะทำให้กระแสหยุดไหลลง ยิ่งใช้ตัวต้านทานค่ามาก กระแสก็จะหยุดไหลเร็วขึ้น แต่ทรานซิสเตอร์ต้องทนโวลท์เตจคร่อมคอลเลคเตอร์-อิมิตเตอร์ เพิ่มขึ้นด้วย



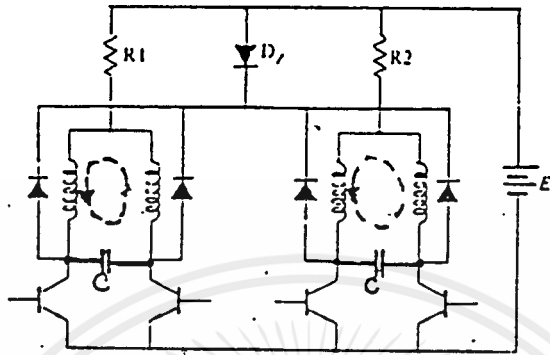
รูปที่ 2.20 วงจร Diode/Resistor Suppressor

3. โดยใช้ซีเนอร์ไดโอด (Zener Diode Suppressor) โดยการต่อซีเนอร์ไดโอดดังรูปที่ 2.21 การต่อแบบนี้เมื่อเปรียบเทียบกับ 2 แบบแรก แล้วจะมีประสิทธิภาพมากกว่า ซึ่งสามารถลดกระแสที่เกิดขึ้นได้เร็วกว่า



รูปที่ 2.21 วงจร Zener Diode Suppressor

4. โดยใช้คอนเดนเซอร์ (Condenser Suppressor) วิธีนี้มักจะใช้กับสเตปป์มอเตอร์แบบไบฟีลา การต่อวงจรดังรูปที่ 2.22 ของมอเตอร์ 4 เฟส ตัวคอนเดนเซอร์ที่ต่ออยู่ระหว่างเฟส 1 กับเฟส 3 และ เฟส 2 กับเฟส 4 จะทำหน้าที่ดังนี้ ในตอนที่ทรานซิสเตอร์ T1 หยุดทำงาน ในการทำงานแบบกระตุ้นเฟสเดียว T2 หรือ T4 เริ่มทำงานแต่ T3 ยังไม่เริ่มทำงาน เนื่องจากเฟส 1 เฟส 3 ต่ออยู่ในลักษณะไบฟีลา กระแสที่เกิดขึ้นหลังจากที่ T1 หยุดทำงานจะไหลวนไปตามเส้นไขปลาดังรูป ถ้า T3 หรือ T4 เริ่มทำงานแต่ T3 ยังไม่เริ่มทำงาน ประจุที่ชาร์จ (Charge) เก็บไว้ในคอนเดนเซอร์จะช่วยเพิ่มทอร์คให้ โดยการปล่อยกระแสไหลผ่านเฟส 1 ทั้งนี้การต่อคอนเดนเซอร์สามารถใช้กับการกระตุ้นแบบ 2 เฟสได้ด้วย สำหรับค่า R_1, R_2 ไลไว้เพื่อปรับกระแสให้เหมาะสม นอกจากนั้นคอนเดนเซอร์ยังช่วยลดการสั่นของมอเตอร์ โดยเปลี่ยนพลังงานกลเป็นพลังงานความร้อนแทน



รูปที่ 2.22 วงจร Condenser Suppressor

2.4 ทฤษฎีการแปลงสัญญาณ Analog เป็นสัญญาณ Digital (ADC)0

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล นั้นเป็นสิ่งจำเป็นมากเพราะจะทำให้เครื่องมือทางดิจิทัลหรือเครื่องคอมพิวเตอร์รับรู้และทำการตอบสนองกับสัญญาณอนาลอกที่เข้ามาได้ ดังนั้นจึงต้องทำการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิทัลเสียก่อน วิธีที่ใช้ในการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิทัลที่สำคัญมีดังนี้คือ

1. แบบ Flash Converters
2. แบบ Dual Slope Converters
3. แบบ Successive Approximation Converters

1. แบบ Flash Converters

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลด้วยวิธี flash converters ได้แสดงไว้ในรูปที่

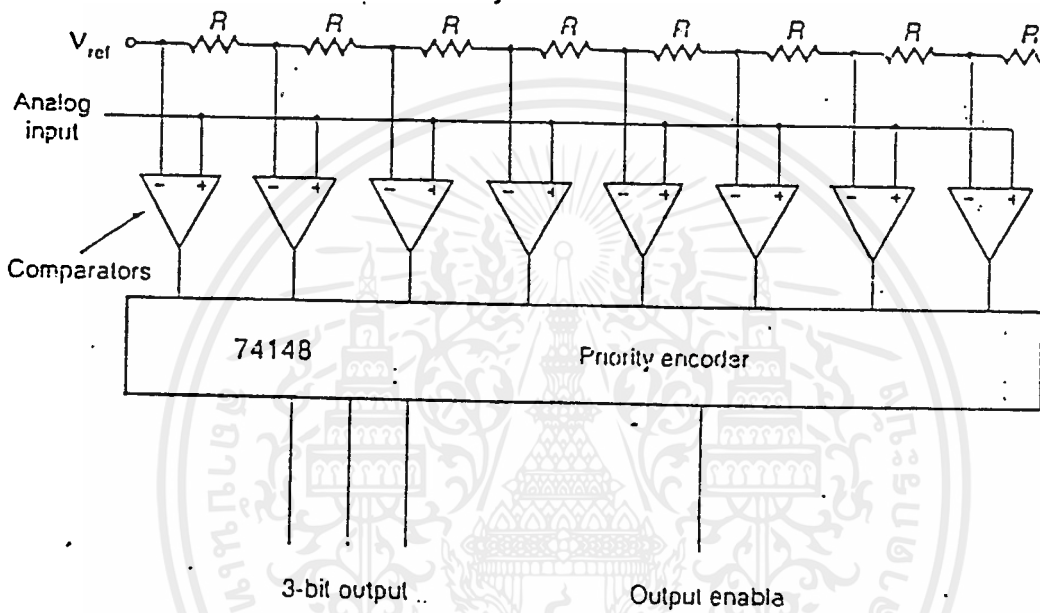
2.23 เป็นวงจร ADC ขนาด 3 บิตมีส่วนประกอบที่สำคัญดังต่อไปนี้

- Resistors Network
- วงจรเปรียบเทียบแรงดัน (Comparators) ใช้ ออปแอมป์ สร้าง
- วงจรเข้ารหัส (Encoder)

หลักการทำงาน ตัวต้านทาน 8 บิตต่อกันเป็นวงจรแบบแรงดัน โดยแบ่งค่าแรงดันของแรงดันอ้างอิงออกเป็น 8 ส่วนถ้า V_{ref} มีค่าเท่ากับ +10 V แรงดันถูกแบ่งออกเป็นค่าต่างๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10,8.75,7.50,6.25,5.00,3.75,2.50,1.25 ค่าของแรงดันที่แบ่งแล้วถูกต่อเข้ากับขา (-) ของ ออปแอมป์ ส่วนขา (+) ของ ออปแอมป์ ต่อถึงกันทุกตัวและยังใช้ขานี้เป็นขาอินพุทโวลต์เดจของวงจรอีกด้วย ออปแอมป์ที่เห็นในวงจรทุกตัวทำหน้าที่เป็นวงจรเปรียบเทียบแรงดันทั้งหมด โดยจะเปรียบเทียบระหว่างแรงดันอินพุทกับแรงดันที่ได้จากการแบ่งของ resistors network สมมติว่าขณะนี้แรงดัน V_{in} มีค่าเท่ากับ 0 V เห็นได้ว่าขณะนี้แรงดันที่ขาบวก (+) ของออปแอมป์ทุกตัวจึงมีค่าเป็น "0" เอาท์พุท ของ ออปแอมป์ มีค่าเป็น "1" ได้ก็ต่อเมื่อแรงดันที่ขา (+) มีมากกว่าแรงดันที่ขา (-) ตารางที่ 2.1 แสดงว่า เอาท์พุท ของ ออปแอมป์ เมื่อ V_{in} มีค่าต่างๆ



รูปที่ 2.23 ADC แบบ flash converters

	Vo1	Vo2	Vo3	Vo4	Vo5	Vo6	Vo7	Vo8
$V_{in} < V1$	0	0	0	0	0	0	0	0
$V2 < V_{in} < V1$	1	0	0	0	0	0	0	0
$V3 < V_{in} < V2$	1	1	0	0	0	0	0	0
$V4 < V_{in} < V3$	1	1	1	0	0	0	0	0
$V5 < V_{in} < V4$	1	1	1	1	0	0	0	0
$V6 < V_{in} < V5$	1	1	1	1	1	0	0	0
$V7 < V_{in} < V6$	1	1	1	1	1	1	0	0
$V8 < V_{in} < V7$	1	1	1	1	1	1	1	0
$V_{in} < V8$	1	1	1	1	1	1	1	1

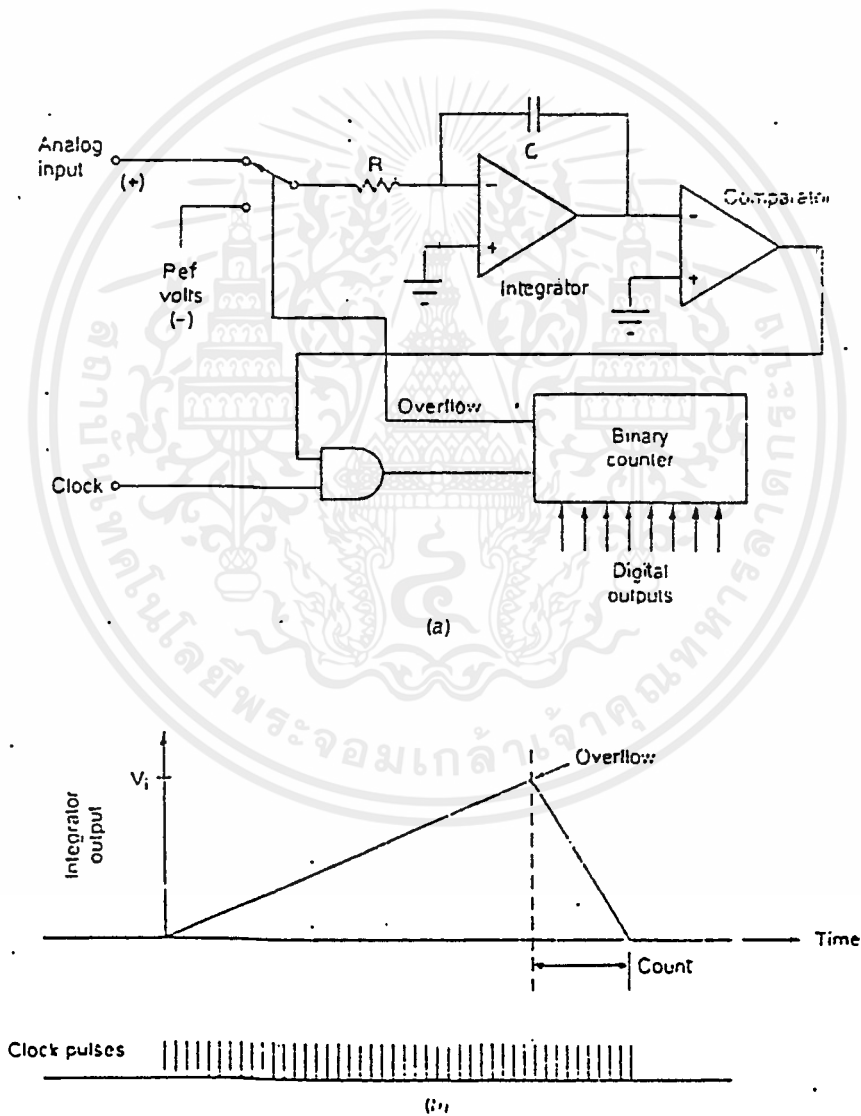
ตารางที่ 2.1 เอาท์พุท ของ ออปแอมป์ ที่ V_{in} ค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเอาท์พุทที่ได้จากออปแอมป์ก็จะถูกส่งให้กับวงจรเข้ารหัสเพื่อเข้ารหัสให้เป็นเลขฐาน 16 ต่อกับ ADC แบบ flash หรือ parallel เป็น ADC ที่มีความเร็วสูงที่สุดเมื่อเทียบกับ ADC ชนิดอื่น และยังมีค่าความเป็นเชิงเส้นดีกว่า ADC ชนิดอื่นมาก ยกตัวอย่างถ้าเอาท์พุทมี 8 บิตจะต้องใช้ออปแอมป์ถึง 225 ตัวหรือ $2^8 - 1$ นั่นเอง

2. Dual Slope Converters

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลแบบดูอัลสโลป แสดงไว้ในรูปที่ 2.24



รูปที่ 2.24 ADC แบบ dual slope (a) circuit diagram (b) timing diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC แบบ dual slope สามารถเขียนการทำงานได้ดังต่อไปนี้

เมื่อเริ่มต้นนับจะถูกทำให้ทุกบิตมีค่าเป็น “0” และอนาล็อกสวิตช์ จะต่อแรงดันอินพุตที่ต้องการ วัดเข้ากับวงจรรีจิสเตอร์ผ่านเข้าไปที่วงจรมับได้ จากนั้นวงจรมับจะนับค่าเพิ่มขึ้นจนกระทั่งทุกบิตมีค่าเป็น “1” ก็เกิดสถานะโอเวอร์โฟลว์

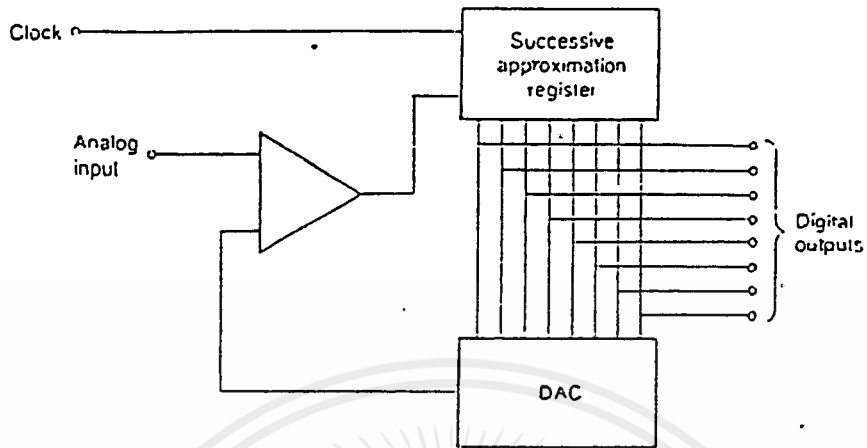
เมื่อเกิดการโอเวอร์โฟลว์ขึ้นอนาล็อกสวิตช์ก็จะเปลี่ยนตำแหน่งมาต่อกับแรงดัน ref (-) แทน ตอนนี้ทุกบิตของตัวนับจะมีค่าเป็น “0” (เปลี่ยนจาก 11111111 เป็น 00000000 นั้นเอง) เมื่ออินพุตของวงจรรีจิสเตอร์ ต่อกับ ref (-) ก็จะทำให้แรงดันมีค่าเอาท์พุทมีค่าลดลง แต่ขณะนี้เอาท์พุทของวงจรรีจิสเตอร์ ยังมีค่าเป็น “1” อยู่วงจรมับคงนับค่าต่อไปโดยเริ่มจาก 00000000 ขึ้นไปเมื่อเอาท์พุทของวงจรรีจิสเตอร์ ลดลงมีค่าเป็น “0” ก็จะทำให้วงจรรีจิสเตอร์หยุดนับ การแปลงสัญญาณได้เสร็จเรียบร้อยแล้ว ค่าที่ปรากฏที่เอาท์พุทของวงจรมับขณะนี้ก็คือค่าดิจิตอลเอาท์พุทนั่นเอง

ADC แบบ dual slope นี้มีข้อดีคือ สามารถสร้างให้มีจำนวนบิตทางเอาท์พุทสูงซึ่งในปัจจุบันนี้สามารถสร้างได้สูงถึง 22 บิต ทำให้ค่าที่แปลงได้มีความเที่ยงตรงและเชื่อถือได้ ดังนั้น ADC ชนิดนี้จึงถูกนำมาใช้งานด้านเครื่องมือวัด เครื่องมือแพทย์ เป็นส่วนใหญ่ แต่ ADC ชนิดนี้มีข้อเสียคือ มีความเร็วในการแปลงสัญญาณต่ำ เบอร์ที่นิยมใช้ขณะนี้ก็คือเบอร์ ICL7106, ICL7107, ICL7109, T500, TL505 เป็นต้น

3. Successive approximation Converters

Successive approximation Converters แสดงดังรูปที่ 2.25 หัวใจของวงจรรีจิสเตอร์ก็คือ DAC และยังมีส่วนควบคุมลอจิก ซึ่งเรียกว่า Successive approximation register (SAR) จะทำให้บิตที่มีนัยสำคัญมากที่สุดเป็น high เมื่อได้รับพัลส์ลจิกแรก ส่วนบิตที่เหลือทั้งหมดจะเป็น low เลขไบนารี จะถูกส่งจาก SAR ไปยัง DAC ทำหน้าที่เปลี่ยนไบนารีเป็นโวลต์เดจ สัญญาณเอาท์พุทของ DAC จะถูกนำไปเปรียบเทียบกับสัญญาณอินพุทอนาล็อก ถ้าแรงดันของ DAC สูงกว่าแรงดันของอินพุทอนาล็อก คอมแพเรเตอร์เอาท์พุทจะเป็น low ไปรีเซ็ตบิตให้เป็น 0 ถ้าแรงดันเอาท์พุทของ DAC น้อยกว่าแรงดันอินพุท MBS จะคงสถานะ 1

เมื่อมี clock ลูกใหม่เข้ามา บิตที่มีนัยสำคัญมากที่สุดถัดมาจะถูกเซ็ตเป็น 1 การเปรียบเทียบเกิดขึ้นเหมือนเดิม จนกระทั่งบิตสุดท้าย (LSB) เป็นอันล้นสุด ที่จุดนี้ SAR จะให้เอาท์พุทเรียกว่า end-of-converters (EOC) จำนวนไบนารีถูกส่งออกที่เอาท์พุทรีจิสเตอร์ของ SAR



รูปที่ 2.25 ADC แบบ successive approximation

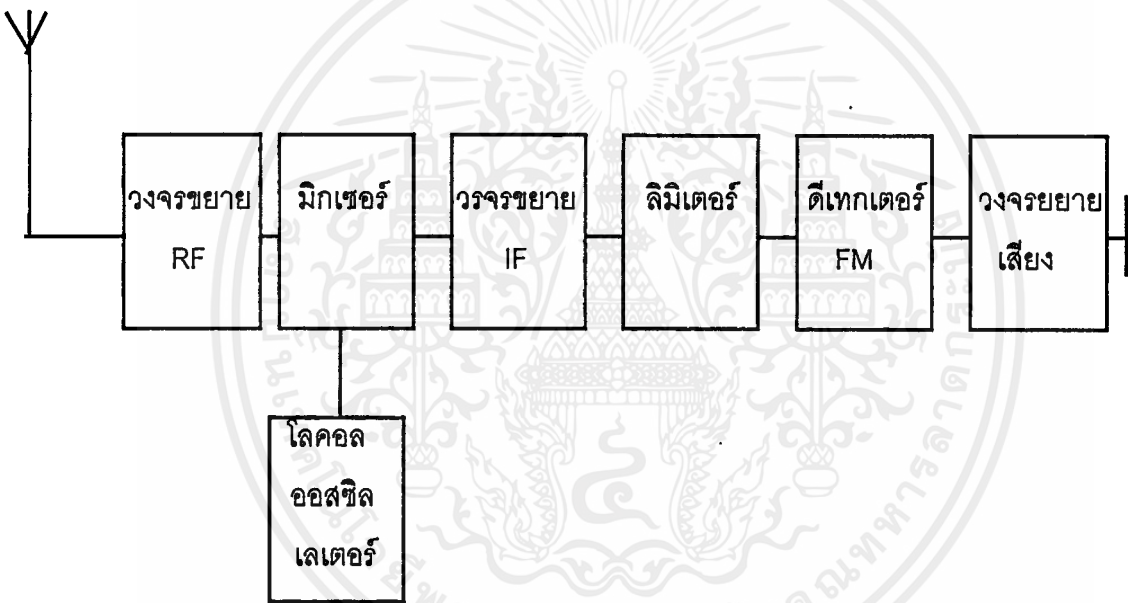
2.5 เครื่องรับวิทยุ

ทฤษฎีเครื่องรับวิทยุ

ระบบเครื่องรับวิทยุมีการวิวัฒนาการมาเรื่อยๆ จากเครื่องรับระบบปริเจนเนอร์ชั่น เป็นเครื่องรับระบบ TRF (Turned Radio Frequency) ซึ่งในปัจจุบันระบบวิทยุแบบนี้อาจจะไม่มีใช้แล้วในที่นี้แล้วในที่นี้จะกล่าวถึงเครื่องรับระบบซูเปอร์เฮเทอโรไดน์ ซึ่งนิยมใช้กันอยู่ในปัจจุบันนี้

2.5.1 เครื่องรับระบบซูเปอร์เฮเทอโรไดน์

คำว่าซูเปอร์เฮเทอโรไดน์ (Superheterodynes) หมายถึงความถี่ที่บิดออกมาจากวงจรเฮเทอโรไดน์นั้นมีความถี่สูงที่หูคนได้ยิน (Supersonic) เพราะอยู่ในย่านความถี่วิทยุแล้ว แทนที่จะเป็นความถี่เสียง โดยระบบซูเปอร์เฮเทอโรไดน์จะมีหลักการเบื้องต้นอยู่ในรูปที่ 2.26

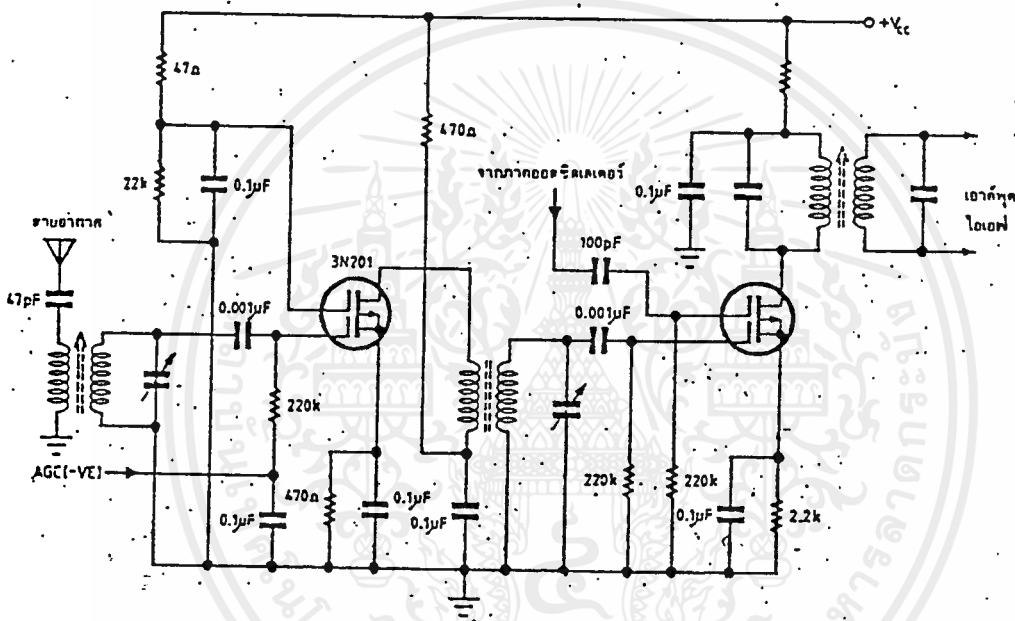


รูปที่ 2.26 แผนผังของเครื่องรับ FM

จากหลักการเบื้องต้นของระบบซูเปอร์เฮเทอโรไดน์ในรูปที่ 2.26 เมื่อเครื่องรับวิทยุผ่านสายอากาศเข้ามา จะถูกเลือกคลื่นวิทยุสถานีที่ต้องการโดยวงจรจูนอันประกอบด้วยขดลวดและตัวเก็บประจุปรับค่าได้ คลื่นวิทยุที่ต้องการจะถูกนำไปขยายให้แรงขึ้น โดยภาคขยายคลื่นวิทยุ (RF Amplifier) ดังรูปที่ 3.2 ในขณะที่ทำการจูนเพื่อเลือกคลื่นวิทยุ จะเป็นการเปลี่ยนความถี่ของภาคกำเนิดความถี่ (Local oscillator) ด้วย เนื่องจากตัวเก็บประจุปรับค่าได้ที่ใช้เป็นแบบแกนร่วม ที่ต้องหมุนไปพร้อมกัน

ถ้าคลื่นวิทยุที่เลือกรับเข้ามามีความถี่ 2445 kHz จะทำให้ภาคกำเนิดความถี่สร้างความถี่ขึ้นมาเท่ากับ 2900 kHz ความถี่ทั้งสองความถี่จะไปผสมกันที่ภาคมิกเซอร์ ดังรูปที่ 2.28 และ 2.29 ได้เป็นความถี่ใหม่ที่ประกอบด้วยความถี่ 4 ความถี่ ได้แก่ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความถี่เดิมแรกเท่ากับ 2445 kHz
2. ความถี่ที่มาผสมมีค่าเท่ากับ 2900 kHz
3. ความถี่ที่เป็นผลรวมของความถี่ทั้งสองเรียกว่า up-converted มีค่าเท่ากับ $2445 + 2900 = 5345$ kHz
4. ความถี่ที่เป็นผลต่างของความถี่ทั้งสอง เรียกว่า ความถี่ปานกลาง (intermediate frequency) ที่เรียกกันว่า สัญญาณ IF นั้นเอง มีค่าเท่ากับ $2900 - 2445 = 455$ kHz

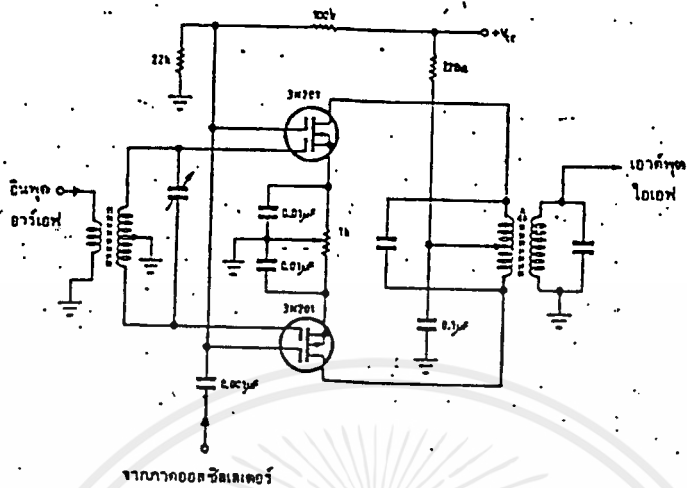


รูปที่ 2.27 วงจรขยาย RF และวงจร Mixer ที่ใช้ MOSFET

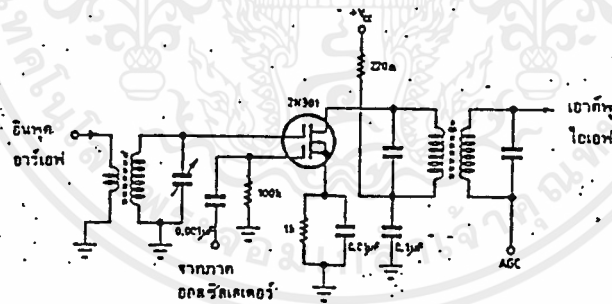
ความถี่ที่ออกจากภาคมิกเซอร์จะถูกส่งต่อไปยังภาคขยายที่ถูกจูนไว้ให้ขยายเฉพาะความถี่ปานกลางอย่างเดียว ดังนั้นความถี่อื่น ๆ คือ ความถี่คลื่นวิทยุที่รับเข้ามา ความถี่ที่ภาคกำเนิดสร้างขึ้นและความถี่ที่เป็นผลรวมก็จะถูกกำจัดออกไป เหลือไว้แต่ความถี่ปานกลางที่ถูกขยายไอเอฟหลาย ๆ ภาคจะยิ่งทำให้การเลือกรับและขยายสัญญาณดีมากยิ่งขึ้น

ความถี่ปานกลางที่ได้จากภาคขยายไอเอฟ จะถูกแปลงเป็นสัญญาณเสียงโดยภาคดีมอดูเลเตอร์ (demodulator) หรือเรียกอีกชื่อหนึ่งว่า ภาคดีเทกเตอร์ (detector) ในภาคดีมอดูเลเตอร์อาจจะประกอบไปด้วย ไดโอดแบบเจอร์เมเนียม และตัวเก็บประจุทำงานร่วมกันโดยสัญญาณเสียงที่ได้ส่งต่อไปขยายให้แรงขึ้นด้วยภาคขยายเสียงหรือสามารถที่จะต่อเข้าหูฟังเพื่อฟังเสียงที่ออกมาโดยตรงก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 วงจร Mixer แบบซิงเกิลบาลานซ์ที่ใช้ MOSFET

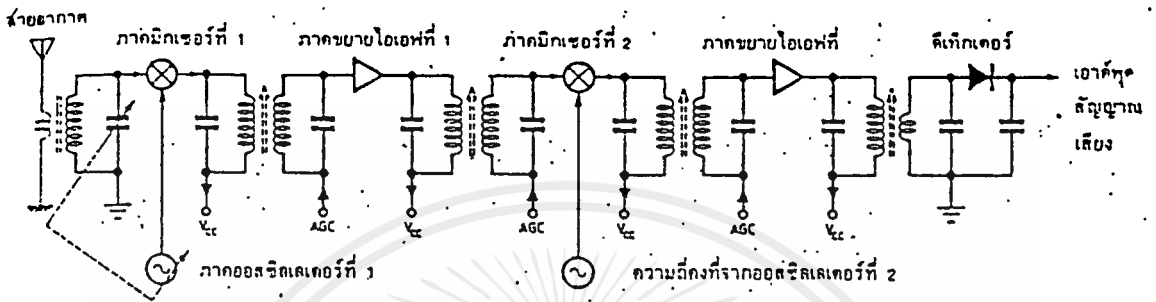


รูปที่ 2.29 วงจร Mixer แบบอินบาลานซ์ที่ใช้ MOSFET

2.5.2 ระบบดับเบิลซูเปอร์เฮเทอโรไดน์

จากปัญหาเรื่องความถี่เสียงที่ทำให้คุณภาพของเครื่องรับระบบซูเปอร์เฮเทอโรไดน์ ก็มีการคิดค้นวิธีที่จะแก้ไขความถี่นี้ โดยการกำหนดค่าความถี่ไอเอฟขึ้นมา 2 ค่า เป็นความถี่ไอเอฟค่าที่ 1 และความถี่ไอเอฟค่าที่ 2 ให้ชื่อระบบนี้ว่า ระบบดับเบิลซูเปอร์เฮเทอโร (Double superhete) ซึ่งมีหลักการดังรูปที่ 2.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 หลักการเบื้องต้นของระบบดับเบิลซูเปอร์เฮต

จากรูปที่ 2.30 วิธีการของระบบดับเบิลซูเปอร์เฮต เมื่อสายอากาศรับเอาคลื่นวิทยุเข้ามา วงจรจูนจะทำการเลือกคลื่นวิทยุสถานีที่ต้องการผ่านเข้าไปที่ภาคมิกเซอร์ 1 ในขณะที่เดียวกันภาคออสซิลเลเตอร์ชุดที่ 1 จะสร้างความถี่ขึ้นมาผสมคลื่นวิทยุที่รับเข้ามาได้ความถี่ไอเอฟค่าที่ 1 ความถี่ปานกลาง ค่าแรกจะมีค่าเท่ากับเท่าไรขึ้นอยู่กับระบบที่ใช้งาน

เมื่อได้ความถี่ไอเอฟค่าแรกมาแล้วจะถูกขยายให้แรงขึ้นโดยภาคขยายไอเอฟที่ 1 สัญญาณที่ถูกขยายจะส่งต่อไปยังภาคมิกเซอร์ 2 โดยนำไปผสมกับความถี่ค่าคงที่จากภาคออสซิลเลเตอร์ที่ 2 ที่สร้างมาจากวงจรกำเนิดความถี่แบบคริสตอล สัญญาณที่ออกจากภาคมิกเซอร์ที่ 2 จะเป็นความถี่ไอเอฟที่ 2 ความถี่ไอเอฟที่ได้นี้จะมามีค่า 455 kHz จากนั้นก็นำเอาความถี่ไอเอฟไปทำการดีเทกเตอร์เป็นคลื่นเสียงต่อไป

ระบบดับเบิลซูเปอร์เฮตจะช่วยกำจัดความถี่เงาให้หมดไปและยังช่วยให้ประสิทธิภาพในการเลือกรับและความไวของเครื่องสูงมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การสื่อสารข้อมูลแบบซิงโครนัส

เป็นการสื่อสารข้อมูลที่ทั้งด้านส่งและด้านรับต้องมีสัญญาณนาฬิกาที่เหมือนกัน ทั้งความถี่และเฟส ต้องรู้ว่าจุดเริ่มต้นในการส่งบิตแรกจากสายว่างนั้นเมื่อใดและหลังจากรับบิตแรกแล้ว จะรับบิตที่ 2,3 เมื่อใด ซึ่งเราสามารถทำได้โดยการส่งนาฬิกาเข้าตัวรับและตัวส่งพร้อม ๆ กัน ซึ่งในบางระบบอาจใช้วิธีส่งสัญญาณนาฬิกาด้านส่งไปยังด้านรับด้วย เพื่อให้ทางด้านรับนำสัญญาณที่รับ มาสร้างเป็นสัญญาณนาฬิกาใหม่ ทั้งนี้เพื่อทางด้านส่งและทางด้านรับมีเฟสที่ตรงกันเพื่อป้องกันข้อผิดพลาดของสัญญาณ

2.7 การสื่อสารข้อมูลแบบอะซิงโครนัส

คือระบบการรับส่งข้อมูลที่ไม่มีการกำหนดเวลาแน่นอน นั่นคือระยะเวลาระหว่างข้อมูลแต่ละครั้งที่ถูกส่งออกไปมีค่าไม่แน่นอน ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัสนั้น โครงสร้างของข้อมูลที่ส่งจะมีลักษณะเป็นบล็อก ๆ ซึ่งแต่ละบล็อกประกอบด้วยบิตเริ่มต้น บิตข้อมูล และบิตสิ้นสุด โดยบิตเริ่มต้นจะแสดงถึงการเริ่มต้นของกลุ่มข้อมูล แล้วตามด้วยส่วนของกลุ่มข้อมูล และบางกรณีอาจจะมีการเพิ่มบิตพาริตี เพื่อให้ตรวจสอบความถูกต้องของข้อมูลและบิตสิ้นสุด

บิตเริ่มต้น (Start bit)

ในการรับส่งบิตข้อมูลอนุกรมแบบอะซิงโครนัส ขณะที่สายส่งยังว่างจากการใช้งานอยู่นั้นสายส่งจะถูกกำหนดให้มีสถานะมาร์ค (Marking state) เป็นสัญญาณลอจิก "1" เมื่อทางด้านส่งจะทำการส่งข้อมูลก็จะต้องส่งบิตเริ่มต้นโดยแทนด้วยสถานะสเปซ หรือสัญญาณลอจิก "0" จำนวน 1 บิตไปก่อน ซึ่งจะทำให้ทางด้านรับตีเท็กซ์สถานะของสายส่งได้ว่าขณะนั้นสายส่งกำลังมีข้อมูลส่งมา สำหรับปัญหาด้านรับจะมีส่วยของวงจร spike detection ทำหน้าที่สุ่มจับสัญญาณสถานะของสายส่ง ด้วยความถี่ของการสุ่มค่าหนึ่งในระหว่างบิตต่อบิต ซึ่งอาจจะเป็น 2,4,16 ครั้งในหนึ่งวินาทีก็ได้ ดังนั้น หน้าที่ของบิตเริ่มต้นนั้นจะเป็นตัวบอกว่าข้อมูลที่เริ่มต้นตรงไหนและความกว้างของบิตเริ่มต้นจะมีขนาดเท่ากับความกว้างของ 1 บิต

บิตข้อมูล (Data bit)

จะถูกส่งออกมาต่อบิตเริ่มต้น ซึ่งอาจมีขนาด 5 บิต จนถึง 8 บิต โดยบิตที่มีนัยสำคัญน้อยที่สุด (LSB) จะถูกส่งออกก่อนไล่ไปจนถึงบิตที่มีนัยสำคัญมากที่สุด (MSB)

บิตตรวจสอบ (parity bit)

ทำหน้าที่ตรวจสอบความถูกต้องของข้อมูลที่ได้รับ บิตตรวจสอบมี 2 ประเภท คือ แบบคู่ และแบบคี่ หมายความว่า ถ้าหากบิตตรวจสอบเป็นคู่ ผู้ส่งจะต้องทำหน้าที่ตรวจสอบข้อมูลว่าข้อมูลที่ส่งไปนั้นมีบิตที่เป็น "1" เป็นจำนวนคู่หรือไม่ ถ้าหากไม่ใช่ก็ต้องเพิ่มบิตที่เป็น "1" ลงไปในบิตตรวจสอบ

ฝ่ายรับเมื่อรับแล้วจะต้องตรวจสอบว่าเป็นจริงดังสถานการณ์ที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่าสัญญาณที่รับนั้นผิดพลาดไป

· บิตสิ้นสุด (stop bit)

บิตสิ้นสุดเป็นบิตที่บอกให้ทราบถึงการจบของข้อมูลที่ส่ง ความกว้างของบิตสิ้นสุดนี้จะมี ความกว้างเป็น 1, 1.5, 2 ของสัญญาณนาฬิกา แล้วแต่ในการใช้งาน ข้อกำหนดก่อนใช้งานพอร์ตอนุกรม

1. ความเร็วในการส่งข้อมูลโดยทั่วไปนิยมใช้ 1200, 2400, 9600
2. ความยาวของบิตข้อมูลโดยทั่วไปใช้ 5-8 บิต
3. บิตตรวจสอบที่ใช้ คู่หรือคี่
4. จำนวนและความกว้างของบิตสิ้นสุด

2.8 ระดับลอจิกของ RS-232

ระดับของสัญญาณที่ใช้ในระบบ TTI ระดับลอจิก "1" แทนด้วย 5V ลอจิก "0" แทนด้วย 0V การ อินเตอเฟส ตามมาตรฐานของ RS-232 ระดับลอจิก "1" แทนด้วย -12V ลอจิก "0" แทนด้วย +12V

Line drivers and receivers

โดยทั่วไปในระบบคอมพิวเตอร์จะมี chip IC ทำหน้าที่ที่จัดการเกี่ยวกับการสื่อสารข้อมูลโดยเฉพาะ ตัวอย่างเช่น ในเครื่อง IBM PC ใช้ IC เบอร์ 8250 เป็นตัวจัดการเกี่ยวกับการสื่อสารข้อมูลแต่ระดับสัญญาณที่เข้าหรือออกจาก IC 8250 ยังคงเป็นสัญญาณระดับเดียวกับที่ใช้ใน CPU คือ +5V แต่ในมาตรฐานของ RS-232C กำหนดระดับสัญญาณคือ +12V, -12V ดังนั้นในการติดต่อกับโลกภายนอกของ 8250 จำเป็นจะต้องแปลงระดับของสัญญาณให้อยู่ในระดับเดียวกับ RS-232C เสียก่อน ซึ่งการแปลงสามารถทำได้โดยการใส่ IC สำเร็จรูปมาเป็นตัวจัดการ IC ที่ทำหน้าที่นี้มีอยู่หลายเบอร์ แต่เบอร์ที่พบกันบ่อย ๆ ก็คือ MAX232, AD232

คุณสมบัติที่สำคัญของ RS-232

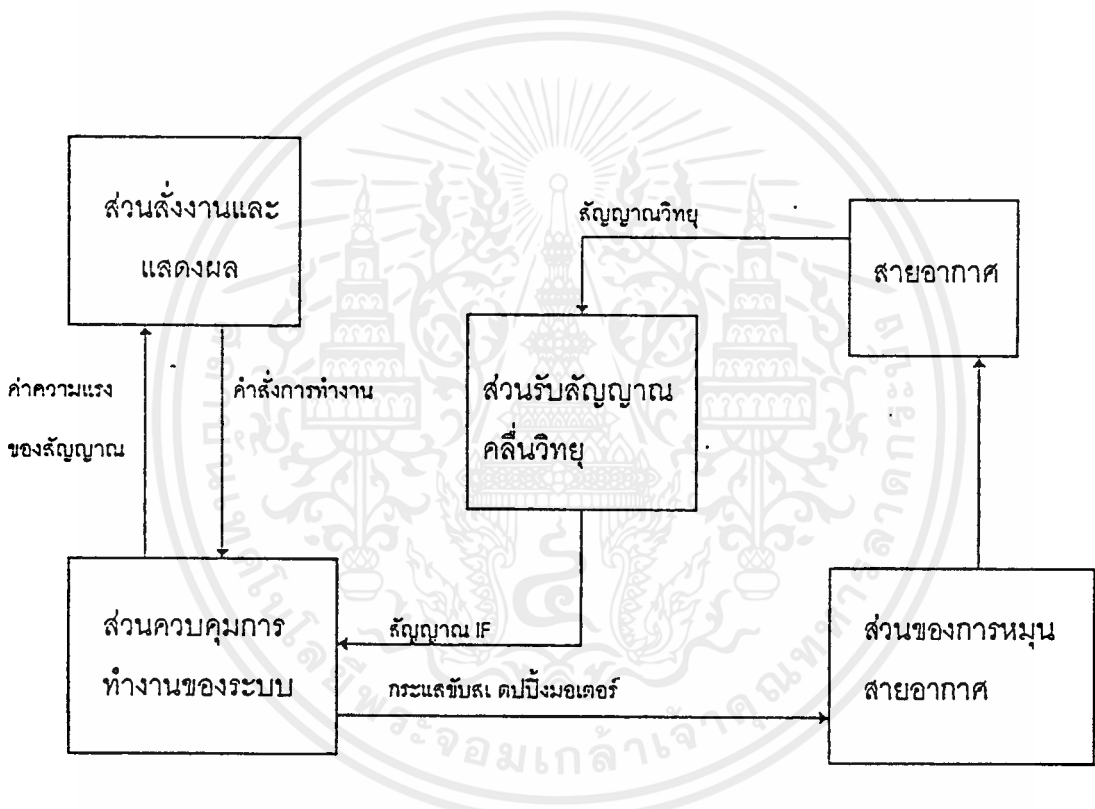
- ใช้ในการ interface แบบ unbalanced
- อัตราเร็วในการส่งข้อมูลสูงสุดไม่เกิน 20,000 bit/sec
- ระยะทางสูงสุดที่ใช้ไม่เกิน 50 ฟุต
- ทำให้เกิด crosstalk ที่มีค่ามาก

บทที่ 3

การคำนวณและการออกแบบ

3.1 หลักการทำงานของระบบ

ระบบการหาทิศทางของสัญญาณคลื่นสามารถแบ่งออกเป็นส่วนๆ ได้ตามบล็อกไดอะแกรม ที่แสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบการหาทิศทางของสัญญาณคลื่น

ส่วนการปฏิบัติงานและแสดงผล ทำหน้าที่ส่งคำสั่งตามการเลือกใช้ของผู้ใช้มายังชุดควบคุมระบบและนำผลของสัญญาณคลื่นที่รับได้ในแต่ละทิศทางไปแสดงเป็นกราฟให้ถูกต้อง ซึ่งในส่วนนี้จะใช้เครื่องคอมพิวเตอร์ต่างๆ ไปมาเป็นส่วนปฏิบัติงานและแสดงผล โดยไม่มีการแก้ไขทางด้านฮาร์ดแวร์ของเครื่องเลย การทำงานของส่วนนี้สร้างจากโปรแกรมที่เขียนด้วยภาษา C

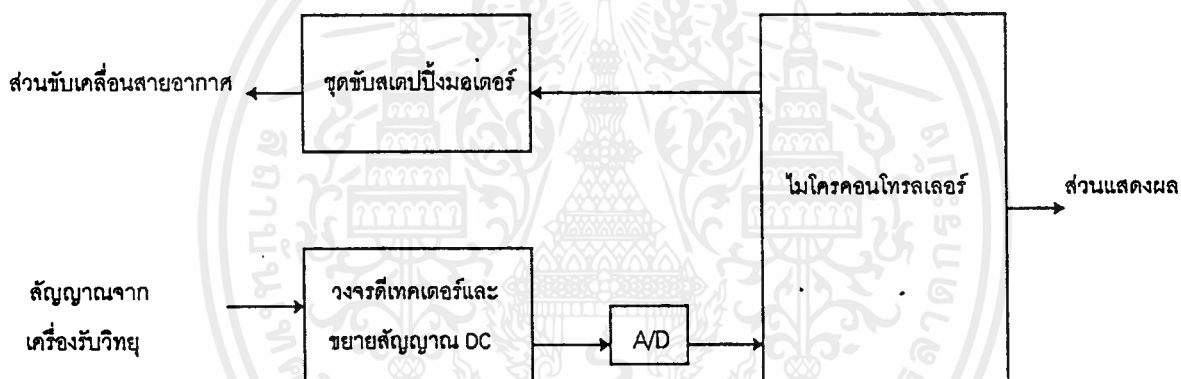
ส่วนควบคุมการทำงานจากระบบ ทำหน้าที่ควบคุมการทำงานจะเริ่มจาก เมื่อรับคำสั่งที่ส่งมาจากส่วนปฏิบัติงานแล้ว ก็จะมีการตรวจสอบคำสั่งเพื่อเลือกฟังก์ชันการทำงาน จากนั้นก็จะเริ่มส่งสัญญาณไปที่ส่วนของการหมุนสายอากาศ ขณะที่สายอากาศหมุนนั้น ชุดควบคุมการทำงานก็จะสั่งให้ชุดรับสัญญาณอ่านสัญญาณที่รับได้ทางสายอากาศเข้ามาแล้วส่งผลที่รับได้นั้นไปแสดงที่ส่วนแสดงผลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุตบแต่งลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการแก้ไข

เครื่องคอมพิวเตอร์สำหรับชุดควบคุมการทำงานนี้ประกอบด้วยส่วนฮาร์ดแวร์คือชุดไมโครคอนโทรลเลอร์ และส่วนที่เป็นโปรแกรมของชุดควบคุมการทำงาน ซึ่งเขียนขึ้นด้วยภาษาแอสเซมบลี

ส่วนของการหมุนสายอากาศ ประกอบไปด้วยชุดขับสเตปป์มอเตอร์ ซึ่งทำหน้าที่ขับกระแสให้แก่สเตปป์มอเตอร์ และชุดหมุนสายอากาศ ซึ่งประกอบไปด้วยมอเตอร์ ชุดเฟืองและฐาน มีหน้าที่หมุนสายอากาศให้หันไปตามทิศทางที่ต้องการโดยการควบคุมของชุดไมโครคอนโทรลเลอร์

ชุดรับสัญญาณ ทำหน้าที่รับสัญญาณคลื่นเข้ามาแล้วทำการแปลงจากสัญญาณคลื่นให้เป็นสัญญาณดิจิทัล ขนาด 8 บิต แล้วส่งต่อไปให้กับชุดไมโครคอนโทรลเลอร์ทำการประมวลผลเปรียบเทียบหาทิศทางที่มีสัญญาณคลื่นสูงที่สุด

3.2 ส่วนควบคุมการทำงานของระบบ ประกอบด้วยส่วนต่างๆดังนี้



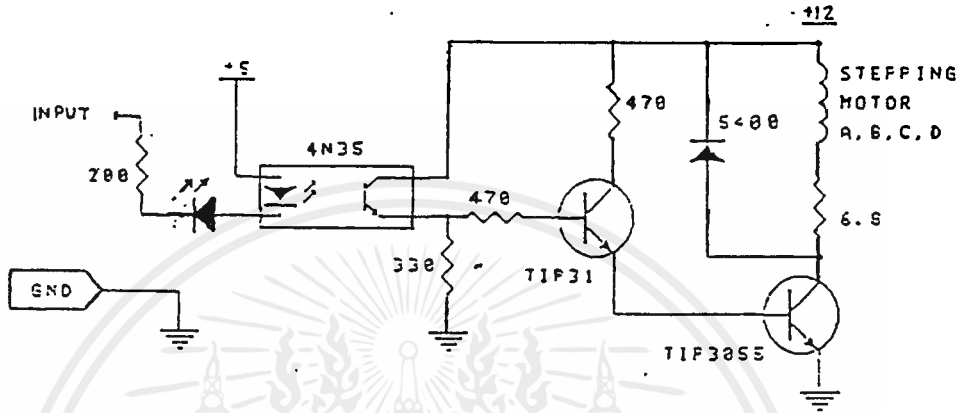
รูปที่ 3.2 ส่วนประกอบของส่วนควบคุมการทำงานของระบบ

3.2.1 ชุดขับสเตปป์มอเตอร์

มีหน้าที่คือ รับสัญญาณจากไมโครคอนโทรลเลอร์ แล้วสร้างกระแสเพื่อไปขับเคลื่อนสเตปป์มอเตอร์ ซึ่งมอเตอร์ที่ใช้เป็นแบบ 4 เฟส จึงต้องใช้วงจรที่ขับเคลื่อนที่เหมือนกัน 4 วงจร โดยแต่ละวงจรจะมีลักษณะดังรูปที่ 3.3

วงจรขับสเตปป์มอเตอร์ จะแยกเป็น 4 วงจร เหมือนกันโดยอินพุทของ OPTO 4w 35 จะมี LED และความต้านทาน 200 โอห์ม ต่ออนุกรมอยู่ที่ขา 2 ส่วนขา 1 จะต่ออยู่กับไฟบวก 5 โวลท์ ในสภาวะที่อินพุทเป็น "1" จะทำให้ไม่มีกระแสไหลผ่านไดโอดที่อยู่ภายในตัวของ OPTO ทำให้เอาท์พุททรานซิสเตอร์ไม่ทำงาน ก็จะไม่มีการไหลผ่านทรานซิสเตอร์ TIP31 และ TIP 3055 ทำให้เกิดสนามแม่เหล็กที่ขดลวดของสเตปป์มอเตอร์ ในภาวะที่อินพุทเป็น "0" จะทำให้มีกระแสไหลผ่านไดโอด ภายใน

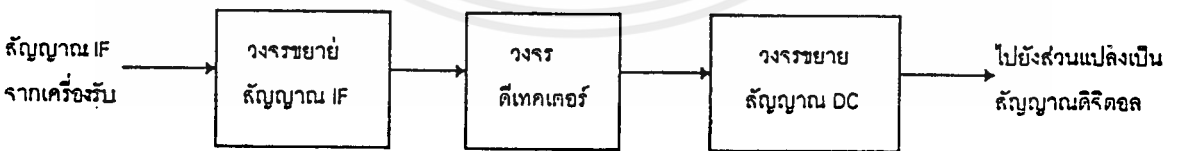
ตัวของ OPTO ทำให้เอาท์พุททรานซิสเตอร์ทำงานมีกระแสเอาท์พุทออกมาไบอัสให้กับทรานซิสเตอร์ TIP31 ซึ่งทำหน้าที่ขับกระแสให้กับทรานซิสเตอร์ TIP 3055 เมื่อทรานซิสเตอร์ TIP3055 ทำงาน จะทำให้มีกระแสไหลผ่านขดลวดของสเต็ปมอเตอร์ ทำให้เกิดสนามแม่เหล็กขึ้นที่ขดลวดนั้นไดโอดเบอร์ 1N5401 ที่ต่อขนานกับขดลวดของสเต็ปมอเตอร์ ทำหน้าที่ป้องกัน back emf ที่เกิดจากขดลวด



รูปที่ 3.3 วงจรขับสเต็ปมอเตอร์

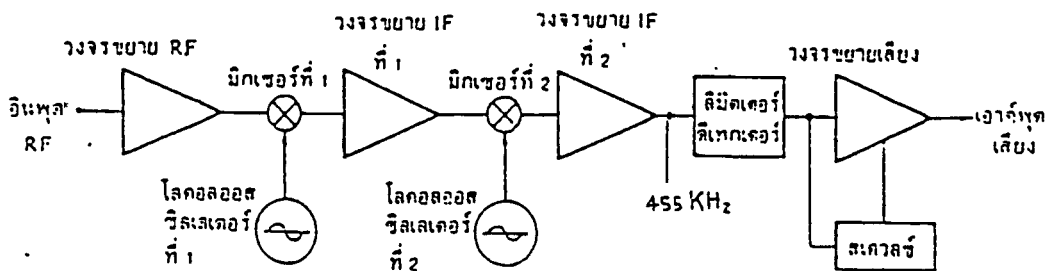
3.2.2 วงจรดีเทคเตอร์และขยายสัญญาณ DC

วงจรส่วนนี้จะรับสัญญาณ IF ความถี่ 455 kHz จากเครื่องรับวิทยุแล้วนำมาแปลงเป็นสัญญาณกระแสตรงที่มีขนาดแปรผันตามความแรงของสัญญาณที่เข้ามา เพื่อส่งไปยังส่วนแปลงสัญญาณเป็นดิจิตอลให้กับไมโครคอนโทรลเลอร์และนำไปแสดงผลต่อไป ซึ่งมีบล็อกไดอะแกรมดังรูปที่ 3.4



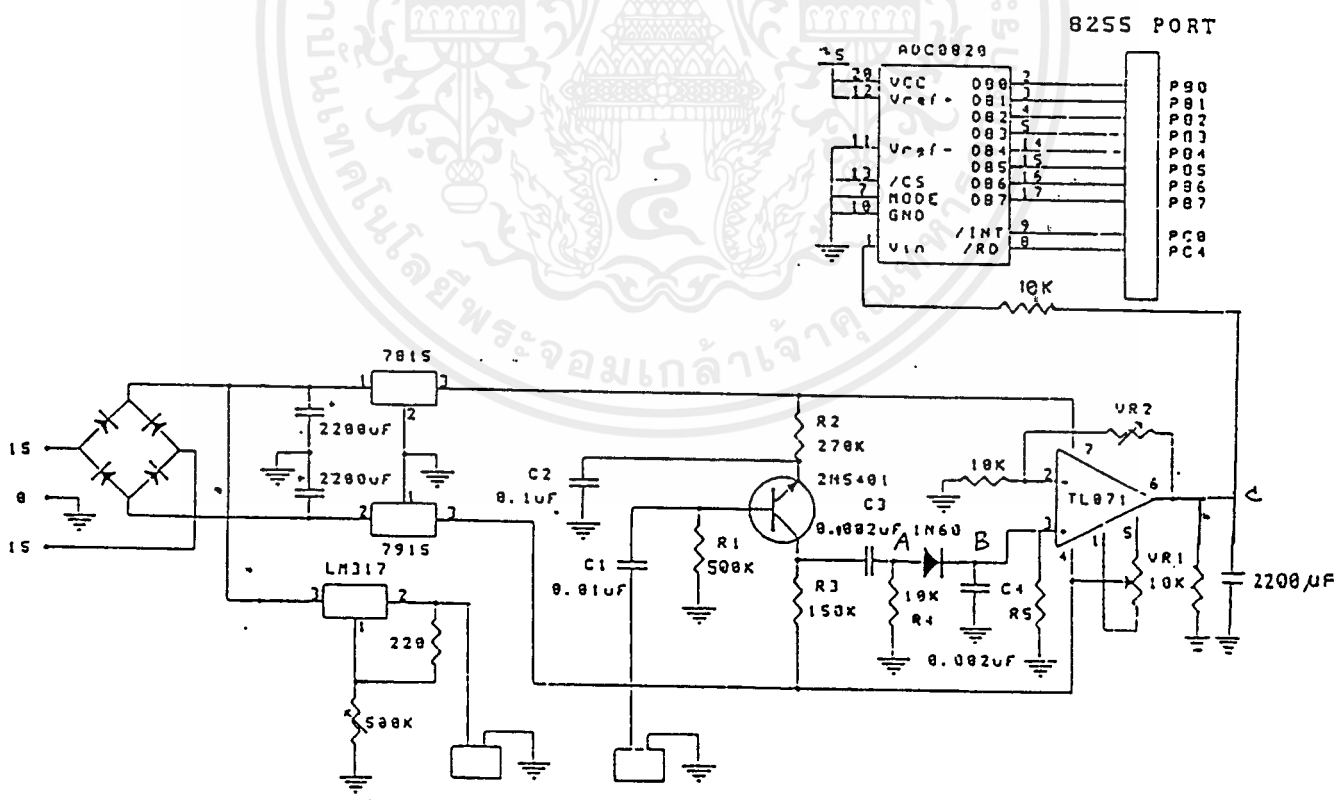
รูปที่ 3.4 บล็อกไดอะแกรมของวงจรดีเทคเตอร์และขยายสัญญาณ DC

สำหรับการรับสัญญาณวิทยุจะใช้เครื่องรับวิทยุ VHF/FM ซึ่งมีวงจรภาครับเป็นดับเบิลคอนเวอร์ชัน ดังแสดงในรูปที่ 3.5



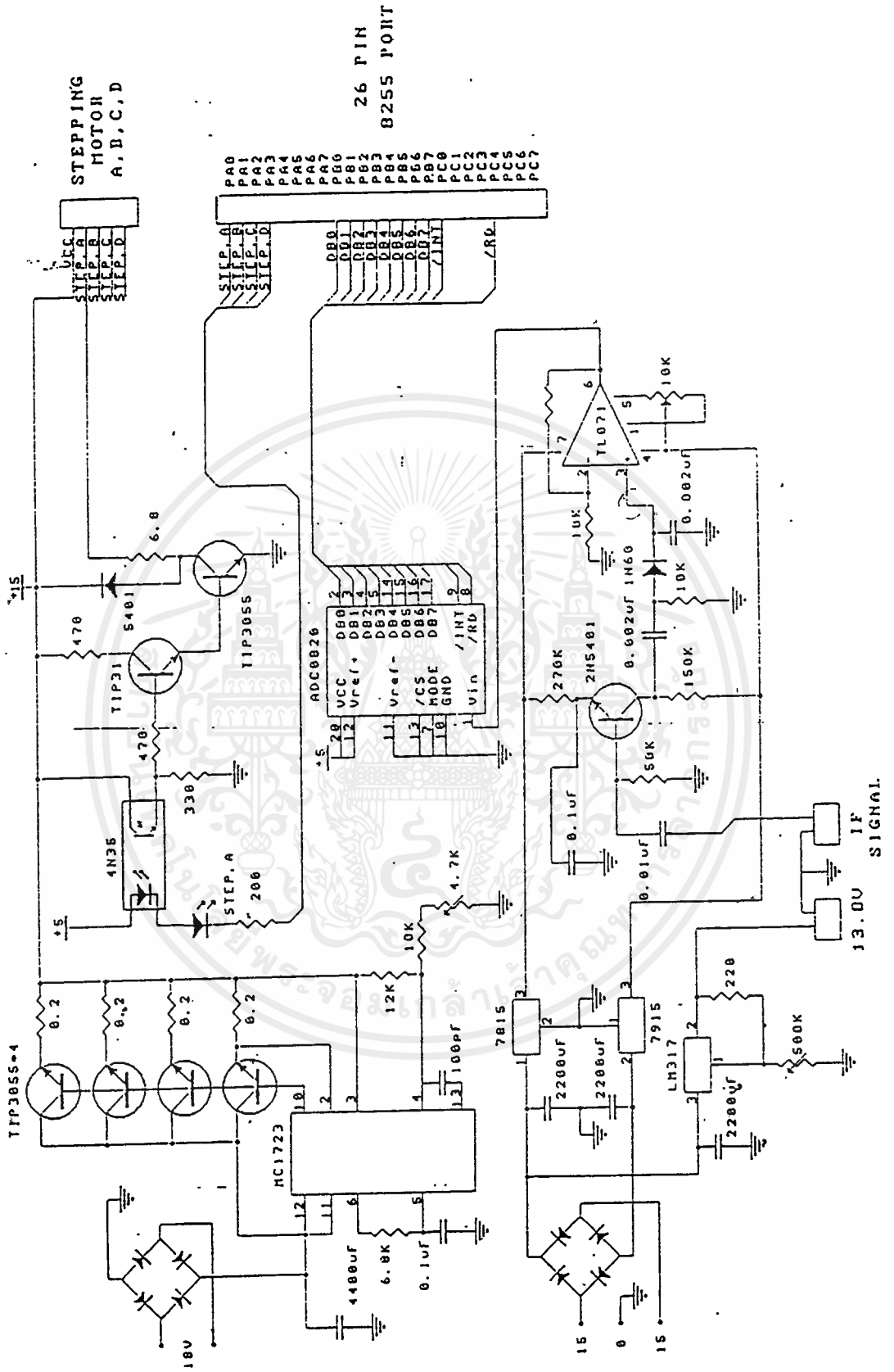
รูปที่ 3.5 แสดงเครื่องรับ FM แบบดับเบิลคอนเวอร์ชัน

IF ภาคที่ 1 ความถี่ 10.7 MHz IF ภาคที่ 2 ความถี่ 455 kHz ในการนำสัญญาณมาทำการ ดีเทคเตอร์ โดยการนำสัญญาณออกมาจากเอาต์พุตของภาคขยาย IF ภาคที่ 2 ซึ่งมีความถี่ 455 kHz สำหรับวงจรดีเทคเตอร์และวงจรรขยายสัญญาณ DC จะเป็นดังนี้



รูปที่ 3.6 แสดงวงจรดีเทคเตอร์และวงจรรขยายสัญญาณ DC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงวงจรรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

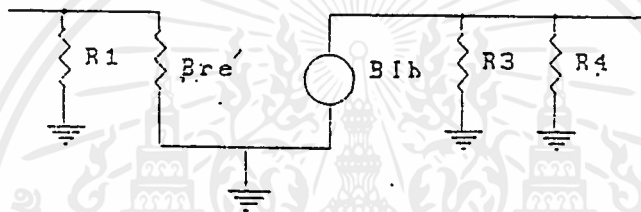
มีทรานซิสเตอร์ 2N 5401 ทำหน้าที่ขยายสัญญาณ IF ที่รับมาจากภาคขยาย IF ภาคที่ 2 ของเครื่องรับวิทยุ VHF FM มีการต่อเป็นวงจรขยายคลาส A แบบอิมิตเตอร์ร่วม (Common Emitter) โดยใช้แหล่งจ่ายไฟ 2 ชุด C_1 เป็นคาปาซิเตอร์คัปปลิ่งสัญญาณ C_2 เป็นตัวลัดวงจรสำหรับสัญญาณ AC การกำหนดกระแสไบอัสด้วย R_1 (R_B) และ R_2 (R_E)

จากวงจร $V_{EE} = 15 \text{ V}, R_E(R_2) = 270 \text{ K}$

จะได้ $I_E \approx \frac{15 \text{ V}}{270 \text{ K}} = 0.05 \text{ mA}$

จุดทำงาน (VCE) = $R_C I_E = 150 \text{ K} \times 0.05 \text{ mA} = 8.33 \text{ V}$

เมื่อมองการทำงานด้าน AC สามารถเขียนวงจรสมมูลได้ดังนี้



รูปที่ 3.8 แสดงวงจรสมมูลของทรานซิสเตอร์ # 2N5401

$$r_e' = 500 \Omega$$

$$I_E = 0.05 \text{ mA}$$

อินพุทอิมพีแดนซ์ คือ

$$R_1 // \beta r_e' = 500 \text{ k} // 40 \times 50$$

อัตราขยายแรงดัน

$$A_V = \frac{R_L}{r_e'}$$

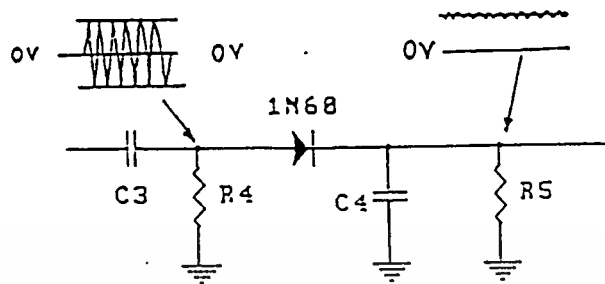
เมื่อ $R_L = R_3 // R_4 = 150 \text{ k} // 10 \text{ k} = 9.375 \text{ k}\Omega$

$$A_V = \frac{9.375 \text{ k}}{500} = 18.75 \text{ เท่า}$$

ในส่วนของการดีเทคเตอร์ มี C_3 และ R_4 เป็น ไฮพาสฟิลเตอร์โดยมีความถี่คัทออฟ ประมาณ 8 kHz ไดโอดทำหน้าที่ตัดสัญญาณซีกลบออก สัญญาณที่ผ่านออกมาจะถูกกรองให้เป็น DC ด้วย C_4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ R_5 จะทำหน้าที่คายประจุให้ C_4 เพื่อไม่ให้เกิดความผิดพลาดเนื่องจากค่าแรงดันที่ประจุอยู่ใน C_4 มีค่ามากกว่าสัญญาณอินพุตที่ช็อบก ทำให้ได้เอาต์พุต DC ที่ได้มีค่ามากกว่าความเป็นจริง



รูปที่ 3.9 แสดงสัญญาณตามจุดต่าง ๆ ของวงจรถืดเทคเดออร์

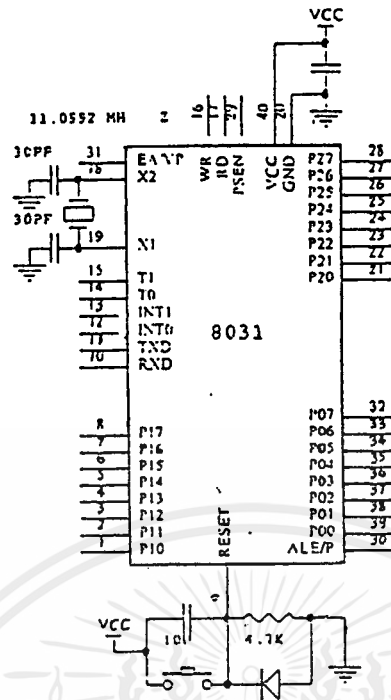
ส่วนของการขยายสัญญาณ DC ประกอบด้วย IC ออปแอมป์ 071 ต่อเป็นวงจขยายแบบนอนอินเวอร์ตติ้งแอมป์ ที่ขา Offset Null มีการต่อ VR_1 เพื่อทำการปรับเอาต์พุตให้เป็นศูนย์ขณะที่ไม่มีการรับสัญญาณวิทยุเข้ามามี RF เป็น VR_2 เพื่อไว้เผื่อมีการปรับแต่งเกณฑ์การขยายให้เหมาะสม สัญญาณ DC ที่ได้จะเปลี่ยนค่าความแรงของสัญญาณวิทยุที่รับได้ ซึ่งจะนำสัญญาณนี้ไปแปลงเป็นค่าทางดิจิทัลด้วยวงจรแปลงอนาลอกเป็นดิจิทัลอีกทีหนึ่ง

3.3 หลักการทำงานและการออกแบบบอร์ดไมโครคอนโทรลเลอร์.

บอร์ดไมโครคอนโทรลเลอร์ทำหน้าที่ควบคุมการทำงานของระบบมีหน้าที่ติดต่อบริส่งข้อมูลกับเครื่องคอมพิวเตอร์ซึ่งเป็นส่วนสั่งงานและแสดงผล ส่วนประกอบต่าง ๆ และการออกแบบวงจรของบอร์ดไมโครคอนโทรลเลอร์มีดังนี้

หน่วยประมวลผลกลาง (Central Unit, CPU) ใช้ไมโครคอนโทรลเลอร์เบอร์ 8031 ทำหน้าที่เป็นประมวลผลข้อมูล ควบคุมการทำงานในส่วนอื่น ๆ ของระบบและทำหน้าที่ติดต่อสื่อสารในการรับส่งข้อมูลกับเครื่องคอมพิวเตอร์

ในการใช้งานที่ขา 18,19 ซึ่งเป็นขา XATL1,XATL2 ของไมโครคอนโทรลเลอร์ 8031 จะติดต่อกับ Crystal 11.0592 MHz. และตัวเก็บประจุ 30 pF อีก 2 ตัว เพื่อทำหน้าที่สร้างสัญญาณนาฬิกา ให้แก่ 8031 และที่ขา 9 ซึ่งเป็น Reset ของ 8031 จะถูกต่อร่วมอยู่กับวงจร Reset ซึ่งใช้ตัวเก็บประจุ 10 μF ต่อแบบอนุกรมกับความต้านทาน 10 k Ω เป็นการ Reset แบบอัตโนมัติ เมื่อมีการป้อนไฟให้แก่วงจรเป็นครั้งแรก สำหรับสวิทช์ใช้สำหรับการ Reset ครั้งต่อ ๆ ไป เมื่อต้องการ

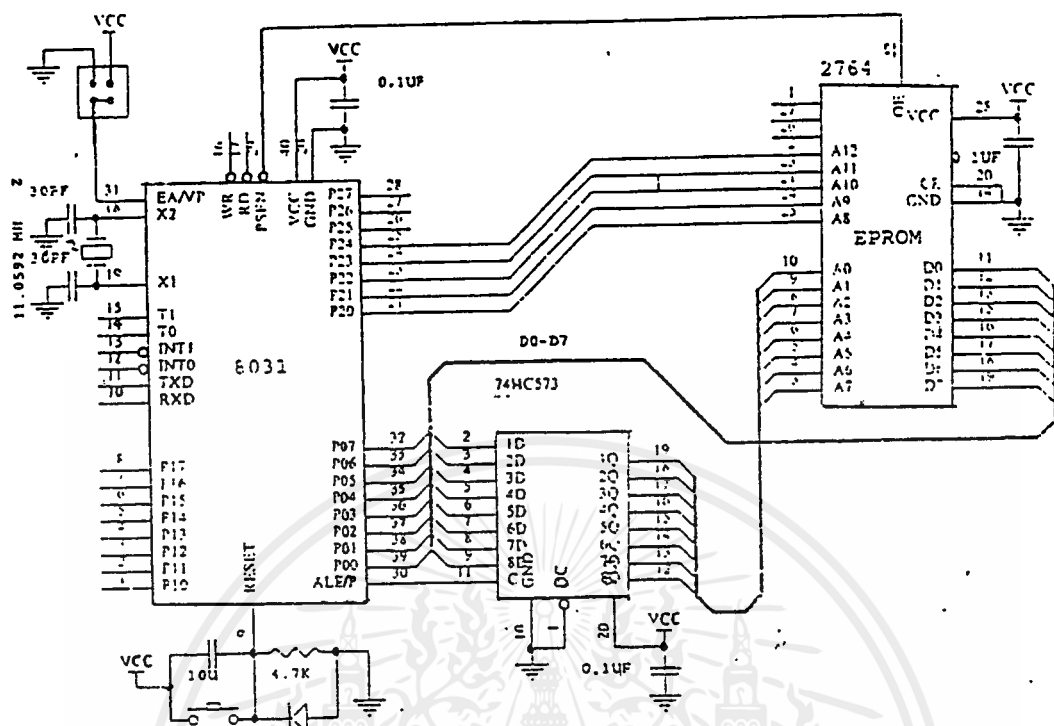


รูปที่ 3.10 แสดงการต่อวงจร Reset และวงจรสร้างสัญญาณนาฬิกาให้กับ 8031

หน่วยความจำภายนอก หน่วยความจำที่ใช้ในบอร์ดไมโครคอนโทรลเลอร์แบ่งเป็น 2 แบบ คือ หน่วยความจำสำหรับโปรแกรม ใช้สำหรับเก็บโปรแกรมการทำงานของระบบโดยใช้ EPROM เบอร์ 2764 ซึ่งมีขนาด 64 kbyte การออกแบบการติดต่อระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำสำหรับโปรแกรมแสดงดังรูปที่ 3.11

สัญญาณที่ 8031 ใช้ในการควบคุมหน่วยความจำสำหรับโปรแกรม จะมีสัญญาณจากขา PSEN ต่อเข้ากับขา OE ของหน่วยความจำ พอร์ต P0 ของ 8031 จะต่อกับขา D1-D8 ของ IC 74573 เพื่อใช้สำหรับแยกสัญญาณ Address ไบท์ต่ำ (A0-A7) ที่ขาทางออกของสัญญาณ (Q0-Q8) ของ IC 74573 จะต่อเข้ากับขา Address (A0-A7) ของหน่วยความจำ 2764 ที่พอร์ต 2 (P2.0-P2.4) จะต่อเข้ากับขา Address ไบท์สูงของหน่วยความจำ 2764 ที่ขา D0-D7 ของหน่วยความจำ 2764 จะถูกต่อเข้าโดยตรงกับพอร์ต P0 เพื่อให้ในการรับส่งข้อมูล

ในการแยกสัญญาณ Address ไบท์ต่ำกับสัญญาณข้อมูลพอร์ต P0 นั้นจะใช้ขา ALE ของ 8031 ต่อเข้ากับ Chip Enable (C) ของ 74573 เมื่อ 8031 ส่งสัญญาณที่เป็น Address ออกที่พอร์ต P0 สัญญาณที่ขา ALE จะมีสถานะเป็น High ทำให้ 74573 ทำการ Latch สัญญาณ Address ไบท์ต่ำไว้ จากนั้นเมื่อ 8031 ส่งสัญญาณข้อมูลออกมาที่พอร์ต P0 ขา ALE จะมีสถานะเป็น Low ซึ่งไม่มีผลใด ๆ เกิดขึ้นกับ 74573



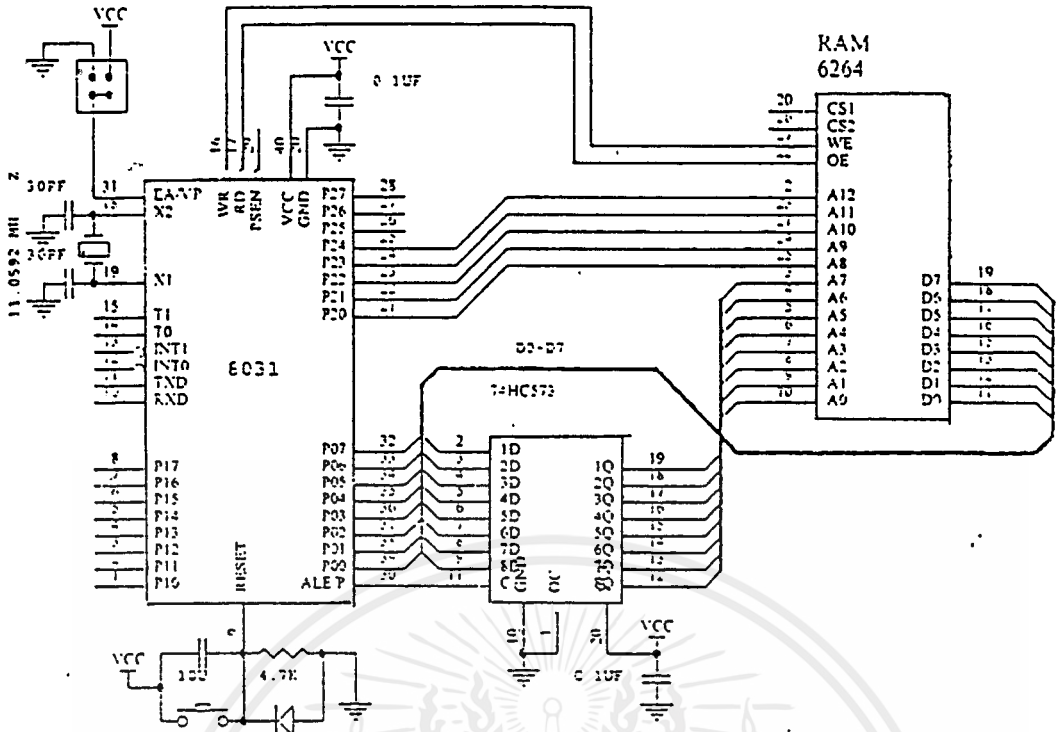
รูปที่ 3.11. แสดงการติดต่อระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำสำหรับโปรแกรม

หน่วยความจำสำหรับข้อมูล มีหน้าที่เก็บข้อมูล โดยใช้ RAM เบอร์ 6264 ขนาด 64 Kbyte การออกแบบการติดต่อระหว่างไมโครคอนโทรลเลอร์ กับหน่วยความจำสำหรับข้อมูลแสดงดังรูปที่ 3.11

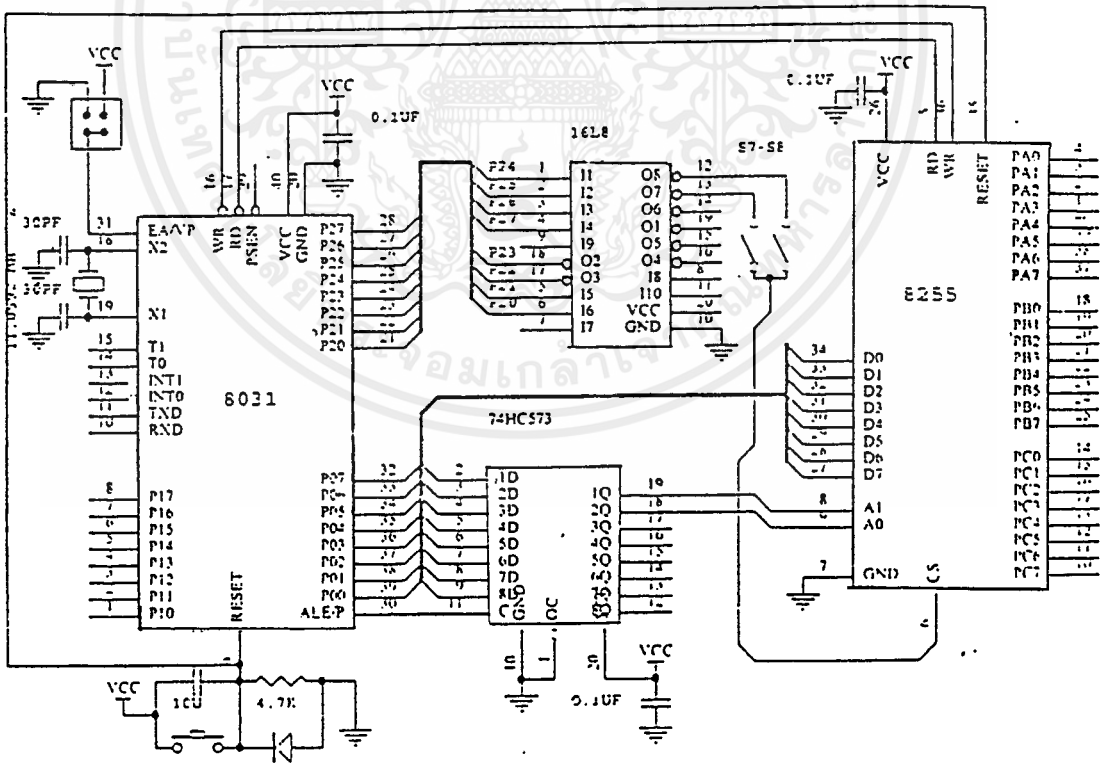
สัญญาณที่ใช้ในการควบคุมหน่วยความจำข้อมูลจะมีการต่อสัญญาณเหมือนกับการต่อหน่วยความจำสำหรับโปรแกรม แต่มีข้อแตกต่างกันที่ขา RD ของ 8031 จะต่อกับขา OE ของ 6264 และขา WR ของ 8031. จะต่อกับขา WR ของ 6264 เพื่อใช้ในการอ่านและเขียนข้อมูล

พอร์ตภายนอกแบบขนาน (External Port) เป็นพอร์ตขนานที่ขยายเพิ่มออกมาจากพอร์ต P0 เพื่อให้มีพอร์ตเพียงพอต่อการใช้งาน การออกแบบการติดต่อระหว่างไมโครคอนโทรลเลอร์กับพอร์ตภายนอกแบบขนานแสดงดังรูป 3.13

สัญญาณที่ใช้ในการควบคุมพอร์ตภายนอกมีขา RD ของ 8031 จะต่อกับขา OE ของ 8255 เพื่อใช้ในการควบคุมการอ่านข้อมูล ขา WR ของ 8031 จะต่อกับขา WR ของ 8255 เพื่อควบคุมการเขียนข้อมูล ขา D0-D7 ของพอร์ต 8031 จะต่อกับขา D0-D7 ของ 8255 เพื่อรับ-ส่งข้อมูลขา A0 และ A1 ของ 8255 จะต่อเข้ากับขา Q0,Q2 ของ 74573 ซึ่งเป็น Address ที่ใช้ในการเลือกพอร์ตใช้งาน



รูปที่ 3.12 แสดงการต่อไมโครคอนโทรลเลอร์กับหน่วยความจำสำหรับข้อมูล



รูปที่ 3.13 แสดงการต่อระหว่างไมโครคอนโทรลเลอร์กับพอร์ทภายนอกแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเลือกตำแหน่ง 8255 นั้นจะใช้ PAL เป็นตัวถอดรหัส ซึ่งในตัว PAL ได้โปรแกรมไว้ให้สามารถใช้งาน 8255 ได้ 2 ตำแหน่ง คือ ที่ Address FC00-FC03 และ FF00-FF03 ในการเลือกใช้งานจะใช้ DIP SW เป็นตัวเลือกตำแหน่งที่ต้องการ

โปรแกรมที่ใช้ออกแบบ PAL 16L8 สำหรับวงจรถอดรหัสภายนอกแบบขนาน

PAL16L8

DESIGNED BY KMIT

DECODE PORT

TI TIBPAL

A12 A13 A14 A15 A9 A8 NC NC NC GNP NC C1 C2 NC NC NC A10 A11

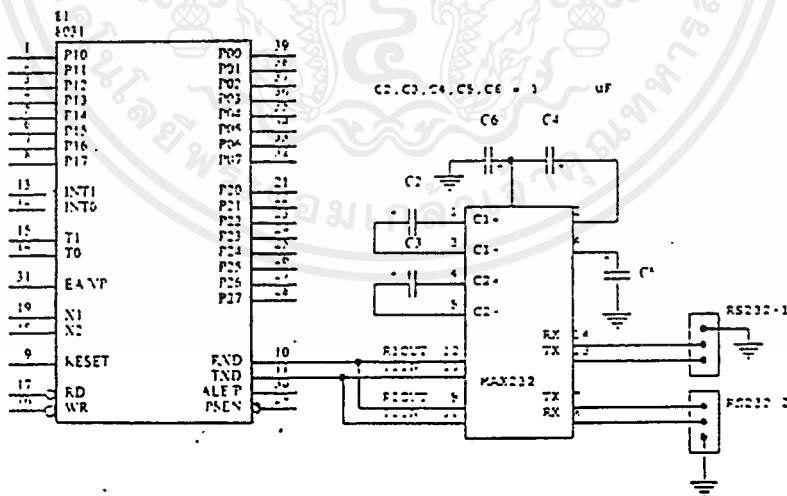
NC VCC

$$/C1 = A15 * A14 * A138 * A12 * A11 * A10 * /A9 * /A8$$

$$/C2 = A15 * A14 * A138 * A12 * A11 * A10 * A9 * A8$$

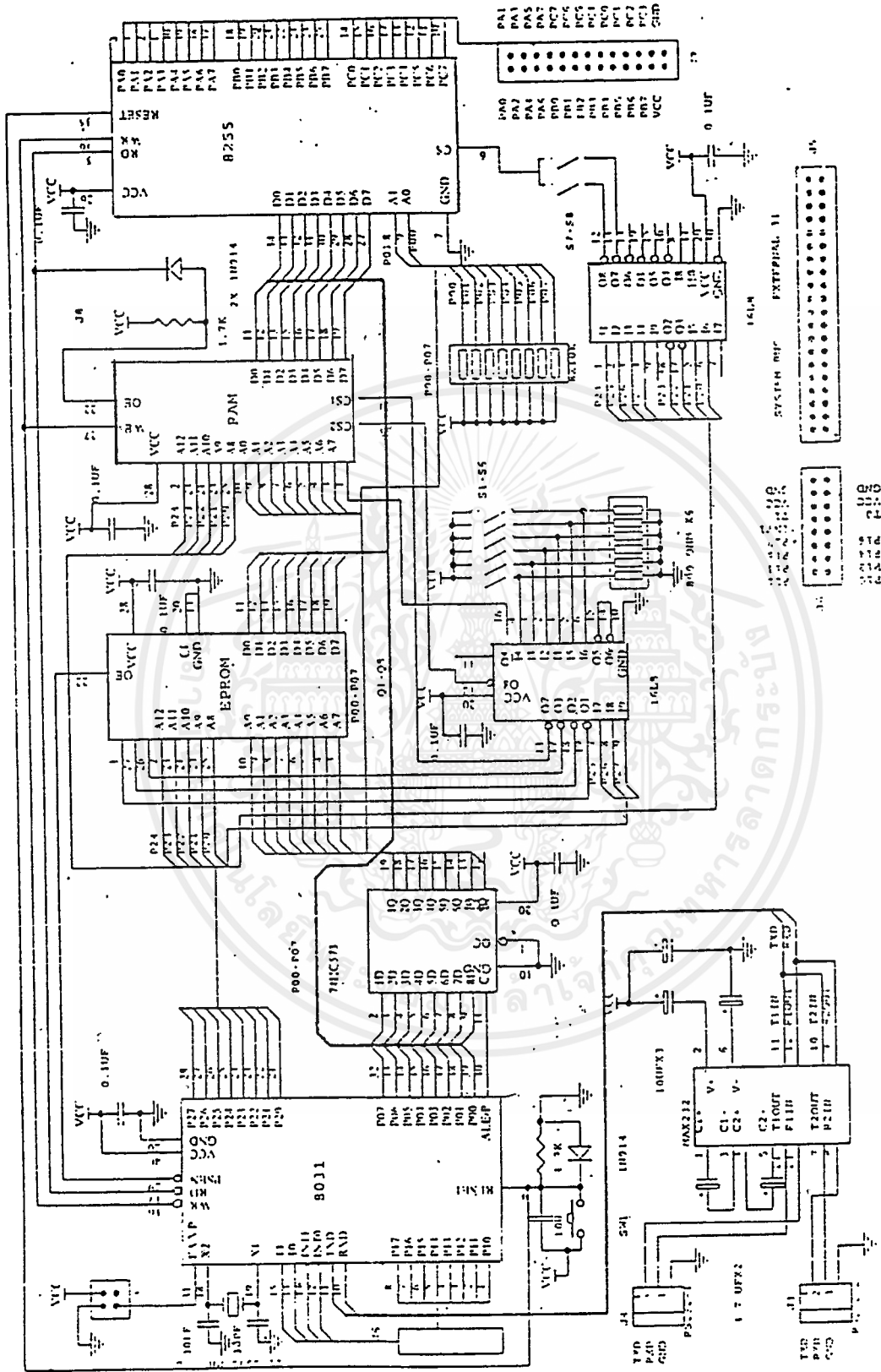
DESCRIPTION

พอร์ทการสื่อสารแบบอนุกรม (Serial Port) ทำหน้าที่รับส่งข้อมูลทางอนุกรมที่ใช้การติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ การออกแบบการต่อใช้งานพอร์ทการสื่อสารแบบอนุกรมแสดงดังรูปที่ 3.13



รูปที่ 3.14 แสดงการต่อใช้งานพอร์ทที่สื่อสารแบบอนุกรม

สัญญาณที่ขา TXD ของ 8031 จะต่ออยู่กับขา T1in และ T2in ของ MAX232 ซึ่งใช้เป็นตัวขับในการส่งสัญญาณข้อมูลแบบมาตรฐาน EIA Standard RS-232 และที่ขา RXD ของ 8031 จะต่ออยู่กับขา R1out และ R2out ของ MAX232 ซึ่งใช้เป็นตัวรับสัญญาณข้อมูลแบบมาตรฐานเดียวกับการส่งเอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงวงจรของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การรับและส่งข้อมูลแบบอนุกรม

รูปแบบการติดต่อสื่อสารแบบอนุกรม แบ่งได้ 3 ลักษณะ คือ

1. แบบ Simplex เป็นการรับส่งข้อมูลได้ในทางเดียวเท่านั้น
2. แบบ Half duplex สามารถรับและส่งข้อมูลได้ แต่การรับ/ส่งจะต้องผลัดกันทำงาน จะทำงานพร้อมกันไม่ได้
3. แบบ full duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

การรับและส่งข้อมูลแบบอนุกรมแบ่งได้เป็น 2 แบบ คือ

1. การรับส่งข้อมูลแบบซิงโครนัส (Synchronous)
2. การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous)

3.5 ชุดหมุนสายอากาศ

ชุดหมุนสายอากาศ ประกอบด้วย

3.5.1 ฐานตั้งเสาอากาศ เป็นฐานสำหรับติดตั้งเสาอากาศ มอเตอร์ และชุดเฟือง

3.5.2 สเตปป์มอเตอร์ ลักษณะคุณสมบัติของสเตปป์มอเตอร์ที่ใช้มีดังนี้

Step Angle	1.8	Degree/Step
Voltage	2.9	V
Current	3.8	A/Phase
Resistance	0.95	Ω /Phase
Inductance	3.8	mH/Phase
Torque	10.8	kg.cm
Weight	1400	g

3.5.3 ชุดเฟือง ประกอบด้วย เฟือง 2 ตัว มีอัตราในการทดเฟือง โดยที่

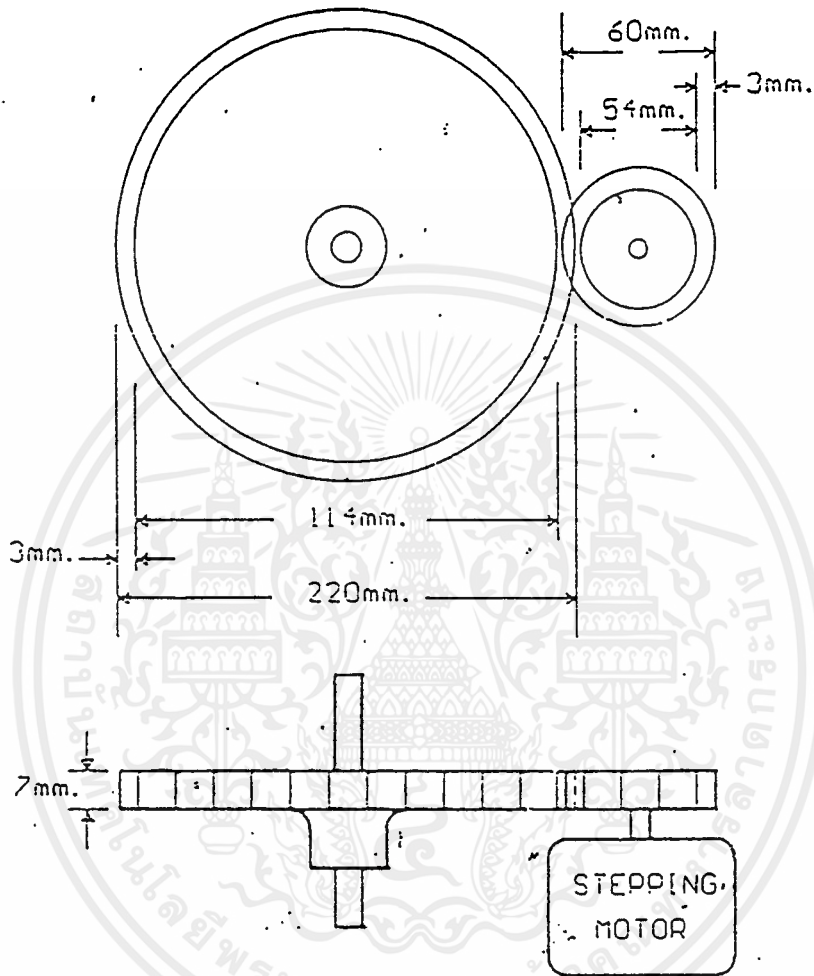
Z_1 คือ จำนวนฟันของเฟืองที่อยู่กับแกนของสเตปป์มอเตอร์ มีจำนวน 40 ฟัน

Z_2 คือ จำนวนฟันของเฟืองที่อยู่กับแกนของเสาอากาศ มีจำนวน 144 ฟัน

จากอัตราทดเฟืองที่กำหนด เมื่อเฟืองที่อยู่กับสเตปป์มอเตอร์หมุนไป 3.6 รอบ จะทำให้สายอากาศหมุนไป 1 รอบ เนื่องจากสเตปป์มอเตอร์มีมุมต่อสเตปเท่ากับ 1.8 องศา ดังนั้น การหมุนสเตปป์ 1 รอบของสายอากาศมีจำนวนเท่ากับ 720 สเตป ดังนั้นจะได้ว่า

$$1 \text{ สเตปของสเตปป์มอเตอร์} = 0.5 \text{ องศาของสายอากาศ}$$

$$\text{หรือ } 1 \text{ องศาของสายอากาศ} = 2 \text{ สเตปของสเตปป์มอเตอร์}$$



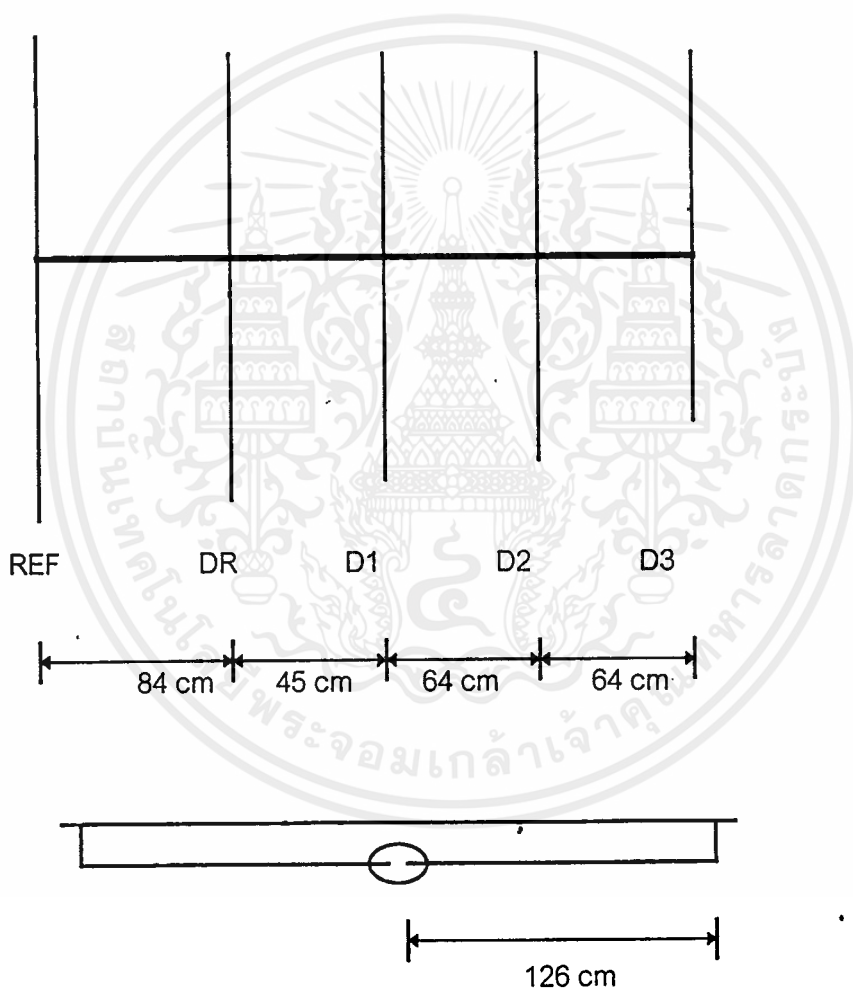
รูปที่ 3.16 แสดงเฟืองชุดหมุนสายอากาศ

3.6 โครงสร้างสายอากาศ

สายอากาศที่ใช้จะเป็นสายอากาศ YAGI 3 Element ซึ่งมีขนาดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

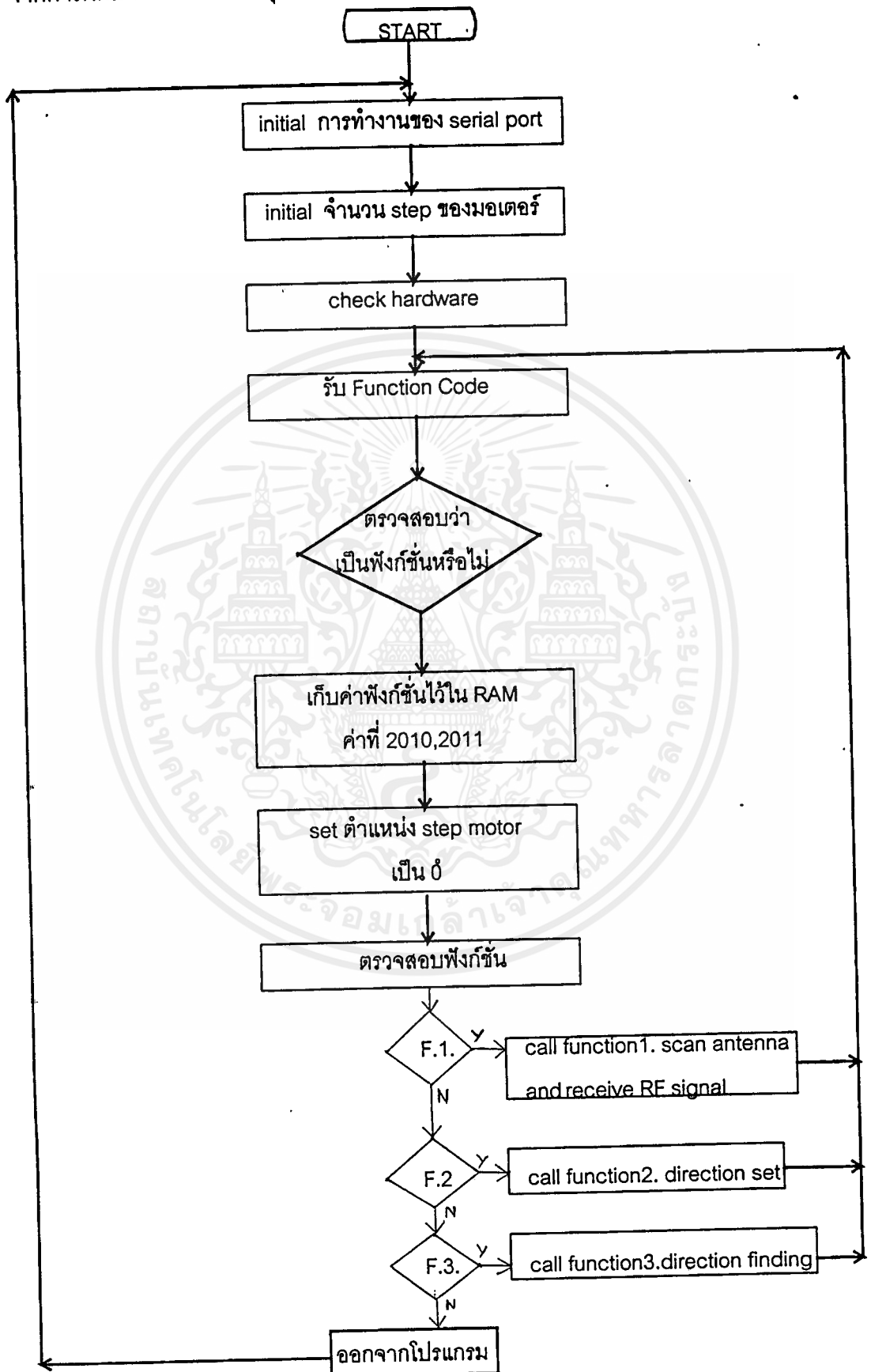
	ความยาว (cm)	เส้นผ่าศูนย์กลาง (cm)
Reflector	311	2
Driven	255	2
Director1	199	2
Director2	196	2
Director3	193	2



รูปที่ 3.17 แสดงโครงสร้างของสายอากาศ YAGI 3 Element

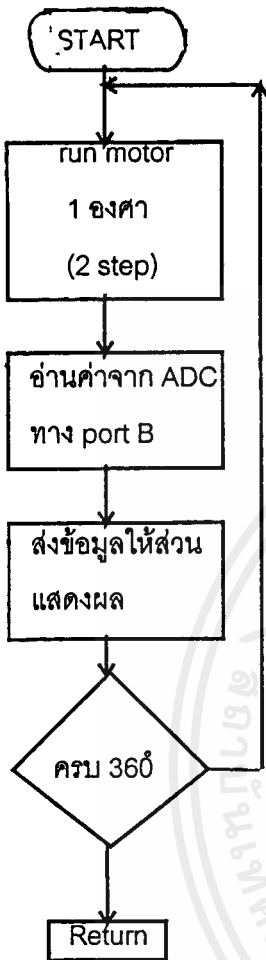
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการศึกษาโปรแกรมควบคุม ซึ่งเขียนด้วยภาษาแอสเซมบลีจะได้เป็น Flow chart ดังนี้

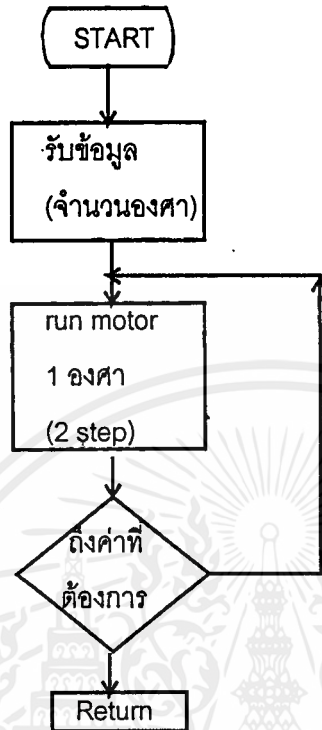


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

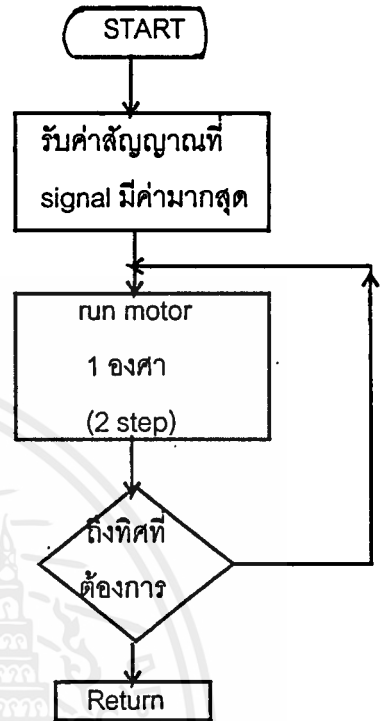
Function 1.



Function 2.



Function 3.



และจำนวนวงจรขับเคลื่อนมอเตอร์ และวงจรดีเทคเตอร์ขยายสัญญาณ DC ไปต่อรวมกับบอร์ดไมโครคอนโทรลเลอร์ให้เป็นวงจรรวมของระบบ ซึ่งโปรแกรมนี้จะอยู่ในไมโครคอนโทรลเลอร์

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองวงจรสแตปป์มอเตอร์

จากการทดลองหาค่ากระแสที่ได้จากด้านเอาต์พุต ซึ่งจะเป็นกระแสที่ใช้ขับสแตปป์มอเตอร์ โดยในการทดลองจะใช้ตัว L 3.8 mH ทำหน้าที่เป็นโหลดแทนสแตปป์มอเตอร์ แล้วทำการวัดกระแสที่ผ่านตัว L นี้

การทดลอง

1. ทำการต่อวงจรตามรูป 3.2
2. ป้อนแรงดันอินพุต 5 โวลต์ (แทนรหัส "1") แล้วใช้แอมมิเตอร์วัดกระแส
3. ป้อนแรงดันอินพุตเป็น 0 โวลต์ (ต่อกราวด์) แทนรหัส "0" วัดกระแสที่ผ่าน L 3.8 mH

ผลการทดลอง

รหัส	แรงดันอินพุต	กระแสที่ได้ทางเอาต์พุต
1	5 V	0 A
0	0 V	1.36 A

สรุปผลการทดลอง

ผลการทดลองที่ได้เป็นไปตามต้องการคือ เมื่อป้อนรหัส "1" วงจรจะไม่ส่งกระแสไปขับสแตปป์มอเตอร์ ในขณะที่เมื่อป้อนรหัส "0" วงจรจะปล่อยกระแสไปขับสแตปป์มอเตอร์ให้ทำงาน

เนื่องจากวงจรนี้เป็นวงจรขับสแตปป์มอเตอร์ ซึ่งจะต้องขยายกระแสให้มีค่ามากพอที่จะขับสแตปป์มอเตอร์ได้ คือระดับแอมแปร์ จากผลการทดลองสามารถสร้างกระแสได้ถึง 1.36 แอมแปร์ และจะนำค่านี้ไปใช้ในการเลือกให้สแตปป์มอเตอร์ที่เหมาะสมต่อไป

4.2 การทดลองวงจรดีเทคเตอร์และขยายสัญญาณ DC

วงจรนี้จะรับสัญญาณ IF จากเครื่องรับวิทยุมาทำการแปลงเป็นสัญญาณ DC ทำการขยาย แล้วแปลงเป็นสัญญาณดิจิตอลส่งไปแสดงผลที่คอมพิวเตอร์ ดังนั้นในการทดลองนี้จะใช้สัญญาณอินพุตที่สร้างจากฟังก์ชันเจนเนอเรเตอร์ ความถี่ 455 kHz (ซึ่งเท่ากับความถี่ของสัญญาณ IF จากภาคขยาย IF ภาคที่สองของเครื่องรับวิทยุแบบดับเบิลคอนเวอร์ชัน) ที่มีขนาดต่าง ๆ แทนสัญญาณจริง

อุปกรณ์ที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้ในการทดลอง

1. ฟังก์ชันเจนเนอเรเตอร์ HP 33120A
2. แหล่งกำเนิดไฟกระแสดรง HP E3611A
3. ออสซิลโลสโคป HP 54600B
4. วงจรที่ใช้ในการทดลอง

การทดลอง

1. ต่อดวงจร ตามรูปที่ 3.4
2. บ้อนสัญญาณรูปซายน์ ความถี่ 455 kHz ขนาด 200 mV จากฟังก์ชันเจนเนอเรเตอร์ เข้าที่ขั้วอินพุทของวงจร
3. ใช้ออสซิลโลสโคปวัดสัญญาณที่ขั้วแอนโอดของไดโอด 1N60 (จุด A) ซึ่งเป็นสัญญาณที่ผ่านการขยายโดยทางทรานซิสเตอร์เบอร์ 2N5401 ก่อนเข้าสู่ส่วนดีเทคเตอร์
4. วัดสัญญาณที่ขา 3 ของออปแอมป์ (จุด B) เป็นสัญญาณที่ออกจากส่วนดีเทคเตอร์ ซึ่งทำการแปลงเป็นสัญญาณ DC แล้ว
5. วัดสัญญาณที่ขา 6 ของออปแอมป์ (จุด C) ซึ่งเป็นสัญญาณ DC ที่ผ่านการขยายเพื่อจะเข้าสู่ส่วนแปลงสัญญาณจากอนาลอกเป็นดิจิตอล
6. ตรวจสอบดูรหัสสัญญาณดิจิตอลที่ได้รับจาก IC ADC0820 (DB_0 - DB_7)
7. เปลี่ยนค่าแอมพลิจูดของสัญญาณอินพุทเป็นค่าต่าง ๆ ตามตาราง แล้วทำการทดลองซ้ำตามข้อที่ 3 - 6

ผลการทดลอง

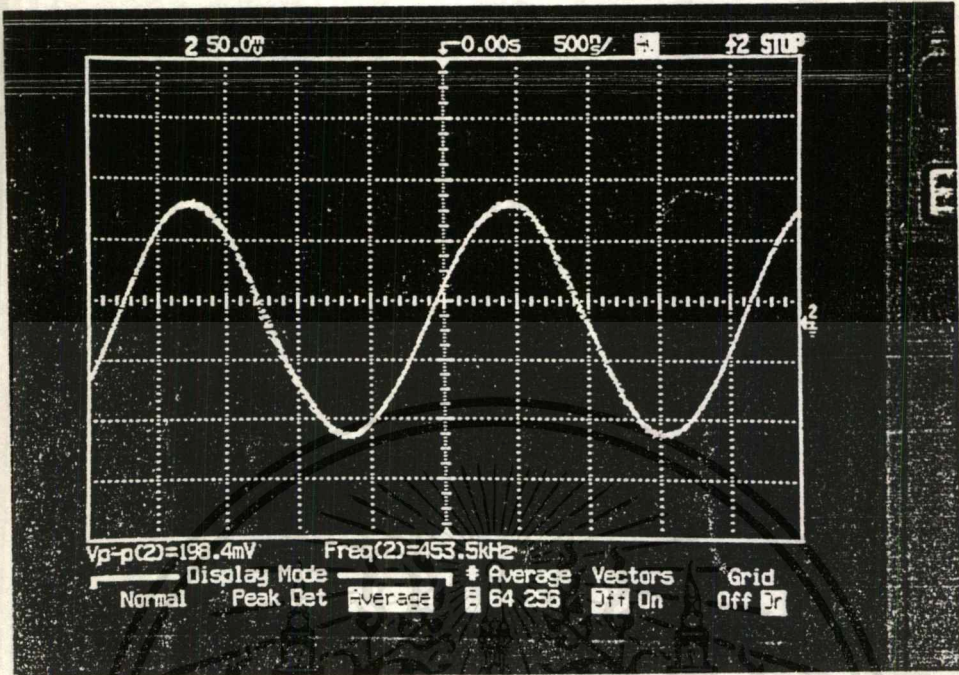
สัญญาณอินพุท	สัญญาณที่ถูกขยายโดยทรานซิสเตอร์ (สัญญาณที่จุด A)	สัญญาณที่ได้จากส่วนดีเทคเตอร์ (สัญญาณที่จุด B)	สัญญาณที่ได้จากส่วนขยายสัญญาณ DC (สัญญาณที่จุด C)
200 mV _{pp}	506.3 mV _{pp}	189.2 mV	2.12 V
400 mV _{pp}	556.2 mV _{pp}	242.7 mV	2.63 V
600 mV _{pp}	612.5 mV _{pp}	277.0 mV	2.92 V
800 mV _{pp}	691.9 mV _{pp}	287.5 mV	3.10 V
1 V _{pp}	750.0 mV _{pp}	296.9 mV	3.28 V
2 V _{pp}	796.9 mV _{pp}	353.7 mV	3.65 V
4 V _{pp}	937.5 mV _{pp}	421.9 mV	4.11 V
6 V _{pp}	969.7 mV _{pp}	443.6 mV	4.44 V
8 V _{pp}	1.16 V _{pp}	476.2 mV	4.71 V
10 V _{pp}	3.94 V _{pp}	968.8 mV	10.90 V

สรุปผลการทดลอง

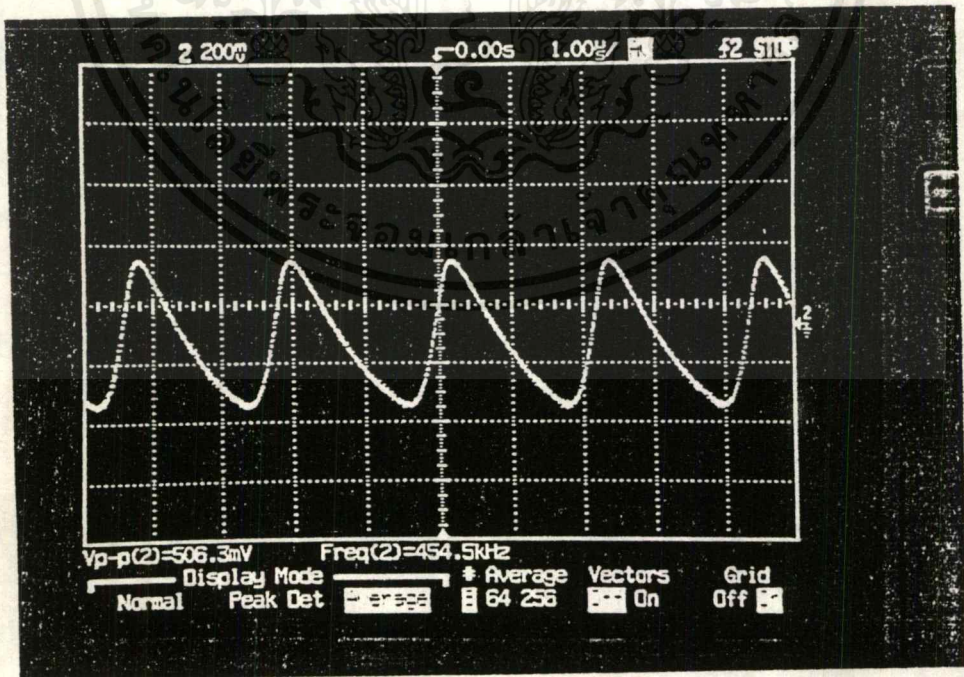
จากผลการทดลองจะเห็นว่าค่าที่ได้ทางเอาต์พุท (จุด C) ที่จะส่งไปแปลงเป็นสัญญาณดิจิตอลนั้นมีค่าแปรเปลี่ยนตามขนาดของสัญญาณอินพุท และมีความเรียบค่อนข้างสูง สามารถส่งไปแปลงเป็นสัญญาณดิจิตอลได้

รูปสัญญาณผลการทดลอง

- อินพุท 200 mV_{pp}

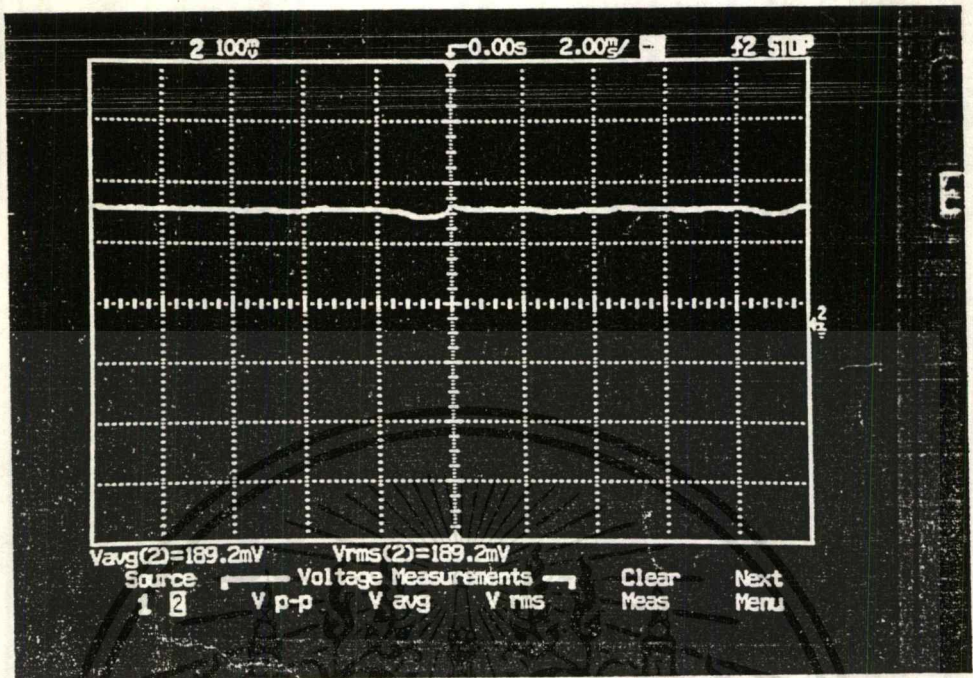


รูปที่ 4.1 สัญญาณอินพุท 200 mV_{pp}

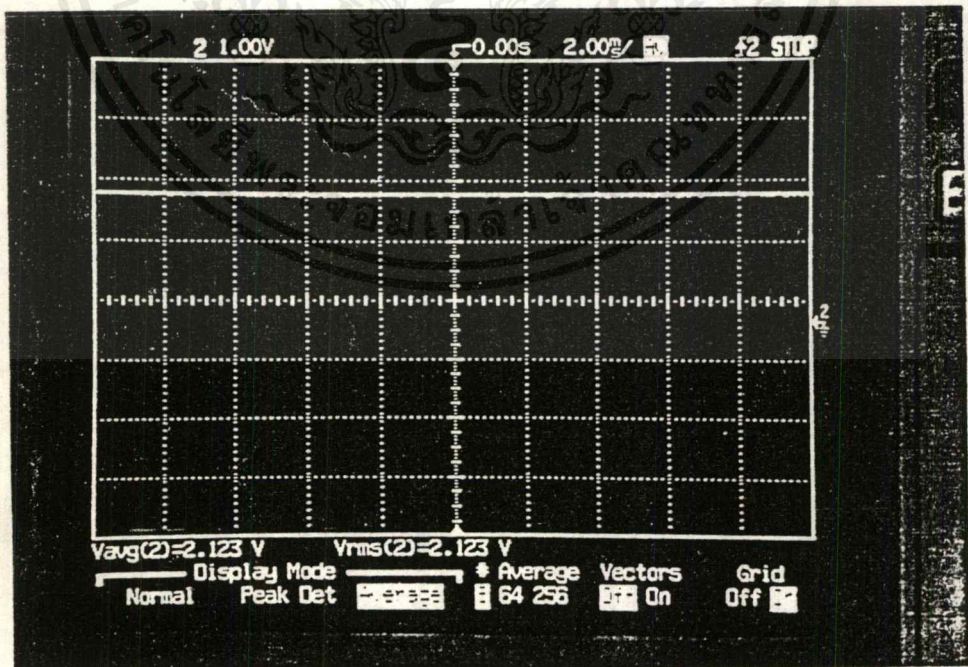


รูปที่ 4.2 สัญญาณความถี่ IF หลังจากผ่านการขยายโดยทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

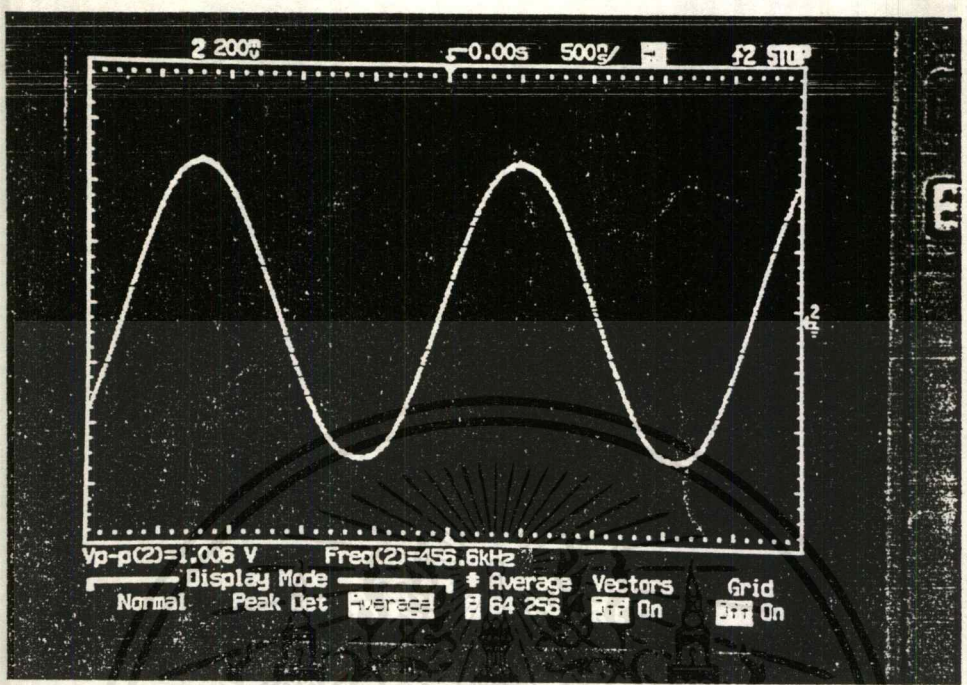


รูปที่ 4.3 สัญญาณความถี่ IF ที่ผ่านวงจรถัดขึ้น

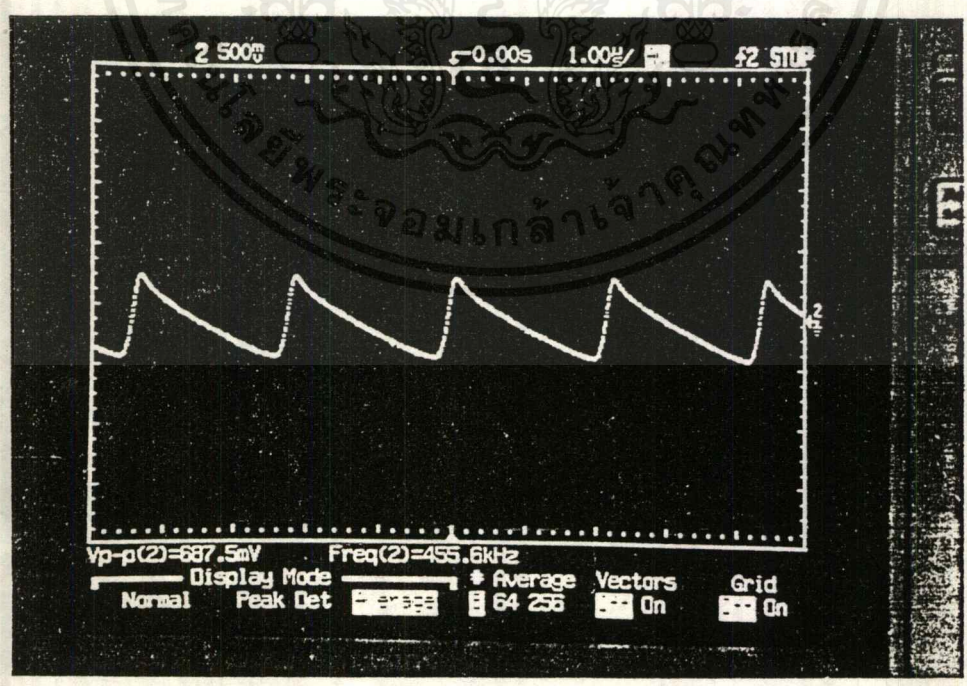


รูปที่ 4.4 สัญญาณเอาต์พุตที่ผ่านการขยายและวงจรรองทางด้านเอาต์พุตแล้ว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อินพุท 1 V_{pp}

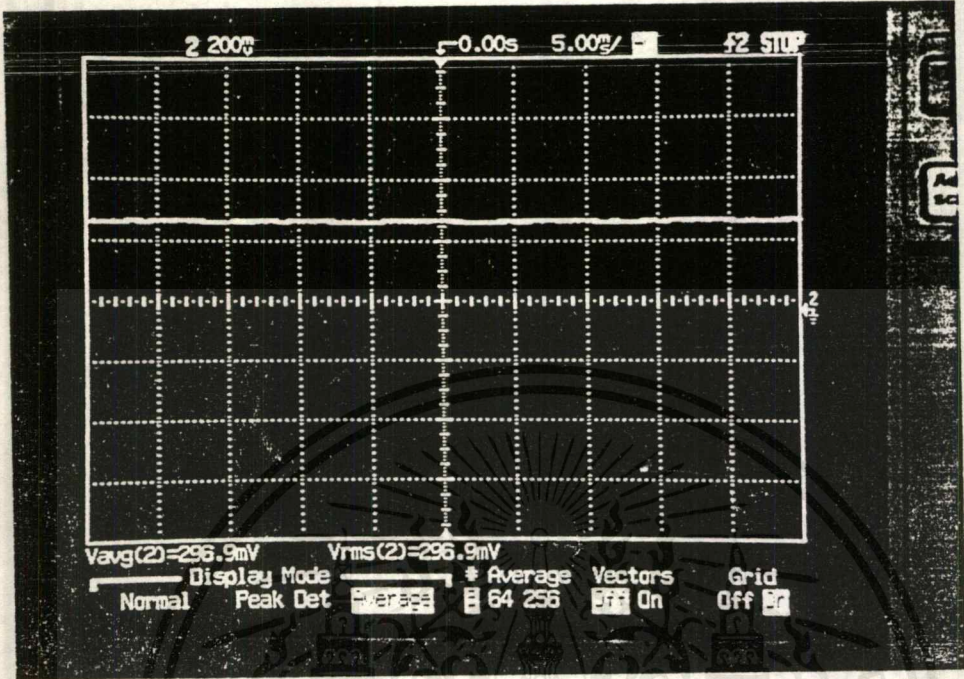


รูปที่ 4.5 สัญญาณอินพุท 1 V_{pp}

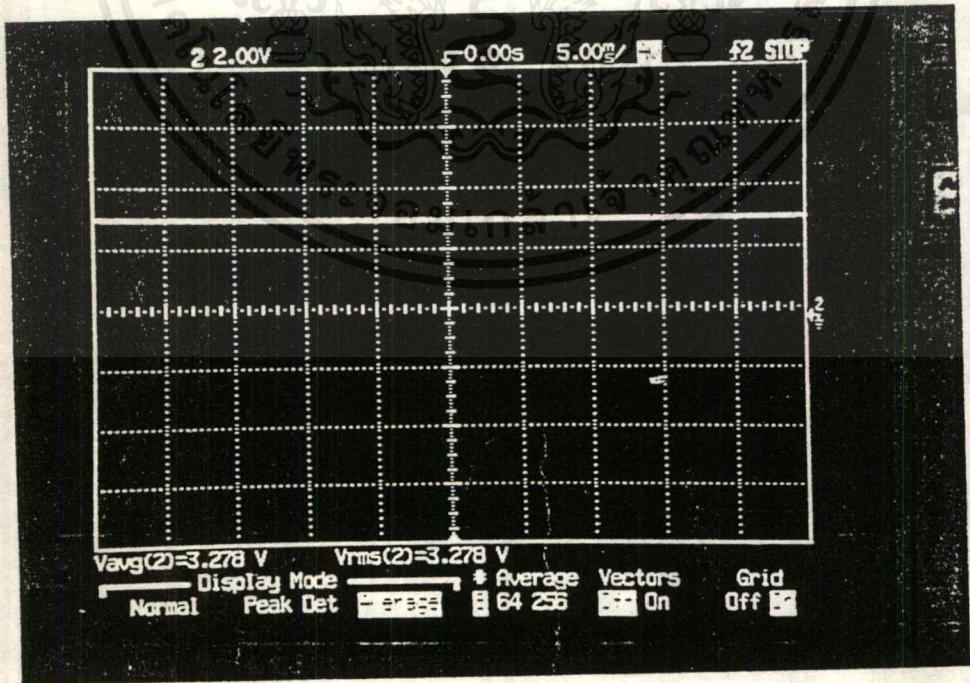


รูปที่ 4.6 สัญญาณความถี่ IF หลังจากผ่านการขยายโดยทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 สัญญาณความถี่ IF ที่ผ่านวงจรถัดเทคเตอร์



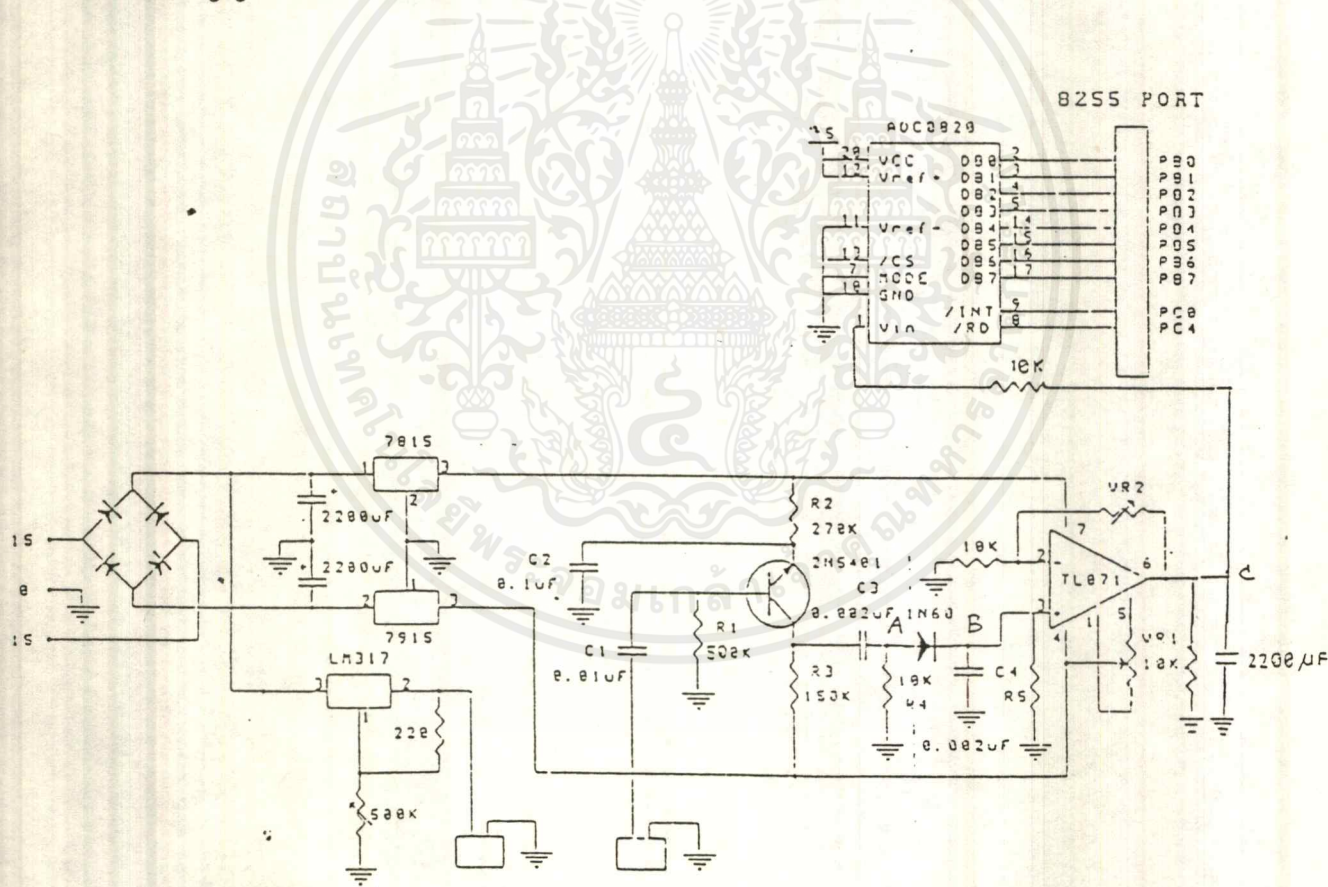
รูปที่ 4.8 สัญญาณแอมพลิจูดที่ผ่านวงจรขยายและวงจรรองทางด้านเอาต์พุตแล้ว

เอกสารนี้เป็นเอกสารของบริษัทสงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น มิได้อยู่ดัดแปลงไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองเพื่อหาระดับสัญญาณของอินพุท

เนื่องจากค่าความแรงของสัญญาณที่ได้จากเอาต์พุทของ ADC นั้น ไม่ใช่ค่าที่แท้จริงของสัญญาณ IF อินพุท ดังนั้นเพื่อที่จะทราบค่าที่แท้จริง สิ่งต้องมีการทดลองเพื่อหาความสัมพันธ์ระหว่างขนาดของสัญญาณอินพุทกับข้อมูลที่ได้ทางเอาต์พุท โดยมีขั้นตอนการทดลองดังนี้

1. ป้อนสัญญาณ IF ความถี่ 455 kHz ให้กับอินพุทของวงจร
2. อ่านข้อมูลที่ได้ทางเอาต์พุทของ ADC (ขา DB₀-DB₇) แล้วบันทึกผล
3. ทำการทดลองข้อ 1,2 ซ้ำอีกหลาย ๆ ครั้ง โดยเปลี่ยนขนาดของสัญญาณ อินพุทไปหลาย ๆ ค่า จนได้ข้อมูลมากเพียงพอ
4. นำข้อมูลที่ได้มาคำนวณหาความสัมพันธ์ ระหว่างอินพุทและเอาต์พุท ในรูปของสมการ
5. จากความสัมพันธ์ที่ได้ นำไปแทนค่าระดับสัญญาณเดิมในโปรแกรมแสดงผล เพื่อใช้แสดงค่าระดับของสัญญาณที่แท้จริงออกมา



รูปที่ 4.9 วงจรดีเทคเตอร์และวงจรขยายสัญญาณ DC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

อินพุท mVpp	เอาต์พุทของ ADC								เอาต์พุทของ ADC ในรูปของเลขฐานสิบ
	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	
		6	5	4	3	2		0	
100.0	1	0	0	0	0	0	0	1	129
139.1	1	0	0	1	0	1	0	1	149
178.1	1	0	1	0	1	0	1	0	170
212.5	1	1	0	0	0	0	0	1	193
231.2	1	1	0	0	0	1	1	1	199
253.1	1	1	0	0	1	1	0	1	205
275.0	1	1	0	0	1	1	1	1	207
300.0	1	1	1	0	0	1	0	0	222

สรุปผลการทดลอง

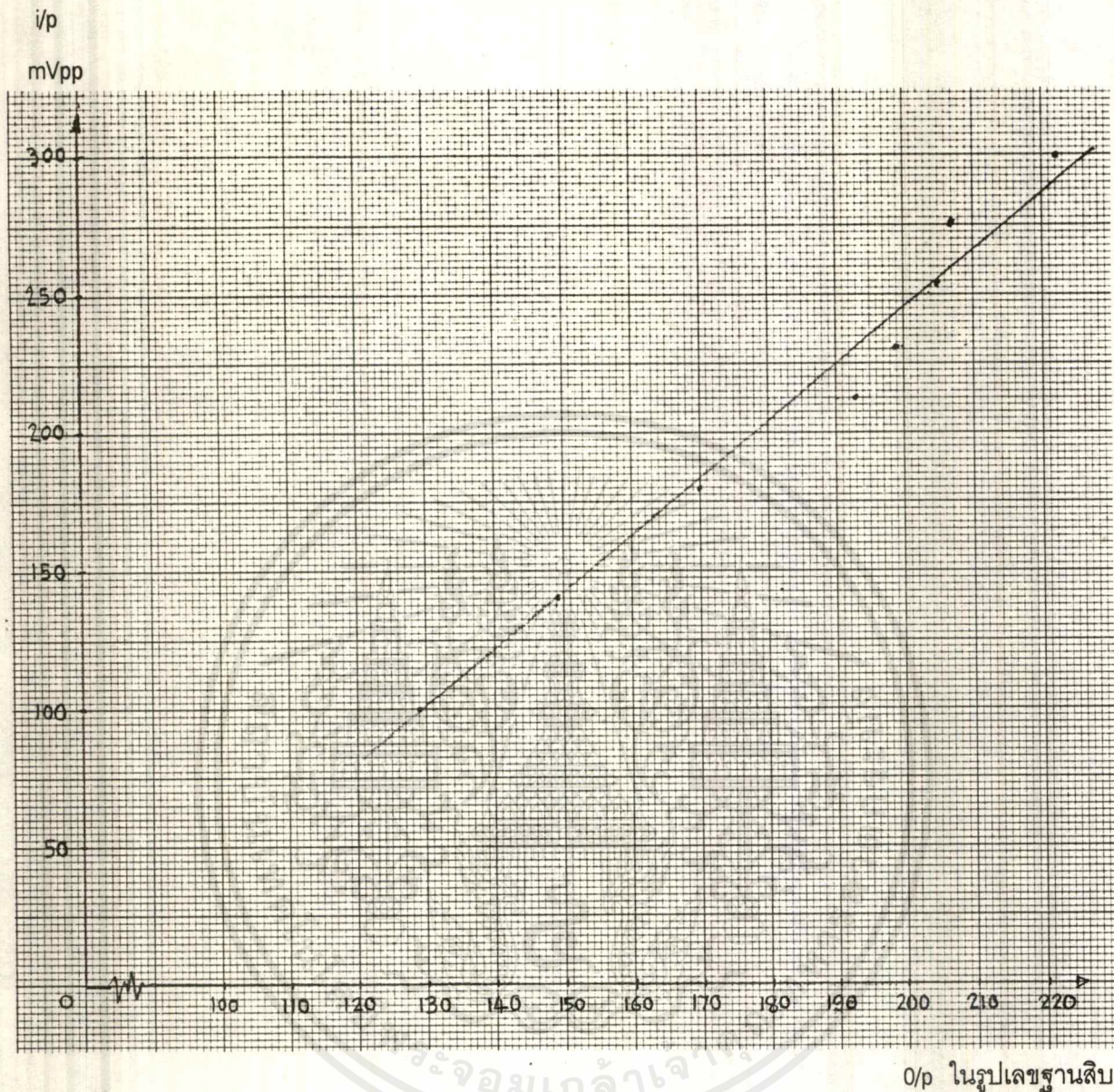
จากการทดลองที่ได้นำมาเขียนเป็นกราฟได้ดังรูปที่ 4.10 เมื่อแกน y คือขนาดของสัญญาณอินพุทและแกน X คือ ข้อมูลที่ได้ทางเอาต์พุท สมการของกราฟเส้นตรงนี้คือ

$$y = 1.955X - 152.195$$

ดังนั้นความสัมพันธ์ที่ได้คือ

$$i/p = 1.955(o/p) - 152.195$$

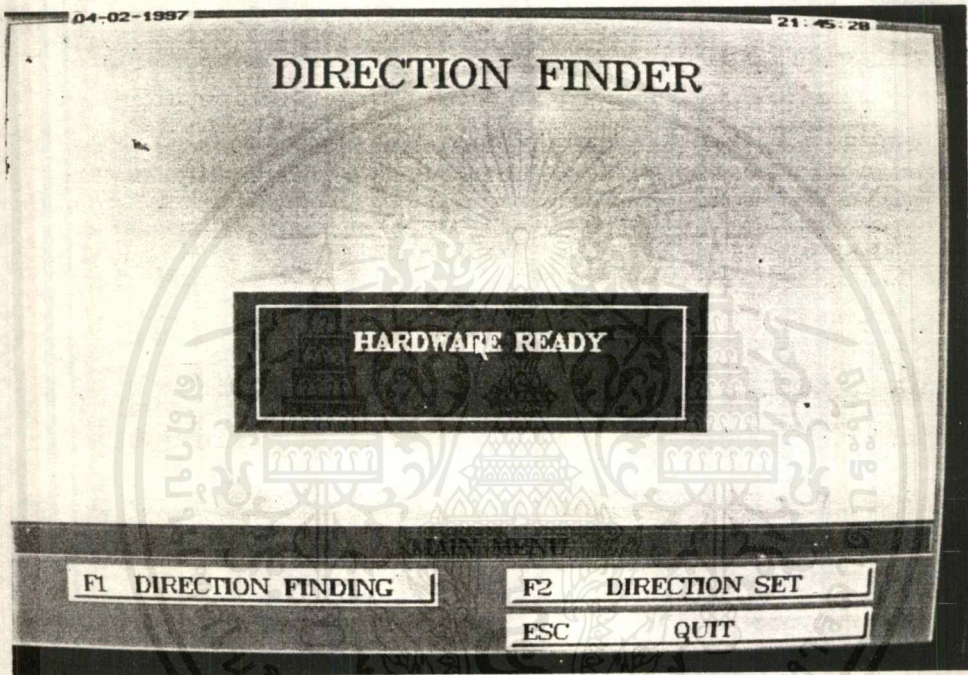
เมื่อนำค่าเอาต์พุทที่อ่านได้มาแทนในสมการจะทำให้ได้ค่าอินพุทตามต้องการ



รูปที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างสัญญาณ IF อินพุท และ เอาท์พุท ของวงจร ADC ในรูปเลขฐานสิบ

4.4 การทดสอบโปรแกรมการหาทิศทางคลื่น

เมื่อทำการต่อส่วนต่าง ๆ ของระบบเรียบร้อยแล้ว จากนั้นก็เปิดเครื่องคอมพิวเตอร์และเครื่อง Direction finder แล้วทำการรันโปรแกรม จอภาพของเครื่องคอมพิวเตอร์จะปรากฏดังรูป 4.11 แต่ถ้าเครื่องคอมพิวเตอร์ไม่สามารถติดต่อกับส่วนที่เป็น Hardware ได้ ที่จอภาพของเครื่องคอมพิวเตอร์จะแสดงข้อความแจ้งข้อผิดพลาดออกมา ดังรูปที่ 4.12



รูปที่ 4.11 แสดงเมนูสำหรับเลือกการทำงานของโปรแกรม

สำหรับรายการในเมนูมีดังนี้

Direction finding (F1) เป็นฟังก์ชันที่ใช้ในการหาทิศทางของสัญญาณคลื่นที่มีความแรงสูงสุด

Direction set (F2) เป็นฟังก์ชันที่ใช้หมุนสายอากาศไปในทิศทางองศาต่าง ๆ ตามที่กำหนดให้

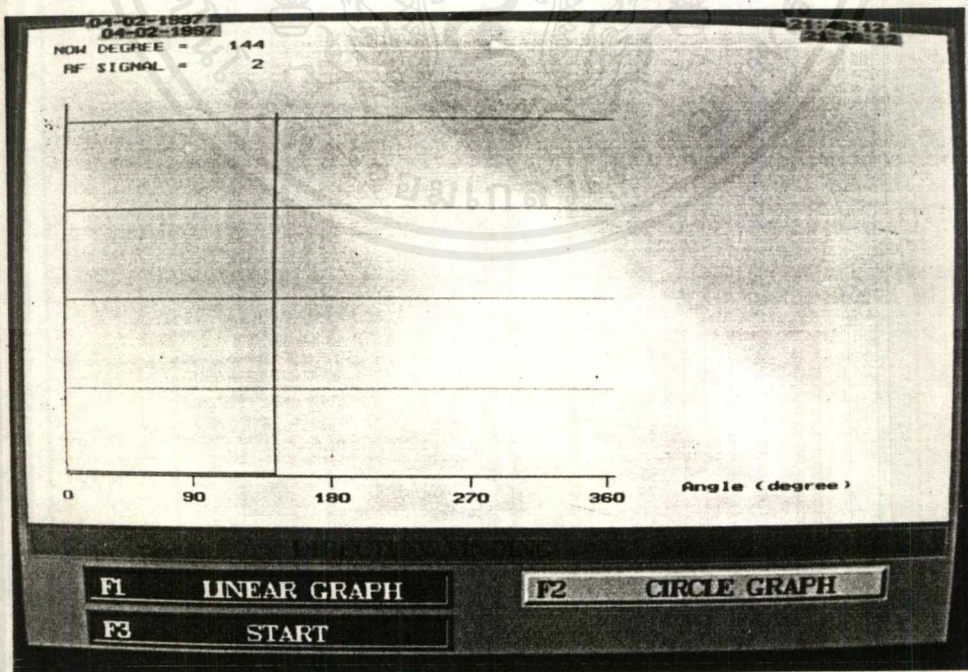
Quit (Esc) จบการทำงาน หรือต้องการออกจากโปรแกรม



รูปที่ 4.12 แสดงข้อผิดพลาดเมื่อเครื่องคอมพิวเตอร์ไม่สามารถติดต่อกับฮาร์ดแวร์ได้

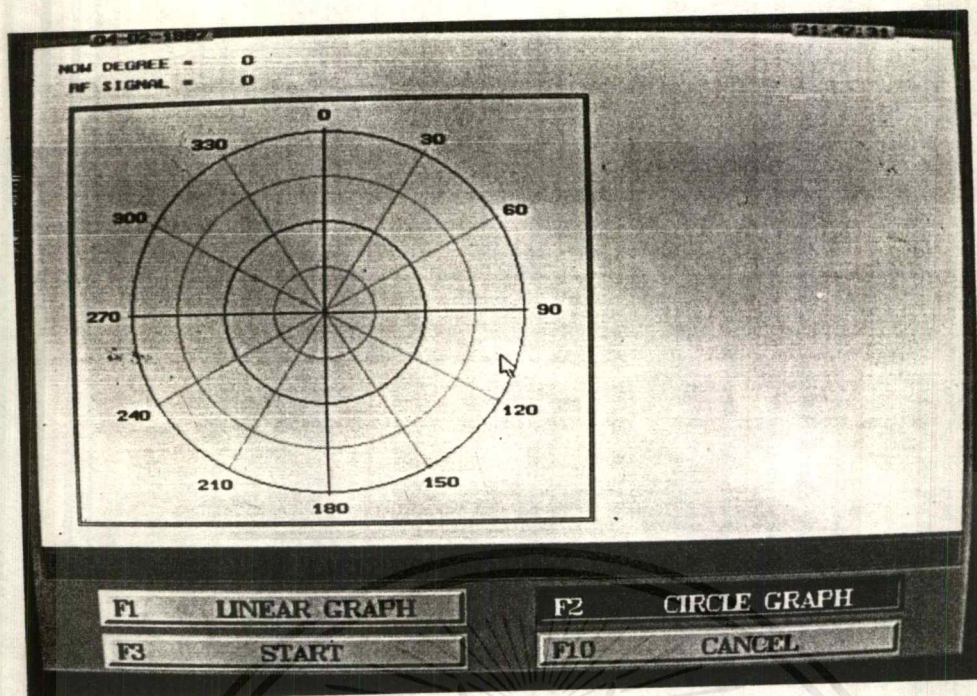
4.4.1 ฟังก์ชัน Direction Finding

เมื่อมีการคลิกเมาส์ที่ปุ่ม "Direction Finding" หรือคีย์ F1 จะเป็นการหาทิศทางของสัญญาณคลื่นที่มีความแรงของสัญญาณสูงสุด ซึ่งในการแสดงระหว่างสัญญาณนั้นสามารถเลือกให้แสดงเป็นกราฟเส้นตรงหรือกราฟวงกลม ก็ได้ โดยการคลิกเมาส์ที่ปุ่มหรือคีย์ F1 หรือ F2 ตามลำดับ



รูปที่ 4.13 แสดงภาพเมื่อเลือกกราฟเส้นตรงขณะกำลังรับสัญญาณ

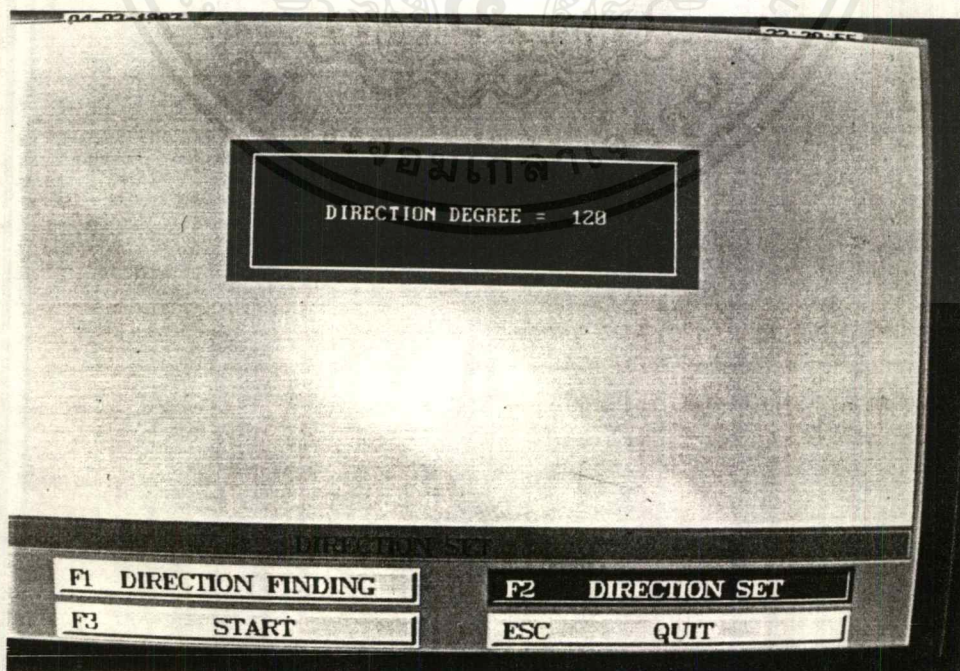
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงภาพเมื่อเลือกกราฟวงกลมขณะรับสัญญาณ

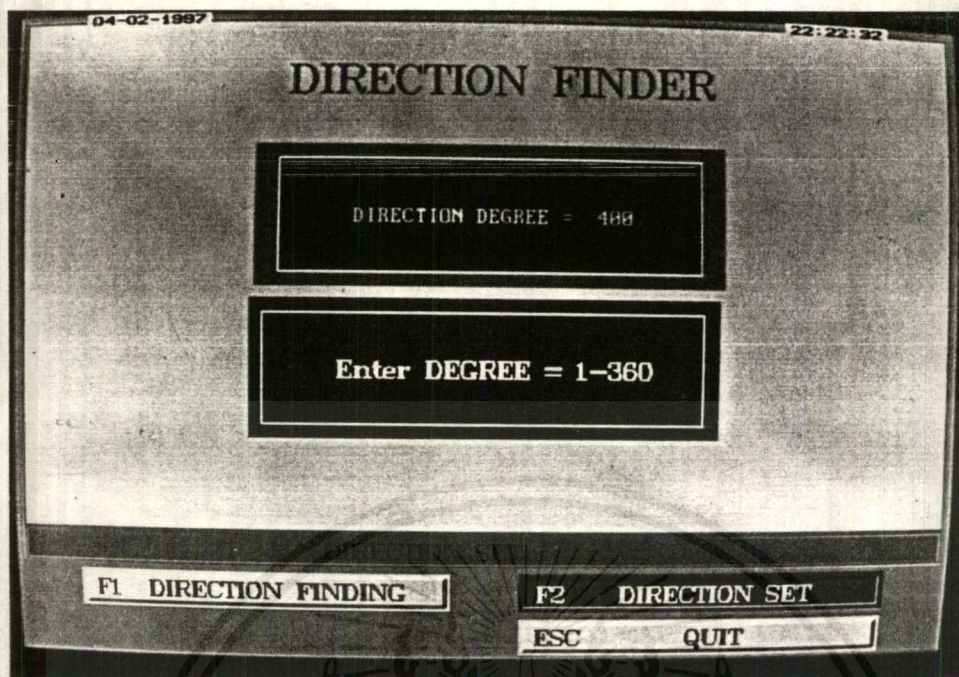
4.4.2 ฟังก์ชัน Direction Set

เมื่อมีภาคคลิกเมาส์ที่ปุ่ม "Direction Set" หรือกดคีย์ F2 ดังรูปที่ 4.15 จะเป็นการกำหนดให้สายอากาศหมุนไปยังทิศทางที่กำหนดให้เป็นองศา ซึ่งในการหาทิศทางนั้นสามารถเป็นการแสดงกราฟวงกลมก็ได้โดยการคลิกเมาส์ที่ปุ่มหรือกดคีย์ F1 หรือ F2 จากนั้นเมื่อต้องการเริ่มต้นก็คลิกเมาส์ที่ปุ่ม "START" หรือกดคีย์ F3 สายอากาศก็จะเริ่มหมุน เครื่องคอมพิวเตอร์ก็จะทำการแสดงทิศทางดังรูปที่ 4.17



รูปที่ 4.15 แสดงการเลือกฟังก์ชัน Direction Set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงความผิดพลาดเมื่อป้อนค่ามุมที่ไม่ถูกต้อง



รูปที่ 4.17 แสดงการหมุนของสายอากาศไปยังตำแหน่งที่ตั้งค่าไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และสรุป

จากการทดลองการสร้างและทดสอบวงจรในส่วนของการขับกระแสให้แก่สเตปป์มอเตอร์ และ วงจรดีเทคเตอร์สัญญาณ DC ซึ่งจากการทดลองนั้นได้ผลดังที่ได้สรุปไว้แล้ว คือ สามารถสร้างกระแสจำนวน เพียงพอให้แก่สเตปป์มอเตอร์ได้ และสามารถที่จะดีเทคค่าความแรงของสัญญาณอินพุทขนาดต่าง ๆ ให้เป็น แรงดันกระแสตรง ซึ่งจะส่งต่อไปยังส่วนของการแปลงให้เข้าเป็นสัญญาณดิจิทัล ซึ่งจะได้ผลการทดลองตาม หัวข้อ 4.3 เพื่อต่อไปยังพอร์ทของไมโครคอนโทรลเลอร์และส่งต่อไปยังคอมพิวเตอร์ต่อไป

ซึ่งจากการใช้โปรแกรมสั่งงานที่เขียนด้วยภาษานั้น ก็จะเป็นไปตามการทดลอง โดยนำสัญญาณ IF จากเครื่องรับวิทยุไปเข้าวงจรดีเทคเตอร์ และแสดงผลของสัญญาณที่รับได้ออกทางหน้าจคอมพิวเตอร์ โดย เป็นการแสดงสัญญาณที่รับได้ในทิศทางต่าง ๆ ที่มอเตอร์และเสาอากาศหมุนไป ซึ่งจะบอกความแรงของ สัญญาณในแต่ละทิศทางที่เข้ามาถึงเครื่องรับ ก็จะทำให้ทราบว่าสัญญาณวิทยุ นั้นส่งมาจากทิศทางใด



บรรณานุกรม

- [1] Takashi Kenjo, "Stepping motor and their microprocessor controls", Oxford University Press, 1986, P 121-165.
- [2] เขมิกอนดักเตอร์ อิเล็กทรอนิกส์, "เจาะกินหุ่นยนต์", ซีเอ็ดดูเคชั่น, 2535, หน้า 102-110
- [3] ทรงยศ บุญญฤทธิ์วงษ์, พงษ์พันธ์ วิมุกติบุตร, พิเชษฐ์ ร่วมสมัคร, "การหาทิศทางของคลื่น", ปรินญา นิพนธ์สาขาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2538
- [4] ปรีชา สุริย์แสง, สิทธิเดช ปิยะชาติ, "เครื่องปรับทิศทางสายอากาศด้วยสเตปปีงมอเตอร์", ปรินญา นิพนธ์อุตสาหกรรมบัณฑิต สาขาเทคโนโลยีอิเล็กทรอนิกส์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2536
- [5] ปฏิภาณ แสงเจริญ, ประเสริฐ มงคลรัตน์ชัย, เมธี ลิวโรจน์สกุล, "ระบบค้นหาทิศทางเครื่องส่งวิทยุ", ปรินญานิพนธ์สาขาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2536



ภาคผนวก ก.
โปรแกรมสั่งงานและแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#include <bios.h>
#include <alloc.h>
#include <math.h>
#include <string.h>
```

```
#define code      227
#define FUNC1    129
#define FUNC2    130
#define FUNC3    131
#define QUIT     138
#define CONTINUE 6
#define CANCEL   7
#define RUN      8
#define MERROR   245
#define DATA_END 170
```

```
union REGS regs;
struct time t;
struct date da;
```

```
int th_str,tm_str,ts_str,th_end,tm_end,ts_end;
int day_str,mon_str,yea_str;
int port=0,count=0,x,y,button,f,num,d,status,b,contin,g;
int datax,datay,err,i,index1,rfddata,rfdmax,degmax,rf_new[365];
int rfdmax_file,degmax_file,data_file[365];
int rfdmax_view,degmax_view,data_view[365];
int fth_str,ftm_str,fts_str,fth_end,ftm_end,fts_end;
int vth_str,vtm_str,vts_str,vth_end,vtm_end,vts_end;
int fdays_str,fmon_str,fyea_str,vdays_str,vmon_str,vyea_str;
int cx[365],cy[365],px[365],py[365];
char name[20],name1[50],fname[50],vname[50],s1[20],s2[20];
unsigned char key;
unsigned int size,size1,size2,size3;
```

```
void initg();
void clr_view(),view1(),view2(),view3();
void iconu(int x1,int y1,int x2,int y2,char ch[20]);
void iconn(int x1,int y1,int x2,int y2);
void icond(int x1,int y1,int x2,int y2,char ch[20]);
void init_port(),send_port(int data);
void send_function(int func,int data1,int data2);
void send_command1(int data);
void click_mouse(),key_function();
void init_mouse(),show_mouse(),hide_mouse();
void function_main(),function_1(),function_2();
void function_3(),function_4();
void function_5(),function_quit(),function_text();
void show_function(char ch[50]);
```

เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void sel_graph(),linear_graph(),circle_graph(),sel_graph_file();
void read_name(char ch[40]);
void in_data(int x,int y,int ind,char ch1[50],char ch2[10]);
void in_rfddata(int n),datamax(),in_error();
void plot_graph(int gr),get_dataxy(int gr);
void m_error(int x1,int y1,int x2,int y2,char ch[50]);
void m_ok(int x1,int y1,int x2,int y2,char ch[50]);
void m_operate(int col,char ch1[50]);
void message(int x1,int y1,int x2,int y2,char ch1[50],char ch2[15]);
void select_page(),clear_page();
void icon_menu(),icon_graph(int fn),icon_view();
void readoper(),inittime(),runtime(),run(int fn,int g);
void position1(int x,int color),position2(int rad,int input,int color);
void scan1(int x,int input,int px1,int py1);
void scan2(int rad,int input,int cx1,int cy1);
void cir_line(int fn),linear_line(int fn),cir(int fn),linear(int fn);
void deg_level(int fn),show_deg_level(int fn,int n),prindeg();
void deg_view(int page),data_trans(int fn),title_view();
void save_file(char ch[50]),load_file(char ch[50]);
void *view_data,*view_data1,*view_data2,*view_data3;

```

```

void main()
{
    initg();
    view1();
    view2();
    inittime();
    icon_menu();
    readoper();
    if(err==1) {
        message(120,300,530,400,"PLEASE CHECK EXTERNAL
HARDWARE","BEFORE RUN PROGRAM");
        delay(100);
        goto off_prog;
    }
    m_ok(167,209,472,309,"          HARDWARE READY          ");
    init_mouse();
    function_main();
off_prog:closegraph();
    function_text();
}

```

```

void initg()
{
    int driver,mode;
    detectgraph(&driver,&mode);
    if(driver!=9) {
        clrscr();
        initgraph(&driver,&mode,"\\tc\\bgi");
        putchar('\007');
        sleep(5);
        closegraph();
        exit(1);
    }
}

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    initgraph(&driver, &mode, "\\tc\\bgi");
    cleardevice();
    setcbrk(1);
    init_port();
}

void view1()
{
    setviewport(0, 0, 639, 479, 0);
    setbkcolor(BLACK);
    setcolor(MAGENTA);
    rectangle(1, 1, 638, 478);
    rectangle(2, 2, 637, 477);
    rectangle(7, 7, 632, 379);
    rectangle(8, 8, 631, 378);
    setfillstyle(SOLID_FILL, CYAN);
    floodfill(3, 3, MAGENTA);
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    floodfill(103, 30, MAGENTA);
    settextstyle(1, 0, 4);
    setcolor(BLUE);
    outtextxy(190, 35, "DIRECTION FINDER");
}

void readoper()
{
    int o;
    o = read_port();
    if(err==1) goto readout;
    send_port(o);
readout:;
}

void view2()
{
    setviewport(0, 0, 639, 479, 0);
    setcolor(MAGENTA);
    rectangle(7, 7, 632, 379);
    rectangle(8, 8, 631, 378);
    rectangle(7, 380, 632, 405);
    rectangle(8, 381, 631, 404);
    setfillstyle(SOLID_FILL, LIGHTGRAY);
    floodfill(10, 10, MAGENTA);
    setfillstyle(SOLID_FILL, GREEN);
    floodfill(10, 382, MAGENTA);
}

void icon_menu()
{
    setviewport(9, 382, 630, 403, 1);
    clearviewport();
    setviewport(0, 0, 639, 479, 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(MAGENTA);
rectangle(7,380,632,405);
rectangle(8,381,631,404);
setfillstyle(SOLID_FILL, GREEN);
floodfill(10,382, MAGENTA);
settextstyle(1,0,1);
setcolor(BLUE);
outtextxy(280,383, "MAIN MENU");
iconu(47,410,297,440, "F1  DIRECTION FINDING");
iconu(342,410,592,440, "F2  DIRECTION SET");
iconu(342,445,592,475, "ESC      QUIT");
}

```

```
void iconu(int x1,int y1,int x2,int y2,char ch[20])
```

```
/** not pushed icon **/
```

```

{
setfillstyle(SOLID_FILL, LIGHTGRAY);
setlinestyle(0,0,1);
bar(x1,y1,x2,y2);
setcolor(DARKGRAY);
moveto(x1,y1);
lineto(x1,y2);
lineto(x2,y2);
lineto(x2,y1);
lineto(x1,y1);
setcolor(8);
setlinestyle(0,0,3);
line(x2-4,y1+4,x2-4,y2-4);
line(x1+4,y2-4,x2-4,y2-4);
setcolor(WHITE);
setlinestyle(0,0,1);
line(x1+4,y1+4,x2-5,y1+4);
line(x1+4,y1+4,x1+4,y2-5);
setcolor(BLUE);
settextstyle(1,0,1);
outtextxy(x1+12,y1+5,ch);
}

```

```
void iconn(int x1,int y1,int x2,int y2)
```

```
/** remove icon **/
```

```

{
setfillstyle(SOLID_FILL, CYAN);
bar(x1,y1,x2,y2);
}

```

```
void icond(int x1,int y1,int x2,int y2,char ch[20])
```

```
/** pushed icon **/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    setfillstyle(SOLID_FILL,RED);
    bar(x1,y1,x2,y2);
    setcolor(DARKGRAY);
    moveto(x1,y1);
    lineto(x1,y2);
    lineto(x2,y2);
    lineto(x2,y1);
    lineto(x1,y1);
    setcolor(0);
    setlinestyle(0,0,3);
    line(x1+4,y1+4,x2-4,y1+4);
    line(x1+4,y1+4,x1+4,y2-4);
    setcolor(LIGHTGRAY);
    setlinestyle(0,0,1);
    line(x2-4,y1+4,x2-4,y2-4);
    line(x1+4,y2-4,x2-4,y2-4);
    setcolor(BLACK);
    settextstyle(1,0,1);
    outtextxy(x1+15,y1+8,ch);
    outtextxy(x1+14,y1+7,ch);
    setcolor(YELLOW);
    settextstyle(1,0,1);
    outtextxy(x1+13,y1+6,ch);
}

void save_view(int x1,int y1,int x2,int y2)
{
    size = imagesize(x1,y1,x2,y2);
    view_data=malloc(size);
    getimage(x1,y1,x2,y2,view_data);
}

void save_view1(int x1,int y1,int x2,int y2)
{
    size1 = imagesize(x1,y1,x2,y2);
    view_data1 = malloc(size1);
    getimage(x1,y1,x2,y2,view_data1);
}

void init_mouse()

/**/ initial mouse /**/

{
    regs.x.ax=0;
    int86(0x33,&regs,&regs);
    if(regs.x.ax==0) {
        m_error(167,209,472,309,"      No Mouse Available !");
        button=3;
        exit(0);
    }
}
regs.x.ax=7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

regs.x.cx=3;
regs.x.dx=635;
int86(0x33,&regs,&regs);
regs.x.ax=8;
regs.x.cx=3;
regs.x.dx=475;
int86(0x33,&regs,&regs);
show_mouse();
}

void show_mouse()
{
regs.x.ax=1;
int86(0x33,&regs,&regs);
}

void hide_mouse()
{
regs.x.ax=2;
int86(0x33,&regs,&regs);
}

void init_port()
/** initial port */
{
regs.h.ah=0;
regs.h.al=code;
regs.x.dx=port;
int86(0x14,&regs,&regs);
}

void send_port(int data)
/** send the data to serial port */
{
err=0;
regs.h.ah=1;
regs.h.al=data;
regs.x.dx=port;
int86(0x14,&regs,&regs);
if(regs.h.ah&128) {
m_error(167,209,472,309," SERIAL PORT SEND ERROR !");
err=1;
return;
}
runtime();
}

read_port()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/**/ read the data from serial port ***/

{
  unsigned long time=0;
  err=0;
  do {
    if(time>50000) {
      m_error(167,209,472,309," SERIAL PORT READ ERROR !");
      err=1;
      return(0);
    }

    time++;
  }
  while(!(check_stat()&256));
  regs.h.ah=2;
  regs.x.dx=port;
  int86(0x14,&regs,&regs);
  if(regs.h.ah&128) {
    m_error(167,209,472,309," DATA TRANSFER ERROR !");
    printf("read %d\n",regs.h.al);
    err=1;
    return(0);
  }
  return regs.h.al;
}

```

check_stat()

```

/**/ check the status of serial port ***/

```

```

{
  regs.h.ah=3;
  regs.x.dx=port;
  int86(0x14,&regs,&regs);
  return regs.x.ax;
}

```

void send_function(int func,int data1,int data2)

```

/**/ send the function to microcontroller ***/

```

```

{
  int r,dt;
  err=0;
  send_port(func);
  r = read_port();
  if(r!=func) {
    m_error(167,209,472,309," DATA TRANSFER ERROR !");
    err=1;
    return;
  }
  delay(500);
  send_port(data1);
}

```

```

dt = read_port();
if(dt!=data1) {
    m_error(167,209,472,309," DATA TRANSFER ERROR !");
    err=1;
    return;
}
delay(500);
send_port(data2);
dt = read_port();
if(dt!=data2) {
    m_error(167,209,472,309," DATA TRANSFER ERROR !");
    err=1;
    return;
}
}
}

```

```

void send_command1(int data)

```

```

/** send the command to microcontroller */

```

```

{
    int dt;
    err=0;
    send_port(data);
    dt = read_port();
    if(dt!=data) {
        m_error(167,209,472,309," DATA TRANSFER ERROR !");
        err=1;
        return;
    }
}

```

```

void inittime()

```

```

/** display the time at initial state */

```

```

{
    char s[20];
    gettime(&t);
    setfillstyle(SOLID_FILL,LIGHTGRAY);
    bar(519,1,589,11);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    setcolor(BLACK);
    sprintf(s,"%02d:%02d:%02d",t.ti_hour,t.ti_min,t.ti_sec);
    outtextxy(523,3,s);
    getdate(&da);
    setfillstyle(SOLID_FILL,LIGHTGRAY);
    bar(51,1,137,11);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    setcolor(BLACK);
    sprintf(s,"%02d-%02d-%d",da.da_mon,da.da_day,da.da_year);
}

```

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(55,3,s);
    day_str = da.da_day;
    mon_str = da.da_mon;
    yea_str = da.da_year;
}

void runtime()

/** display the current time */

{
    char s[20];
    gettime(&t);
    setfillstyle(SOLID_FILL,LIGHTGRAY);
    bar(519,1,589,11);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    setcolor(BLACK);
    sprintf(s,"%02d:%02d:%02d",t.ti_hour,t.ti_min,t.ti_sec);
    outtextxy(523,3,s);
    if(t.ti_hour==0 && t.ti_min==0 && t.ti_sec==0); {
        getdate(&da);
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        bar(51,1,137,11);
        setusercharsize(2,5,2,5);
        settextstyle(0,0,USER_CHAR_SIZE);
        setcolor(BLACK);
        sprintf(s,"%02d-%02d-%d",da.da_mon,da.da_day,da.da_year);
        outtextxy(55,3,s);
    }
}

void click_mouse()

/** receive the command from mouse */

{
    regs.x.ax=3;
    int86(0x33,&regs,&regs);
    button=regs.x.bx&3;
    x=regs.x.cx;
    y=regs.x.dx;
    if(count==500) {
        runtime();
        count=0;
    }
    count=count++;
}

void key_function()

/** receive command from keyboard */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    regs.h.ah=0x0c;
    int86(0x21,&regs,&regs);
    key=inport(0x60)&255;
}

void function_main()

/** select function */

{
    do {
        click_mouse();
        key_function();
        if(x<=297 && x>=47 && y<=440 && y>=410) {
            if(button&1) {
                hide_mouse();
                f=1;
                icond(47,410,297,440,"F1 DIRECTION FINDING");
                function_1();
                icon_menu();
                show_mouse();
            }
        }
        if(x<=592 && x>=342 && y<=440 && y>=410) {
            if(button&1) {
                hide_mouse();
                f=2;
                icond(342,410,592,440,"F2 DIRECTION SET");
                function_2();
                icon_menu();
                show_mouse();
            }
        }
        if(x<=592 && x>=342 && y<=475 && y>=445) {
            if(button&1) {
                hide_mouse();
                icond(342,445,592,475,"ESC QUIT");
                f=10;
                button=3;
                function_quit();
                delay(500);
                show_mouse();
            }
        }
        if(key==0x3b) {
            hide_mouse();
            f=1;
            icond(47,410,297,440,"F1 DIRECTION FINDING");
            function_1();
            icon_menu();
            show_mouse();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(key==0x3c) {
            hide_mouse();
            f=2;
            icond(342,410,592,440,"F2    DIRECTION SET");
            function_2();
            icon_menu();
            show_mouse();
        }
        if(key==0x10||key==0x01) {
            hide_mouse();
            icond(342,445,592,475,"ESC          QUIT");
            f=10;
            button=3;
            function_quit();
            delay(500);
            show_mouse();
        }
    }
    while(button!=3);
}

void clr_view()
{
    if(num!=f) return;
    setviewport(9,9,630,377,1);
    clearviewport();
    setviewport(0,0,639,479,0);
}

void function_quit()
{
    send_command1(QUIT);
}

void function_1()

/**/ direction finding /**/

{
    char s[50];
    int data1=4;
    int data2=0;
    rfddata=0;
    err=0;
    num=f;
    view2();
    show_function("DIRECTION FINDING");
    m_operate(2,"TRANSFERING DATA");
    delay(10);
    send_function(FUNC1,data1,data2);
    delay(10);
    free(view_data);
    if(err!=1) m_ok(167,209,472,309,"    DATA TRANSFER SUCCESS");
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น มิใช่เพื่อเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    icon_graph(f);
    show_mouse();
    sel_graph();
    hide_mouse();
}

void function_2()

/** direction set */

{
    char s[50];
    int degree,data1,data2;
    int z1=0;
    rfddata=0;
    num=f;
    view2();
    show_function("DIRECTION SET");
test:  in_data(30,10,20,"DIRECTION DEGREE = ", "%d");
    degree = d;
    if(degree<1||degree>360) {
        in_error();
        z1=1;
        goto test;
    }
    if(z1==1) {
        free(view_data);
    }
    data1 = degree/90;
    data2 = degree%90;
    m_operate(2,"TRANSFERING DATA");
    delay(10);
    send_function(FUNC2,data1,data2);
    free(view_data);
    delay(10);
    if(err!=1) m_ok(167,209,472,309," DATA TRANSFER O.K. !");
    icon_graph(f);
    circle_graph();
    show_mouse();
    sel_graph();
    hide_mouse();
}

```

```
void data_trans(int fn)
```

```
/** transfer data from running part to view part */
```

```

{
    if(fn==1) {
        for(i=1;i<=360;i++) {
            data_view[i]=rf_new[i];
        }
        rfmax_view=rfmax;
    }
}

```

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยส่วนงานที่เกี่ยวข้องกับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

degmax_view=degmax;
vth_str=th_str;
vtm_str=tm_str;
vts_str=ts_str;
vth_end=th_end;
vtm_end=tm_end;
vts_end=ts_end;
vday_str=day_str;
vmon_str=mon_str;
veya_str=yea_str-1900;
strcpy(vname,"DIRECTION FINDING");
}
}

```

```
void function_5()
```

```
/** function list data in numeric value **/
```

```

{
num=f;
setfillstyle(SOLID_FILL,WHITE);
bar(1,1,639,479);
view3();
inittime();
select_page();
setfillstyle(SOLID_FILL,WHITE);
bar(1,1,639,479);
view1();
view2();
inittime();
}

```

```
void select_page()
```

```
/** select the page of data to list on screen **/
```

```

{
int pag=1;
title_view();
icon_view();
deg_view(pag);
show_mouse();
do {
click_mouse();
key_function();
if(x<=373 && x>=263 && y<=470 && y>=440) {
if(button&1) {
hide_mouse();
icond(263,440,373,470,"F5  PREV");
if(pag > 1) {
pag=pag-1;
clear_page();
deg_view(pag);
}
}
}
}

```

```

    }
    iconu(263,440,373,470,"F5  PREV");
    show_mouse();
}
}
if(x<=497 && x>=387 && y<=470 && y>=440) {
    if(button&1) {
        hide_mouse();
        icond(387,440,497,470,"F6  NEXT");
        if(pag < 3) {
            pag=pag+1;
            clear_page();
            deg_view(pag);
        }
        iconu(387,440,497,470,"F6  NEXT");
        show_mouse();
    }
}
if(x<=621 && x>=511 && y<=470 && y>= 440) {
    if(button&1) {
        hide_mouse();
        icond(511,440,621,470,"F10  EXIT");
        button=5;
        delay(200);
        show_mouse();
    }
}
if(key==0x3f) {
    hide_mouse();
    icond(263,440,373,470,"F5  PREV");
    if(pag > 1) {
        pag=pag-1;
        clear_page();
        deg_view(pag);
    }
    iconu(263,440,373,470,"F5  PREV");
    show_mouse();
}
if(key==0x40) {
    hide_mouse();
    icond(387,440,497,470,"F6  NEXT");
    if(pag < 3) {
        pag=pag+1;
        clear_page();
        deg_view(pag);
    }
    iconu(387,440,497,470,"F6  NEXT");
    show_mouse();
}
if(key==0x44) {
    hide_mouse();
    icond(511,440,621,470,"F10  EXIT");
    button=5;

```



```

        delay(200);
        show_mouse();
    }
}
while(button!=5);
hide_mouse();
}

void view3()

/**/ data list screen ***/

{
    setcolor(MAGENTA);
    rectangle(1,1,638,478);
    rectangle(2,2,637,477);
    rectangle(7,7,631,53);
    rectangle(8,8,630,52);
    rectangle(7,61,631,432);
    setcolor(BLUE);
    rectangle(8,62,630,431);
    setfillstyle(SOLID_FILL,CYAN);
    floodfill(3,3,MAGENTA);
    setfillstyle(SOLID_FILL,WHITE);
    floodfill(10,10,MAGENTA);
    setfillstyle(SOLID_FILL,WHITE);
    floodfill(10,100,BLUE);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    setcolor(BLUE);
    moveto(8,80);
    lineto(630,80);
    outtextxy(20,68,"DEG.    INPUT");
    outtextxy(144,68,"DEG.    INPUT");
    outtextxy(268,68,"DEG.    INPUT");
    outtextxy(392,68,"DEG.    INPUT");
    outtextxy(516,68,"DEG.    INPUT");
    outtextxy(20,18,"NAME");
    outtextxy(20,37,"DATE");
    outtextxy(400,18,"START TIME");
    outtextxy(400,37,"  END TIME");
    outtextxy(170,18,"DEGREE MAXIMUM =");
    outtextxy(170,37,"DATA MAXIMUN =");
    outtextxy(575,37,"PAGE");
}

void title_view()          /****    CHANGE THE VALUE    *****/

/**/ view the value of degree and level at maximun point ***/

{
    char s[20];
    setusercharsize(2,5,2,5);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

settextstyle(0,0,USER_CHAR_SIZE);
setcolor(BROWN);
sprintf(s,"%s",vname);
outtextxy(65,18,s);
sprintf(s,"%02d-%02d-%4d",vmon_str,vday_str,vyea_str+1900);
outtextxy(65,37,s);
sprintf(s,"%3d",degmax_view);
outtextxy(307,18,s);
sprintf(s,"%3d",rfmax_view);
outtextxy(307,37,s);
sprintf(s,"%02d:%02d:%02d",vth_str,vtm_str,vts_str);
outtextxy(490,18,s);
sprintf(s,"%02d:%02d:%02d",vth_end,vtm_end,vts_end);
outtextxy(490,37,s);
}

void icon_view()
{
    iconu(263,440,373,470,"F5    PREV");
    iconu(387,440,497,470,"F6    NEXT");
    iconu(511,440,621,470,"F10  EXIT");
}

void clear_page()
{
    setfillstyle(SOLID_FILL,WHITE);
    bar(10,85,627,428);
}

void deg_view(int page)          /***** THIS FUNCTION TOO *****/

/**/ display the value of degree and signal level of every direction
***/

{
    int deg,dx,dy;
    char s[8];
    setcolor(LIGHTGRAY);
    moveto(132,62);
    lineto(132,431);
    moveto(256,62);
    lineto(256,431);
    moveto(380,62);
    lineto(380,431);
    moveto(504,62);
    lineto(504,431);
    setfillstyle(SOLID_FILL,WHITE);
    bar(610,34,620,48);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    setcolor(BROWN);
    sprintf(s,"%d",page);
    outtextxy(613,37,s);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (deg=1;deg<=120;deg++) {
    if(deg<=24) {
        dx=20;
        dy=78+(14*deg);
    }
    if(deg>=25 && deg<=48) {
        dx=144;
        dy=78+(14*(deg-24));
    }
    if(deg>=49 && deg<=72) {
        dx=268;
        dy=78+(14*(deg-48));
    }
    if(deg>=73 && deg<=96) {
        dx=392;
        dy=78+(14*(deg-72));
    }
    if(deg>=97 && deg<=120) {
        dx=516;
        dy=78+(14*(deg-96));
    }
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    if(page==1) {
        sprintf(s1,"%3d",deg);
        sprintf(s2,"%3d",data_view[deg]);
    }
    if(page==2) {
        sprintf(s1,"%3d",deg+120);
        sprintf(s2,"%3d",data_view[deg+120]);
    }
    if(page==3) {
        sprintf(s1,"%3d",deg+240);
        sprintf(s2,"%3d",data_view[deg+240]);
    }
    setcolor(BLUE);
    outtextxy(dx,dy,s1);
    setcolor(BLUE);
    outtextxy(dx+32,dy,"=");
    setcolor(BLUE);
    outtextxy(dx+45,dy,s2);
}
}

```

```
void function_6()
```

```
/**/ function select port number /**/
```

```

{
    char p;
    num=f;
    view2();
    show_function(" PORT SELECT");
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setlinestyle(0,0,1);
setcolor(GREEN);
rectangle(250,150,595,250);
setfillstyle(SOLID_FILL, GREEN);
floodfill(272,202, GREEN);
setcolor(WHITE);
rectangle(265,160,580,240);
setfillstyle(SOLID_FILL, 0);
floodfill(300,220, WHITE);
textcolor(WHITE);
gotoxy(37,13);
printf("SELECT PORT (1=com1 or 2=com2) : ");
p=getche();
if(p=='1') {
    port=0;
}
if(p=='2') {
    port=1;
}
init_port();
setfillstyle(SOLID_FILL, LIGHTGRAY);
floodfill(400,200, MAGENTA);
}

```

```
void linear_graph()
```

```
/** build the linear graph structure */
```

```

{
    setviewport(9,9,630,377,1);
    setfillstyle(SOLID_FILL, WHITE);
    bar(1,1,620,367);
    linear_line(f);
    setcolor(BLUE);
    setlinestyle(0,0,2);
    moveto(120,330);
    lineto(120,338);
    moveto(210,330);
    lineto(210,338);
    moveto(300,330);
    lineto(300,338);
    moveto(390,330);
    lineto(390,338);
    setcolor(BLUE);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    outtextxy(27,343,"0");
    outtextxy(113,343,"90");
    outtextxy(199,343,"180");
    outtextxy(289,343,"270");
    outtextxy(379,343,"360");
    outtextxy(440,335,"Angle (degree)");
    setusercharsize(1,2,1,2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือมีการขังในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    deg_level(f);
    setviewport(0,0,639,479,0);
}

void circle_graph()

/** build the circle graph structure */

{
    setviewport(9,9,630,377,1);
    setfillstyle(SOLID_FILL,WHITE);
    bar(1,1,620,367);
    setcolor(BLUE);
    setlinestyle(0,0,2);
    rectangle(30,50,370,350);
    rectangle(28,48,372,352);
    cir_line(f);
    setcolor(BLUE);
    setusercharsize(2,5,2,5);
    settextstyle(0,0,USER_CHAR_SIZE);
    outtextxy(197,57,"0");
    outtextxy(337,197,"90");
    outtextxy(189,337,"180");
    outtextxy(40,197,"270");
    outtextxy(263,76,"30");
    outtextxy(316,127,"60");
    outtextxy(313,268,"120");
    outtextxy(261,319,"150");
    outtextxy(115,319,"210");
    outtextxy(60,268,"240");
    outtextxy(60,127,"300");
    outtextxy(115,76,"330");
    setcolor(BLUE);
    setlinestyle(0,0,2);
    moveto(200,200);
    lineto(200,70);
    deg_level(f);
    setviewport(0,0,639,479,0);
}

void in_data(int x,int y,int ind,char ch1[50],char ch2[10])

/** get the value of degree from keyboard (use in "direction set") */

{
    char s[6];
    save_view(170,100,475,200);
    setlinestyle(0,0,1);
    setcolor(GREEN);
    rectangle(170,100,475,200);
    setfillstyle(SOLID_FILL,GREEN);
    floodfill(172,152,GREEN);
    setcolor(WHITE);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rectangle(185,110,460,190);
setfillstyle(SOLID_FILL,0);
floodfill(200,170,WHITE);
textcolor(15);
gotoxy(x,y);
printf(ch1);
gotoxy(x+ind,y);
gets(s);
sscanf(s,ch2,&d);
runtime();
free(view_data);
}

```

```
void m_error(int x1,int y1,int x2,int y2,char ch[50])
```

```
/** show the interrupt message when found the error ***/
```

```

{
err=1;
save_view(x1,y1,x2,y2);
sound(2000);
setlinestyle(0,0,1);
setcolor(RED);
rectangle(x1,y1,x2,y2);
setfillstyle(SOLID_FILL,RED);
floodfill(x1+20,y1+20,RED);
setcolor(LIGHTGRAY);
rectangle(x1+15,y1+10,x2-15,y2-10);
nosound();
setcolor(LIGHTGRAY);
settextstyle(1,0,1);
outtextxy(x1+5,y1+25,ch);
setcolor(GREEN);
settextstyle(0,0,1);
outtextxy(x1+35,y1+60,"Press any key to continue");
getch();
runtime();
free(view_data);
}

```

```
void m_ok(int x1,int y1,int x2,int y2,char ch[50])
```

```
/** show the message when complete the hardware check ***/
```

```

{
save_view(x1,y1,x2,y2);
setlinestyle(0,0,1);
setcolor(MAGENTA);
rectangle(x1,y1,x2,y2);
setfillstyle(SOLID_FILL,GREEN);
floodfill(x1+2,y1+2,MAGENTA);
setcolor(YELLOW);
rectangle(x1+15,y1+10,x2-15,y2-10);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    setcolor(YELLOW);
    settextstyle(1,0,1);
    outtextxy(x1+5,y1+25,ch);
    delay(200);
    free(view_data);
}

```

```
void in_error()
```

```
/** show the message when enter incorrect degree ***/
```

```

{
    save_view1(167,209,472,309);
    setlinestyle(0,0,1);
    setcolor(RED);
    rectangle(167,209,472,309);
    setfillstyle(SOLID_FILL,RED);
    floodfill(190,250,RED);
    setcolor(WHITE);
    rectangle(175,219,462,299);
    setcolor(YELLOW);
    settextstyle(1,0,1);
    outtextxy(205,249," Enter DEGREE = 1-360");
    runtime();
}

```

```
void m_operate(int col,char ch1[50])
```

```
/** show the message while something operate ***/
```

```

{
    setviewport(0,0,639,479,0);
    save_view(167,209,472,309);
    setlinestyle(0,0,1);
    setcolor(col);
    rectangle(167,209,472,309);
    setfillstyle(SOLID_FILL,col);
    floodfill(170,212,col);
    setcolor(YELLOW);
    rectangle(182,219,457,299);
    setcolor(YELLOW);
    settextstyle(1,0,1);
    outtextxy(230,260,ch1);
    delay(100);
    runtime();
}

```

```
void show_function(char ch[50])
```

```
/** show the menu title ***/
```

```

{
    setviewport(9,382,630,403,1);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clearviewport();
setviewport(0,0,639,479,0);
setcolor(MAGENTA);
rectangle(7,380,632,405);
rectangle(8,381,631,404);
setfillstyle(SOLID_FILL, GREEN);
floodfill(9,382, MAGENTA);
settextjustify(CENTER_TEXT, TOP_TEXT);
settextstyle(1, 01);
setcolor(BLUE);
outtextxy(280,383, ch);
settextjustify(LEFT_TEXT, TOP_TEXT);
}

```

```
void icon_graph(int fn)
```

```
/** show the menu */
```

```

{
  if(fn==1) {
    iconu(47,410,297,440,"F1 LINEAR GRAPH");
    iconu(342,410,592,440,"F2 CIRCLE GRAPH");
    iconn(47,445,297,475);
    iconu(342,445,592,475,"F10 CANCEL");
  }
  if(fn==2) {
    iconn(47,410,297,440);
    iconn(342,410,592,440);
    iconu(47,445,297,475,"F3 START");
    iconu(342,445,592,475,"F10 CANCEL");
  }
  if(fn==3) {
    iconu(47,410,297,440,"F1 LINEAR GRAPH");
    iconu(342,410,592,440,"F2 CIRCLE GRAPH");
    iconu(47,445,297,475,"F3 VIEW DATA");
    iconu(342,445,592,475,"F10 CANCEL");
  }
}
}

```

```
void sel_graph()
```

```
/** select the type of graph to show (while finding the direction) */
```

```

{
  int g=0;
  err=0;
  if(f==2) g=2;
  do {
graph:   click_mouse();
         key_function();
         if(x<=297 && x>=47 && y<=440 && y>=410) {
           if(button&1 && f==1) {
             hide_mouse();

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        iconu(342,410,592,440,"F2      CIRCLE GRAPH");
        icond(47,410,297,440,"F1      LINEAR GRAPH");
        iconu(47,445,297,475,"F3      START");
        g=1;
        linear_graph();
        show_mouse();
    }
}
if(x<=297 && x>=47 && y<=440 && y>=410) {
    if(button&1 && f==1) {
        hide_mouse();
        iconu(47,410,297,440,"F1      LINEAR GRAPH");
        icond(342,410,592,440,"F2      CIRCLE GRAPH");
        iconu(47,445,297,475,"F3      START");
        g=2;
        circle_graph();
        show_mouse();
    }
}
if(x<=297 && x>=47 && y<=475 && y>=445) {
    if(button&1) {
        if(g==0)goto graph;
        hide_mouse();
        icond(47,445,297,475,"F3      START");
        iconn(342,445,592,475);
        send_command1(RUN);
        show_mouse();
        run(f,g);
        hide_mouse();
        if(f==2) {
            show_mouse();
            goto past;
        }
        data_trans(1);
        icon_graph(3);
        show_mouse();
        sel_graph_file();
        button=4;
        delay(500);
    }
}
if(x<=592 && x>=342 && y<=475 && y>=445) {
    if(button&1) {
        hide_mouse();
        icond(342,445,592,475,"F10      CANCEL");
        send_command1(CANCEL);
        button=4;
        err=15;
        delay(300);
        show_mouse();
    }
}
}
if(key==0x3b&&f==1) {

```

past:

```

hide_mouse();
iconu(342,410,592,440,"F2          CIRCLE GRAPH");
icond(47,410,297,440,"F1          LINEAR GRAPH");
iconu(47,445,297,475,"F3          START");
g=1;
linear_graph();
show_mouse();
}
if(key==0x3c&&f==1) {
hide_mouse();
iconu(47,410,297,440,"F1          LINEAR GRAPH");
icond(342,410,592,440,"F2          CIRCLE GRAPH");
iconu(47,445,297,475,"F3          START");
g=2;
circle_graph();
show_mouse();
}
if(key==0x3d) {
if(g==0)goto graph;
hide_mouse();
icond(47,445,297,475,"F3          START");
icon(342,445,592,475);
send_command1(RUN);
show_mouse();
key=0x00;
run(f,g);
hide_mouse();
if(f==2) {
show_mouse();
goto past1;
}
data_trans(1);
icon_graph(3);
show_mouse();
sel_graph_file();
past1:
button=4;
delay(500);
}
if(key==0x44) {
hide_mouse();
icond(342,445,592,475,"F10          CANCEL");
send_command1(CANCEL);
button=4;
err=15;
delay(500);
show_mouse();
}
}
while(button!=4);
}

```

```
void sel_graph_file()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/** select the type of data to show (after the end of finding) */
```

```
{
  int i,a=0;
  f=3;
  show_function(vname);
  do {
    click_mouse();
    key_function();
    if(x<=297 && x>=47 && y<=440 && y>=410) {
      if(button&1) {
view_1:          a=1;
                  hide_mouse();
                  iconu(342,410,592,440,"F2      CIRCLE GRAPH");
                  iconu(47,445,297,475,"F3      VIEW DATA");
                  icond(47,410,297,440,"F1      LINEAR GRAPH");
                  linear_graph();
                  setviewport(9,9,630,377,1);
                  px[1]=31;
                  py[1]=330-data_view[1];
                  putpixel(px[1],py[1],RED);
                  for(i=2;i<=360;i++) {
                    scan1(i,data_view[i],px[i-1],py[i-1]);
                  }
                  position1(degmax_view,GREEN);
                  show_deg_level(3,degmax_view);
                  setviewport(0,0,639,479,0);
                  show_mouse();
                }
            }
    if(x<=592 && x>=342 && y<=440 && y>=410) {
      if(button&1) {
view_2:          a=2;
                  hide_mouse();
                  iconu(47,410,297,440,"F1      LINEAR GRAPH");
                  iconu(47,445,297,475,"F3      VIEW DATA");
                  icond(342,410,592,440,"F2      CIRCLE GRAPH");
                  circle_graph();
                  setviewport(9,9,630,377,1);
                  setcolor(DARKGRAY);
                  moveto(200,200);
                  lineto(200,70);
                  cx[0]=(data_view[360]/2)*cos((360+270)*M_PI/180.0)+200;
                  cy[0]=(data_view[360]/2)*sin((360+270)*M_PI/180.0)+200;
                  putpixel(cx[0],cy[0],RED);
                  for(i=1;i<=360;i++) {
                    scan2(i,data_view[i],cx[i-1],cy[i-1]);
                  }
                  position2(degmax_view,130,GREEN);
                  show_deg_level(3,degmax_view);
                  setvie-5Xwport(0,0,639,479,0);
                  show_mouse();
                }
            }
          }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(x<=297 && x>=47 && y<=475 && y>=445) {
    if(button&1) {
        hide_mouse();
        iconu(47,410,297,440,"F1      LINEAR GRAPH");
        iconu(342,410,592,440,"F2      CIRCLE GRAPH");
        icond(47,445,297,475,"F3      VIEW DATA");
        function_5();
        icon_graph(3);
        show_function(vname);
        show_mouse();
        if(a==1) {
            button=0;
            goto view_11;
        }
        if(a==2) {
            button=0;
            goto view_12;
        }
    }
}
if(x<=592 && x>=342 && y<=475 && y>=445) {
    if(button&1) {
        hide_mouse();
        icond(342,445,592,475,"F10     CANCEL");
        send_command1(CANCEL);
        button=4;
        delay(300);
        show_mouse();
    }
}
if(key==0x3b) {
view_11:
    a=1;
    hide_mouse();
    iconu(342,410,592,440,"F2      CIRCLE GRAPH");
    iconu(47,445,297,475,"F3      VIEW DATA");
    icond(47,410,297,440,"F1      LINEAR GRAPH");
    linear_graph();
    setviewport(9,9,630,377,1);
    px[1]=31;
    py[1]=330-data_view[1];
    putpixel(px[1],py[1],RED);
    for(i=2;i<=360;i++) {
        scanl(i,data_view[i],px[i-1],py[i-1]);
    }
    position1(degmax_view,GREEN);
    show_deg_level(3,degmax_view);
    setviewport(0,0,639,479,0);
    show_mouse();
}
if(key==0x3c) {
view_12:
    a=2;
    hide_mouse();

```

```

iconu(47,410,297,440,"F1      LINEAR GRAPH");
iconu(47,445,297,475,"F3      VIEW DATA");
icond(342,410,592,440,"F2      CIRCLE GRAPH");
circle_graph();
setviewport(9,9,630,377,1);
setcolor(DARKGRAY);
moveto(200,200);
lineto(200,70);
cx[0]=(data_view[360]/2)*cos((360+270)*M_PI/180.0)+200;
cy[0]=(data_view[360]/2)*sin((360+270)*M_PI/180.0)+200;
putpixel(cx[0],cy[0],RED);
for(i=1;i<=360;i++) {
    scan2(i,data_view[i],cx[i-1],cy[i-1]);
}
position2(degmax_view,130,GREEN);
show_deg_level(3,degmax_view);
setviewport(0,0,639,479,0);
show_mouse();
}
if(key==0x3d) {
    hide_mouse();
    iconu(47,410,297,440,"F1      LINEAR GRAPH");
    iconu(342,410,592,440,"F2      CIRCLE GRAPH");
    icond(47,445,297,475,"F3      VIEW DATA");
    delay(50);
    function_5();
    icon_graph(3);
    show_function(vname);
    show_mouse();
    if(a==1) {
        key=0x00;
        delay(50);
        button=0;
        goto view_1;
    }
    if(a==2) {
        key=0x00;
        delay(50);
        button=0;
        goto view_2;
    }
}
if(key==0x44) {
    hide_mouse();
    icond(342,445,592,475,"F10      CANCEL");
    send_command1(CANCEL);
    button=4;
    delay(500);
    show_mouse();
}
}
while(button!=4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void in_rfdata(int n)

/**/ read the signal level data from ADC to rf_new ***/

{
  setviewport(0,0,639,479,0);
  index1=read_port();
  delay(50);
  send_port(index1);
  if(index1==DATA_END) goto dtend;
  rfdata=read_port();
  rf_new[n]=rfdata;
dtend: delay(3);
  setviewport(9,9,630,377,1);
}

void run(int fn,int g)

/**/ finding the direction ***/

{
  if(g==1) {
    hide_mouse();
    linear(fn);
    show_mouse();
  }
  if(g==2) {
    hide_mouse();
    cir(fn);
    show_mouse();
  }
}

void datamax()

/**/
/* compare the value from each direction to find the direction */
/* of the strongest signal , display this degree and value to */
/* the motor and then send the command to control antenna goes */
/* to that direction */
/**/

{
  int ax,ay,l,deg,data1,data2;
  char s[10];
  ay=rf_new[1];
  ax=1;
  for(l=2;l<=360;l++) {
    if(ay<rf_new[l]) {
      ay=rf_new[l];
      ax=l;
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
deg=360-ax;
data1=deg/90;
data2=deg%90;
setusercharsize(2,5,2,5);
settextstyle(0,0,USER_CHAR_SIZE);
setcolor(BLUE);
outtextxy(210,15,"MAX DEGREE =");
outtextxy(210,30," RF SIGNAL =");
sprintf(s,"%3d",ax);
outtextxy(333,15,s);
sprintf(s,"%3d",ay);
outtextxy(333,30,s);
send_function(FUNC3,data1,data2);
delay(100);
send_command1(RUN);
rfmax=ay;
degmax=ax;
}

```

```
void cir(int fn)
```

```

/*****
/* display the circle graph, */
/* receive the signal level of each direction */
/* and display at the monitor as circle graph */
*****/
{
  if(fn==1) {
    int rp,l;
    int n=1;
    rp=130;
    gettime(&t);
    th_str = t.ti_hour;
    tm_str = t.ti_min;
    ts_str = t.ti_sec;
    day_str = da.da_day;
    mon_str = da.da_mon;
    yea_str = da.da_year;
    in_rfdata(n);
    cx[n]=(rfdata/2)*cos(270*M_PI/180.0)+200;
    cy[n]=(rfdata/2)*sin(270*M_PI/180.0)+200;
    setviewport(9,9,630,377,1);
    position2(n,rp,GREEN);
    putpixel(cx[n],cy[n],RED);
    show_deg_level(fn,n);
    send_port(rfdata);
    do {
      n=n+1;
      in_rfdata(n);
      if(index1==DATA_END) {

```

```
        button=5;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        n=n-1;
        goto outc1;
    }
    position2(n-1,rp,WHITE);
    position2(n,rp,GREEN);
    scan2(n,rfddata,cx[n-1],cy[n-1]);
    if(n==2||n==182) {
        setcolor(DARKGRAY);
        moveto(200,70);
        lineto(100,330);
    }
    if(n==92||n==272) {
        setcolor(DARKGRAY);
        moveto(70,200);
        lineto(330,200);
    }
    if(n==31||n==211) {
        setcolor(LIGHTGRAY);
        moveto(135,312);
        lineto(256,87);
    }
    if(n==62||n==242) {
        setcolor(LIGHTGRAY);
        moveto(87,264);
        lineto(312,134);
    }
    if(n==332||n==152) {
        setcolor(LIGHTGRAY);
        moveto(135,87);
        lineto(256,312);
    }
    if(n==301||n==121); {
        setcolor(LIGHTGRAY);
        moveto(87,135);
        lineto(312,265);
    }
}
if((n%10)==0) cir_line(fn);
if((n%10)!=0) delay(15);
show_deg_level(fn,n);
send_port(rfddata);
outc1:
    if(n>4) {
        for(l=n-3;l<=n;l++) {
            setcolor(RED);
            moveto(cx[l],cy[l]);
            lineto(cx[l-1],cy[l-1]);
        }
    }
}
while(button!=5);
button=0;
gettime(&t);
th_end = t.ti_hour;
tm_end = t.ti_min;

```

```

ts_end = t.ti_sec;
position2(n, rp, GREEN);
cx[0]=cx[360];
cy[0]=cy[360];
datamax();
do {
    index1=read_port();
    if(index1==DATA_END) {
        button=5;
        goto outc2;
    }
    n=n-1;
    position2(n+1, rp, WHITE);
    position2(n, rp, GREEN);
    cir_line(fn);
    show_deg_level(fn, n);
    setcolor(RED);
    if(n<358) {
        for(l=n;l<=n+2;l++) {
            moveto(cx[l], cy[l]);
            lineto(cx[l+1], cy[l+1]);
        }
    }
    setviewport(0, 0, 639, 479, 0);
outc2:    send_port(index1);
    setviewport(9, 9, 630, 377, 1);
}
while(button!=5);
setviewport(0, 0, 639, 479, 0);
}
if(fn==2) {
    int rp, l;
    int n=1;
    rp=130;
    setviewport(9, 9, 630, 377, 1);
    setcolor(WHITE);
    moveto(200, 200);
    lineto(200, 70);
    do {
        index1=read_port();
        if(index1==DATA_END) {
            button=5;
            n=n-1;
            goto outc3;
        }
        position2(n-1, rp, WHITE);
        position2(n, rp, GREEN);
        cir_line(fn);
        show_deg_level(fn, n);
        n=n+1;
        setviewport(0, 0, 639, 479, 0);
outc3:    send_port(index1);
        delay(28);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setviewport(9,9,630,377,1);
    }
    while(button!=5);
    position2(n,rp,GREEN);
    setviewport(0,0,639,479,0);
}
}

```

```
void position2(int rad,int input,int color)
```

```
/** draw the line to show current degree (circle graph) */
```

```

{
    int cx,cy;
    cx=input*cos((rad+270)*M_PI/180.0)+200;
    cy=input*sin((rad+270)*M_PI/180.0)+200;
    setcolor(color);
    setlinestyle(0,0,2);
    moveto(200,200);
    lineto(cx,cy);
}

```

```
void scan2(int rad,int input,int cx1,int cy1)
```

```
/** draw the line from previous point to current point (circle graph)
***/
```

```

{
    cx[rad]=(input/2)*cos((rad+270)*M_PI/180.0)+200;
    cy[rad]=(input/2)*sin((rad+270)*M_PI/180.0)+200;
    setcolor(RED);
    setlinestyle(0,0,1);
    moveto(cx[rad],cy[rad]);
    lineto(cx1,cy1);
}

```

```
void cir_line(int fn)
```

```
/** draw the axis of circle graph */
```

```

{
    if(fn==1||fn==3) {
        setcolor(DARKGRAY);
        setlinestyle(0,0,2);
        moveto(70,200);
        lineto(330,200);
        moveto(200,70);
        lineto(200,330);
        circle(200,200,128);
        circle(200,200,64);
        setcolor(LIGHTGRAY);
        setlinestyle(0,0,1);
        circle(200,200,96);
    }
}

```

```

    circle(200,200,32);
    moveto(135,87);
    lineto(265,312);
    moveto(87,135);
    lineto-5X(312,265);
    moveto(135,312);
    lineto(265,87);
    moveto(87,264);
    lineto(312,134);
}

if(fn==2) {
    setcolor(DARKGRAY);
    setlinestyle(0,0,2);
    moveto(70,200);
    lineto(85,200);
    moveto(315,200);
    lineto(330,200);
    moveto(200,70);
    lineto(200,85);
    moveto(200,315);
    lineto(200,330);
    moveto(135,87);
    lineto(142,100);
    moveto(257,299);
    lineto(265,312);
    moveto(87,135);
    lineto(100,142);
    moveto(299,257);
    lineto(312,265);
    moveto(135,312);
    lineto(142,299);
    moveto(257,100);
    lineto(265,87);
    moveto(87,264);
    lineto(100,257);
    moveto(299,142);
    lineto(312,134);
    setcolor(LIGHTGRAY);
    setlinestyle(0,0,1);
    circle(200,200,128);
}
}

```

```
void position1(int x,int color)
```

```
/** draw the vertical line to show current degree (linear graph) ***/
```

```

{
    setcolor(color);
    setlinestyle(0,0,2);
    moveto(x+30,329);
    lineto(x+30,70);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void scan1(int x,int input,int px1,int py1)

/**/ draw the line from previous point to current point (linear graph )
***/

{
    px[x]=30+x;
    py[x]=330-input;
    setcolor(RED);
    setlinestyle(0,0,1);
    moveto(px[x],py[x]);
    lineto(px1,py1);
}

```

```

void linear(int fn)

```

```

/*****/
/* display the linear graph, */
/* receive the signal level of each direction */
/* and display it at the monitor as linear graph */
/*****/

{
    if(fn==1) {
        int l;
        int n=1;
        gettime(&t);
        th_str = t.ti_hour;
        tm_str = t.ti_min;
        ts_str = t.ti_sec;
        day_str = da.da_day;
        mon_str = da.da_mon;
        yea_str = da.da_year;
        in_rfddata(n);
        px[1]=31;
        py[1]=330-rfddata;
        setviewport(9,9,630,377,1);
        position1(1,GREEN);
        putpixel(31,py[n],RED);
        show_deg_level(fn,n);
        linear_line(fn);
        send_port(rfddata);
        do {
            n=n+1;
            in_rfddata(n);
            if(index1==DATA_END) {
                button=5;
                n=n-1;
                goto outp1;
            }
            position1(n-1,WHITE);
            position1(n,GREEN);

```

```

scan1(n, rfddata, px[n-1], py[n-1]);
show_deg_level(fn, n);
linear_line(fn);
delay(10);
send_port(rfddata);
outp1: setcolor(RED);
if(n>3) {
    for(l=n-2; l<=n; l++) {
        moveto(px[l], py[l]);
        lineto(px[l-1], py[l-1]);
    }
}
while(button!=5);
button=0;
position1(n, GREEN);
gettime(&t);
th_end = t.ti_hour;
tm_end = t.ti_min;
ts_end = t.t
datamax();
do {
    index1=read_port();
    if(index1==DATA_END) {
        button=5;
        goto outp2;
    }
    n=n-1;
    position1(n+1, WHITE);
    position1(n, GREEN);
    show_deg_level(fn, n);
    linear_line(fn);
    setcolor(RED);
    delay(28);
    if(n<355) {
        for(l=n; l<=n+5; l++) {
            moveto(px[l], py[l]);
            lineto(px[l+1], py[l+1]);
        }
    }
    setviewport(0, 0, 639, 479, 0);
outp2: send_port(index1);
    setviewport(9, 9, 630, 377, 1);
}
while(button!=5);
setviewport(0, 0, 639, 479, 0);
}
if(fn==2) {
    int l;
    int n=1;
    setviewport(9, 9, 630, 377, 1);
    do {
        index1=read_port();

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(index1==DATA_END) {
            button=5;
            n=n-1;
            goto outp3;
        }
        position1(n-1,WHITE);
        position1(n,GREEN);
        show_deg_level(fn,n);
        linear_line(fn);
        n=n+1;
        setviewport(0,0,639,479,0);
outp3:    send_port(index1);
        setviewport(9,9,630,377,1);
    }
    while(button!=5);
    position1(n,GREEN);
    setviewport(0,0,639,479,0);
}
}

```

```

void linear_line(int fn)

```

```

/*****
/*  draw the axis and grid of linear graph  */
/*  fn = 1 or 3 : horizontal line          */
/*  fn = 2      : vertical line            */
*****/

```

```

{
    setcolor(BLUE);
    setlinestyle(0,0,2);
    moveto(30,60);
    lineto(30,330);
    lineto(395,330);
    if(fn==1 || fn==3) {
        setcolor(LIGHTGRAY);
        setlinestyle(0,0,1);
        moveto(30,74);
        lineto(393,74);
        moveto(30,138);
        lineto(393,138);
        moveto(30,202);
        lineto(393,202);
        moveto(30,266);
        lineto(393,266);
    }
    if(fn==2) {
        setcolor(LIGHTGRAY);
        setlinestyle(0,0,1);
        moveto(120,329);
        lineto(120,70);
        moveto(210,329);
        lineto(210,70);
    }
}

```

```

        moveto(300,329);
        lineto(300,70);
        moveto(390,329);
        lineto(390,70);
        moveto(30,202);
        lineto(393,202);
    }
}

```

```
void deg_level(int fn)
```

```
/** show the initial value of degree and signal level before start **/
```

```

{
    if(fn==1) {
        setusercharsize(2,5,2,5);
        settextstyle(0,0,USER_CHAR_SIZE);
        setcolor(GREEN);
        outtextxy(20,15,"NOW DEGREE = 0");
        outtextxy(20,30," RF SIGNAL = 0");
    }
    if(fn==2) {
        setusercharsize(2,5,2,5);
        settextstyle(0,0,USER_CHAR_SIZE);
        setcolor(GREEN);
        outtextxy(120,20,"NOW DEGREE = 0");
    }
}

```

```
void show_deg_level(int fn,int n)          /***** CHANGE VARIABLE rf_new
*****/
```

```

/*****
/* show value of degree and signal level */
/* fn=1 : show degree and level between running motor */
/* fn=2 : show the degree only (use in function Direction set) */
/* fn=3 : show degree and level at maximum point */
*****/

```

```

{
    if(fn==1) {
        char s[10];
        setfillstyle(SOLID_FILL,WHITE);
        bar(135,8,170,45);
        setusercharsize(2,5,2,5);
        settextstyle(0,0,USER_CHAR_SIZE);
        setcolor(GREEN);
        sprintf(s,"%3d",n);
        outtextxy(142,15,s);
        sprintf(s,"%3d",rf_new[n]);
        outtextxy(142,30,s);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char s[10];
setfillstyle(SOLID_FILL,WHITE);
bar(233,10,280,30);
setusercharsize(2,5,2,5);
settextstyle(0,0,USER_CHAR_SIZE);
setcolor(GREEN);
sprintf(s,"%3d",n);
outtextxy(238,20,s);
}
if(fn==3) {
char s[30];
setfillstyle(SOLID_FILL,WHITE);
bar(15,12,380,43);
setusercharsize(2,5,2,5);
settextstyle(0,0,USER_CHAR_SIZE);
setcolor(LIGHTBLUE);
outtextxy(210,15,"(MAX) DEGREE =");
outtextxy(210,30,"RF SIGNAL =");
outtextxy(20,15,"START TIME");
outtextxy(20,30,"END TIME");
setcolor(GREEN);
sprintf(s,"%3d",n);
outtextxy(335,15,s);
sprintf(s,"%3d",rfmax_view);
outtextxy(335,30,s);
sprintf(s,"%02d:%02d:%02d",vth_str,vtm_str,vts_str);
outtextxy(112,15,s);
sprintf(s,"%02d:%02d:%02d",vth_end,vtm_end,vts_end);
outtextxy(112,30,s);
}
}

```

```
void message(int x1,int y1,int x2,int y2,char ch1[50],char ch2[15])
```

```
/** show the any message **/
```

```

{
runtime();
save_view(x1,y1,x2,y2);
setlinestyle(0,0,1);
setcolor(RED);
rectangle(x1,y1,x2,y2);
setfillstyle(SOLID_FILL,RED);
floodfill(x1+20,y1+20,RED);
setcolor(YELLOW);
rectangle(x1+15,y1+10,x2-15,y2-10);
setcolor(YELLOW);
settextjustify(CENTER_TEXT, TOP_TEXT);
setusercharsize(3,5,2,5);
settextstyle(1,0,USER_CHAR_SIZE);
outtextxy(((x2-x1)/2)+x1,y1+22,ch1);
setcolor(YELLOW);
outtextxy(((x2-x1)/2)+x1,y1+46,ch2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประกอบการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(YELLOW);
settextstyle(0,0,1);
outtextxy(((x2-x1)/2)+x1,y2-26,"press any key to continue");
settextjustify(LEFT_TEXT, TOP_TEXT);
getch();
runtime();
free(view_data);
}

```

```
void function_text()
```

```
/**/ ending window /**/
```

```

{
clrscr();
textcolor(GREEN+BLINK);
gotoxy(13,12);
cprintf("=====");
gotoxy(13,13);
cprintf("|");
gotoxy(13,14);
cprintf("|");
gotoxy(13,15);
cprintf("=====");
textcolor(GREEN+BLINK);
gotoxy(17,13);
cprintf("PROGRAM DIRECTION FINDER");
gotoxy(17,14);
cprintf("THANK YOU");
gotoxy(1,8);
delay(300);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM CONTROL MOTOR

```

    ORG    0000H
    LJMP   MAIN

    ORG    0030H
MAIN:    CLR     IE.7
    MOV    SCON,#50H
    MOV    TCON,#40H
    MOV    TMOD,#20H
    MOV    TH1,#0FAH
    MOV    PCON,#80H
    MOV    DPTR,#0FC03H
    MOV    A,#83H
    MOV    R3,#00H

CONTRWD: MOVX   @DPTR,A
    DJNZ   R3,CONTRWD
    DEC    DPL
    MOVX   A,@DPTR
    SETB   0E4H
    MOVX   @DPTR,A
    MOV    R4,#00H
    MOV    R7,#04H
    MOV    R6,#00H
    LCALL  SAVESTEP

    MOV    DPTR,#0FC00H
CCC:     MOV    A,#03H
    MOVX   @DPTR,A

    LCALL  TRANSMIT
    CLR    RI
    LCALL  RECEIVE
    CJNE   A,#03H,CCC

MENU:    LCALL  RECCODE

    LCALL  CHKFUNCT
    CJNE   R4,#0AH,MENU
    AJMP   MAIN

```

```

;*****
;*****          < TRANSMIT >          *****
;*****  TRANSMIION DATA TO COMPUTER  *****
;*****

```

```

TRANSMIT:  MOV    SBUF,A
           JNB    TI,$
           CLR    TI
           RET

```

```

;*****
;*****          < RECEIVE >          *****
;*****  RECEIVE DATA FROM COMPUTER  *****
;*****

```

```

RECEIVE:   JNB    RI,$

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    A,SBUF
CLR    RI
RET

```

```

;*****
;*****      < CHKHWRDY >      *****
;*****      RECEIVE AND TRANSMIT " READY " CODE      *****
;*****

```

```

CHKHWRDY:    CLR    RI
CHKRDY:      LCALL  RECEIVE
              CJNE  A,#08DH,NOREADY
              LCALL  TRANSMIT
              RET
NOREADY:     LCALL  MERROR
              AJMP  CHKRDY

```

```

;*****
;*****      < MERROR >      *****
;*****      MESSAGE ERROR      *****
;*****

```

```

MERROR:     MOV    A,#0F5H
              LCALL  TRANSMIT
              RET

```

```

;*****
;*****      < RECCODE >      *****
;*****      RECEIVE " FUNCTION AND DATA " CODE      *****
;*****

```

```

RECCODE:     PUSH  01H
              MOV   R1,#02H
ISFUNCT:     LCALL  RECEIVE
              JNB   0E7H,NISFUNCT
              CLR   0E7H
              MOV   B,A
CHKQUIT:     CJNE  A,#0AH,CHKF1
              SETB  0E7H
              AJMP  RETMAIN
CHKF1:       CJNE  A,#01H,CHKF2
              AJMP  ISDATA
CHKF2:       CJNE  A,#02H,CHKF3
              AJMP  ISDATA
CHKF3:       CJNE  A,#03H,NISFUNCT
ISDATA:      MOV   DPTR,#2010H
              SETB  0E7H
RECDATAC:    LCALL  TRANSMIT
              LCALL  RECEIVE
              MOVX  @DPTR,A
              INC   DPL
              DJNZ  R1,RECDATAC
              LCALL  SETZERO
RETMAIN:     LCALL  TRANSMIT
              POP   01H
              RET
NISFUNCT:    LCALL  TRANSMIT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AJMP ISFUNCT

```

;*****
;*****      < CHKFUNCT >      *****
;*****      CHECK FUNCTION CODE      *****
;*****

```

```

CHKFUNCT:      MOV      A,B
CHECKF0:      CJNE      A,#0AH,CHECKF1
               LCALL      FQUIT
               RET
CHECKF1:      CJNE      A,#01H,CHECKF2
               LCALL      FUNCT1
               RET
CHECKF2:      CJNE      A,#02H,CHECKF3
               LCALL      FUNCT2
               RET
CHECKF3:      CJNE      A,#03H,NOTFUNCT
               LCALL      FUNCT3
               RET
NOTFUNCT:     MOV      R4,#0AH
CHKFTRT:      RET

```

```

;*****
;*****      < FQUIT >      *****
;*****      FUNCTION 10 " QUIT FROM PROGRAM "      *****
;*****

```

```

FQUIT:      MOV      DPTR,#0FC00H
             MOV      A,#0FFH
             MOVX     @DPTR,A
             LCALL      DELAY
             LCALL      DELAY
             MOV      R4,#0AH
             RET

```

```

;*****
;*****      < FUNCT1 >      *****
;*****      FUNCTION 1 SCAN ANTENNA 360 AND CHECK RF SIGNAL      *****
;*****

```

```

FUNCT1:      LCALL      RECEIVE
CANCEL1:     CJNE      A,#07H,CODERUN1
             LCALL      TRANSMIT
             RET
CODERUN1:    CJNE      A,#08H,NRUNFT1
RUNFT1:      LCALL      TRANSMIT
             PUSH      01H
             PUSH      02H
             PUSH      03H
             LCALL      RESTDATA
             MOV      R0,#00H
             MOV      R5,#00H
             LCALL      SAVEDATA
             LCALL      READSTEP

```

```

        LCALL  READDATA

        CJNE  R1,#00H,LOADR0R1
LOADR0R2: LCALL  R2TOR0
        AJMP  RUNMOTOR
LOADR0R1: LCALL  READNUMBER
RUNMOTOR: LCALL  READVALUE

        LCALL  INC6DEC7
        LCALL  DELAY1
        LCALL  READVALUE
        LCALL  INC6DEC7
        LCALL  DELAY2

        LCALL  ADC
        CJNE  R4,#07,FT1CONT
        AJMP  FT1RETURN

FT1CONT: DJNZ  R0,RUNMOTOR
        DJNZ  R1,LOADR0R1
        CJNE  R2,#00H,LOADR0R2
        MOV  A,#0AAH
        LCALL  TRANSMIT
        CLR  RI
        LCALL  RECEIVE
FT1RETURN: LCALL  SAVESTEP

        POP  03H
        POP  02H
        POP  01H
        RET

NRUNFT1: LCALL  TRANSMIT
        AJMP  FUNCT1

;*****
;***** < FUNCT2 > *****
;***** FUNCTION 2 " DIRECTION SET " *****
;*****

        FUNCT2: LCALL  RECEIVE
CANCEL2: CJNE  A,#07H,CODERUN2
        LCALL  TRANSMIT
        RET

CODERUN2: CJNE  A,#08H,NRUNFT2
RUNFT2: LCALL  TRANSMIT

        PUSH  01H
        PUSH  02H
        PUSH  03H

        LCALL  RESTDATA
        MOV  R0,#00H
        MOV  R5,#00H
        LCALL  SAVEDATA

        LCALL  READSTEP

        LCALL  READDATA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                CJNE   R1, #00H, LOAD2R0R1
LOAD2R0R2:                    LCALL  R2TOR0
                                AJMP  RUNMOTOR2

LOAD2R0R1:                    LCALL  READNUMBER

RUNMOTOR2:                    LCALL  READVALUE
                                LCALL  INC6DEC7
                                LCALL  DELAY3
                                LCALL  READVALUE
                                LCALL  INC6DEC7
                                LCALL  DELAY3

                                LCALL  SENDDEG

FT2CONT:                      DJNZ   R0, RUNMOTOR2
                                DJNZ   R1, LOAD2R0R1
                                CJNE   R2, #00H, LOAD2R0R2
                                MOV    A, #0AAH
                                LCALL  TRANSMIT
                                CLR    RI
                                LCALL  RECEIVE
FT2RETURN:                    LCALL  SAVESTEP

                                POP    03H
                                POP    02H
                                POP    01H
                                RET

NRUNFT2:                      LCALL  TRANSMIT
                                AJMP  FUNCT2

;*****
;*****      < FUNCT3 >      *****
;*****      FUNCTION 3  " DIRECTION FINDING "      *****
;*****

FUNCT3:                      LCALL  RECEIVE
CANCEL3:                    CJNE   A, #07H, CODERUN3
                                LCALL  TRANSMIT
                                RET

CODERUN3:                    CJNE   A, #08H, NRUNFT3
RUNFT3:                      LCALL  TRANSMIT

                                PUSH   01H
                                PUSH   02H
                                PUSH   03H

                                LCALL  RESTDATA
                                MOV    R0, #00H
                                MOV    R5, #00H
                                LCALL  SAVEDATA

                                LCALL  READSTEP
                                LCALL  CONVERT

                                LCALL  READDATA

```

```

                                CJNE   R1, #00H, LOAD3R0R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOAD3R0R2:      LCALL  R2TOR0
                AJMP   RUNMOTOR3

LOAD3R0R1:      LCALL  READNUMBER

RUNMOTOR3:      LCALL  SETMOTOR
                LCALL  INC6DEC7
                LCALL  DELAY3
                LCALL  SETMOTOR
                LCALL  INC6DEC7
                LCALL  DELAY3

                LCALL  SENDDEG

FT3CONT:        DJNZ   R0, RUNMOTOR3
                DJNZ   R1, LOAD3R0R1
                CJNE   R2, #00H, LOAD3R0R2
                MOV    A, #0AAH
                LCALL  TRANSMIT
                CLR    RI
                LCALL  RECEIVE
FT3RETURN:      LCALL  CONVERT
                LCALL  SAVESTEP

                POP    03H
                POP    02H
                POP    01H
                RET

NRUNFT3:        LCALL  TRANSMIT
                AJMP   FUNCT3

;*****
;*****          < RESTDATA >          *****
;*****          RESTORE DATA CODE FROM RAM TO R1 AND R2          *****
;*****

RESTDATA:       MOV    DPTR, #2010H
                MOVX   A, @DPTR
                MOV    R1, A
                INC    DPL
                MOVX   A, @DPTR
                MOV    R2, A
                RET

;*****
;*****          < DATABRK >          *****
;*****          DATA BREAKING          *****
;*****

DATABRK:        LCALL  RECEIVE
CONTINUE:       CJNE   A, #06H, BRCANCEL
                MOV    R4, #06H
                LCALL  TRANSMIT
                RET
BRCANCEL:       CJNE   A, #07H, NDATABRK
                MOV    R4, #07H
                LCALL  TRANSMIT

```

```

RET
NDATABRK:      LCALL TRANSMIT
              AJMP  DATABRK

```

```

;*****
;***** < DELAY > *****
;***** DELAY 256*256 *****
;*****

```

```

DELAY:        PUSH  01H
              PUSH  02H
              MOV   R1,#00H
DD:           MOV   R2,#00H
              DJNZ  R2,$
              DJNZ  R1,DD
              POP   02H
              POP   01H
              RET

```

```

;*****
;***** < DELAY1 > *****
;***** DELAY FOR EACH STEP *****
;*****

```

```

DELAY1:       PUSH  01H
              PUSH  02H
              MOV   R1,#20H
DD1:          MOV   R2,#20H
              DJNZ  R2,$
              DJNZ  R1,DD1
              POP   02H
              POP   01H
              RET

```

```

;*****
;***** < DELAY2 > *****
;***** DELAY FOR EACH STEP *****
;*****

```

```

DELAY2:       PUSH  01H
              PUSH  02H
              MOV   R1,#01H
DD2:          MOV   R2,#01H
              DJNZ  R2,$
              DJNZ  R1,DD2
              POP   02H
              POP   01H
              RET

```

```

;*****
;***** < DELAY3 > *****
;***** DELAY FOR EACH STEP *****
;*****

```

```

DELAY3:       PUSH  01H
              PUSH  02H
              MOV   R1,#20H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DD3:  MOV    R2,#20H
      DJNZ   R2,$
      DJNZ   R1,DD3
      POP    02H
      POP    01H
      RET

```

```

;*****
;*****          < SETZERO >          *****
;*****          SET POSITION DEGREE OF STEPPING MOTER TO ZERO          *****
;*****

```

```

SETZERO:  LCALL  READSTEP
          LCALL  CONVERT
          MOV    DPTR,#0FC02H
          MOVX   A,@DPTR
          SETB  0E7H
          MOVX   @DPTR,A
ZEROCONT: MOVX   A,@DPTR
          JBC   0E3H,ZEROOK
          LCALL  SETMOTOR
          LCALL  INC6DEC7
          AJMP  ZEROCONT
ZEROOK:   LCALL  CONVERT
          LCALL  SAVESTEP
          MOV    DPL,02H
          MOVX   A,@DPTR
          CLR    0E7H
          MOVX   @DPTR,A
          RET

```

```

;*****
;*****          < SENDDEG >          *****
;*****          SEND POSITION DEGREE OF STEPPING MOTOR          *****
;*****

```

```

SENDDEG:  MOV    A,#01H
          LCALL  TRANSMIT
          CLR    RI
          LCALL  RECEIVE
          RET

```

```

;*****
;*****          < ADC >          *****
;*****          ANALOG TO DIGITAL CONVERTOR          *****
;*****

```

```

ADC:      MOV    DPTR,#0FC02H
          LCALL  SENDDEG
RUNADC:   MOVX   A,@DPTR
          CLR    0E4H
          MOVX   @DPTR,A
INTADC:   MOVX   A,@DPTR
          JB    0E0H,RUNADC
          DEC    DPL
          MOVX   A,@DPTR
          MOV    R3,A
          LCALL  TRANSMIT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADCREC:      LCALL RECEIVE
             CJNE  A, #0F9H, ADCDATA
             LCALL TRANSMIT
             LCALL DATABRK
             CJNE  R4, #07H, ADCREC
             RET

```

```

ADCDATA:    CJNE  A, 03H, ADCERROR
ADCCONT:    MOV   DPL, #02H
             MOVX  A, @DPTR
             SETB  0E4H
             MOVX  @DPTR, A
             MOV   R4, #06H
             RET

```

```

ADCERROR:   LCALL MERROR
             MOV   R4, #07H
             RET

```

```

;*****
;*****      < SAVEDATA >      *****
;*****      SAVE DATA R0,R1,R2,R5 TO RAM      *****
;*****

```

```

SAVEDATA:   MOV   DPTR, #2020H
             MOV   A, R0
             MOVX  @DPTR, A
             INC   DPL
             MOV   A, R1
             MOVX  @DPTR, A
             INC   DPL
             MOV   A, R2
             MOVX  @DPTR, A
             INC   DPL
             MOV   A, R5
             MOVX  @DPTR, A
             RET

```

```

;*****
;*****      < READDATA >      *****
;*****      READ DATA R0,R1,R2,R5 TO RAM      *****
;*****

```

```

READDATA:   MOV   DPTR, #2020H
             MOVX  A, @DPTR
             MOV   R0, A
             INC   DPL
             MOVX  A, @DPTR
             MOV   R1, A
             INC   DPL
             MOVX  A, @DPTR
             MOV   R2, A
             INC   DPL
             MOVX  A, @DPTR
             MOV   R5, A
             RET

```

```

;*****
;*****      < SAVESTEP >      *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**** SAVE STEP OF MOTOR R6,R7 TO RAM ****
;*****
```

```
SAVESTEP:      MOV     DPTR,#2030H
                MOV     A,R6
                MOVX   @DPTR,A
                INC     DPL
                MOV     A,R7
                MOVX   @DPTR,A
                RET
```

```
;*****
;**** < READSTEP > ****
;**** READ STEP OF MOTOR R6,R7 FROM RAM ****
;*****
```

```
READSTEP:      MOV     DPTR,#2030H
                MOVX   A,@DPTR
                MOV     R6,A
                INC     DPL
                MOVX   A,@DPTR
                MOV     R7,A
                RET
```

```
;*****
;**** < READVALUE > ****
;**** READ VALUE FOR OUT TO STEPPING MOTOR (3,6,C,9) ****
;*****
```

```
READVALUE:     MOV     DPTR,#1980H
                MOV     A,R6
                MOVC   A,@A+DPTR
                MOV     DPTR,#0FC00H
                MOVX   @DPTR,A
                RET
```

```
;*****
;**** < SETMOTOR > ****
;**** OUT VALUE FOR SET STEPPING MOTOR TO ZERO AND MAX ****
;*****
```

```
SETMOTOR:      MOV     DPTR,#1970H
                MOV     A,R6
                MOVC   A,@A+DPTR
                MOV     DPTR,#0FC00H
                MOVX   @DPTR,A
                RET
```

```
;*****
;**** < INC6DEC7 > ****
;**** CHECK STEP OF MOTOR ( R6 = 0-3 , R7 < 4 ) ****
;*****
```

```
INC6DEC7:      INC     R6
                DJNZ   R7,R6CONT
                MOV     R6,#00H
                MOV     R7,#04H

R6CONT:        RET
```

```

;*****
;****          < READNUMBER >          *****
;****          READ NUMBER 84,84,84,84 TO R0 BY INDEX R5          *****
;*****

```

```

READNUMBER:      MOV      DPTR,#1990H
                  MOV      A,R5
                  MOVC     A,@A+DPTR
                  MOV      R0,A
                  INC      R5
                  RET

```

```

;*****
;****          < R2TOR0 >          *****
;****          LOAD DATA FROM R2 TO R0          *****
;*****

```

```

R2TOR0:          MOV      A,R2
                  MOV      R0,A
                  MOV      R2,#00H
                  INC      R1
                  RET

```

```

;*****
;****          < CONVERT >          *****
;****          CONVERT DATA R6,R7 FOR STEP OF MOTOR          *****
;*****

```

```

CONVERT:         LCALL   READSTEP
                  CJNE   R6,#01H,R606H
                  MOV    R6,#03H
                  MOV    R7,#01H
R606H:           CJNE   R6,#03H,CONVRET
                  MOV    R6,#01H
                  MOV    R7,#03H
CONVRET:         LCALL   SAVESTEP
                  RET

```

```

;*****
;****          DATA FOR OUT TO STEPPING MOTOR          *****
;****          RIGHT HAND 1980 H , LEFT HAND 1970          *****
;*****

```

```

ORG      1970H
DB       03H,09H,0CH,06H

```

```

ORG      1980H
DB       03H,06H,0CH,09H

```

```

;*****
;****          DATA NUMBER FOR MOV TO R0          *****
;****          ADDRESS 1990 H          *****
;*****

```

```

ORG      1990H
DB       5AH,5AH,5AH,5AH

```

END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC1723 MC1723C



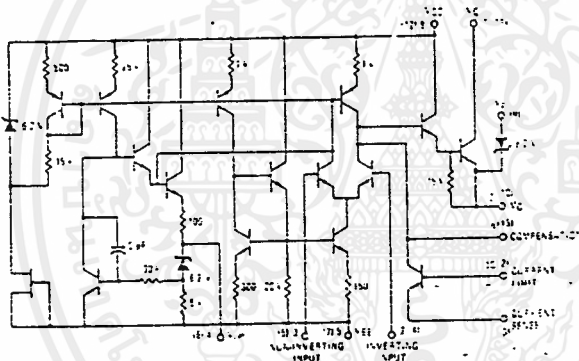
MOTOROLA

MONOLITHIC VOLTAGE REGULATOR

The MC1723 is a positive or negative voltage regulator designed to deliver load current to 150 mA dc. Output current capability can be increased to several amperes through use of one or more external pass transistors. MC1723 is specified for operation over the military temperature range (-55°C to +125°C) and the MC1723C over the commercial temperature range (0 to +70°C)

- Output Voltage Adjustable from 2 Vdc to 37 Vdc
- Output Current to 150 mA dc Without External Pass Transistors
- 0.01% Line and 0.03% Load Regulation
- Adjustable Short-Circuit Protection

FIGURE 1 - CIRCUIT SCHEMATIC



ALL NUMBERS COEFFICIENT TO TERMINALS ARE FOR THE METAL PACKAGE
PIN NUMBERS IN PARENTS ARE FOR DUAL IN LINE PACKAGES

VOLTAGE REGULATOR

SILICON MONOLITHIC
INTEGRATED CIRCUIT



(Top View)

P SUFFIX
PLASTIC PACKAGE
CASE 646-05

(Top View)

G SUFFIX
METAL PACKAGE
CASE 603-04
(TO-100)



L SUFFIX
CERAMIC PACKAGE
CASE 632-02
MO-001AA

ORDERING INFORMATION

Device	Alternate	Temperature Range	Package
MC1723CG	LM723CM, LA723CM	0°C to 70°C	Plastic
MC1723CL	LM723CS, LA723CS	0°C to 70°C	Plastic
MC1723CG	LM723CM, LA723CM	0°C to 125°C	Plastic
MC1723C		-55°C to +125°C	Plastic
MC1723L		-55°C to +125°C	Ceramic
MC1723L		-55°C to +125°C	Ceramic

FIGURE 2 - TYPICAL CIRCUIT CONNECTION

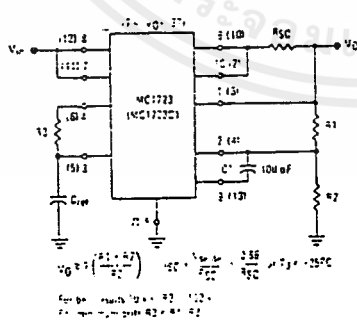
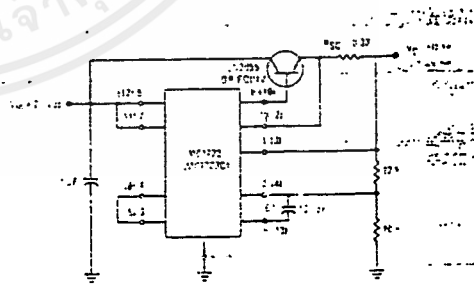


FIGURE 3 - TYPICAL NPN CURRENT BOOST CONNECTION



MOTOROLA LINEAR INTERFACE DEVICES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

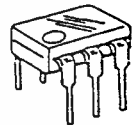
6-Pin DIP Optoisolators
Transistor Output

These devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Convenient Plastic Dual-In-Line Package
- High Current Transfer Ratio — 100% Minimum at Spec Conditions
- Guaranteed Switching Speeds
- High Input-Output Isolation Guaranteed — 7500 Volts Peak
- UL Recognized. File Number E54915
- VDE approved per standard 0883/6.80 (Certificate number 41853), with additional approval to DIN IEC380/VDE0806, IEC435/VDE0805, IEC65/VDE0860, VDE0110b, covering all other standards with equal or less stringent requirements, including IEC204/VDE0113, VDE0160, VDE0832, VDE0833, etc.
- Meets or Exceeds All JEDEC Registered Specifications
- Special lead form available (add suffix "T" to part number) which satisfies VDE0883/6.80 requirement for 8 mm minimum creepage distance between input and output solder pads.
- Various lead form options available. Consult "Optoisolator Lead Form Options" data sheet for details.

4N35
4N36
4N37

6-PIN DIP
OPTOISOLATORS
TRANSISTOR
OUTPUT



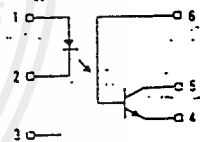
CASE 730A-02
PLASTIC

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

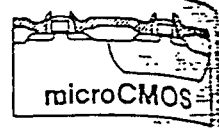
Rating	Symbol	Value	Unit
INPUT LED			
Reverse Voltage	V_R	6	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Output Detector Derate above 25°C	P_D	120	mW
		1.41	$\text{mW}/^\circ\text{C}$
OUTPUT TRANSISTOR			
Collector-Emitter Voltage	V_{CE0}	30	Volts
Emitter-Base Voltage	V_{EB0}	7	Volts
Collector-Base Voltage	V_{CB0}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Input LED Derate above 25°C	P_D	150	mW
		1.76	$\text{mW}/^\circ\text{C}$
TOTAL DEVICE			
Isolation Source Voltage (1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	250	mW
		2.94	$\text{mW}/^\circ\text{C}$
Ambient Operating Temperature Range	T_A	-55 to +100	$^\circ\text{C}$
Storage Temperature Range	T_{stg}	-55 to +150	$^\circ\text{C}$
Soldering Temperature (10 seconds, 1/16" from case)	—	260	$^\circ\text{C}$

(1) Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

SCHEMATIC



1. LED ANODE
2. LED CATHODE
3. N.C.
4. EMITTER
5. COLLECTOR
6. BASE



MM54HC573/MM74HC573 TRI-STATE® Octal D-Type Latch

General Description

These high speed octal D-type latches utilize microCMOS Technology, 3.5 micron silicon gate P-well CMOS. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

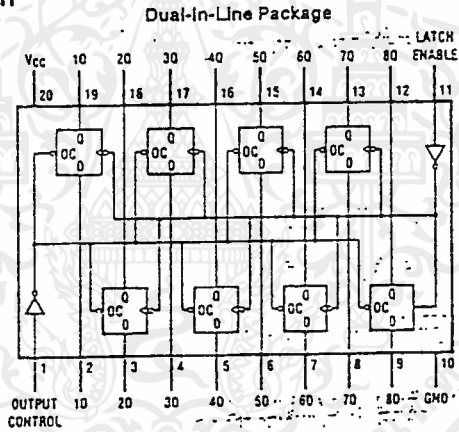
When the LATCH ENABLE (LE) input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL OC input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

The 54HC/74HC logic family is speed, function and pin compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 13 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μA maximum
- Low quiescent current: 80 μA maximum (74HC series)
- Compatible with bus-oriented systems
- Output drive capability: 15 LS-TTL loads

Connection Diagram

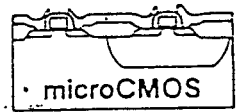


TOP VIEW TU/F/5212-1
MM54HC573/MM74HC573
54HC573 (J) 74HC573 (J,N)

Truth Table

Output Control	Latch Enable	Data	Output
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

H = high level, L = low level
Q₀ = level of output before steady-state input conditions were established.
Z = high impedance
X = Don't care



ADC0820 8-Bit High Speed μ P Compatible A/D Converter with Track/Hold Function

General Description

Using a half-flash conversion technique, the 8-bit ADC0820 CMOS A/D offers a 1.5 μ s conversion time and dissipates only 75 mW of power. The half-flash technique consists of 32 comparators, a most significant 4-bit ADC and a least significant 4-bit ADC.

The input to the ADC0820 is tracked and held by the input tracking circuitry eliminating the need for an external sample-and-hold for signals moving at less than 100 mV/ μ s.

For ease of interface to microprocessors, the ADC0820 has been designed to appear as a memory location or I/O port and the need for external interfacing logic.

Key Specifications

- Resolution: 8 Bits
- Conversion Time: 2.5 μ s Max (RD Mode)
1.5 μ s Max (WR-RD Mode)
- Input signals with slew rate of 100 mV/ μ s converted without external sample-and-hold to 8 bits
- Low Power: 75 mW Max
- Total Unadjusted Error: $\pm 1/2$ LSB and ± 1 LSB

Features

- Built-in track-and-hold function
- No missing codes
- No external clocking
- Single supply—5 V_{DC}
- Easy interface to all microprocessors, or operates stand-alone
- Latched TRI-STATE^s output
- Logic inputs and outputs meet both MOS and T²L voltage level specifications
- Operates ratiometrically or with any reference value equal to or less than V_{CC}
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Overflow output available for cascading
- 0.3" standard width 20-pin DIP

Connection Diagram

Functional Diagram

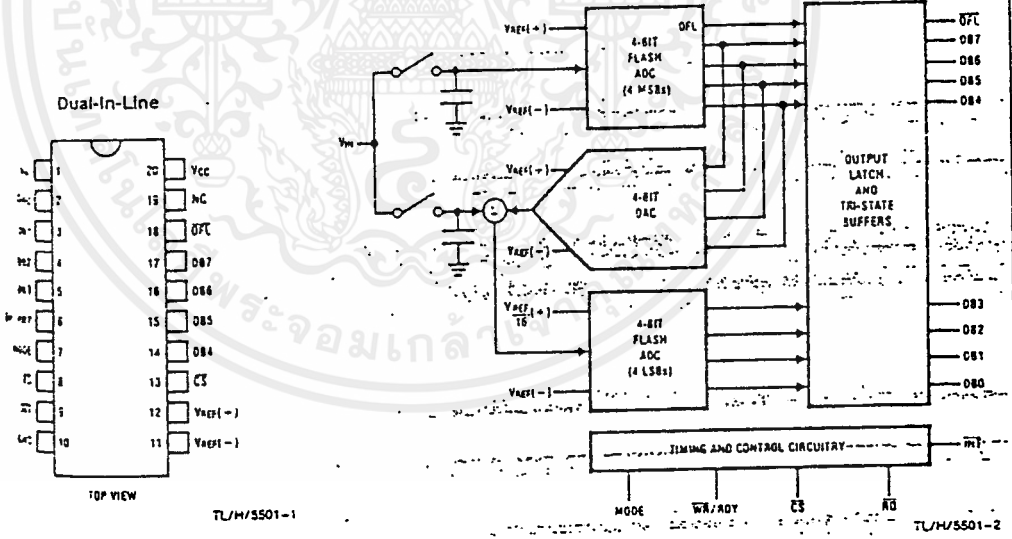
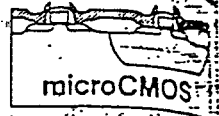


FIGURE 1



MM54HC573/MM74HC573 TRI-STATE® Octal D-Type Latch

General Description

These high speed octal D-type latches utilize microCMOS Technology, 3.5 micron silicon gate P-well CMOS. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

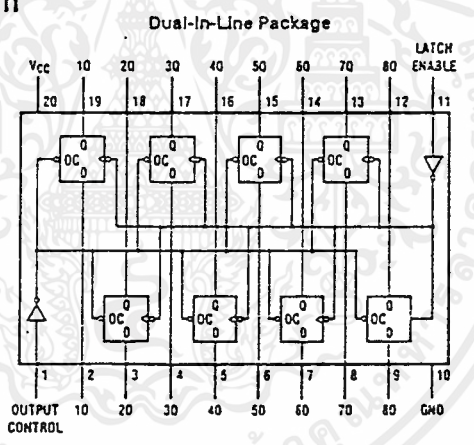
When the LATCH ENABLE(LE) input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL OC input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

The 54HC/74HC logic family is speed, function and price compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 13 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μA maximum
- Low quiescent current: 80 μA maximum (74HC series)
- Compatible with bus-oriented systems
- Output drive capability: 15 LS-TTL loads

Connection Diagram



TOP VIEW
MM54HC573/MM74HC573
54HC573 (J) 74HC573 (J,N)

Truth Table

Output Control	Latch Enable	Data	Output
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

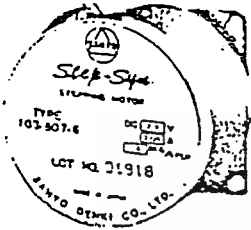
H = high level, L = low level
Q₀ = level of output before steady-state input conditions were established.
Z = high impedance
X = Don't care

2 Degree Step

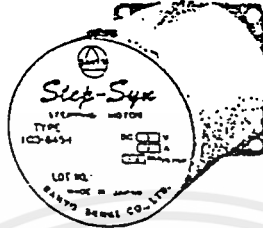
2 Degree Step
2.5 Degree Step

type 80

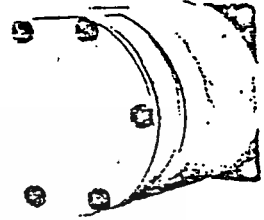
TWO
Phase



103-807-6



103-845-1



103-807-14S0 (Drip proof type)

Model number	Shaft	Step Angle Degree/Step	Voltage V	Current A/Phase	Resistance Ω/Phase	Inductance mH/Phase	Holding torque kg·cm	Rotor inertia kg·cm ²	Weight g	Lead code	Availability
3-807-4 (41)	Single shaft (Double shaft)	1.8	12	0.8	14.7	45	12	560	1400	I	○
3-807-5 (51)	Single shaft (Double shaft)	1.8	5	1.9	265	82	12	560	1400	I	○
3-807-6 (61)	Single shaft (Double shaft)	1.8	2.9	3.05	0.95	3.8	10.8	560	1400	II	○
3-807-7 (71)	Single shaft (Double shaft)	1.8	1.3	4.8	0.39	1.5	10.8	560	1400	II	△
3-807-14S0	Single shaft	1.8	3.5	2.7	1.3	8.2	15	560	1400	IV	○
3-380-014i	Double shaft	2	4.8	1.4	3.4	8	8	500	1400	III	x
3-810-1 (11)	Single shaft (Double shaft)	1.8	3	4	0.75	4.5	17	900	2500	I	○
3-815-2 (21)	Single shaft (Double shaft)	1.8	2.5	4.6	0.55	2.5	21.6	1125	2500	II	○
3-815-08S0	Single shaft	1.8	3.5	3.5	1	10	21.5	1125	2500	IV	○
3-865-3 (31)	Single shaft (Double shaft)	2.5	3	5	0.63	2.4	28	1550	4500	II	△
3-845-1 (11)	Single shaft (Double shaft)	1.8	5	4	1.18	6.8	35	1550	4500	I	○
3-845-2 (21)	Single shaft (Double shaft)	1.8	3	6.7	0.45	2.0	35	1550	4500	I	○
3-845-08S0	Single shaft	1.8	3.4	5.8	0.6	6.8	43	1550	4500	IV	○

J* at the end of the model number indicates drip proof type.

General specifications

Dielectric strength : 900 VAC

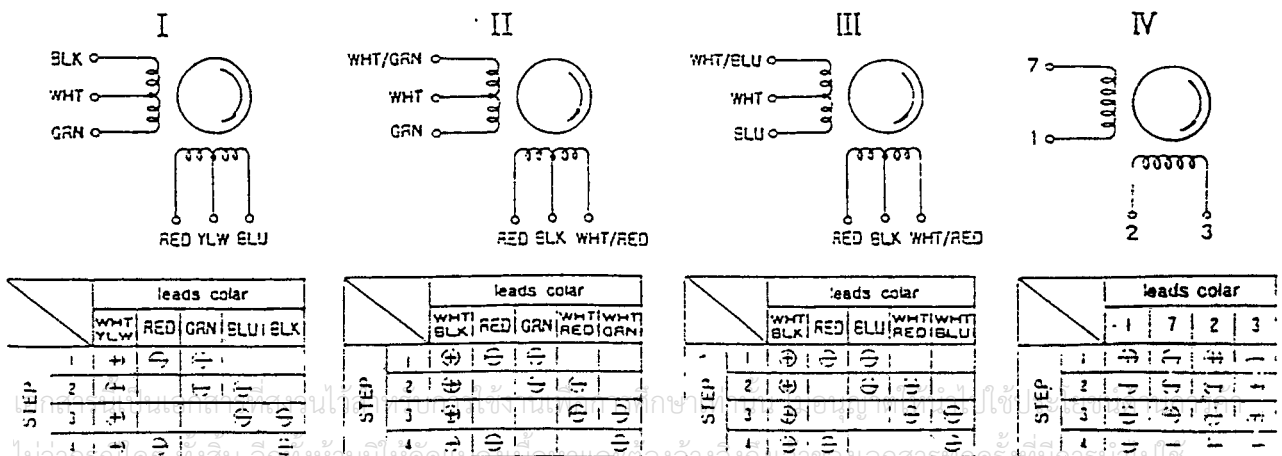
Insulation resistance : 100M (500VDC)

Insulation type : class B

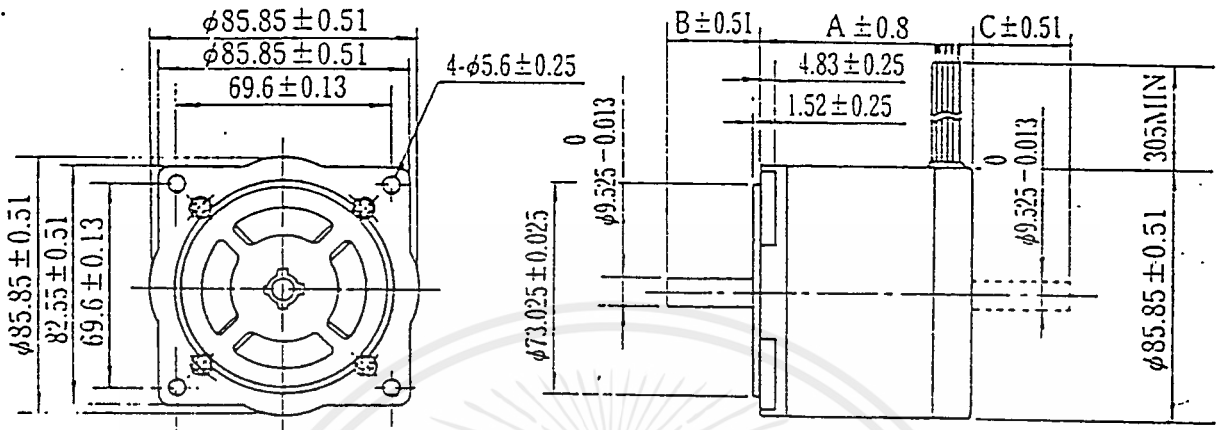
Radial play 0.025mm or below (Load 450g)

Axial play 0.075mm or below (Load 920g)

Monitor internal wiring and rotating direction (viewed from mounting side)



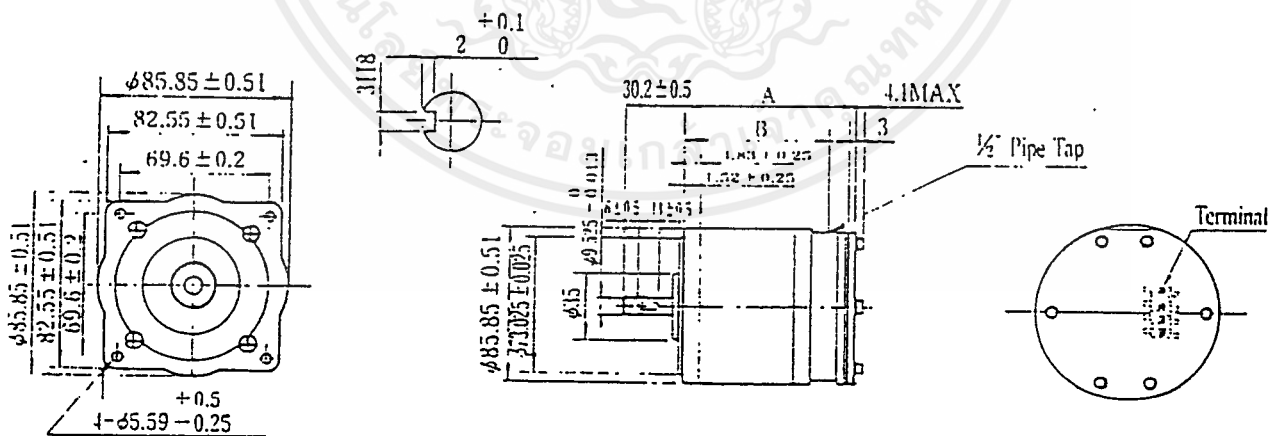
Dimensions



DIMENSION (mm)

Model number	A	B	C
103-807-	62	30.2	28.5
103-815-	93.5	30.2	30.2
103-810-	93.5	31.8	31.2
103-845-	135	25.4	25.4
103-865-			
103-880-0141	62	14.2	28.5

(Drip-proof type)



DIMENSION (mm)

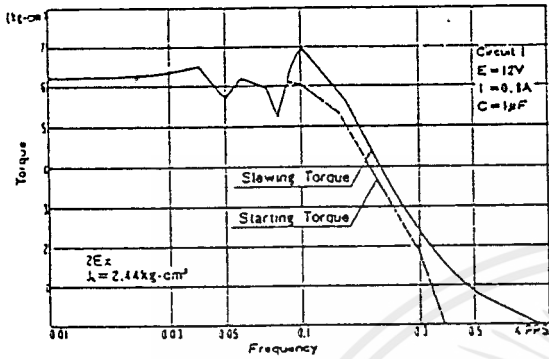
Model number	A	B
103-807-14S0	90	71.8
103-815-08S0	121.5	103.3
103-845-08S0	163	144.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

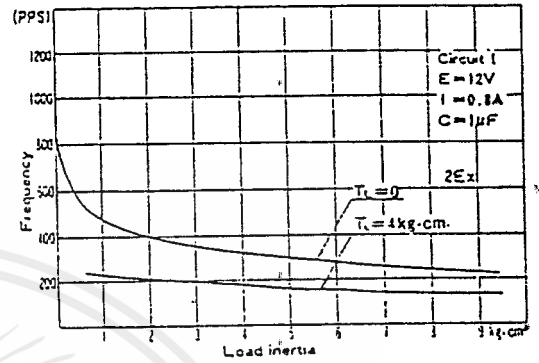
J Torque vs. Frequency

Load inertia vs. Starting rate

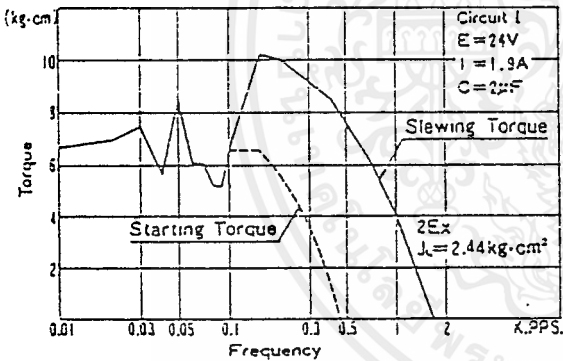
103-807-4



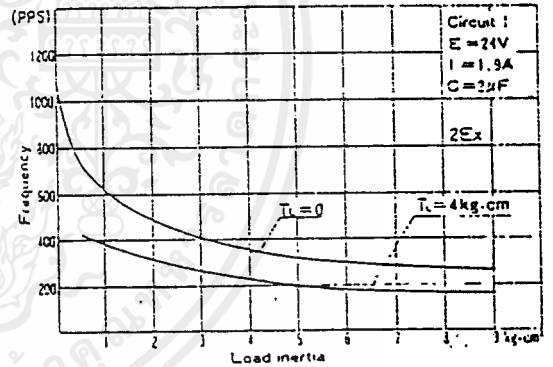
103-807-4



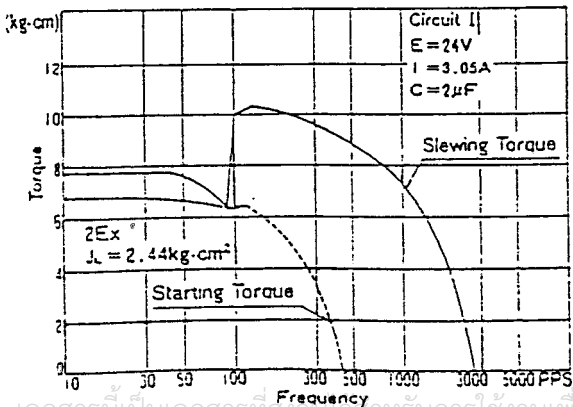
103-807-5



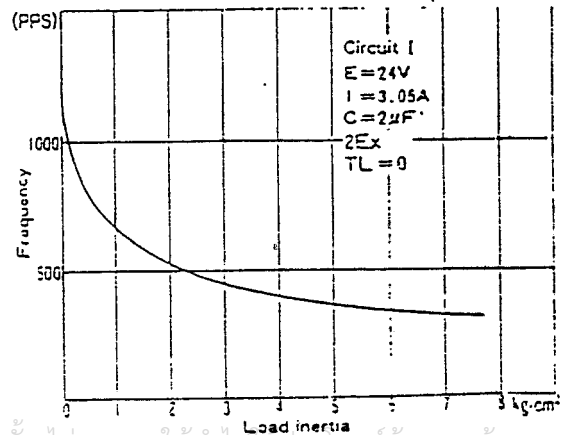
103-807-5



103-807-6



103-807-6

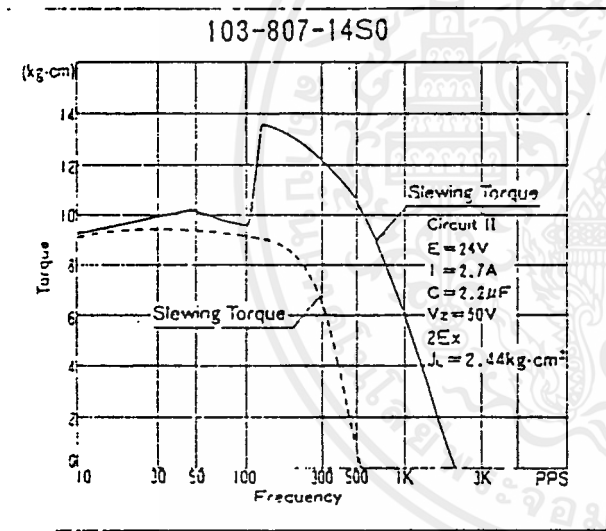
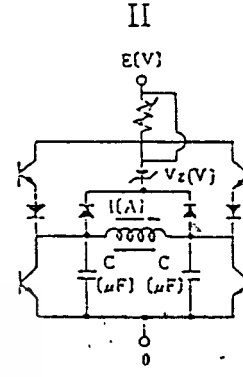
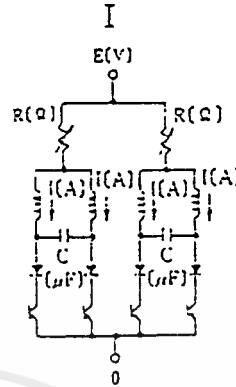
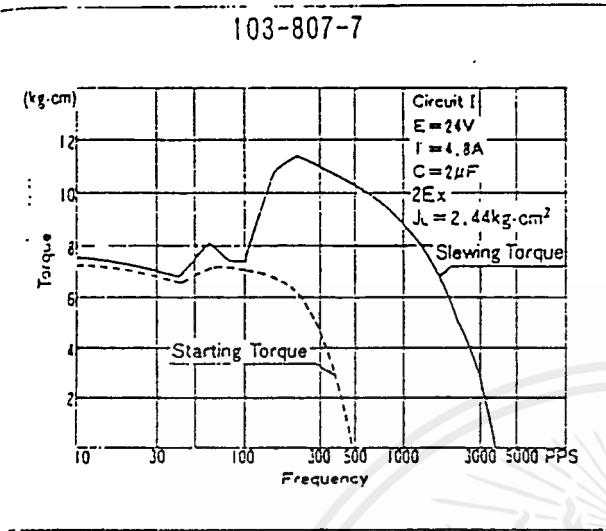


เอกสารนี้เป็นเอกสารที่สละไว้ในเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

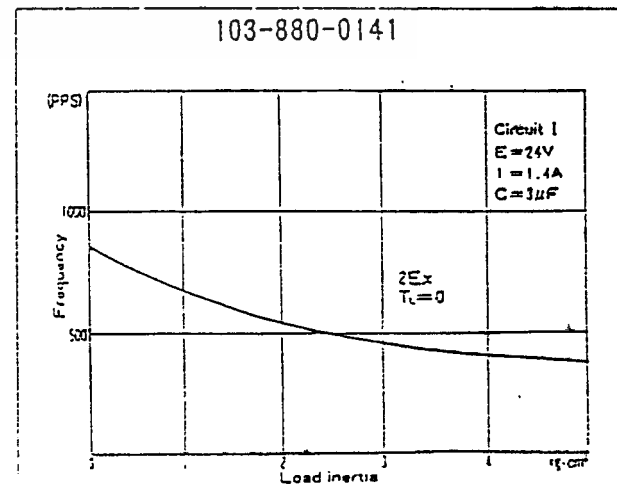
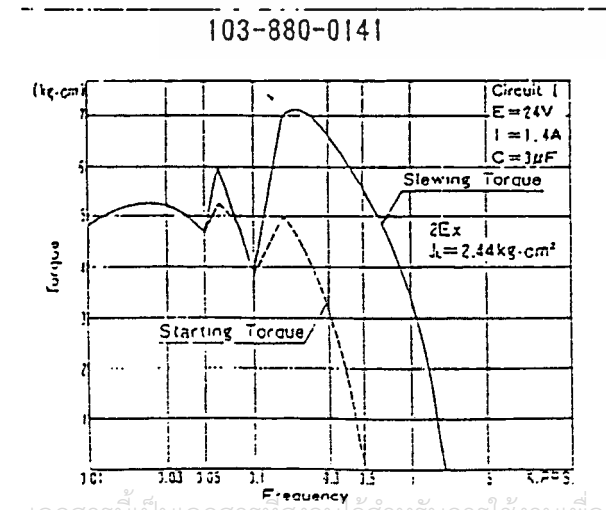
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Torque vs. Frequency

Driving Circuit



Load inertia vs. Starting rate



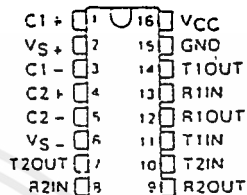
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232 DUAL EIA-232 DRIVER/RECEIVER

DJ170, FEBRUARY 1989 REVISED, JUNE 1989

- Operates with Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typ
- Meets ANSI/EIA-232-D-1986 Specifications (Revision of EIA Standard RS-232-C)
- Designed to be Interchangeable with Maxim MAX232
- Applications
 - EIA-232 Interface
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

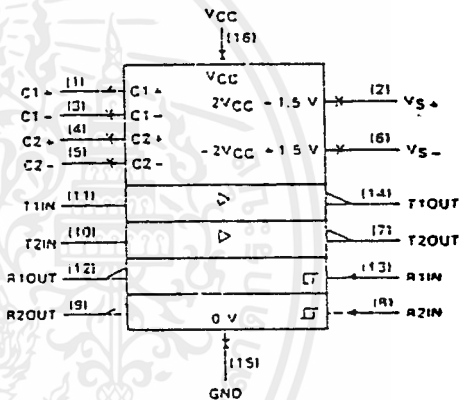
D OR N PACKAGE
(TOP VIEW)



description

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

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

Input supply voltage, V_{CC} (see Note 1)	-0.3 V to 6 V
Positive output supply voltage, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage, V_{S-}	0.3 V to -15 V
Input voltage range: Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range: T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: V_{S+}	30 s
V_{S-}	30 s
T1OUT, T2OUT	unlimited
Operating free-air temperature range	0°C to 70°C
Storage temperature range	-65°C to 150°C
Lead temperature 1.5 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to network ground terminal.

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.

PRODUCTION DATA documents contain information current as of publication date. Product compliance is specified per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1989, Texas Instruments Incorporated

TEXAS
INSTRUMENTS

POST OFFICE BOX 655302 • DALLAS, TEXAS 75265

2-43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232
DUAL EIA-232 DRIVER/RECEIVER

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	V
High-level input voltage, V_{IH} (T1IN, T2IN)	2			V
Low-level input voltage, V_{IL} (T1IN, T2IN)			0.8	V
Receiver input voltage, R1IN, R2IN			± 30	V
Operating free-air temperature, T_A	0		70	$^{\circ}\text{C}$

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP [†]	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	5	7		V
		R1OUT, R2OUT	3.5			
V_{OL}	Low-level output voltage [‡]	T1OUT, T2OUT		-7	-5	V
		R1OUT, R2OUT			0.4	
V_{T+}	Receiver positive-going input threshold voltage	R1IN, R2IN		1.7	2.4	V
V_{T-}	Receiver negative-going input threshold voltage	R1IN, R2IN	0.8	1.2		V
V_{hys}	Input hysteresis	R1IN, R2IN	0.2	0.5	1	mV
r_i	Receiver input resistance	R1IN, R2IN	3	5	7	k Ω
r_o	Output resistance	T1OUT, T2OUT			300	Ω
I_{OS}^{\dagger}	Short-circuit output current	T1OUT, T2OUT			± 10	mA
I_{IS}^{\dagger}	Short-circuit input current	T1IN, T2IN			200	μA
I_{CC}	Supply current	$V_{CC} = 5.5\text{ V}$, All outputs open, $T_A = 25^{\circ}\text{C}$		8	10	mA

[†]All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^{\circ}\text{C}$.

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

[†]Not more than one output should be shorted at a time.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^{\circ}\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low-to-high-level output		500		ns
$t_{PHL(R)}$	Receiver propagation delay time, high-to-low-level output		500		ns
SR	Driver slew rate			30	V/ μs
$SR(tr)$	Driver transition region slew rate		3		V/ μs

TYPICAL APPLICATION DATA

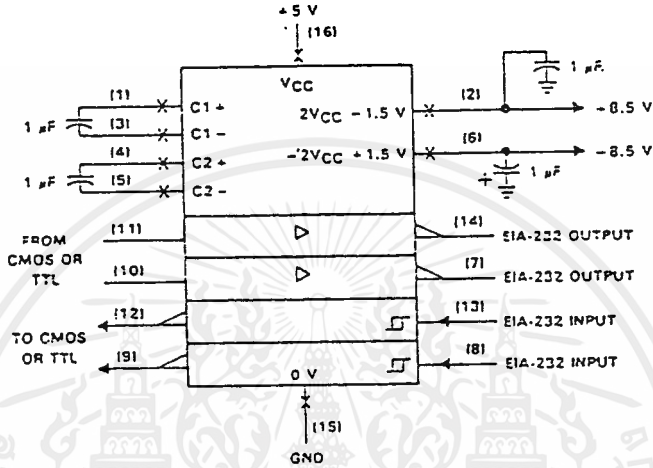
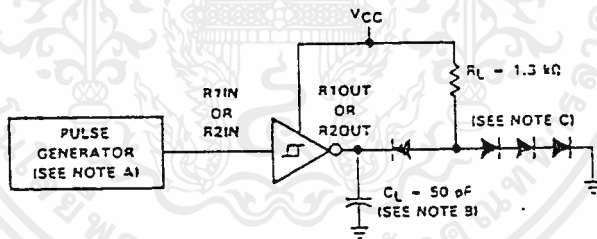
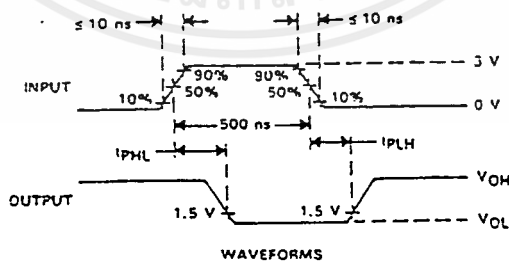


FIGURE 1. TYPICAL OPERATING CIRCUIT

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



WAVEFORMS

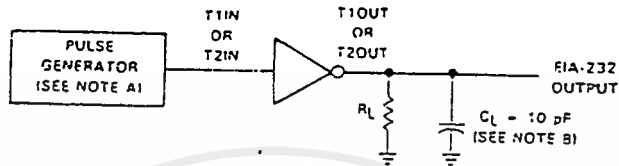
- NOTES: A. The pulse generator has the following characteristics: $Z_{out} = 50 \Omega$, Duty Cycle $\leq 50\%$.
 B. C_L includes probe and jig capacitance.
 C. All diodes are 1N3064 or equivalent.

FIGURE 2. RECEIVER TEST CIRCUIT AND WAVEFORMS FOR t_{PHL} AND t_{PLH} MEASUREMENT

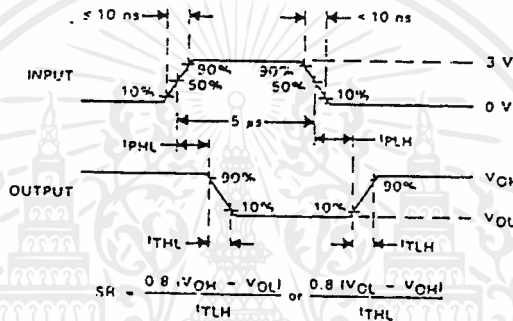
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232
DUAL EIA-232 DRIVER/RECEIVER

PARAMETER MEASUREMENT INFORMATION



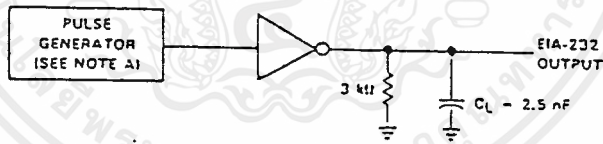
TEST CIRCUIT



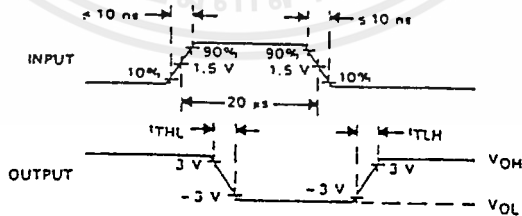
WAVEFORMS

NOTES A The pulse generator has the following characteristics: $Z_{OUT} = 50 \Omega$, Duty Cycle = 50%
B C_L includes probe and jig capacitance

FIGURE 3. DRIVER TEST CIRCUIT AND WAVEFORMS FOR t_{PHL} AND t_{PLH} MEASUREMENT (5- μ s INPUT)



TEST CIRCUIT



WAVEFORMS

NOTE A: The pulse generator has the following characteristics: $Z_{OUT} = 50 \Omega$, Duty Cycle $\leq 50\%$

FIGURE 4. TEST CIRCUIT AND WAVEFORMS FOR t_{THL} AND t_{TLH} MEASUREMENT (20- μ s INPUT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้