

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจวัดและบันทึกความแรงของสัญญาณวิทยุ

FIELD STRENGTH MEASURING AND RECORDING SYSTEM



โดย
ว่าที่ร.ต.วรัญญ์ วิชาคุณ
นายเผ่าพันธุ์ ชีรนนท์ธนพัฒน์

รฟ.
๗๒13๖
๒5๕๙

เลขหมู่.....
เลขทะเบียน..... 72694
วัน,เดือน,ปี 21 ต.ย. 2558

b. 117 146x
i.

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว

ผ่านการตรวจชิ้นงานแล้ว

เอกสารนี้เป็น (ลงชื่อ)..... ผู้ตรวจ สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ในด้านการค้า
(ลงชื่อ)..... ผู้ตรวจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาแก้ไข

ระบบตรวจวัดและบันทึกความแรงของสัญญาณวิทยุ
FIELD STRENGTH MEASURING AND RECORDING SYSTEM

โดย

ว่าที่ร.ต.วรชัญญ์ วิชาคุณ 47015737

นายเผ่าพันธุ์ ชีรนนท์ชนพัฒน์ 47015747

อาจารย์ที่ปรึกษา

รศ.นิภา สีสารุจิ

รศ.ณรงค์ เหมกรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

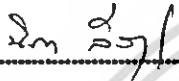
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง ระบบตรวจวัดและบันทึกความแรงของสัญญาณวิทยุ

FIELD STRENGTH MEASURING AND RECORDING SYSTEM

ผู้จัดทำ

1. ว่าที่ร.ต.วรพันธ์ วิชาคุณ 47015737
2. นายผ่่าพันธุ์ นีร์นันท์ธนพัฒน์ 47015747


.....
(รศ.นิภา ดิธารุจิ)


.....
(รศ.ณรงค์ เหมกรณ์)

อาจารย์ที่ปรึกษา

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจวัดและบันทึกความแรงของสัญญาณวิทยุ
Field Strength Measuring and Recording System

โดย ว่าที่ร.ต.วรัญญ์ วิชาคุณ 47015737

นายเผ่าพันธุ์ ธีรนนท์ธนพัฒน์ 47015747

อาจารย์ที่ปรึกษา รศ. นิภา ลีลารุจิ
รศ. ณรงค์ เหมกรณ์

บทคัดย่อ

โครงการนี้เป็นกรนำเสนอบทตรวจวัดความแรงของสัญญาณวิทยุ โดยจะสามารถวัดระดับความแรงของสัญญาณที่ใช้งานความถี่ย่าน VHF ให้แสดงผลความแรงของสัญญาณและสามารถบันทึกความเปลี่ยนแปลงของระดับสัญญาณเมื่อเวลาเปลี่ยนไปพร้อมกับประมวลผลและแสดงผลนั้นผ่านทางคอมพิวเตอร์ เพื่อคุณผลกระทบค่อสัญญาณที่เกิดจากสิ่งแวดล้อมโดยรอบบริเวณจุดทดสอบเช่น เมื่อมีเครื่องบินบินผ่าน, ตึกสูงและผลกระทบอื่นๆ

ABSTRACT

This project presented the RF field strength measuring system. It can measure the received signal level in VHF frequency. The signal level is recorded and processed on computer. This system observes the effect of environment around test point such as when the airplane flies over, the high building.

กิตติกรรมประกาศ

ขอขอบพระคุณ รองศาสตราจารย์ นิภา สีลาจฺจ เป็นอย่างยิ่งที่ได้ให้โอกาส คำแนะนำแนวทาง ในการค้นคว้าในการเขียนโครงงานฉบับนี้ รวมทั้งการตรวจทานแก้ไขโครงงานพิเศษนี้ให้ถูกต้อง สมบูรณ์และการให้คำแนะนำปรึกษาทุกๆ อย่าง ไม่ว่าจะเป็นเรื่องใดก็ตามตลอดมา

ขอขอบพระคุณ รองศาสตราจารย์ ณรงค์ เหมภรณ์ อาจารย์ที่ปรึกษาร่วมเป็นอย่างสูง ที่ได้ให้ คำแนะนำและคำปรึกษาในทุกๆ เรื่องสำหรับการวิจัยอีกทั้งยังเอื้อให้ความช่วยเหลือเรื่องเครื่องมือ ทุกๆ ชิ้น

ขอขอบพระคุณ ศาสตราจารย์ ดร.วิวัฒน์ กิรานนท์ ประธานคณะกรรมการสอบโครงงาน ที่ได้ ช่วยตรวจทานแก้ไขและให้คำชี้แนะในการทำโครงงานฉบับนี้

ขอขอบพระคุณ อาจารย์คณะกรรมการทุกๆ ท่านสำหรับการสอบวิชาโครงงาน

ขอขอบพระคุณ นายสุภกิจ แก้วดวงตา รุ่นพี่นักศึกษาปริญญาโท ภาควิชาวิศวกรรม โทโรคมณฑล ที่ได้คอยให้คำแนะนำในการทำงานให้สำเร็จลุล่วงไปด้วยดี

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ด้วยความเคารพรักยิ่ง สำหรับโอกาส ความรัก คำแนะนำและกำลังใจในการศึกษาและการทำโครงงานพิเศษ สิ่งที่ท่านมอบให้มากเกินจะเขียนได้

สุดท้ายขอขอบคุณเพื่อนๆ พี่ๆ ทุกท่านที่มีได้กล่าวนามมา ณ ที่นี้ ที่ได้คำปรึกษา การสนับสนุน ช่วยเหลือรวมทั้งเป็นกำลังใจในการทำการศึกษาค้นคว้ามาโดยตลอด รวมถึงมีส่วนช่วยให้โครงงานฉบับ นี้เสร็จสมบูรณ์

ว่าที่ร.ต.วรธันย์ วิชาคุณ
นายเผ่าพันธุ์ ธีรนนท์ธนพัฒน์

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
ที่มาและความสำคัญของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 จูนเนอร์เครื่องรับโทรทัศน์ (Tuner)	2
2.1.1 หน้าที่และคุณสมบัติของจูนเนอร์เครื่องรับโทรทัศน์	2
2.1.1.1 หน้าที่ของจูนเนอร์	2
2.1.1.2 คุณสมบัติของวงจรจูนเนอร์	2
2.1.2 การทำงานของจูนเนอร์เครื่องรับโทรทัศน์	2
2.1.3 หลักการของอิเล็กทรอนิกส์จูนเนอร์	4
2.2 วงจรเอจีสตี	6
2.2.1 หน้าที่ของเอจีสตี	6
2.2.2 หลักการทำงานของวงจรเอจีสตี	7
2.3 วงจรขยายสัญญาณไอเอฟภาพรวม	9
2.3.1 หน้าที่ของวงจรขยายสัญญาณไอเอฟภาพรวม	9
2.3.2 แบบคิวิตซ์และการตอบสนองความถี่ของภาคขยายความถี่ไอเอฟภาพรวม	9
2.4 วงจรขยายสัญญาณอาร์เอฟ (RF Amplifiers)	11
2.4.1 ประโยชน์ของการใช้ RF Amplifier	11
2.4.2 ประโยชน์ของเฟส	12
2.5 วงจรออสซิลเลเตอร์ (Oscillator Circuit)	13
2.5.1 หลักการของวงจรออสซิลเลเตอร์	13
2.6 วงจรมิกเซอร์ (Mixer Circuit)	14
2.7 เซรามิกฟิลเตอร์ (Ceramic Filter)	17
2.8 ไมโครคอนโทรลเลอร์	18
2.8.1 ไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์	18
2.8.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยบริษัทอินเทล	18
2.8.2 หน่วยความจำภายในชิพ	18
2.8.2.1 โครงสร้างภายในของ 8051	18
2.8.3 พอร์ตของ 8051	20
2.8.4 วงจรคล็อกของ MCS-51	23
2.8.5 การแปลงสัญญาณอนาลอกเป็นดิจิตอล	24

สารบัญ (ต่อ)

	หน้า
2.8.6 การเชื่อมโยง ADC 0804 เข้ากับไมโครคอนโทรลเลอร์ MCS-51	27
2.9 การใช้งานพอร์ตสื่อสารอนุกรมแบบ Single Processor ของพอร์ตสื่อสารอนุกรม	28
2.9.1 การเขียนโปรแกรมควบคุมการรับและส่งข้อมูล	29
2.9.1.1 การเชื่อมต่อไมโครโปรเซสเซอร์เพื่อรับส่งข้อมูล อนุกรม (UART)	29
2.9.1.2 การเชื่อมต่อแบบมาตรฐาน RS-232C	31
2.10 การใช้งาน LCD Module โครงสร้าง LCD Module	32
2.10.1 การเชื่อมต่อ LCD Module เข้ากับไมโครคอนโทรลเลอร์	33
2.11 สเต็ปป์มอเตอร์ (stepping Motor)	37
2.11.1 โครงสร้างและหลักการทำงานของสเต็ปป์มอเตอร์	38
2.11.2 การทำงานพื้นฐาน	39
บทที่ 3 การออกแบบและการสร้าง	42
3.1 บล็อกไดอะแกรมของระบบตรวจวัดและบันทึกความแรงสัญญาณวิทยุ	42
3.2 วงจรขยายสัญญาณ IF ของจูนเนอร์	42
3.3 วงจรมิกเซอร์	42
3.4 วงจรกรองความถี่ผ่าน 10.7 เมกะเฮิร์ตซ์	43
3.5 การออกแบบวงจรขยายความถี่กลางที่ความถี่ 10.7 เมกะเฮิร์ตซ์ (IF Amplifier 10.7 MHz)	43
3.6 การออกแบบวงจรวัดความแรงของสัญญาณ	46
3.7 การแปลงสัญญาณอนาล็อกเป็นดิจิตอล	47
3.8 การออกแบบวงจรรับ-ส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ของ ไมโครคอนโทรลเลอร์ MCS – 51	48
3.9 การแสดงข้อมูลบนโมดูล LCD เข้ากับไมโครคอนโทรลเลอร์	49
3.10 การออกแบบวงจรควบคุมสเต็ปป์มอเตอร์และรับ-ส่งข้อมูลผ่านคอนโทรลเลอร์	50
3.11 โปรแกรมการตรวจวัดสัญญาณและการบันทึกผลบนคอมพิวเตอร์	51
บทที่ 4 การทดลองและผลการทดลอง	54
4.1 ผลการทดลองวงจรขยายสัญญาณ RF	54
4.2 ผลการทดลองวงจรมิกเซอร์ (Mixer)	55
4.3 ผลการทดลองวงจรกรองแถบความถี่ 10.7 เมกะเฮิร์ตซ์	57
4.4 ผลการทดลองวงจรขยายสัญญาณความถี่กลางที่ 10.7 เมกะเฮิร์ตซ์	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.5 การทดลองวงจรเอจีสซีของจูนเนอร์	59
4.6 ผลการทดลองวงจรวัดความแรงสัญญาณ	60
4.7 ผลการทดลองวงจรแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิทัล (Analog to Digital Converter) แล้วแสดงผลที่หน้าจอ LCD	61
4.8 แสดงผลของโปรแกรมการหาทิศทางของสัญญาณ เมื่อ ทำการรับค่าจากไมโครคอนโทรลเลอร์	63
4.9 แสดงผลของการเก็บบันทึกค่าสถานะต่างๆขณะที่ทำการวัด	62
บทที่ 5 บทสรุปและวิจารณ์	65
ภาคผนวก	66
1. โปรแกรมคอนโทรลเลอร์ในการแสดงผลที่จอ LCD ส่งออกพอร์ตอนุกรม	66
2. โปรแกรมการแสดงผลที่คอมพิวเตอร์	69
กิตติกรรมประกาศ	78
หนังสืออ้างอิง	79

สารบัญรูปภาพ

รูปที่	หน้า
1.1 บล็อกไดอะแกรมแสดงการทำงานของระบบ	1
2.1 แสดงบล็อกไดอะแกรมการทำงานของจูนเนอร์เครื่องรับโทรทัศน์	2
2.2 การเกิดค่าความจุและการใช้งานวาริแคปไดโอด	5
2.3 บล็อกไดอะแกรม	5
2.4 บล็อกไดอะแกรมอิเล็กทรอนิกส์จูนเนอร์ VHF/UHF (Philips)	6
2.5 บล็อกไดอะแกรมตำแหน่งเอจี้ซี	7
2.6 กราฟการทำงานของเอจี้ซี	8
2.7 กราฟการไหลของกระแสคอลเลกเตอร์กับอัตราขยาย	8
2.8 สเปกตรัมสัญญาณโทรทัศน์ช่อง 7 ที่รับเข้าทางจูนเนอร์	10
2.9 ขอบเขตความถี่สัญญาณไอเอฟภาพรวมในเครื่องรับโทรทัศน์	10
2.10 เส้นโค้งตอบสนองความถี่ไอเอฟภาพรวมที่ต้องการ	11
2.11 วงจร MOSFET RF แบบ dual – gate	12
2.12 วงจรออสซิลเลเตอร์เบื้องต้น	14
2.13 ตัวอย่างการทำงานของวงจรมิกเซอร์	15
2.14 วงจรบาลานซ์มิกเซอร์ชนิดพาสซีฟ	16
2.15 วงจรมิกเซอร์แบบไม่สมดุล	16
2.16 หม้อแปลงเรโซแนนซ์และเซรามิกฟิลเตอร์	17
2.17 โครงสร้างภายใน 8051	19
2.18 การจัดวางขาของ 8051	20
2.19 แสดงโครงสร้าง พอร์ต 0 (บิต)	21
2.20 โครงสร้างของพอร์ต 1 (บิต)	21
2.21 โครงสร้างของ พอร์ต 2 (บิต)	22
2.22 โครงสร้างของพอร์ต 3 (บิต)	23
2.23 การต่อขาเรซีตให้กับ 8051	23
2.24 วงจรสร้างค็ลคของ 8051	24
2.25 รายละเอียดและการจัดวางขาของ ADC 0804	25
2.26 ผังโครงสร้างภายในของ ADC 0804	25
2.27 ผังเวลาการสาร์ท A/D	26
2.28 ผังเวลาการอ่านข้อมูลไปเก็บ	27
2.29 การต่อ ADC 0804 เข้ากับ MCS-51	28
2.30 การใช้งาน โหมด 1	29
2.31 ผังการทำงาน Serial Port (Mode 1)	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
2.32 RS-232-C Voltage Range	31
2.33 การเชื่อมต่อ MCS-51 กับ RS-232	31
2.34 แสดงลักษณะของตัว LCD Module	32
2.35 แสดงการเชื่อมต่อ LCD Module เข้ากับ 8255	33
2.36 แสดงรหัสตัวอักษรที่ใช้กับ LCD Module	37
2.37 โครงสร้างของสเต็ปปีงมอเตอร์	38
2.38 เส้นแรงแม่เหล็กที่ทำให้เกิดแรงบิด	38
2.39 แสดงการทำงานของสเต็ปปีงมอเตอร์	38
2.40 การทำงานพื้นฐานของสเต็ปปีงมอเตอร์	40
2.41 ชนิดการขับของสเต็ปปีงมอเตอร์	41
3.1 บล็อกไดอะแกรมของวงจรส่วนรับและตรวจจับสัญญาณ	42
3.2 วงจรขยายสัญญาณ IF ของจูนเนอร์	42
3.3 วงจรสมมูลของวงจรมิกเซอร์	43
3.4 วงจรกรองความถี่ผ่าน 10.7 เมกะเฮิร์ตซ์	43
3.5 วงจรสมมูลทาง DC ของวงจรคอมมอนอีมิเตอร์	43
3.6 วงจรขยาย 10.7 เมกะเฮิร์ตซ์	46
3.7 วงจรวัดความแรงของสัญญาณความถี่กลาง 10.7 เมกะเฮิร์ตซ์	46
3.8 การเชื่อมต่อ ADC 0804 เข้ากับ ไมโครคอนโทรลเลอร์	47
3.9 วงจรการสื่อสารข้อมูลผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	48
3.10 แสดงการเชื่อมต่อโมดูล LCD 16 ตัวอักษร 1 บรรทัดเข้ากับ MCS 51	49
3.11 วงจรการควบคุมสเต็ปปีงมอเตอร์และรับส่งข้อมูลเข้าคอมพิวเตอร์	50
3.12 ไดอะแกรมการทำงานของโปรแกรมหลักของคอมพิวเตอร์	51
3.13 ไดอะแกรมการทำงานของโปรแกรมส่วนของการพล็อตกราฟวงกลม	52
3.14 ไดอะแกรมการทำงานของโปรแกรมส่วนของการพล็อตกราฟเส้นและกราฟบาร์	53
4.1 สัญญาณทางด้านอินพุตก่อนเข้าวงจรขยายสัญญาณ	54
4.2 สัญญาณทางด้านเอาต์พุตหลังจากที่ผ่านวงจรขยายสัญญาณ	54
4.3 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุตของวงจรขยาย	55
4.4 สเปกตรัมของสัญญาณภาพที่ออกจากจูนเนอร์(IF Tuner) 37.96 เมกะเฮิร์ตซ์	56
4.5 สเปกตรัมของสัญญาณจากเครื่องส่งเคราะห์ความถี่(Signal Generator) 48.66 เมกะเฮิร์ตซ์	56
4.6 สเปกตรัมของสัญญาณเอาต์พุตที่ได้จากวงจรมิกเซอร์	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปร่าง (ต่อ)

รูปที่	หน้า
4.7 กราฟผลตอบแทนความถี่ของวงจรกรองความถี่ 10.7 เมกะเฮิร์ตซ์	58
4.8 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุต ของวงจรขยายสัญญาณความถี่กลาง 10.7 เมกะเฮิร์ตซ์	59
4.9 กราฟแสดงความสัมพันธ์ระหว่างอัตราขยายของจูนเนอร์กับความ ความแรงของสัญญาณ	60
4.10 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุต ของวงจรวัดความแรงสัญญาณ	61
4.11 แสดงโปรแกรมการหาทิศทางและการแสดงสถานะของค่าที่ทำกรวัด	63
4.12 แสดงการบันทึกค่าสถานะต่างๆที่ทำการตรวจวัดสัญญาณ	63
4.13 รูปของการบันทึกค่าสถานะต่างๆที่ทำการตรวจวัดสัญญาณ	64



สารบัญตาราง

ตารางที่	หน้า
2.1 รุ่นของ 8051 จาก ATMEL (All ROM Flash)	19
2.2 Various Speeds of 8051 From Atmel	20
2.3 ขาดต่างๆ ของ ADC 0804	26
2.4 SMO, SM1 บิตเลือกโหมดการทำงาน	29
2.5 ค่าสัญญาณดิจิทัลที่ตีเทคจากแรงดัน	31
2.6 คำสั่งเคลียร์ตัวแสดงผล	32
2.7 คำสั่ง Cursor at Home	34
2.8 คำสั่งโหมดในการป้อนข้อมูล	34
2.9 คำสั่งควบคุมการแสดงผล	34
2.10 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร	35
2.11 คำสั่งการกำหนดฟังก์ชันการทำงาน	35
2.12 คำสั่งเลือกแอดเดรสของ CGRAM	35
2.13 คำสั่งเลือกแอดเดรสของ DDRAM	36
2.14 คำสั่งอ่านแฟลช Busy และ Address	36
4.1 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับระดับสัญญาณเอาต์พุต ที่ความถี่ 38 เมกะเฮิร์ตซ์	55
4.2 ตารางผลตอบสนองของเวลาที่ของวงจรองแถบความถี่ 10.7 เมกะเฮิร์ตซ์	57
4.3 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับสัญญาณเอาต์พุต	58
4.4 การเปรียบเทียบระหว่างอัตราขยายสัญญาณของจูนเนอร์และความแรงของสัญญาณ	59
4.5 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับระดับสัญญาณเอาต์พุต	60
4.6 ตารางเปรียบเทียบอินพุตที่มีค่าเป็นแรงดันกับเอาต์พุตที่ได้ค่าเป็นดิจิทัล	61

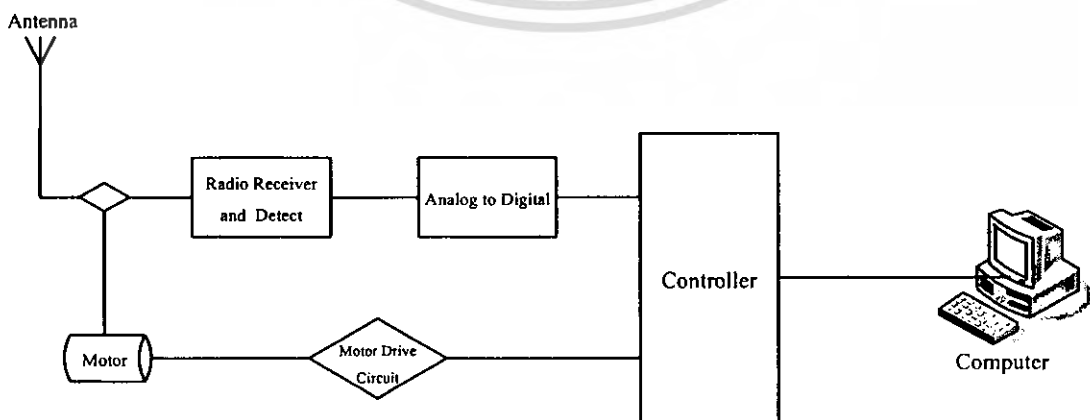
บทที่ 1

บทนำ

ที่มาและความสำคัญของโครงการ

ในชีวิตประจำวันปัจจุบันเราฟังวิทยุ ดูโทรทัศน์กันแทบทุกคน เสียงวิทยุหรือภาพจากโทรทัศน์ที่ส่งไปตามบ้านจากสถานีส่งกระจายเสียงและแพร่ภาพนั้นจะอยู่ในรูปของคลื่นวิทยุซึ่งจัดเป็นคลื่นแม่เหล็กไฟฟ้า (Electromagnetic Wave) ที่กระจายออกจากสายอากาศที่วางอยู่ใกล้ผิวโลกและเดินทางออกไปทุกทิศในทุกระนาบ การกระจายของคลื่นวิทยุนี้มีลักษณะเป็นการขยายตัวของพลังงานออกเป็นวงกลม โดยทั่วไปคลื่นวิทยุอาจแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือ คลื่นดิน (Ground Wave) กับคลื่นฟ้า (Sky Wave) การเคลื่อนที่ของคลื่นวิทยุนี้เกิดภายในบรรยากาศโดยมีต้องอาศัยสายไฟฟ้า ดังนั้นถ้าเกิดการเปลี่ยนแปลงของบรรยากาศจะทำให้เกิดการเปลี่ยนแปลงหลายอย่างในการเคลื่อนที่ของคลื่นวิทยุ เช่น ระดับเสียงของคลื่นวิทยุสูงบ้างต่ำบ้าง การจางหาย (Fading) เกิดพายุแม่เหล็ก การได้ยินเสียงสะท้อน (Echo) และเสียงรบกวนอาจเกิดขึ้นเนื่องมาจากปรากฏการณ์ธรรมชาติหรือการเคลื่อนที่ของอากาศยานเช่น เครื่องบินที่บินผ่านบนท้องฟ้าซึ่งอยู่ในระยะที่ใกล้แนวคลื่น

โครงการนี้จึงได้เกิดแนวความคิดในการสร้างระบบตรวจวัดความแรงของสัญญาณวิทยุขึ้นมา โดยจะสามารถวัดระดับความแรงของสัญญาณวิทยุ (RF : Radio Frequency) ที่ใช้งานย่านความถี่ VHF ซึ่งในการตรวจวัดสัญญาณนั้นเพื่อให้เข้าใจได้ง่ายขึ้นจะทำการตรวจวัดสัญญาณที่มีการใช้งานอยู่จริงซึ่งก็คือความถี่ของโทรทัศน์ในช่วง 174 – 223 เมกะเฮิรตซ์ (MHz) ของโทรทัศน์ช่อง 5, 7, 9 และช่อง 11 โดยใช้สายอากาศแบบมีทิศทางในการรับคลื่นและใช้สเต็ปปีงมอเตอร์ที่ควบคุมโดยไมโครคอนโทรลเลอร์หมุนให้เป็นสเต็ป เพื่อหาทิศทางที่แรงที่สุดของสัญญาณแล้วหลังจากนั้นก็ทำการส่งข้อมูลเพื่อไปประมวลผลบนคอมพิวเตอร์และให้แสดงผลความแรงของสัญญาณในรูปแบบกราฟวงกลมรอบทิศทาง 360 องศา ถึงตอนนี้ทำให้เราสามารถที่จะทราบได้แล้วว่าทิศทางไหนของโทรทัศน์แต่ละช่องจะแรงที่สุด จากนั้นทำการปรับสายอากาศให้หันไปทิศทางนั้นค้างไว้แล้วบันทึกการเปลี่ยนแปลงของระดับสัญญาณเมื่อเวลาเปลี่ยนไปตามเวลาบนคอมพิวเตอร์ ถ้าหากมีอากาศยานบินผ่านระหว่างแนวการส่งและรับสัญญาณนั้นก็ทำให้ความแรงของสัญญาณนั้นเปลี่ยนแปลงไป



รูปที่ 1.1 บล็อกไดอะแกรมแสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 จูนเนอร์เครื่องรับโทรทัศน์

2.1.1 หน้าที่และคุณสมบัติของจูนเนอร์เครื่องรับโทรทัศน์

วงจรจูนเนอร์เครื่องรับโทรทัศน์เป็นส่วนที่รับคลื่นสัญญาณโทรทัศน์เข้ามาเพื่อให้ผู้ใช้สามารถเลือกรับรายการของช่องต่างๆ ได้วงจรจูนเนอร์จะทำงานแบบซูเปอร์เฮเทอโรไดน์แบบด์วิคท์ของสัญญาณผ่านเข้าและผ่านออกจากวงจรมีความกว้างตามมาตรฐานของระบบโทรทัศน์ที่รับ เช่น มาตรฐานของระบบ CCIR (PAL-B) ที่ใช้อยู่ในประเทศไทยมีแบนด์วิคท์ของสัญญาณ 7 เมกะเฮิรตซ์

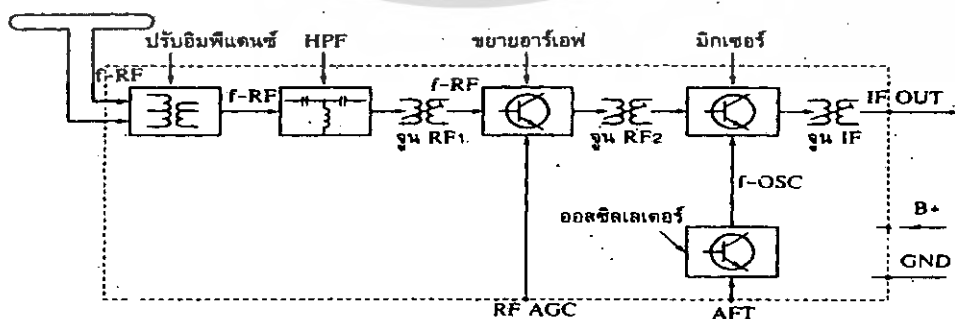
2.1.1.1 หน้าที่ของจูนเนอร์ มีดังนี้

1. ปรับอิมพีแดนซ์ (Impedance) ของเครื่องรับโทรทัศน์ให้แมตช์พอดีกับอิมพีแดนซ์ของสายนำสัญญาณ
2. ทำการแยกทางไฟฟ้า (Electrical Isolation) ของสายอากาศออกจากกราวด์และไฟฟ้าของเครื่องรับ และให้ผู้ชมสามารถเลือกรับสัญญาณจากสถานีส่งในช่องที่ต้องการได้
3. ขยายสัญญาณโทรทัศน์ที่รับเข้ามาให้มีความแรงพอกับความต้องการที่จะส่งให้ภาคขยายความถี่ปานกลางโดยกำจัดสัญญาณรบกวนออกไป
4. เปลี่ยนความถี่คลื่นพาหะของสัญญาณโทรทัศน์ช่องที่รับเข้ามาให้เป็นความถี่กลาง

2.1.1.2 คุณสมบัติของวงจรถูนเนอร์ มีดังนี้

1. ลดและขจัดสัญญาณรบกวนได้ดี ความถี่มีเสถียรภาพและไม่เกิดการออสซิลเลตในตัวเอง (Parasitic Oscillation)
2. ขยายความถี่ได้กว้างพอดำเนินมาตรฐานช่องที่รับครอบคลุมความถี่พาหะภาพและพาหะเสียงของช่องที่รับ ให้ระดับสัญญาณไอเอฟเอาต์พุตที่สม่ำเสมอ

2.1.2 การทำงานของจูนเนอร์เครื่องรับโทรทัศน์



รูปที่ 2.1 บล็อกไดอะแกรมการทำงานของจูนเนอร์เครื่องรับโทรทัศน์

จากบล็อกไดอะแกรมจูนเนอร์เครื่องรับโทรทัศน์ในรูปที่ 2.1 มีหลักการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สายอากาศโทรทัศน์รับสัญญาณโทรทัศน์ช่องต่างๆ ส่งเข้าตามสายนำสัญญาณ เพื่อให้ตัวปรับอิมพีแดนซ์รับสัญญาณนี้ส่งเข้าภาคขยายอาร์เอฟ เนื่องจากสายนำสัญญาณที่ใช้จะมีอยู่ 2 แบบคือ สายแบนด์วิทลิตมีอิมพีแดนซ์ 300 โอห์มกับสายกลมโคแอกเชียลมีอิมพีแดนซ์ 75 โอห์มเพื่อประสิทธิภาพการถ่ายทอดกำลังสัญญาณโทรทัศน์ได้สูงสุดตัวปรับอิมพีแดนซ์จะทำหน้าที่ปรับอิมพีแดนซ์ของสายนำสัญญาณให้พอดีกับอินพุตของจูนเนอร์ซึ่งมีอิมพีแดนซ์ประมาณ 75 โอห์ม ขณะเดียวกันตัวปรับอิมพีแดนซ์นี้ถูกออกแบบให้ทำหน้าที่เป็นตัวแยกไฟฟ้าภายในเครื่องกับสายอากาศด้วย

2. สัญญาณโทรทัศน์ที่ปรับอิมพีแดนซ์แล้วจะเข้าสู่วงจรกรองความถี่สูงผ่าน (HPF: High Pass Filter) ทำหน้าที่กรองความถี่อาร์เอฟ (Radio Frequency) อื่นๆ ที่มีแถบใกล้เคียงกับสัญญาณโทรทัศน์ เช่น ความถี่วิทยุ AM วิทยุ FM และวิทยุสื่อสารไม่ให้เข้ามารบกวนความถี่ของสัญญาณโทรทัศน์ภายในเครื่องรับ ดังนั้นจึงมีเพียงความถี่ของสัญญาณโทรทัศน์เท่านั้นที่ผ่านเข้าไปถึงวงจรขยายอาร์เอฟ

3. สัญญาณโทรทัศน์ที่ผ่านวงจรกรองความถี่สูงผ่านมาแล้วจะถูกวงจรจูน RF1 จูนคัดเลือกเฉพาะความถี่สัญญาณของช่องที่ต้องการรับเท่านั้นให้ผ่านมายังวงจรขยายอาร์เอฟเพื่อขยายสัญญาณโทรทัศน์ให้แรงขึ้น สัญญาณโทรทัศน์ที่ขยายแล้วจะถูกจูนออกทางวงจรจูน RF2 ให้เข้าไปยังมิกเซอร์

4. อัตราขยายวงจรขยายอาร์เอฟถูกควบคุมด้วยแรงดันเอจิสซีเพื่อให้ทำการขยายอย่างเหมาะสม วงจรเอจิสซีที่ควบคุมการขยายของภาคอาร์เอฟจะเป็นดีเลย์เอจิสซี (Delay AGC) ถ้าหากสัญญาณที่เข้าทางสายอากาศมีความแรงต่ำกว่า 60 เดซิเบล (dB) แรงดันที่เอจิสซีจ่ายไบแอสแก่วงจรขยายอาร์เอฟจะมีค่าคงที่พอเหมาะที่จะทำให้ภาคขยายอาร์เอฟของสัญญาณได้แรงที่สุด แต่ถ้าสัญญาณรับเข้ามาทางสายอากาศมีความแรงเกินกว่า 60 เดซิเบล วงจรเอจิสซีจะจ่ายแรงดันไบแอสมาควบคุมการขยายของวงจรขยายอาร์เอฟให้ลดอัตราการขยายลงเพื่อรักษาระดับสัญญาณเอาต์พุตให้สม่ำเสมอไม่ให้เกิดการขยายสัญญาณมีเอาต์พุตแรงเกินไป อันเกิดผลในเครื่องรับคือเกิดอาการภาพคำเข่มจืดมีเสียงครางและภาพคดหรือถูกดึงได้

5. วงจรออสซิลเลเตอร์หรือโลคอลออสซิลเลเตอร์ทำหน้าที่สร้างคลื่นความถี่ VHF (หรือ UHF) ที่มีความถี่สูงกว่าความถี่สัญญาณโทรทัศน์ของช่องที่รับเข้ามาเท่ากับความถี่ปานกลางหรือความถี่ไอเอฟ (IF: Intermediate Frequency) ความถี่ออสซิลเลเตอร์นี้จะถูกส่งเข้าภาคมิกเซอร์ในจูนเนอร์ บางเครื่องจะมีการควบคุมความถี่ของออสซิลเลเตอร์ให้คงที่ด้วยวงจรเอฟที

6. วงจรมิกเซอร์จะรับสัญญาณโทรทัศน์ที่ขยายจากภาคขยายอาร์เอฟและจูนผ่านวงจร RF2 เข้ามาแล้วนำมาผสมกับสัญญาณความถี่ออสซิลเลเตอร์เพื่อเปลี่ยนคลื่นพาหะของสัญญาณภาพและพาหะของสัญญาณเสียงให้เป็นสัญญาณความถี่ไอเอฟภาพและไอเอฟเสียงซึ่งจะถูกจูนออกด้วยวงจรจูน IF เพื่อส่งออกไปเข้าภาคขยายไอเอฟรวมต่อไป

7. สัญญาณไอเอฟเอาต์พุตที่จูนออกจากวงจรมิกเซอร์จะยังคงมีรูปร่างเหมือนกับสัญญาณโทรทัศน์ของช่องที่รับมาจากสายอากาศและต่างกันเฉพาะคลื่นความถี่พาหะ โดยความถี่พาหะ

เสียงจะถูกเปลี่ยนให้มีความถี่ต่ำลงเป็นความถี่ไอเอฟไม่ว่ารับสัญญาณโทรทัศน์จากช่องใดก็ตาม ความถี่ไอเอฟภาพและเสียงจะมีค่าเท่ากันหมดและความถี่ไอเอฟภาพจะสูงกว่าความถี่ไอเอฟเสียงเท่ากับในระบบ CCIR (625 เส้น) และเท่ากับ 4.5 เมกะเฮิร์ตซ์ ในระบบ FCC (525 เส้น)

ความถี่ไอเอฟจะหาได้จากสูตรต่อไปนี้

$$f_{IF} = f_{OSC} - f_{RF} \quad (2.1)$$

โดยที่ f_{IF} คือความถี่ไอเอฟ

f_{OSC} คือความถี่ออสซิลเลเตอร์

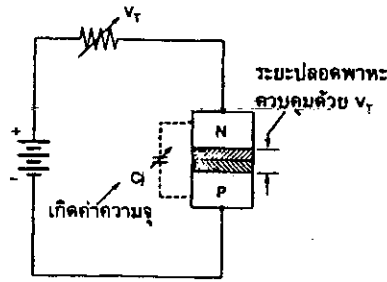
f_{RF} คือความถี่พาหะภาพ (หรือพาหะเสียง)

2.1.3 หลักการของอิเล็กทรอนิกส์จูนเนอร์

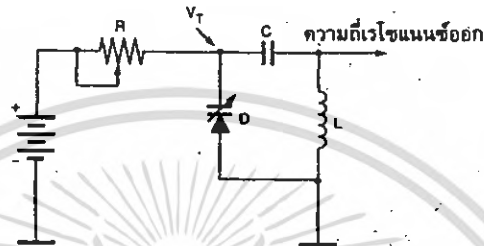
อิเล็กทรอนิกส์จูนเนอร์คือจูนเนอร์ที่ใช้วาร์แคปเตอร์ไดโอด (Varactor Diode) หรือวารีแคปไดโอด (Varicap Diode) ต่อเป็นวงจรรีโซแนนซ์รวมอยู่กับขดลวดและตัวเก็บประจุ การเปลี่ยนแปลงค่าความถี่เรโซแนนซ์ใช้วิธีเปลี่ยนแปลงแรงดันไบแอสกลับที่ต่อตัววารีแคปไดโอดเพื่อให้ค่าความจุภายในวารีแคปไดโอดเปลี่ยนแปลง อิเล็กทรอนิกส์จูนเนอร์จึงสามารถเลือกความถี่หรือเลือกรับสถานีช่องต่างๆ ด้วยการป้อนแรงดันที่เหมาะสมเข้าไปในจูนเนอร์ โดยปกติแรงดันที่ป้อนเข้าไปเลือกช่องถูกตั้งล่วงหน้าไว้พอเหมาะกับช่องที่ต้องการรับแล้วใช้วิธีคัปเปิล วิธีสลับสวิตช์หรือการควบคุมระยะไกลเพื่อเปลี่ยนแปลงช่องที่ต้องการจูนเนอร์แบบอิเล็กทรอนิกส์จึงแพร่หลายมากเพราะสะดวกไกลการจูนและมีขนาดเล็กลงสามารถติดตั้งไว้บนแผ่นวงจรพิมพ์ที่ไกลจากปุ่มเปลี่ยนช่องได้

วารีแคปไดโอดมีลักษณะของโครงสร้างที่ประกอบด้วยสารกึ่งตัวนำ N และ P เชื่อมติดกันแต่สร้างให้รอยต่อ PN (PN junction) มีค่าความจุของตัวเก็บประจุสูงเป็นพิเศษโดยค่าความจูนี่สามารถควบคุมให้เปลี่ยนแปลงได้ในขอบเขตหนึ่ง ด้วยการป้อนแรงดันไบแอสกลับคร่อมตัวไดโอด จำนวนของแรงดันที่มากขึ้นคร่อมไดโอดจะทำให้ระยะปลอดพาหะ (Depletion Region) ตรงรอยต่อกว้างหรือแคบได้ (ทางไฟฟ้า) ตัววารีแคปไดโอดจึงมีลักษณะเป็นตัวเก็บประจุที่สารกึ่งตัวนำ P และ N ทำหน้าที่เป็นแผ่นเพลตมีระยะปลอดพาหะเป็นฉนวนไดอิเล็กตริก (Dielectric) ดังนั้นเมื่อระยะห่างของแผ่นเพลตถูกควบคุมด้วยแรงดันที่ทำให้ระยะปลอดพาหะกว้างหรือแคบได้ ค่าความจุที่เกิดในตัววารีแคปไดโอดจึงสามารถควบคุมให้มีค่าความจุตามที่ต้องการได้ด้วยการป้อนแรงดันไบแอสจากภายนอกเข้าไปบังคับการเกิดค่าความจุและการใช้งานวารีแคปไดโอด แสดงตามรูปที่ 2.2

จากรูปที่ (2.2ข) จะเห็นว่าขดลวด L ต่อเป็นวงจรรีโซแนนซ์ร่วมกับตัวเก็บประจุ C และวาร์แคปเตอร์ไดโอด D แรงดันไฟตรง V_T จะผ่านตัวต้านทาน R เข้าไปไบแอสกลับแก่วารีแคปนอก จากตัวเก็บประจุ C จะทำหน้าที่เรโซแนนซ์แล้วยกกันไฟบวกไม่ให้ลงกราวด์ผ่านทางขดลวดเมื่อปรับค่าแรงดัน V_T จะทำให้ค่าความจุของวารีแคปเปลี่ยนค่าความถี่ของวงจรรีโซแนนซ์ก็จะเปลี่ยนแปลง จึงทำให้วงจรนี้จูนความถี่ค่าต่างๆ ตามที่ต้องการได้โดยการปรับค่าแรงดัน V_T ที่ป้อนเข้าวงจร



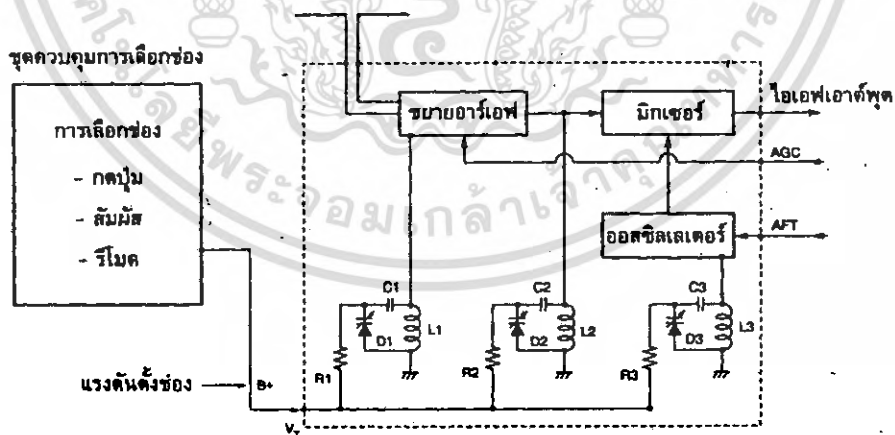
(ก) การเกิดค่าความจุที่รอยต่อ PN



(ข) การใช้วารีแคปในวงจรเรโซแนนซ์

รูปที่ 2.2 การเกิดค่าความจุและการใช้งานวารีแคปไดโอด

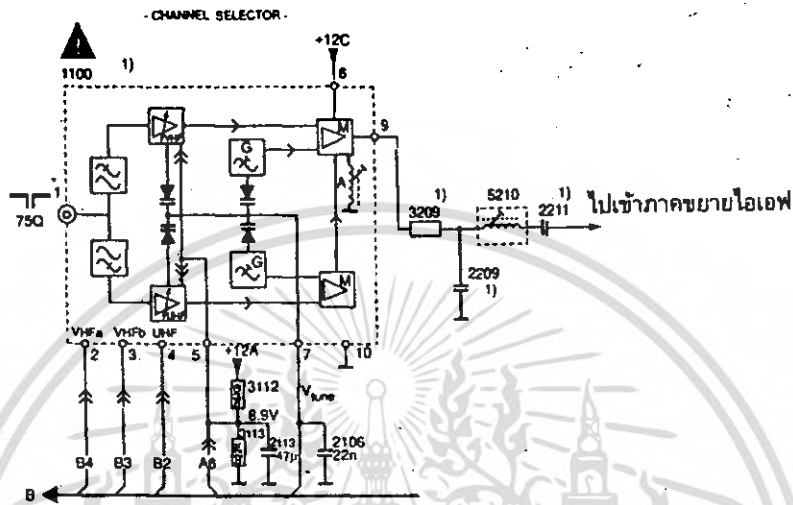
ในการนำวารีแคปไดโอดไปใช้งานในอิเล็กทรอนิกส์จูนเนอร์นั้นต้องใช้วารีแคปไดโอดที่ทางโรงงานสร้างมาเป็นชุดมีคุณสมบัติเหมือนกันนำมาต่อไว้ในวงจรเรโซแนนซ์ของภาคขยายอาร์เอฟ ภาคมิกเซอร์และภาคออสซิลเลเตอร์ การจูนมีแรงดันที่จูนป้อนเข้ามาร่วมกันการเปลี่ยนค่าความถี่ที่จูนป้อนเข้าร่วมกันการเปลี่ยนค่าความถี่ที่จูนได้จึงมีอัตราส่วนสัมพันธ์กันทั้ง 3 วงจรบล็อกไดอะแกรมอิเล็กทรอนิกส์จูนเนอร์แสดงดังรูปที่ 2.3



รูปที่ 2.3 บล็อกไดอะแกรม

จากรูปที่ 2.3 จะเห็นได้ว่าการใช้งานอิเล็กทรอนิกส์จูนเนอร์มีส่วนประกอบอยู่ 2 ชุดคือชุดแรกเป็นชุดจูนเนอร์ที่มีวารีแคปไดโอดประกอบอยู่ในวงจรเรโซแนนซ์ทั้ง 3 ภาคกับชุดที่สองเป็นชุดเลือกแบนด์และจูนเลือกช่องสามารถติดตั้งไว้ห่างจากตัวอิเล็กทรอนิกส์จูนเนอร์ได้แล้วต่อสายไฟเข้าหา

กัน แรงดันจูนเลือกช่อง V_T ป้อนเข้าจูนเนอร์แรงดันนี้จะเข้า R1, R2, R3 เท่าๆ กันเป็นไบแอสวาริแคป วงจรจูนทั้ง 3 ชุดจะได้แรงดันไบแอสเท่ากันความถี่รีโซแนนซ์จึงสัมพันธ์กันทั้งหมดทำให้เกิดความถี่ไอเอฟออกทางเอาต์พุต ในวงจรส่วนอื่นของอิเล็กทรอนิกส์จูนเนอร์ก็ยังคงเหมือนจูนเนอร์แบบกลไกคือมีแรงดันเอจีซีมาควบคุมอัตราขยายของวงจรขยายอาร์เอฟและมีแรงดันเอเอฟที่มัลติเพล็กซ์ของออสซิลเลเตอร์



รูปที่ 2.4 บล็อกโคแอสเทรอนอิเล็กทรอนิกส์จูนเนอร์ VHF/UHF (Philips)

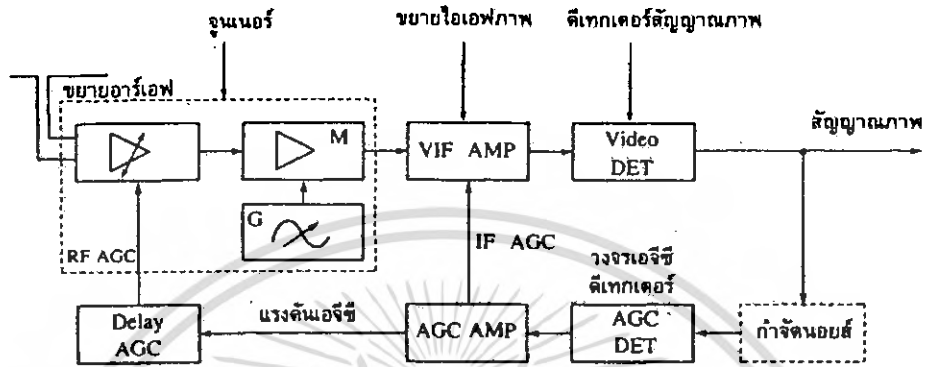
บล็อกโคแอสเทรอนในรูปที่ 2.4 เป็นอิเล็กทรอนิกส์จูนเนอร์ VHF/UHF ของ Philips สายอากาศมีอิมพีแดนซ์ 75 โอห์มป้อนเข้าที่ขา 1 ของจูนเนอร์สายอากาศนี้ใช้ร่วมกันทั้ง VHF/UHF คลื่น VHF จะผ่านวงจรกรองความถี่ (HPF) ไปยังหน่วยขยายคลื่น UHF หน่วยออสซิลเลเตอร์กำเนิดความถี่และมิกเซอร์แยกกันระหว่าง VHF กับ UHF การจูนคลื่นช่องที่รับเข้าและออสซิลเลเตอร์ทั้ง VHF/UHF ใช้จูนด้วยวาริแคปไดโอดโดยใช้แรงดัน V_{tune} จากสาย B5 ป้อนเข้าจูนเนอร์ที่ขา 7 คลื่นโทรทัศน์ที่ผ่านหน่วยมิกเซอร์และเป็นความถี่ไอเอฟแล้วจะถูกจูนออกด้วยขดลวด A ส่งออกจากจูนเนอร์ที่ขา 9 การเลือกแบนด์ใช้แรงดันจากสาย B4 ป้อนเข้าที่ขา 2 เพื่อใช้สำหรับ VHFa คือ VHF แบนด์ต่ำช่อง 2-4 ใช้แรงดันจากสาย B3 ป้อนเข้าที่ขา 3 ใช้สำหรับ VHFb คือ VHF แบนด์สูงช่อง 5-12 และใช้แรงดันจากสาย B2 ป้อนเข้าที่ขา 4 สำหรับ UHF ส่วนสาย A6 ที่ป้อนเข้าขา 5 นั้นเป็น RF AGC การจ่ายแรงดันเลี้ยงตัวจูนเนอร์ใช้แรงดัน +15V ป้อนเข้าที่ขา 6 และที่ขา 10 เป็นกราวด์

2.2 วงจรเอจีซี

2.2.1 หน้าที่ของเอจีซี

วงจรเอจีซีเป็นวงจรควบคุมอัตราขยายสัญญาณอย่างอัตโนมัติคล้ายๆ กับที่มีใช้อยู่กันในเครื่องรับวิทยุหน้าที่ของวงจรเอจีซีคือควบคุมระดับสัญญาณภาพทางเอาต์พุตให้มีขนาดมาตรฐานโดยการตรวจสอบขยายสัญญาณภาพแล้วสร้างแรงดันไปควบคุมบังคับให้ภาคขยายไอเอฟ

ภาพรวมและภาคขยายอาร์เอฟในจูนเนอร์ทำการขยายได้อย่างเหมาะสม ทั้งนี้เพราะสัญญาณภาพจะได้สัญญาณขนาดเล็กภาพทางหน้าจอจะจางบางสถานีมีสัญญาณแรงมากการขยายโอเวอร์โหด การดีเทกต์ภาพจะได้สัญญาณภาพขนาดใหญ่มาก ภาพทางหน้าจอดีมากเกิดอาการภาพคดและมีเสียงรบกวน จำเป็นต้องใช้วงจรเอจิสซีช่วยแก้ปัญหาเหล่านี้รูปที่ 2.5 แสดงบล็อกโคอะแกรมตำแหน่งเอจิสซี



รูปที่ 2.5 บล็อกโคอะแกรมตำแหน่งเอจิสซี

สรุปหน้าที่ของวงจรเอจิสซีได้ดังนี้

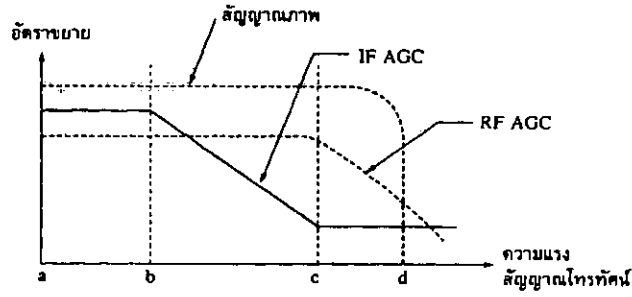
1. ควบคุมอัตราขยายของภาคขยายไอเอฟโดยอัตโนมัติตามความแรงของสัญญาณโทรทัศน์ที่รับเข้ามา
2. ควบคุมการขยายอาร์เอฟของสัญญาณโทรทัศน์ที่จูนเนอร์หลังจากภาคขยายไอเอฟไม่สามารถจะลดสัญญาณได้

2.2.2 หลักการทำงานของวงจรเอจิสซี

การทำงานของวงจรเอจิสซีมี 2 ขั้นตอนคือ

ขั้นตอนที่ 1 ไอเอฟเอจิสซีทำหน้าที่ตรวจสอบแอมพลิจูดของสัญญาณภาพให้ได้ระดับที่ต้องการ ถ้าต่ำกว่ามาตรฐานวงจรเอจิสซีก็จะปล่อยให้ภาคขยายไอเอฟภาพรวมและวงจร ขยายอาร์เอฟได้ขยายสัญญาณอย่างเต็มที่ดูกราฟช่วง a-b ในรูปที่ 2.6 ถ้าวงจรดีเทกเตอร์สัญญาณภาพขยายได้แอมพลิจูดสูงกว่ามาตรฐาน วงจรไอเอฟเอจิสซีจะบังคับให้วงจรอาร์เอฟในจูนเนอร์ยังคงปกติ ดูกราฟช่วง b-c ในรูปที่ 2.6

ขั้นตอนที่ 2 อาร์เอฟเอจิสซีหรือดีเลย์เอจิสซีจะรับรู้ระดับของแรงดันไอเอฟเอจิสซี ถ้าแรงดันไอเอฟเอจิสซีมีสูงถึงจุดหนึ่งหมายความว่าสัญญาณภาพยังแรงอยู่วงจร ไอเอฟไม่สามารถจะลดการขยายได้อีกต่อไป อาร์เอฟเอจิสซีจะรับหน้าที่คือโดยส่งแรงดันออกไปบังคับวงจรขยายอาร์เอฟในจูนเนอร์ให้ลดอัตราขยายลงจนสัญญาณภาพมีขนาดมาตรฐานตามที่ต้องการ ดูกราฟช่วง c-d ในรูปที่ 2.6

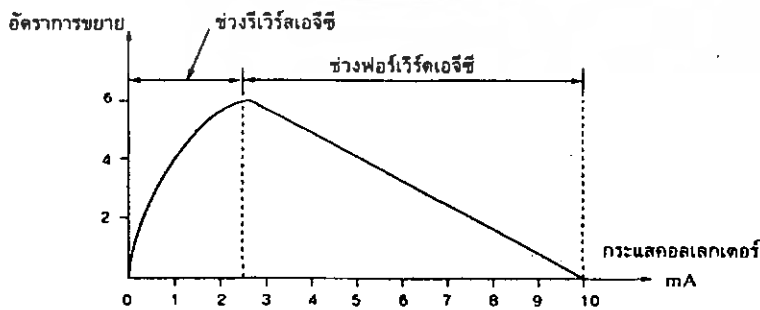


รูปที่ 2.6 กราฟการทำงานของเอจีซี

หลักการใช้แรงดันควบคุมการขยายของทรานซิสเตอร์ในวงจรขยายไอเอฟภาพและ วงจรขยายอาร์เอฟนั้นทำได้ 2 ลักษณะคือแบบแรกเป็นลักษณะรีเวิร์สเอจีซี (Reverse AGC) แบบที่สอง เป็นลักษณะฟอร์เวิร์ดเอจีซีดูรูปที่ 2.7 ประกอบแบบรีเวิร์สเอจีซีจะใช้แรงดันเอจีซีเป็นขั้วตรงข้าม ป้อนเข้าหากล้างให้ไบแอสที่ขาเบสของทรานซิสเตอร์ลดลง ซึ่งเป็นการลดกระแสคอลเลกเตอร์จะได้ อัตราขยายต่ำลงกระแสจะอยู่ในช่วง 0 – 2.5 mA มักใช้ในเครื่องรับวิทยุรีเวิร์สเอจีซีมีข้อเสียคือช่วงการ ควบคุมแคบและถ้าสัญญาณที่เข้ามาอ่อนมากแรงดันเอจีซีมีต่ำจะทำให้กระแสคอลเลกเตอร์ไหลมากเกินไป จุดยอดของกราฟอัตราขยายจะเริ่มลดลงเรื่อยๆ ซึ่งเป็นสิ่งที่ไม่ต้องการ

เมื่อพิจารณากระแสคอลเลกเตอร์ช่วงจาก 2.5 - 10 mA กราฟรูปที่ 2.7 จะเห็นว่าถ้า เพิ่มไบแอสให้กระแสคอลเลกเตอร์ไหลมากขึ้นอัตราขยายจะลดลง จะได้กราฟที่ลดลงค่อนข้างเป็น เส้นตรงและกว้างพอสมควร ดังนั้นถ้าตั้งจุดไบแอสปกติไว้ที่กระแสคอลเลกเตอร์ 2.5 mA ถ้าสัญญาณ แรงให้เพิ่มไบแอสขึ้นกระแสคอลเลกเตอร์จะไหลเพิ่มขึ้นได้อัตราการขยายลดลง หลักการนี้แรงดันเอจีซี ต้องเป็นขั้วชนิดเดียวกับแรงดันไบแอสที่ขาเบสของทรานซิสเตอร์ การนำแรงดันเอจีซีไปเพิ่มไบแอสเพื่อ ลดอัตราขยายนี้เรียกว่า ฟอร์เวิร์ดเอจีซี (Forward AGC) มีใช้อยู่ในเครื่องรับโทรทัศน์ทั่วไป

วงจรเอจีซีหรือวงจรควบคุมการขยายสัญญาณอัตโนมัติที่นิยมใช้ในเครื่องรับโทรทัศน์มี อยู่ 3 ชนิดคือ วงจรเอจีซีที่ทำงานตามค่าเฉลี่ยสัญญาณภาพ วงจรพีคเอจีซี (Peak AGC) และวงจรคีย์ เอ จีซี (Keyed AGC)



รูปที่ 2.7 กราฟการไหลของกระแสคอลเลกเตอร์กับอัตราขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วงจรขยายสัญญาณไอเอฟภาพรวม

2.3.1 หน้าที่ของวงจรขยายสัญญาณไอเอฟภาพรวม

วงจรขยายสัญญาณไอเอฟภาพรวม (VIF: Video Intermediate Frequency) ตั้งอยู่ระหว่างจูนเนอร์กับวงจรเทกเตอร์สัญญาณภาพมีหน้าที่ดังนี้

1. ขยายสัญญาณไอเอฟภาพและเสียงให้มีขนาดแอมพลิจูดของสัญญาณสูงขึ้นพอเพียงที่จะดีเทกต์ได้อย่างมีคุณภาพ โดยมีอัตราขยายประมาณ 2,000 – 3,000 เท่า

2. ขยายสัญญาณไอเอฟได้แถบกว้างประมาณ 7 เมกะเฮิร์ตซ์ เพื่อสัญญาณต่างๆ จะถูกขยายได้หมดและครอบคลุมความถี่พาหะภาพและเสียงที่ห่างกันอยู่ 5.5 เมกะเฮิร์ตซ์ให้ถูกขยายอย่างครบถ้วน อาจจะกว้างหรือแคบกว่า 7 เมกะเฮิร์ตซ์ก็ได้ขึ้นอยู่กับระบบโทรทัศน์

3. กำจัดความถี่พาหะภาพของช่องข้างเคียงด้านสูงไม่ให้เข้ามารบกวนระบบเสียง

4. กำจัดความถี่พาหะเสียงของช่องข้างเคียงด้านต่ำไม่ให้เข้ามารบกวนระบบภาพ

5. ลดความแรงของพาหะเสียงของช่องที่กำลังรับ (33.4 เมกะเฮิร์ตซ์) ให้มีอัตราส่วนพอเหมาะและไม่รบกวนสัญญาณภาพ

6. ขยายสัญญาณไอเอฟได้เอาต์พุตที่มีความแรงสม่ำเสมอโดยการควบคุมของแรงดันสัญญาณความถี่ที่ถูกขยายโดยวงจรขยายไอเอฟภาพรวม มีดังนี้

1. ความถี่สัญญาณไอเอฟเสียง (33.4 เมกะเฮิร์ตซ์)

2. ความถี่สัญญาณไอเอฟภาพ (38.9 เมกะเฮิร์ตซ์)

3. สัญญาณซิงก์แนวนอน (15,625 เมกะเฮิร์ตซ์)

4. สัญญาณซิงก์แนวตั้ง (50 เมกะเฮิร์ตซ์)

5. สัญญาณเบลิงกิ้ง (Blanking)

6. สัญญาณอีควอไลซิง (Equalizing)

การขยายสัญญาณไอเอฟภาพรวมสมัยก่อนจะใช้หลอดสัญญาณเป็นตัวยุขยและจูนกับปลิงด้วยมือแปลงไอเอฟ เมื่อเครื่องรับถูกพัฒนาเป็นทรานซิสเตอร์ช่วงแรกๆ ก็ยังใช้วิธีการจูนด้วยมือแปลงเช่นเดียวกัน เหตุที่ต้องใช้มือแปลงจูนก็เพราะต้องการจำกัดความถี่รบกวนจากปกติจะทำการขยาย 3 หรือ 4 ภาคติดต่อกันเครื่องรับในระยะหลังๆ จะใช้อาร์ชีปลิงสัญญาณหรือใช้ซอร์ฟิลเตอร์ (Saw Filter) และใช้ไอซีเป็นตัวขยายสัญญาณทางด้านอินพุตของภาคไอเอฟมีวงจรถ่าย (trap) เพื่อตัดความถี่ต่างๆ เหมือนกันเช่นแทรป 40.4 เมกะเฮิร์ตซ์ ตัดความถี่พาหะเสียงของช่องข้างเคียงด้านต่ำแทรป 31.9 เมกะเฮิร์ตซ์ สำหรับลดระดับความแรงของสัญญาณเสียงช่องที่กำลังรับให้มีแอมพลิจูดต่ำกว่าสัญญาณภาพ วงจรถ่ายอาจเป็นขดลวดกับตัวเก็บประจุหรือเป็นเซรามิกฟิลเตอร์ก็ได้

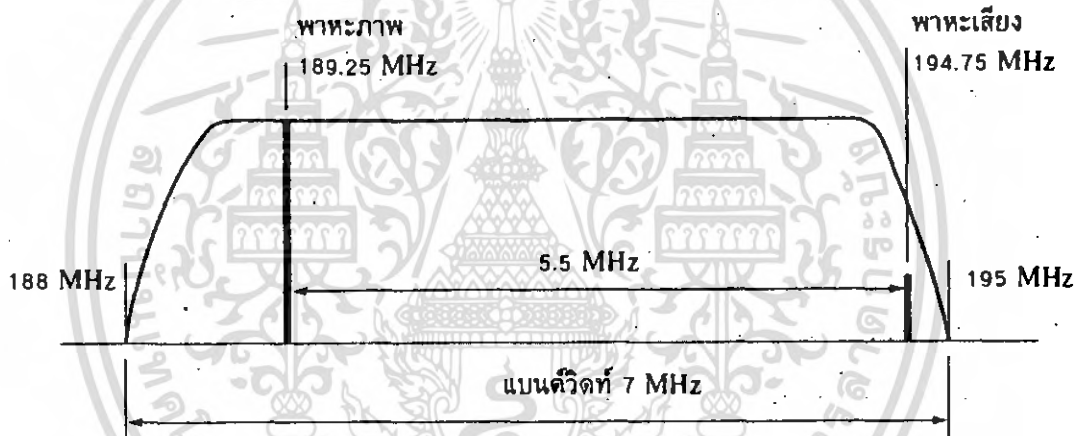
2.3.2 แบนด์วิดท์และการตอบสนองความถี่ของภาคขยายความถี่ไอเอฟภาพรวม

การรับสัญญาณโทรทัศน์เข้ามา 1 ช่องการขยายภายในจูนเนอร์ของเครื่องรับโทรทัศน์จะต้องมีแบนด์วิดท์กว้างมากพอให้สัญญาณรายละเอียดต่างๆ ผ่านไปได้อย่างครบถ้วน เช่น รับสัญญาณโทรทัศน์กองทัพบกช่อง 7 (กทม.) ขอบเขตของความถี่ที่รับเข้ามาทางจูนเนอร์อยู่ระหว่าง 195 – 188

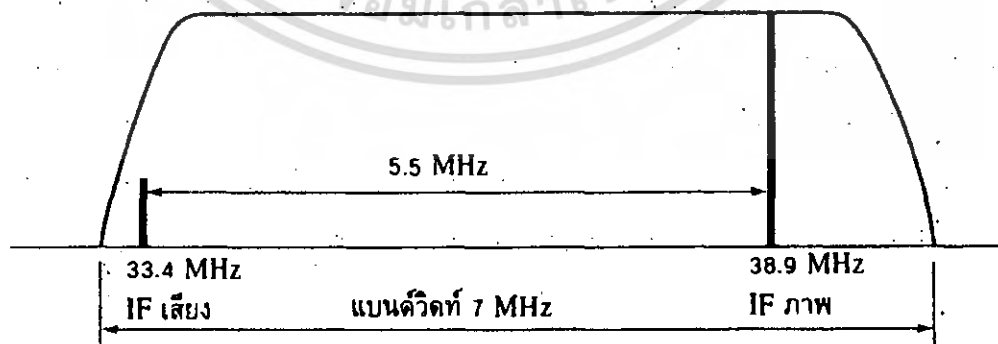
เมกะเฮิร์ตซ์ พหุเสียงมีความถี่ 194.75 เมกะเฮิร์ตซ์ เขียนสเปกตรัมความถี่ได้ดังรูปที่ 2.8 และขอบเขตความถี่สัญญาณไอเอฟภาพรวมในเครื่องรับโทรทัศน์แสดงดังรูปที่ 2.9

เมื่อสัญญาณโทรทัศน์ผ่านจูนเนอร์แล้วจะถูกแปลงเป็นความถี่ไอเอฟตามกระบวนการซูเปอร์เฮเทอโรไดน์โดยเมื่อรับสัญญาณทางช่อง 7 ออสซิลเลเตอร์จะกำเนิดคลื่นความถี่ภายใน 228.15 เมกะเฮิร์ตซ์ เมื่อทำการหักล้างกับความถี่พาหะภาพและพหุเสียงจะได้ไอเอฟภาพและไอเอฟเสียง โดยพาหะภาพจะแทนด้วย IF ภาพ 39.5 เมกะเฮิร์ตซ์ และพหุเสียงจะแทนด้วย IF เสียง 33.4 เมกะเฮิร์ตซ์ ดังความสัมพันธ์ต่อไปนี้

f_{osc}	-	$f_{พาหะภาพ}$	=	IF ภาพ
228.15	-	189.25	=	38.9 เมกะเฮิร์ตซ์
f_{osc}	-	$f_{พหุเสียง}$	=	IF เสียง
228.15	-	194.75	=	33.4 เมกะเฮิร์ตซ์

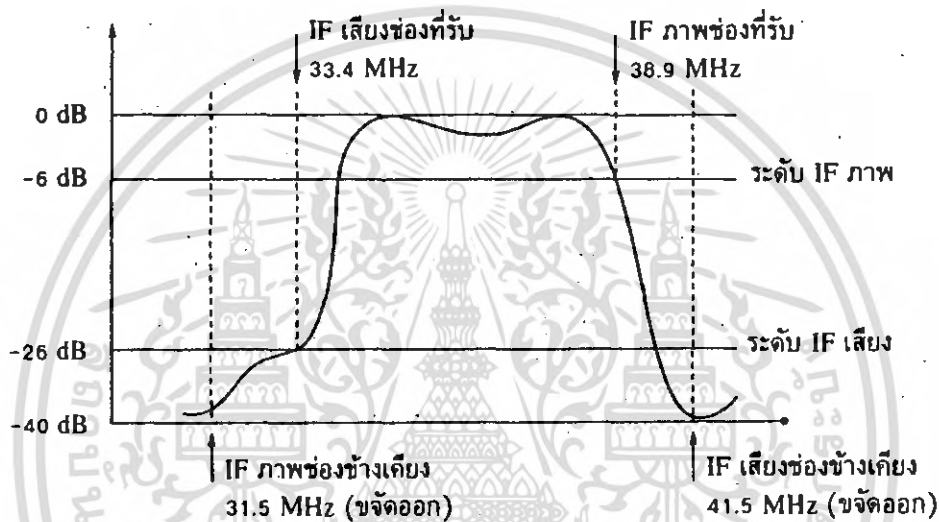


รูปที่ 2.8 สเปกตรัมสัญญาณโทรทัศน์ช่อง 7 ที่รับเข้าทางจูนเนอร์



รูปที่ 2.9 ขอบเขตความถี่สัญญาณไอเอฟภาพรวมในเครื่องรับโทรทัศน์

จึงเห็นได้ว่าสัญญาณภาพและสัญญาณเสียงโทรทัศน์จะผสมอยู่กับความถี่พาหะอันใหม่ ซึ่งเป็น IF และยังคงมีความห่างระหว่าง IF และยังคงมีความห่างระหว่าง IF ภาพกับ IF เสียง 5.5 เมกะเฮิร์ตซ์ แต่จะมีการสลับที่ระหว่างภาพและเสียงคือเมื่อเป็นสัญญาณ IF แล้วความถี่ IF ภาพจะสูงกว่าความถี่ IF เสียงเท่ากับ 5.5 เมกะเฮิร์ตซ์ ให้ลองเปรียบเทียบรูปที่ 2.8 กับรูปที่ 2.9 ส่วนการตอบสนองความถี่ที่ต้องการจริงของภาคขยายไอเอฟนั้นที่ความถี่ IF เสียงจะถูกลดทอนให้มีขนาดต่ำลงและบางความถี่ที่ไม่ต้องการต้องขจัดออกไป เส้นโค้งตอบสนองความถี่ของภาคขยายไอเอฟที่ต้องการแสดงดังรูปที่ 2.10



รูปที่ 2.10 เส้นโค้งตอบสนองความถี่ไอเอฟภาพรวมที่ต้องการ

2.4 วงจรขยายสัญญาณอาร์เอฟ (RF Amplifiers)

การใช้วงจรขยายสัญญาณอาร์เอฟในเครื่องรับซูเปอร์เฮเทอโรไดน์ จะเปลี่ยนแปลงจากการไม่ใช้เลยในการประยุกต์ใช้งานที่ไม่ต้องการ ไปจนถึง 3 หรือ 4 stage ในเครื่องรับที่ซับซ้อน

2.4.1 ประโยชน์ของการใช้วงจรขยายสัญญาณอาร์เอฟ

1. ทำให้สามารถกำจัดความถี่อิมเมจ (Image Frequency) ได้ดีขึ้น
2. มีอัตราขยายสูงขึ้น ทำให้มีความไวในการรับ (Sensitivity) สูงขึ้น
3. ปรับปรุงคุณสมบัติของสัญญาณรบกวน (Noise Characteristic)

วงจรมิกเซอร์ต้องการอุปกรณ์ที่ทำงานในบริเวณ non-linear เมื่อผลิตความถี่ที่แตกต่างที่ต้องการที่เอาต์พุตซึ่งจะมีสัญญาณรบกวนออกมามาก ด้วยการใช้วงจรขยายสัญญาณอาร์เอฟจะทำให้สัญญาณมีระดับเพิ่มขึ้นอย่างเหมาะสมและลดผลของสัญญาณรบกวนที่วงจรมิกเซอร์ไปได้

วงจรรขยายสัญญาณอาร์เอฟโดยปกติจะใช้เฟทที่เป็นอุปกรณ์ประเภทแอคทีฟ (Active Component) ประโยชน์ต่อไปนี้ของเฟทจะนำไปสู่การใช้ทั่วไปของวงจรรขยายสัญญาณอาร์เอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ประโยชน์ของเฟท

1. High Input Impedance จะไม่โหลดคความ (Load Down) ที่ Q ของวงจรรจุนที่มาก่อนถึงเฟทคังนั้นมันจะรักษาการเลือกระดับสัญญาณที่ระดับสูงสุดที่สามารถเป็นได้

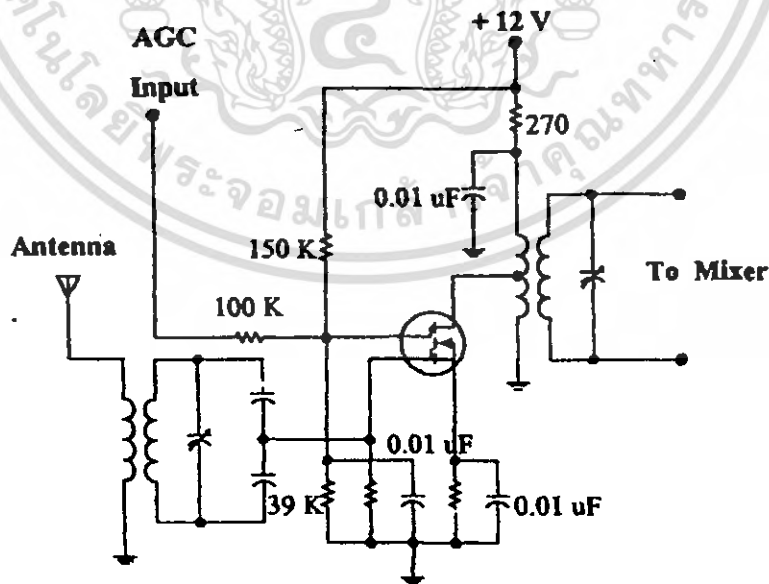
2. การมีเฟทชนิดเกทคู่ (Dual Gate Fet) อย่างเพียงพอนำไปใช้ได้กับ Isolate Injection Point ที่สัญญาณเอจิสซี

3. ความสัมพันธ์ของ Square - Law ของอินพุตและเอาต์พุตโดยจะยอมให้มีความผิดเพี้ยนต่ำความผิดเพี้ยนที่อ้างอิงที่เรียกว่า ครอสมอดูเลชัน (Cross - modulation)

ทำหน้าที่ในการรับสัญญาณ RF และทำการขยายให้มีขนาดสูงขึ้น เนื่องจากสัญญาณ RF ที่รับเข้ามาขอมจะมีสัญญาณรบกวนเข้ามาด้วยเสมอการขยายสัญญาณก็เพื่อไม่ให้สัญญาณ RF ก่อนที่จะส่งเข้าภาคมิกเซอร์ถูกรบกวนและถูกทำลายจากสัญญาณรบกวนที่มีค่าสูง ถ้าหากสัญญาณที่ต้องการรับมีค่าความถี่สูง คือ มากกว่า 1 กิกะเฮิรตซ์ (GHz) ขึ้นไปไม่จำเป็นต้องมีส่วน RF เพราะความถี่เหล่านี้มีความต้านทานของสัญญาณเพิ่มขึ้นในขณะที่เกน (Gain) มีค่าลดลง

วงจขยายสัญญาณอาร์เอฟยังจะช่วยในการลด Local Oscillator Reradiation เมื่อไม่มีวงจขยายสัญญาณอาร์เอฟแล้วสัญญาณของ Local Oscillator จะสะท้อนกลับมายังเสาอากาศรับและรบกวนการส่ง

ดังตัวอย่างของ MOSFET RF Amplifier แสดงในรูปที่ 2.11 วงจรนี้จะมี dual gate ที่ระดับของ AGC ประยุกต์ที่เกท 2 จะให้ค่าเกนที่เปลี่ยนแปลงได้อย่างอัตโนมัติ สัญญาณที่รับได้จากสายอากาศจะถูกป้อนผ่านวงจรรจุนคัปปลิง (Tune Coupling) ไปยังเกท 1 และการเชื่อมที่เอาต์พุตจะถูกรแยกออกเพื่อการคัปปลิงแต่ละเครือข่ายซึ่งรักษาอุปกรณ์จาก self-oscillation ไม่ต้องการ



รูปที่ 2.11 วงจร MOSFET RF แบบ Dual - Gate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 วงจรออสซิลเลเตอร์ (Oscillator Circuit)

ออสซิลเลเตอร์มีความหมายในตัวเองแปลว่าสิ่งที่แกว่งไปมา ในกรณีของวงจรอิเล็กทรอนิกส์ ออสซิลเลเตอร์มีความหมายถึงการเปลี่ยนไปมาของแรงดันหรือกระแสระหว่างค่าสูงสุดและค่าต่ำสุดหรืออาจกล่าวได้ว่าง่าย ๆ ก็คือ วงจรกำเนิดนั่นเอง

วงจรออสซิลเลเตอร์ส่วนใหญ่จะให้อัตราสัญญาณรูปไซน์ ดังนั้นวงจรออสซิลเลเตอร์จึงถูกนำมาใช้งานในวงจรเครื่องส่ง วงจรกำเนิดสัญญาณความถี่ต่างๆ และเป็นวงจรในส่วนประกอบย่อยของวงจรอื่นๆ อีกเป็นจำนวนมาก

การกำเนิดสัญญาณด้วยวงจรออสซิลเลเตอร์มีได้หลายแบบ ซึ่งแต่ละแบบก็จะมีลักษณะสำคัญพิเศษอื่นๆประกอบด้วย เช่น ให้ความถี่ที่ต่ำๆ ดี หรือให้ความถี่ที่สูงๆ ดี มีเสถียรภาพทางความถี่ที่ดี การปรับความถี่เป็นไปได้ยากหรือง่ายเป็นต้น

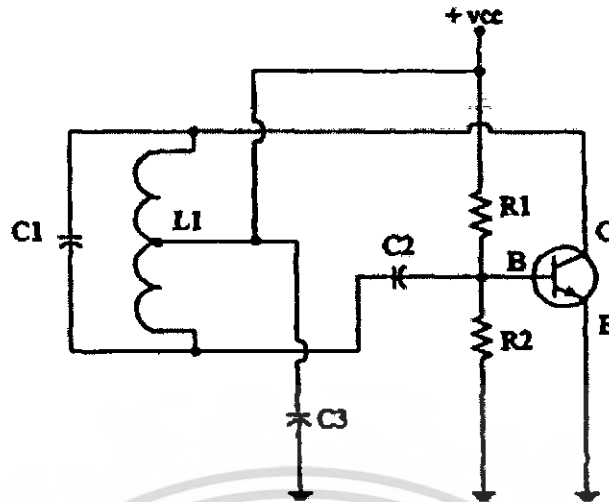
2.5.1 หลักการของวงจรออสซิลเลเตอร์

วงจรออสซิลเลเตอร์ก็คือวงจรขยายชนิดหนึ่งนั่นเองแต่เป็นวงจรขยายชนิดป้อนกลับแบบบวกสิ่งที่จะอธิบายในเรื่องการป้อนกลับแบบบวกให้เข้าใจคือ การพิจารณาในรูปของเฟสกับการป้อนกลับสัญญาณป้อนกลับจากเอาต์พุตจะกลับเฟส 180 องศา แล้วป้อนกลับเข้ามาทางอินพุตใหม่ ดังนั้นสัญญาณที่เข้ามาจะมีทิศทางเสริมกับสัญญาณป้อนกลับซึ่งถ้าเฟสที่ป้อนกลับเข้ามาไม่ตรงกับวงจรขยายก็จะไม่เกิดการออสซิลเลต ซึ่งปกติวงจรขยายสัญญาณทั่วไปสัญญาณเอาต์พุตจะมีเฟสต่างจากอินพุต 180 องศาอยู่แล้ว

การออสซิลเลตหรือการเปลี่ยนแปลงค่าแรงดันเอาต์พุตจากค่าต่ำสุดจนถึงค่าสูงสุดมักจะเกิดการเปลี่ยนในลักษณะคลื่นรูปไซน์ ความถี่ของลูกคลื่นสามารถคำนวณได้จากค่าคงตัวของอุปกรณ์ต่างๆ ภายในวงจร ในวงจรออสซิลเลเตอร์โดยทั่วไปที่ใช้งานกันอยู่จะมีการออสซิลเลตได้ด้วยตัวเองโดยไม่ต้องมีการป้อนสัญญาณอินพุตเข้าไปเลยแม้แต่น้อย ทั้งนี้เพราะสัญญาณที่ช่วยกระตุ้นให้เกิดการออสซิลเลตในขณะเริ่มต้นนั้นคือสัญญาณรบกวนที่มีอยู่ภายในอุปกรณ์นั่นเอง

วงจรออสซิลเลเตอร์ทั่วไปมักจะมีการออสซิลเลตที่ความถี่เดียว ทั้งนี้เพราะวงจรป้อนกลับที่สัญญาณจากเอาต์พุตมาซึ่งอินพุตมักจะเป็นวงจรที่ประกอบด้วยอุปกรณ์จำพวก R, L และ C ซึ่งอุปกรณ์จำพวก L และ C จะมีผลทางด้านเฟสนั่นคือจะมีความถี่ที่เป็นผลทำให้เฟสที่ได้กลับมาเสริมกับอินพุตเกิดการออสซิลเลตขึ้นตัวอย่างวงจรออสซิลเลเตอร์เบื้องต้นที่ใช้ทรานซิสเตอร์แสดงดังรูปที่ 2.12

วงจรออสซิลเลเตอร์ดังรูปที่ 2.12 เรียกอีกอย่างว่าวงจรฮาร์ทเลย์ออสซิลเลเตอร์การทำงานของวงจรอาศัยหลักการป้อนกลับด้วยวงจรรีโซแนนซ์ที่ประกอบด้วยตัวเก็บประจุ C₁ และตัวเหนี่ยวนำ L₁ สัญญาณป้อนกลับจากเอาต์พุตทางขาคอลเล็กเตอร์ของทรานซิสเตอร์จะป้อนกลับผ่านวงจรจูนกลับเฟส 180 องศาเข้ามาทางขาเบสของทรานซิสเตอร์



รูปที่ 2.12 วงจรออสซิลเลเตอร์เบื้องต้น

สำหรับทรานซิสเตอร์ตัวด้านทาน R_1 และ R_2 จะเป็นตัวไปทรานซิสเตอร์ให้ทำงานในลักษณะของวงจรขยายสัญญาณส่วนตัวเก็บประจุ C_2 เปรียบเสมือนเป็นตัวคัปปลิงสัญญาณเข้ามาทางอินพุต โดยปกติวงจรขยายชนิดขั้วอิมิตเตอร์ร่วมนั้นสัญญาณเอาต์พุตที่ได้ทางขาคอลเล็กเตอร์จะมีทิสเฟสตรงข้ามกับสัญญาณทางอินพุตอยู่แล้วก็มีทิสทางเฟสต่างกัน 180 องศา ผลของสัญญาณป้อนกลับจะมีทิสทางทางเสริมกับสัญญาณอินพุตทันทีวงจรก็จะออสซิลเลตได้

การทำงานในแต่ละรอบของการออสซิลเลตสามารถอธิบายได้ดังนี้ ในขณะที่เริ่มแรกของการออสซิลเลตจะมีการเปลี่ยนแปลงของกระแสคอลเล็กเตอร์ ซึ่งเป็นผลป้อนกลับมาทางอินพุตทางเบส กระตุ้นที่เบส โดยถ้ากระแสคอลเล็กเตอร์เพิ่มขึ้นทางค่ามากจะทำให้แรงดันที่คอลเล็กเตอร์ลดลง การป้อนกลับจะกระตุ้นเข้ามาที่เบสด้วยค่าแรงดันน้อยลงเป็นผลทำให้กระแสลดลงและแรงดันที่คอลเล็กเตอร์ของทรานซิสเตอร์จะสูงขึ้นและเป็นในลักษณะนี้เรื่อยไป

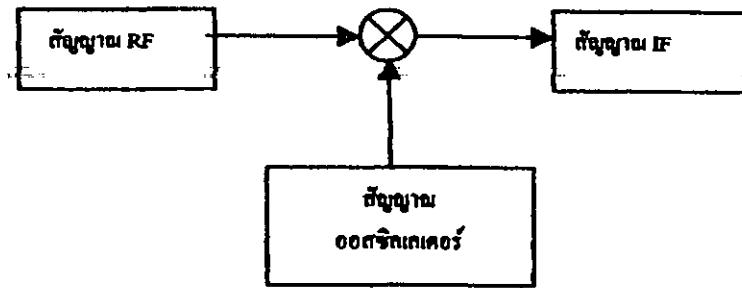
ตัวเก็บประจุ C_3 ของวงจรตามรูปที่ 2.12 นี้เป็นทางผ่าน (By pass) ของสัญญาณไฟสลับและในทำนองเดียวกันก็จะเป็นตัวกันแรงดันไฟโดยตรงเพื่อไม่ให้ลดลงกราวกับความถี่ของการออสซิลเลตของวงจรสามารถคำนวณได้จากค่ารีโซแนนซ์ของ L_1 และ C_1 ที่ต่อป้อนกลับสมการของการหาความถี่จะเป็นดังนี้

$$f_{osc} = \frac{1}{2\pi\sqrt{L_1 C_1}} \quad (2.2)$$

2.6 วงจรมิกเซอร์ (Mixer Circuit)

วงจรมิกเซอร์จะทำหน้าที่ผสมสัญญาณระหว่างสัญญาณความถี่สองสัญญาณเพื่อให้ได้สัญญาณความถี่ใหม่ขึ้นมา โดยที่ข้อมูลที่ผสมมาด้วยนั้นเปลี่ยนแปลงในส่วนของภาครับนี้มิกเซอร์จะทำหน้าที่ผสมสัญญาณระหว่างสัญญาณ RF และสัญญาณจากโวลลอสซิลเลเตอร์เพื่อให้ได้ความถี่กลางออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ตัวอย่างการทำงานของวงจรมิกเซอร์

ถ้าให้สัญญาณความถี่ที่รับเข้ามาเป็น $V_1 = v_1 \cos \omega_1 t$ (2.4)

สัญญาณความถี่จากออสซิลเลเตอร์เป็น $V_2 = v_2 \cos \omega_2 t$ (2.5)

และเอาต์พุตที่ได้เป็น

$$I_0 = A(V_1 + V_2)^2 \quad (2.6)$$

จะได้

$$\begin{aligned} I_0 &= A(v_1 \cos \omega_1 t + v_2 \cos \omega_2 t)^2 \\ &= Av_1^2 \cos^2 \omega_1 t + 2Av_1 v_2 \cos \omega_1 t \cos \omega_2 t + Av_2^2 \cos^2 \omega_2 t \end{aligned} \quad (2.7)$$

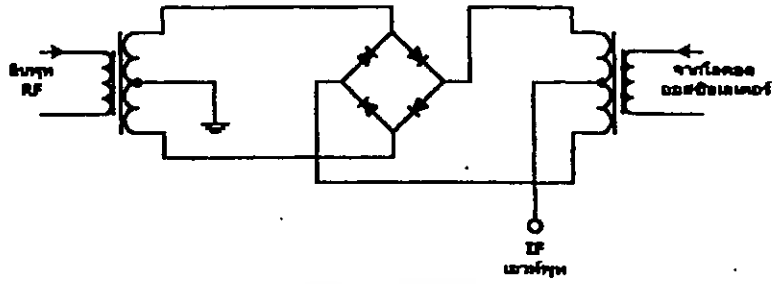
พิจารณาเทอมที่สอง

$$\cos \omega_1 t \cos \omega_2 t = \frac{1}{2} \{ \cos(\omega_1 + \omega_2)t + \cos(\omega_1 - \omega_2)t \} \quad (2.8)$$

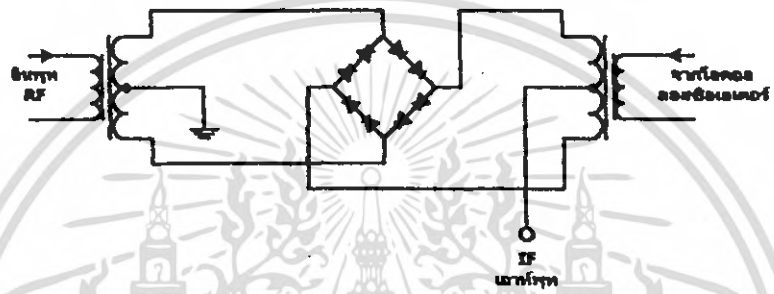
จากสมการที่ 2.8 จะเห็นว่าผลของการผสมสัญญาณนั้นจะทำให้เกิดสัญญาณความถี่ที่น่าสนใจสองความถี่คือ ความถี่ที่เป็นผลบวกและความถี่ที่เป็นผลลบระหว่างความถี่ที่รับเข้ากับความถี่ที่ผลิตขึ้นมาจากออสซิลเลเตอร์

วงจรมิกเซอร์แบ่งได้เป็น 2 ประเภทคือ ประเภทแอคทีฟ (Active) ใช้ทรานซิสเตอร์หรือไอซี รวมทั้งอุปกรณ์อื่นๆที่ให้อัตราขยายในการผสมคลื่นและประเภทพาสซีฟ (Passive) ใช้ไดโอด ซึ่งไม่มีการขยายสัญญาณนอกจากนี้ยังแบ่งวงจรมิกเซอร์ออกได้เป็น 2 ประเภท คือ แบบสมดุลหรือบาลานซ์กับแบบไม่สมดุลหรืออิมบาลานซ์ วงจรมิกเซอร์แบบสมดุลนี้ต้องวงจรให้ขั้วอินพุตหรือเอาต์พุตของวงจรมิกเซอร์ไม่เกิดปฏิกิริยาซึ่งกันและกันสัญญาณไม่เล็ดลอดระหว่างขั้วคุณสมบัตินี้เรียกว่า การแยกระหว่างขั้วหรือไอโซเลชัน (Isolation) ขั้วอินพุตของวงจรมิกเซอร์มี 2 ขั้วคือ สัญญาณ RF กับสัญญาณออสซิลเลเตอร์และมีขั้วเอาต์พุต 1 ขั้ว คือ สัญญาณ IF พิจารณารูปที่ 2.14 จะเห็นว่าการแยกระหว่างขั้ว RF และขั้วออสซิลเลเตอร์จะช่วยให้สัญญาณออสซิลเลเตอร์ย้อนกลับสู่สายอากาศแผ่กระจายคลื่นออกไปได้และการแยกระหว่างขั้ว RF กับ IF จะช่วยให้สัญญาณที่มีความถี่พอดิกับความถี่ IF เล็ดลอดเข้าไปสู่วงจรมิกเซอร์ IF

สำหรับวงจรมิกเซอร์แบบไม่สมดุลนั้น มีคุณสมบัติการแยกแยะระหว่างขั้วไม่ดีเหมือนกับแบบสมดุล ถ้าต้องการให้มีการแยกสัญญาณดีต้องใช้ฟิลเตอร์ช่วยในการกรองสัญญาณต่างหาก



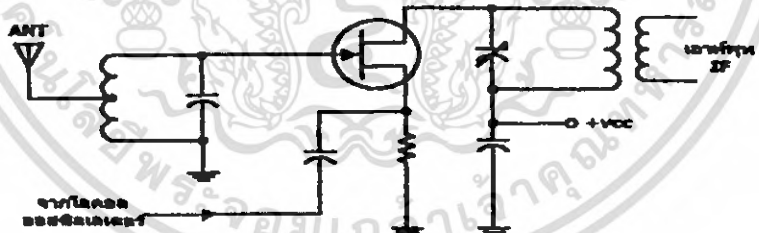
(ก)



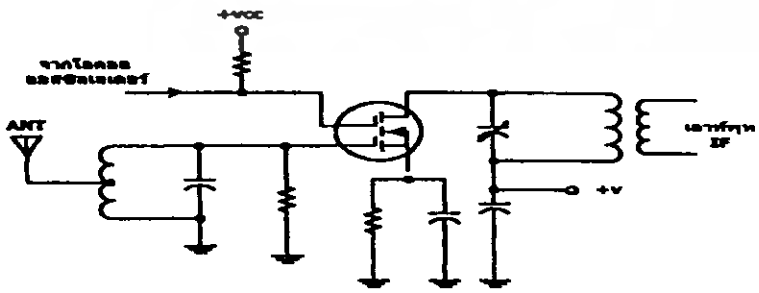
(ข)

รูปที่ 2.14 วงจรบาลานซ์มิกเซอร์ชนิดพาสซีฟ

ตัวอย่างวงจรอับบาลานซ์มิกเซอร์



(ก)



(ข)

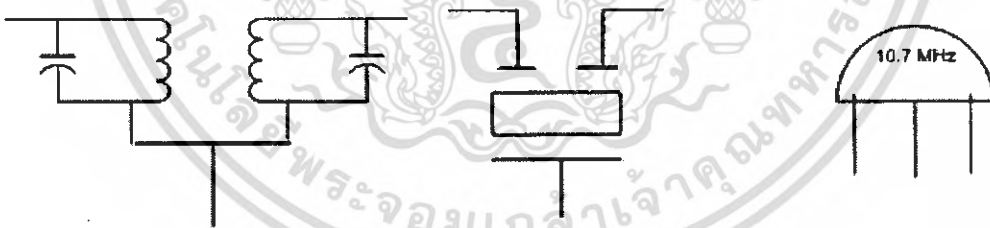
รูปที่ 2.15 วงจรมิกเซอร์แบบไม่สมดุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ MOSFET คุณสมบัติของวงจรคือมีการแยกแยะระหว่างขั้วออสซิลเลเตอร์กับสายอากาศ ก่อนข้างดีแต่ระหว่างขั้ว RF และขั้ว IF ไม่ค่อยดีจำเป็นต้องใช้ฟิลเตอร์ช่วยกรองความถี่เพื่อกำจัด สัญญาณ RF มิให้เล็ดลอดเข้าสู่ขั้วของ IF ได้

2.7 เซรามิกฟิลเตอร์ (Ceramic Filter)

เซรามิกฟิลเตอร์เป็นอุปกรณ์ที่สร้างขึ้นมาสำหรับใช้แทนหม้อแปลงความถี่สูงซึ่งมีความถี่เฉพาะ เช่น หม้อแปลงความถี่กลางของเครื่องรับวิทยุและของเครื่องรับโทรทัศน์ เซรามิกฟิลเตอร์ทำจากสารพวกเพียโซอิเล็กทริกซึ่งจะมีค่าเรโซแนนซ์เฉพาะตัว เมื่อป้อนสัญญาณเข้าไปในเซรามิกฟิลเตอร์สัญญาณที่ผ่านได้คือความถี่ที่เรโซแนนซ์กับตัวมันเท่านั้น หากสัญญาณที่ป้อนเข้าไปมีความถี่ไม่ถูกต้องตามค่าเรโซแนนซ์ทางด้านเอาต์พุตจะไม่มีสัญญาณออกมา ตัวอย่างความถี่ของเซรามิกฟิลเตอร์เช่น 455 กิโลเฮิร์ตซ์ สำหรับภาคความถี่กลางของเครื่องรับสัญญาณเอเอ็ม, ความถี่ 10.7 เมกะเฮิร์ตซ์ ของเครื่องรับสัญญาณเอฟเอ็มและความถี่อื่นๆ ที่ใช้ในเครื่องรับสัญญาณโทรทัศน์ เช่น 5.5 เมกะเฮิร์ตซ์ 4.5 เมกะเฮิร์ตซ์ 38.9 เมกะเฮิร์ตซ์ เป็นต้น เซรามิกฟิลเตอร์มีขนาดเล็กสามารถใช้แทนหม้อแปลงที่เชื่อมโยงเฉพาะความถี่ได้โดยตรงทำให้สะดวกและประหยัดเวลาได้มาก เนื่องจากไม่ต้องถูกจูนหรือปรับแต่งความถี่แต่เซรามิกฟิลเตอร์เป็นอุปกรณ์พาสซีฟขยายสัญญาณไม่ได้ ดังนั้นสัญญาณที่ใส่เข้าไปจะสูญเสียภายในเซรามิกฟิลเตอร์ (Insert Loss) สัญญาณเอาต์พุตที่ออกมาจะถูกบั่นทอนลงไปประมาณ 30 เปอร์เซ็นต์ เซรามิกฟิลเตอร์แบบ 3 ขาใช้สำหรับส่งผ่านสัญญาณจะเรียกว่า ตัวคัปปลิงฟิลเตอร์ อีกแบบหนึ่งมี 2 ขา เป็นตัวบายพาสฟิลเตอร์เพื่อการบายพาสความถี่ที่ไม่ต้องการทิ้ง



(ก) หม้อแปลงเรโซแนนซ์ (ข) สัญลักษณ์เซรามิกฟิลเตอร์ (ค) รูปร่างเซรามิกฟิลเตอร์

รูปที่ 2.16 หม้อแปลงเรโซแนนซ์และเซรามิกฟิลเตอร์

2.8 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ได้ถูกนำมาใช้งานอย่างกว้างขวางในงานด้านต่างๆ เช่น

- เครื่องใช้ในบ้าน เช่น เครื่องซักผ้า เตาอบไมโครเวฟ กล้องถ่ายรูป
- ในสถานที่ทำงาน เช่น FAX, Laser Printer
- ในรถยนต์ เช่น Air Bag, ABS และกล่องควบคุมหัวฉีด

2.8.1 ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์

ไมโครโปรเซสเซอร์ หมายถึง General-Purpose Microprocessor เช่น Z-80, 8086, 80286, 80486 และ Pentium ซึ่งทั้งหมดที่กล่าวมาแล้วนี้ภายในชิปไม่มี RAM, ROM, I/O Port, Timer, Serial Port

ซึ่งเกิลชิปไมโครคอนโทรลเลอร์ คือการนำเอา RAM, ROM, I/O Port, Timer, Serial Port มาใส่เข้าไปในชิป ทำให้ขนาดเล็กลงมากมีผู้ผลิตออกมาหลายราย เช่น Intel, Atmel, Philips, Siemens, Dallas Semiconductor

2.8.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยบริษัทอินเทล

- ต้องการแหล่งจ่ายไฟ + 5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์ สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไป มีถึง 256 ไบต์
- หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทม์เมอร์และเคาน์เตอร์ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รีจิสเตอร์รพีได้ 6 แหล่ง 5 เวกเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปแบบได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต

2.8.2 หน่วยความจำภายในชิป

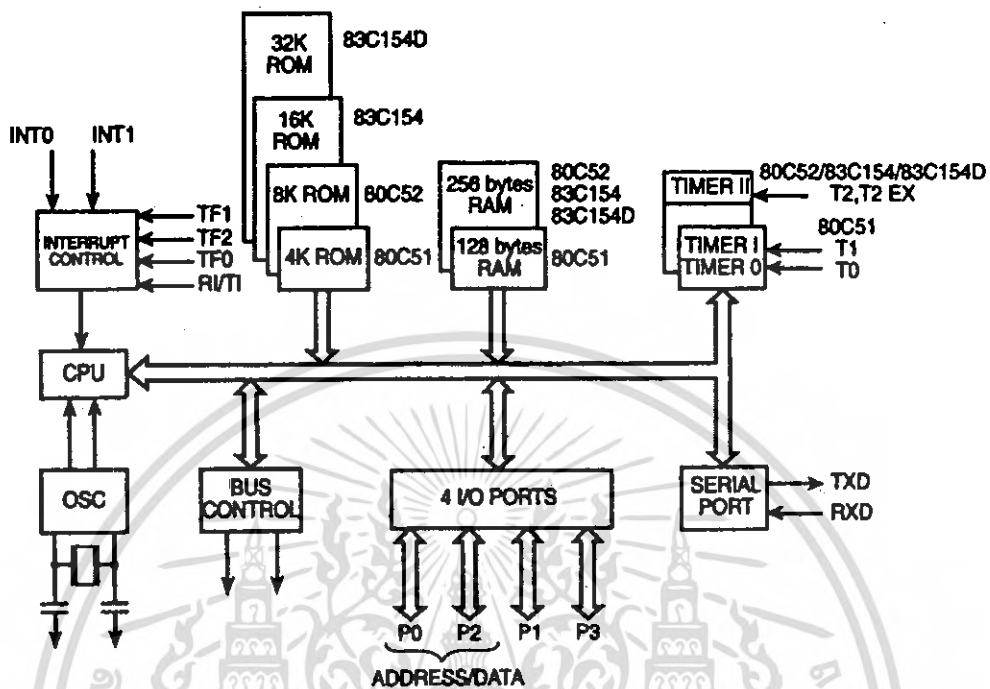
แบ่งได้เป็น 2 ชนิดคือ

- Program Memory (4 Bytes ROM สำหรับเบอร์ 8051)
- DATA Memory (128 Bytes RAM สำหรับเบอร์ 8051)

ข้อเสียของ 8051 คือ ROM ภายในชิปต้องโปรแกรมข้อมูลจากโรงงานโดยจะต้องผลิตครั้งละมากๆ

ข้อดีของ 89C51 คือ เปลี่ยนจาก ROM ภายในชิปเป็น 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM) สามารถลบและโปรแกรมใหม่ได้พันครั้ง ปัจจุบันได้ผลิตขึ้นมาหลายเบอร์ ดังแสดงในตารางที่ 2.1

2.8.2.1 โครงสร้างภายในของ 8051



รูปที่ 2.17 โครงสร้างภายใน 8051

คุณสมบัติของไมโครคอนโทรลเลอร์ โดยบริษัท Atmel

ตารางที่ 2.1 รุ่นของ 8051 จาก ATMEL (All ROM Flash)

Part Number	ROM	RAM	I/O pins	Timer	Interrupt	V _{cc}	Packaging
AT89C51	4K	128	32	2	6	5V	40
AT89LV51	4K	128	32	2	6	3V	40
AT89C1051	1K	64	15	1	3	3V	20
AT89C2051	2K	128	15	2	6	3V	20
AT89V52	8K	128	32	3	8	5V	40
AT89LV52	8K	128	32	3	8	3V	40

Notes: "C" in the part number indicates CMOS.

"LV" in the part number indicates Low Voltage

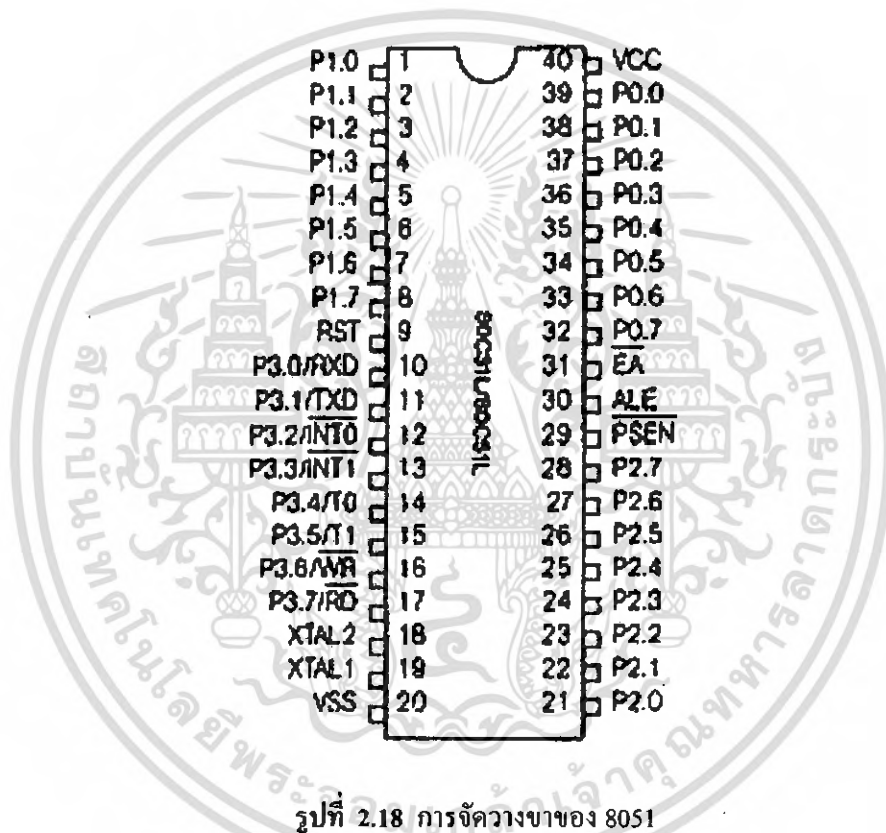
89C51 มีความเร็วของซีพียู 20, 16, 12 เมกะเฮิร์ตซ์ ดังแสดงในตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 Various Speeds of 8051 From Atmel

Part Number	Speed	Pins	Packaging	Use
AT89C51-12PC	12 MHz.	40	Dip Plastic	Commercial
AT89C51-16PC	16 MHz.	40	Dip Plastic	Commercial
AT89C51-20PC	20 MHz.	40	Dip Plastic	Commercial

การใช้โปรแกรม 89 C51 จะต้องควบคุมผ่านทาง P2.6, P2.7, P3.3, P3.6 และ P3.7 ส่วนข้อมูลป้อนผ่าน P0 8 บิต (และ P1, P2 แบบนี้เรียกว่าแบบ Parallel Load)



รูปที่ 2.18 การจัดวางขาของ 8051

2.8.3 พอร์ตของ 8051

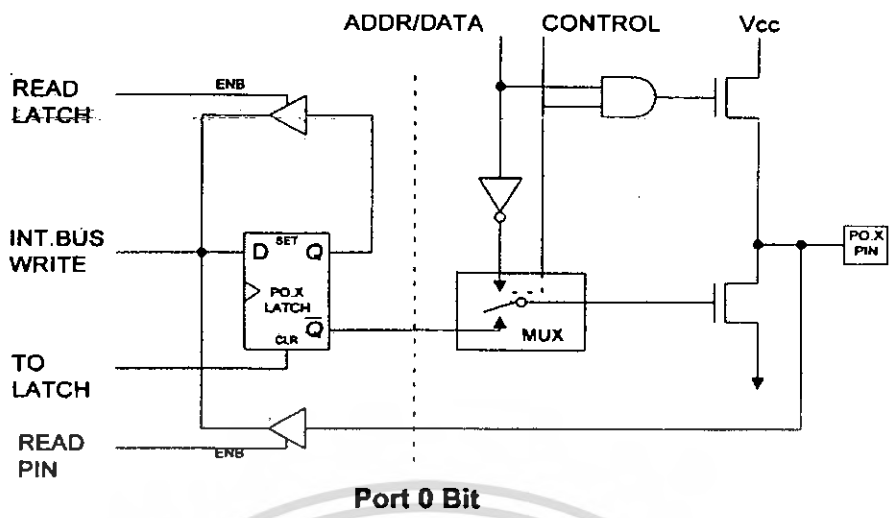
8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ ดังนี้

- Vc (ขา 40) ต่อกับ +5V
- Vss (ขา 20) เป็นขา GND
- พอร์ต 0 (ขา 32 – 39) มีทั้งหมด 8 บิต คือ (P0.7 - P0.0) มีโครงสร้างแบบ Open

- Drain

Bi - directional ไม่มีความต้านทานภายในพูล์อัพภายในดังแสดงในรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



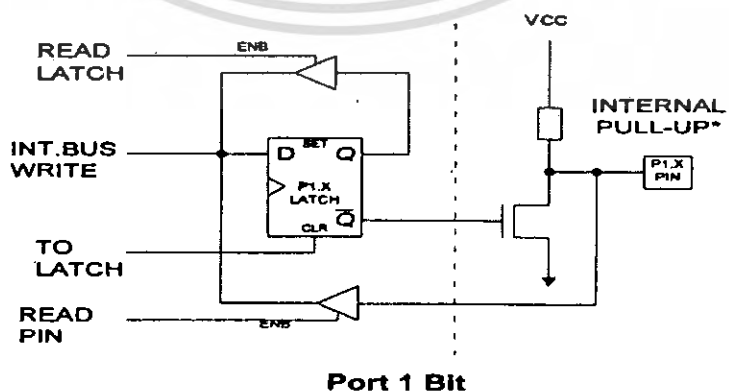
รูปที่ 2.19 แสดงโครงสร้างพอร์ต 0 (บิต)

พอร์ต 0 ขา (32 – 39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ 2 หน้าที่ คือเป็นทางผ่านของแอดเดรสบัสและดาต้าบัส เมื่อต้องการติดต่อกับหน่วยจำภายนอกหรือเป็นไอโอพอร์ต ถ้าต้องการให้ทำงานเป็นอินพุตพอร์ตต้องส่งลอจิก “1” ไปยังพอร์ตนี้ จะมีผลให้ Q ของ D-FF เป็น “0” ทำให้เฟลตั่วกลางมีสถานะ OFF สัญญาที่ใช้อ่านอินพุตพอร์ตเลขชี้โดยส่งสัญญา READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (pin) จะใช้สัญญา Read (pin) พอร์ตนี้ไม่มีควมต้านทานพูลอัฟภายใน

พอร์ต 1 (ขา 1 – 8) มีทั้งหมด 8 บิต คือ (P1.0 - P1.7) มีโครงสร้างคล้ายพอร์ต 0 แต่จะใช้ควมต้านทานพูลอัฟภายในแทน (Internal pull - up) มีโครงสร้างดังรูปที่ 2.20

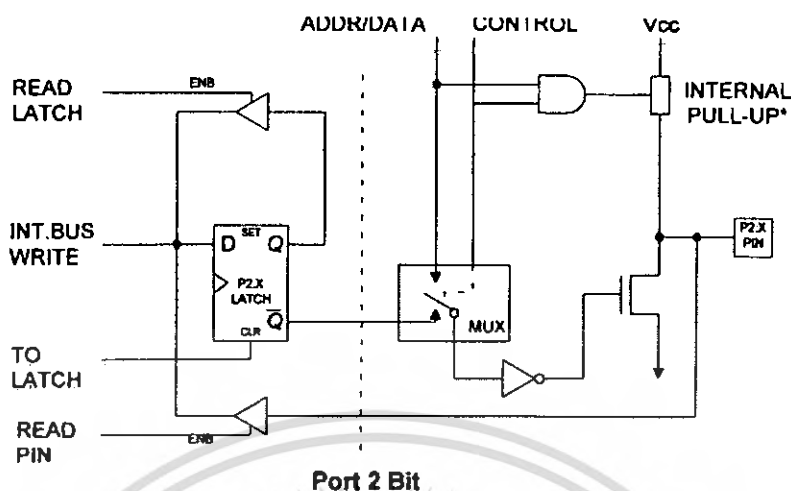
พอร์ต 2 (ขา 21 – 28) มีทั้งหมด 8 บิต คือขา (P2.7 - P2.0) มีโครงสร้างคล้ายพอร์ต 0 โดยมีเฟลตั่วกลางตัวเดียวส่วนด้านบนใช้ควมต้านทานพูลอัฟแทน (Internal Pull Up) พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15 - A8) และเป็นไอโอพอร์ตใช้งานได้ทั่วไปเมื่อจะใช้งานเป็นอินพุตพอร์ตต้องส่งลอจิก “1” มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ เฟลตั่วอยู่ในภาวะ off ดังแสดงในรูปที่

2.20



รูปที่ 2.20 โครงสร้างของพอร์ต 1 (บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

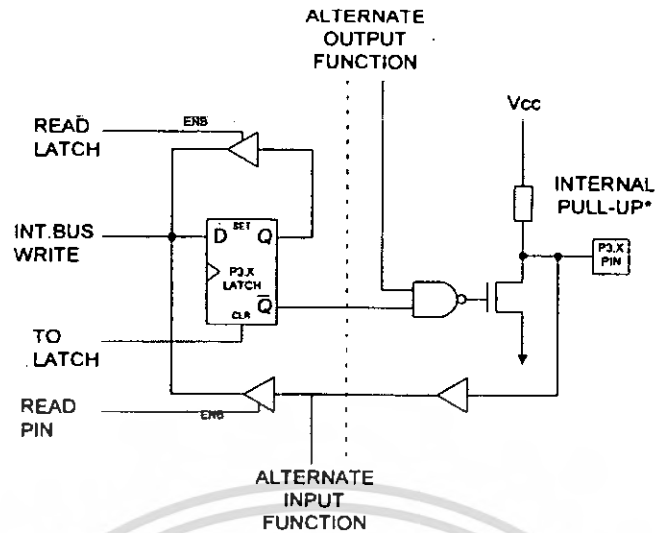


รูปที่ 2.21 โครงสร้างของ พอร์ต 2 (บิต)

พอร์ต 3 (10 - 17) มีทั้งหมด 8 บิต คือ ขา (P3.7 - P3.0) มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ต ถ้าจะโปรแกรมให้เป็นอินพุตพอร์ตต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งก็คือส่งสัญญาณควบคุมออกมาและรับสัญญาณ เข้าไปสัญญาณต่างๆ มีดังนี้

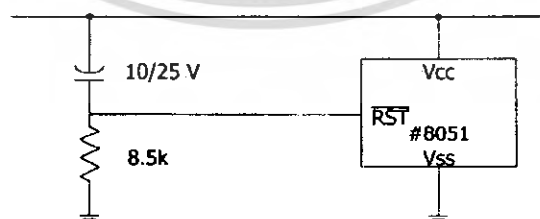
- P3.0/RXD (Serial Input Port) : เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)
- P3.1/TXD (Serial Output Port) : เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)
- P3.2/INT0 (External Interrupt 0) : ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
- P3.3/INT1 (External Interrupt 1) : ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1
- P3.4/T0 (Counter 0 External Input) : ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 0 (เป็นอินพุตโหมดเคาน์เตอร์)
- P3.5/T1 (Counter 1 External Input) : ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 1 (เป็นอินพุตโหมดเคาน์เตอร์)
- P3.6/ WR (External Data Memory Write Strobe) : ขารับสัญญาณใช้ในการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก
- P3.7/ RD (External Data Memory Read Strobe) : ขาสัญญาณใช้ในการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 โครงสร้างของพอร์ต 3 (บิต)

- ALE (ขา 30) : เป็นขาส่งสไตรบสำหรับใช้ในการแลคซ์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอกทีฟทุกๆ 2 ครั้งใน 1 แมกซ์ซินไซเคิล
- PSEN (ขา 29) : เป็นขาสไตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอกสัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมกซ์ซินไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขา
- EA (ขา 31) : ใช้เลือกหน่วยความจำโปรแกรมภายนอก
ป้อน "0" จะอ่านโปรแกรมจากภายนอกชิพ
ป้อน "1" จะอ่านโปรแกรมจากภายในชิพ
- RST (ขา 9) : ขารีเซ็ตจะรีเซ็ตได้ก็ต่อเมื่อป้อนลอจิก "1" เข้าที่ขานี้ นานอย่างน้อย 2 แมกซ์ซินไซเคิล
- XTAL1 (ขา 19) : ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรถอดสซีสเลเตอร์ภายใน
- XTAL 2 (ขา 18) : ใช้ต่อคริสตอลภายนอกโดยเป็นเอาต์พุตของวงจรถอดสซีสเลเตอร์ภายใน



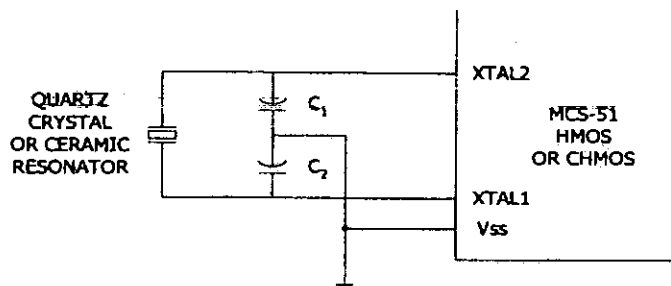
รูปที่ 2.23 การต่อขารีเซ็ตให้กับ 8051

2.8.4 วงจรคล็อกของ MCS-51

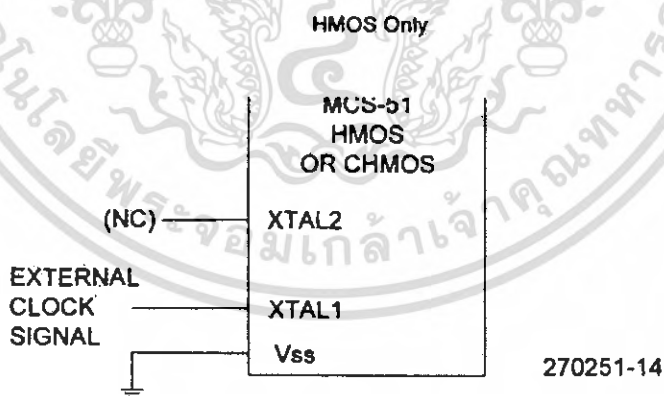
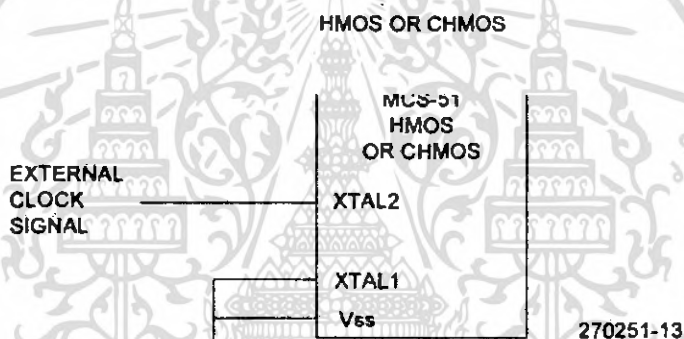
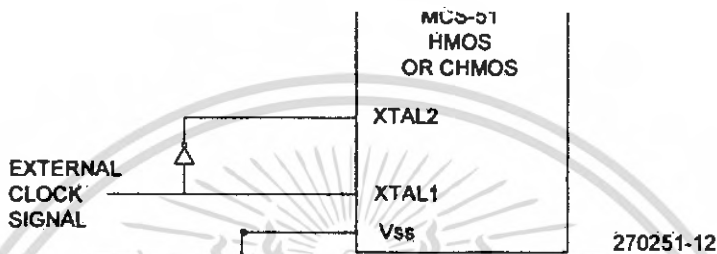
การต่อมีอยู่ด้วยกัน 2 รูปแบบคือแบบใช้คล็อกภายในและคล็อกจากภายนอกมีรูปแบบ

การต่อคังรูป 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) Using the on-chip Oscillator



CMOS Only

(ข) Using the External Clock

รูปที่ 2.24 วงจรสร้างค็ลคของ 8051

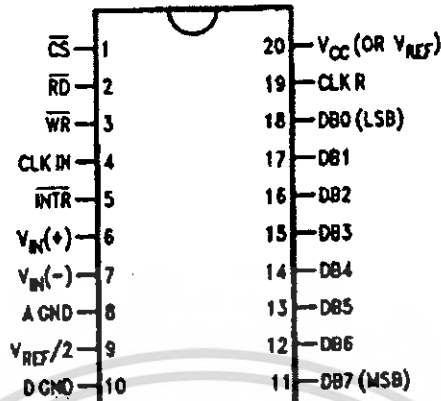
2.8.5 การแปลงสัญญาณอนาลอกเป็นดิจิตอล

ADC 0804 เป็น A/D ขนาด 8 บิต มีความเร็วในการแปลงสัญญาณ 100 s (Conversion

Time = 100 s) มีรายละเอียดและการจัดวางของ ADC 0804 ดังรูปที่ 2.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

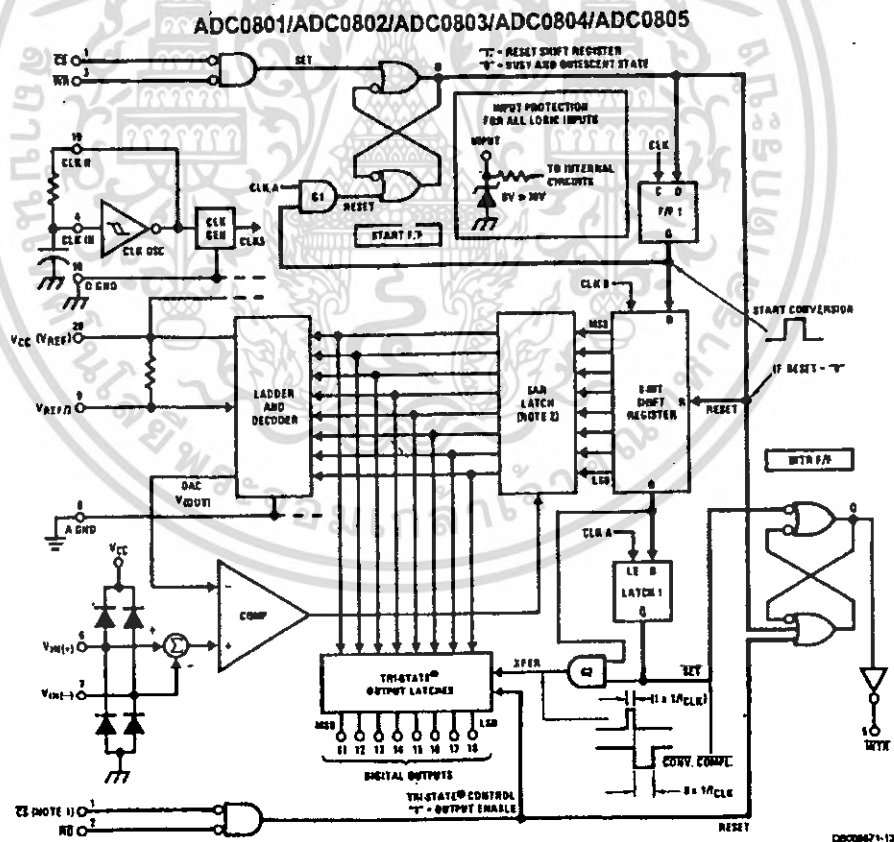
ADC080X Dual-In-Line and Small Outline (SO) Packages



DS005871-30

See Ordering Information

รูปที่ 2.25 รายละเอียดและการจัดวางขาของ ADC 0804



Note 13: CS shown twice for clarity.
 Note 14: SAR - Successive Approximation Register.

FIGURE . Block Diagram

รูปที่ 2.26 ผังโครงสร้างภายในของ ADC 0804

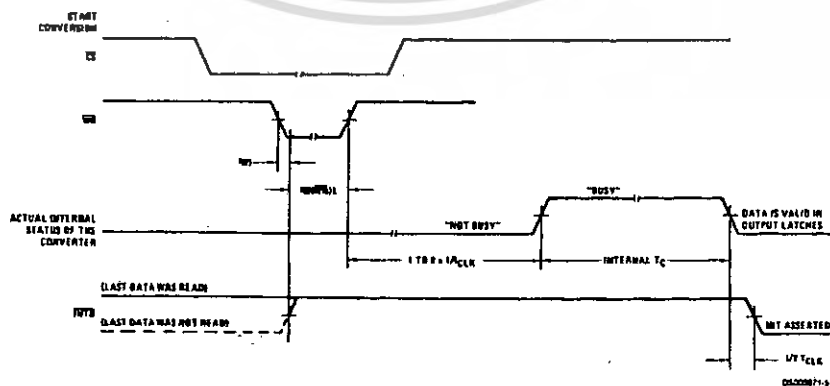
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ขาต่างๆ ของ ADC 0804 มีดังนี้

ขาต่างๆ	ความหมายและการทำงาน
D7-D0	ขาเอาต์พุต 8 บิต
CS	ขาเลือกชิพต้องป้อนด้วย Low
VCC	ขาไฟเลี้ยง 5V
DGND	Digital Ground
AGND	Analog Ground
V _{in+} ถึง V _{in-}	2 ขานี้เป็น Differential Analog Input โดยที่ $V_{in} = V_{in(+)}$ หรือต่อลง ground และต่อ $V_{in(-)}$ กับสัญญาณที่ต้องการ
CLK R และ CLK IN	2 ขานี้จะต้องต่อเข้ากับ R และ C เพื่อทำให้เกิดการอออสซิลเลตภายในค่าความถี่คำนวณจาก $f = (1/1) RC$ ถ้าใช้ $R = 10\text{ k}$ และ $C = 150\text{ pF}$ จะอออสซิลเลตที่ความถี่ 606 กิโลเฮิรตซ์ทำให้ค่า Conversion Time ได้เท่ากับ 110 s
WR	ใช้ Start A/D เพื่อเริ่มการ conversion ขานี้ต้องกระตุ้นด้วย Low ขานี้มักจะต่อ WR ของ CPU เพื่อง่ายในการออกแบบ
RD	ใช้อ่านข้อมูลจาก A/D ไปเก็บโดยต้องป้อนด้วย Low ขานี้มักจะต่อกัน RD ของ CPU เพื่อง่ายในการออกแบบ
V _{ref/2}	ขานี้กำหนดช่วงกว้างแรงดันอินพุตที่เข้า A/D (Span ADJ) ถ้าปล่อยลอย ก็จะรับอินพุตช่วงกว้างถึง 5 V ถ้าป้อน 2V จะรับอินพุตกว้างถึง 4V ถ้าป้อน 1V จะรับอินพุตกว้างถึง 2V

AGND และ DGND ก็คือ Analog Ground และ Digital Ground มีจุดประสงค์เพื่อแยกสัญญาณรบกวนทางอนาลอกไม่ให้ไปกวนทางดิจิตอลเอาต์พุต

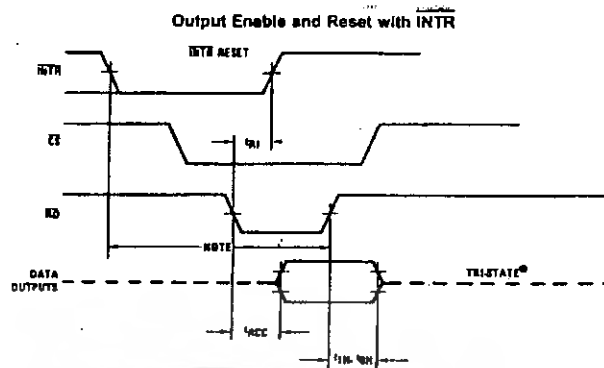
Timing Diagrams (All timing is measured from the 50% voltage points)



รูปที่ 2.27 ผังเวลาการสแตร์ท A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing Diagrams (All timing is measured from the 50% voltage points) (Continued)



รูปที่ 2.28 ผังเวลาการอ่านข้อมูลไปเก็บ

วิธีสตาร์ท A/D ทำได้โดยส่งสัญญาณเข้าขา WR และ CS ด้วยสัปดาห์ (Low) แล้วให้รออีกประมาณ 100 μ s หลังจากนั้น A/D จะส่งสัญญาณออกมาที่ขา INTR (คือสัญญาณที่บอกว่าแปลงสัญญาณเสร็จแล้ว) มีระดับสัญญาณเป็น Low ขานี้มักจะต่อกับขา INTR, INT1 ของ CPU เพื่อบอกให้ CPU ทำขบวนการอินเตอร์รัพท์เพื่อนำข้อมูลที่แปลงแล้วไปเก็บในการอ่านข้อมูลไปเก็บ CPU ต้องส่งสัญญาณ RD มาเข้า RD ของ A/D (โดยเข้า CS ต้องเป็น low อยู่ก่อนหน้า) รายละเอียดช่วงการ Start และอ่านข้อมูลไปเก็บ ดังแสดงในแผนผังเวลาตามรูปที่ 2.27 และ 2.28 ตามลำดับ

2.8.6 การเชื่อมโยง ADC 0804 เข้ากับ MCS-51

การเชื่อมต่อ ADC 0804 เข้ากับ MCS-51 ทำได้ 2 รูปแบบ

1. ติดต่อบนแบบ I/O Port
2. ติดต่อบนแบบ Memory Map

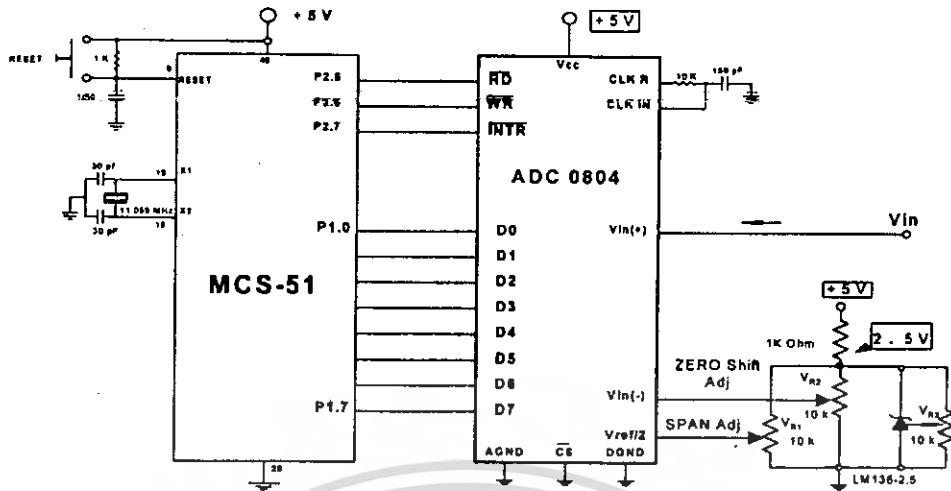
การติดต่อบนแบบ I/O Port

สามารถต่อ ADC 0804 เข้ากับ MCS-51 ผ่านทาง Port 0, Port 1, Port 2, Port 3 เราสามารถใช้พอร์ตไหนเป็น DATA Port ก็ได้ส่วนพอร์ตที่เหลือเป็นสัญญาณควบคุม

การเชื่อมต่อ ADC 0804 เข้ากับ MCS-51 โดยกำหนดให้

ขา DATA	ต่อเข้ากับขา Port 1 (P1)
ขา RD	ต่อเข้ากับขา P2.5
ขา WR	ต่อเข้ากับขา P2.6
ขา INTR	ต่อเข้ากับขา P2.7

วิธีการต่อดังแสดงในรูป 2.29



รูปที่ 2.29 การต่อ ADC 0804 เข้ากับ MCS-51

2.9 การใช้งานพอร์ตสื่อสารอนุกรมแบบ Single Processor ของพอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่า ฟลูดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน

- ทางด้านส่งใช้ขา TxD พอร์ต (3.1)
- ทางด้านรับใช้ขา RxD พอร์ต (3.0)

Serial Port Buffer (SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมโดยมีอยู่ 2 ตัว

การส่งข้อมูล ข้อมูลที่จะส่งให้ใส่ใน SBUF โดยใช้คำสั่ง MOV SBUF, A โดยเตรียมข้อมูลที่ จะส่งเข้า A ก่อน

การรับข้อมูล ข้อมูลที่รับได้จะอยู่ใน SBUF การถ่ายข้อมูลออกมาใช้คำสั่ง MOV A, SBUF แล้วจึงนำข้อมูลใน A ไปใช้

พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม ดังนี้

REN (Receive Enable) บิตควบคุมให้รับหรือไม่รับข้อมูล

- 1 : ให้รับข้อมูลได้
- 0 : ห้ามรับข้อมูล

(หมายเหตุ) การรับข้อมูลสามารถห้ามได้แต่การส่งข้อมูลห้ามไม่ได้

TI แฟล็กซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI แฟล็กซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

2.9.1 การเขียนโปรแกรมควบคุมการรับและส่งข้อมูลทำได้ 2 วิธี

- การตรวจสอบบิต TI หรือ RI โดยใช้คำสั่งตรวจสอบบิตเช่นใช้คำสั่ง WAIT : JNB

TI, WAIT

คำสั่งนี้หมายความว่า ถ้า TI = 0 ใหวนไปยังแอดเดรสชื่อ WAIT
 ถ้า TI = 1 ถือว่าส่งข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมต้องเคลียร์เอง)

- การใช้อินเตอร์รัพต์ควบคุม

โหมด 1 : พอร์ตสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON อัตราโอเวอร์โพล์ของ Timer 1, 2



รูปที่ 2.30 การใช้งานโหมด 1

2.9.1.1 การเชื่อมต่อไมโครโปรเซสเซอร์เพื่อรับส่งข้อมูลอนุกรม (UART) มีอยู่ 2

โหมดด้วยกันคือ

- Single Processor Mode
- Multiprocessor Mode

Single Processor Mode : ในโหมดนี้เราจะใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมเข้าหากัน

Multiprocessor Mode : ในโหมดนี้เราจะใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นตัวแม่ (Master) และอีก 0-256 เป็นตัวลูก (Slave)

รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม SCON

Serial Control Port Register (SCON) อยู่ใน SFR ตำแหน่ง (98H)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

ตารางที่ 2.4 SM0, SM1 บิตเลือกโหมดการทำงาน

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/12) ของ CPU Osc
0	1	1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1, 2
1	0	2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1, 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM2 บิตเลือกการทำงานแบบ Single Processors Environment หรือ Multiprocessor Environment

1 : เลือก Multiprocessor Environment ใช้ได้กับโหมด 2, 3

0 : เลือก Single Processors Environment ใช้ได้กับทุกโหมด

เมื่อเลือกการทำงานรับข้อมูลแบบ Multiprocessors Mode แล้ว

ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 1RI จะเซต

ถ้าข้อมูลบิตที่ 9 ที่รับได้มีค่าเป็น 0RI จะไม่เซต

REN (Receive Enable) บิตควบคุมให้รับหรือไม่รับข้อมูล

1 : ให้รับข้อมูลได้

0 : ห้ามรับข้อมูล

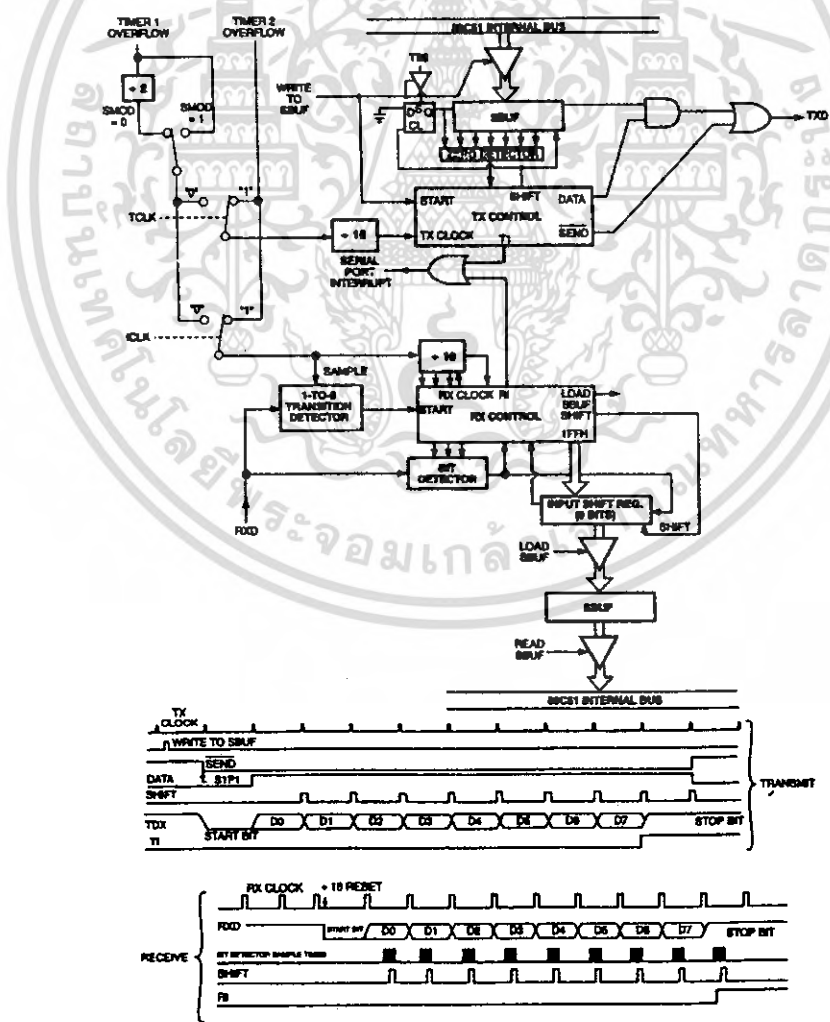
TB8 (Transmit bit D8) ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ส่งในบิตนี้ได้เลย

RB8 (Receive bit D8) ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมากับในบิตนี้

(ข้อมูลบิตที่ 9 ก็คือค่าใน ก็คือค่าใน TB8 ทางด้านส่งนั่นเอง)

TI แฟล็กซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI แฟล็กซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

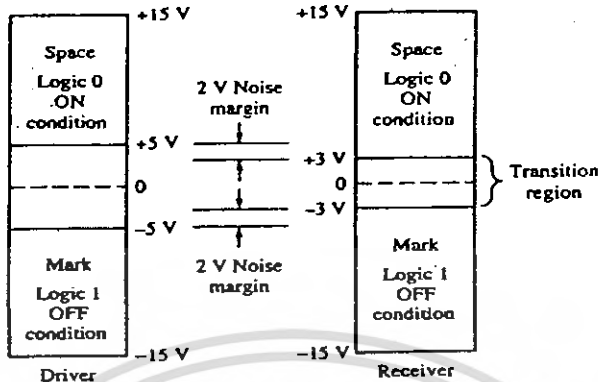


รูปที่ 2.31 ผังการทำงาน Serial Port (Mode 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1.2 การเชื่อมต่อแบบมาตรฐาน RS-232C

RS-232C จะใช้แรงดันไฟเป็นบวกและลบไม่ใช่ TTL ดังนั้นจึงต้องแปลงเสียบก่อนดังแสดงในตารางที่ 2.5

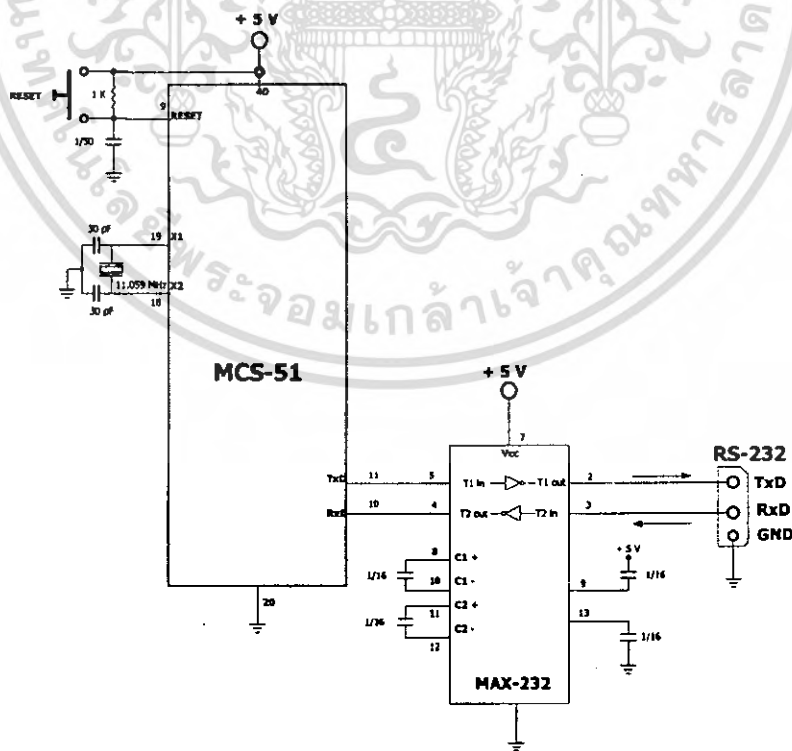


รูปที่ 2.32 RS-232-C Voltage Range

ตารางที่ 2.5 ค่าสัญญาณดิจิทัลที่แตกต่างจากแรงดัน

TTL(V)	RS-232C(V)
0	+(5-15)V
1	-(5-15)V

RS-232C มีรายละเอียดดังรูป 2.32 อุปกรณ์ที่ใช้แปลงระหว่าง TTL และ RS-232C คือ MAX-232 ทำหน้าที่เป็น Drivers/Receiver ได้ด้วย



รูปที่ 2.33 การเชื่อมต่อ MCS-51 กับ RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การใช้งาน LCD Module และ โครงสร้าง LCD Module

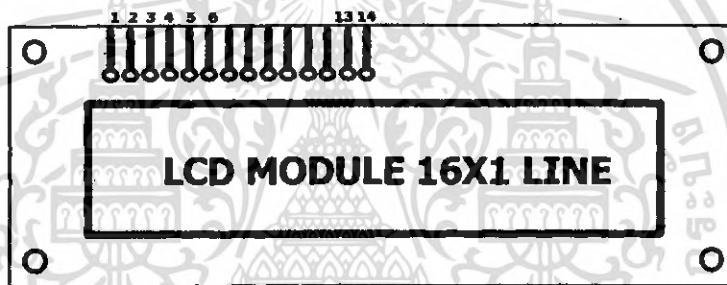
ใน LCD Module จะมีส่วนประกอบหลักๆ 3 ส่วนดังนี้

1. ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมมองของข้อมูลแสดงผลบนจอ

2. ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ LCD Module เช่นลบจอภาพแสดงอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมการทำงานโดยเฉพาะชิปควบคุมนี้ใช้ชิปควบคุมแบบอักษร ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟฟิก

3. ตัวขับ (Driver) เป็นตัวขับรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนดชิปที่ใช้ทำหน้าที่ได้แก่เบอร์ HD44100H และ MSM5229 เป็นต้น

LCD Module มีอยู่หลายรุ่นและคุณสมบัติแตกต่างกันไปซึ่งแบ่งได้เป็น 2 แบบคือ แบบ Dot Matrix และ Graphic โดยแบบ Dot Matrix จะแสดงผลเป็นแบบ 5x8 Dot. หรือ 5x10 Dot. มีตั้งแต่ 1 Line, 2 Line และ 4 Line ซึ่งการใช้งานแต่ละแบบจะใกล้เคียงกัน ลักษณะขาสัญญาณของ LCD Module แบบ 1 Line ดังรูปที่ 2.34



รูปที่ 2.34 แสดงลักษณะของตัว LCD Module

ตัว LCD Module จะมีขาใช้งานทั้งหมด 14 ขาคือขาที่ของแต่ละขามีดังนี้คือ

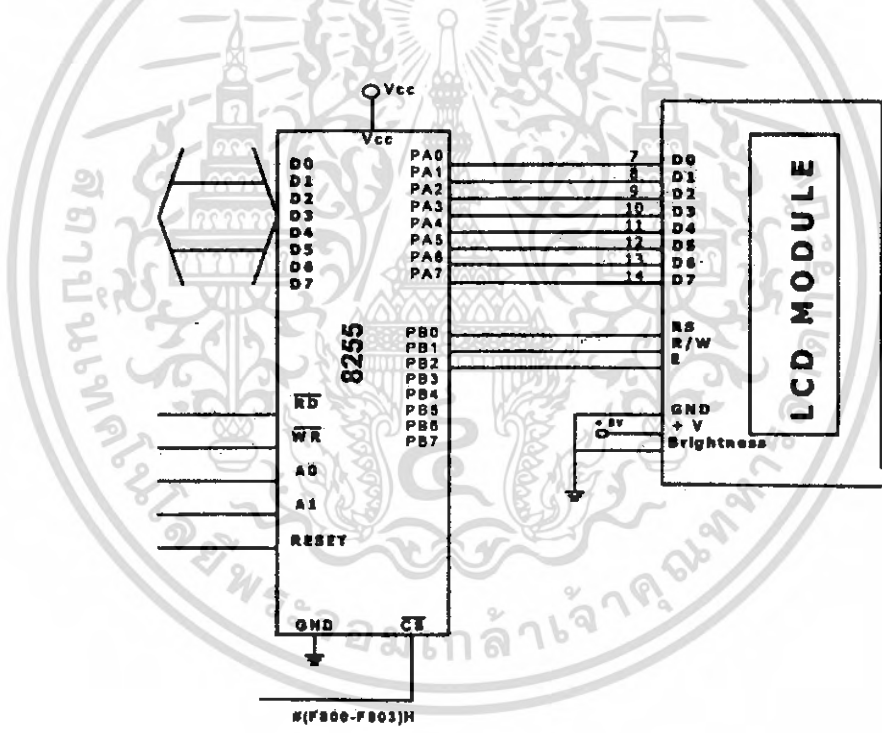
- | | | |
|-------------------------|---|--|
| ขา 1 (GND) | : | เป็น Ground ใช้ต่อกับระบบ Ground ของไมโครคอนโทรลเลอร์ |
| ขา 2 (VCC) | : | เป็นไฟเลี้ยงวงจรของ LCD มีขนาด 5VDC |
| ขา 3 (V _{ee}) | : | เป็นขาสำหรับปรับความเข้มของจอ LCD โดยที่เมื่อต่อกับ VCC จะมีความเข้มต่ำสุด และเมื่อต่อกับ Ground จะมีความเข้มมากที่สุด โดยปกติจะต่อกับ Ground เสมอเพื่อความสะดวกในการต่อ |
| ขา 4 (RS) | : | Register Select ใช้สำหรับบอก LCD ทราบว่าข้อมูลที่ส่งให้มันเป็น Instruction หรือ Data โดยเมื่อนี้
เป็น "0" หมายถึง Instruction
เป็น "1" หมายถึง Data |
| ขา 5 (R/W) | : | ใช้สำหรับกำหนดว่าเป็นการอ่านหรือเขียนข้อมูลให้กับ LCD โดยเมื่อนี้
เป็น "0" หมายถึงเป็นการเขียนข้อมูล
เป็น "1" หมายถึงเป็นการอ่านข้อมูล |

- ขา 6 (E) : เป็นขา Enable ขานี้ เป็น "1" ใช้สำหรับบอก LCD ว่าอุปกรณ์ภายนอก ต้องการติดต่อกับ เป็น "0" ตัว LCD จะไม่สนใจสัญญาณ RS, R/ W และ (DB₇ – DB₀)
- ขา 7-14 : เป็นขา Data Bus สำหรับอ่านหรือเขียนข้อมูลให้กับตัว LCD (DB₇ – DB₀)

2.10.1 การเชื่อมต่อ LCD Module เข้ากับไมโครคอนโทรลเลอร์

การเชื่อมต่อ LCD Module เข้ากับไมโครคอนโทรลเลอร์สามารถทำได้โดยตรงกับตัว MCS-51 หรือต่อผ่าน 8255 ก็ได้ในที่นี่จะต่อโดยผ่าน 8255 ดังรูปที่ 2.35

- ขาสัญญาณข้อมูล D0 – D7 (ขา 7 – 14) ต่อเข้ากับ 8255 พอร์ต A
- ขา RS (ขา 4) ต่อเข้ากับ 8255 พอร์ต B บิต 0
- ขา R/ W (ขา 5) ต่อเข้ากับ 8255 พอร์ต B บิต 1
- ขา E (ขา 6) ต่อเข้ากับ 8255 พอร์ต B บิต 2



รูปที่ 2.35 แสดงการเชื่อมต่อ LCD Module เข้ากับ 8255

ชุดคำสั่งของ LCD Module ใช้ในการทำงานแต่ละคำสั่ง

ตารางที่ 2.6 คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

คำสั่ง Clear Display เป็นคำสั่งที่ใช้เขียนข้อมูลหรือตัวอักษรว่าง (Space) ลงใน DDRAM ทั้งหมด และทำการกำหนดค่า DDRAM Address เป็น 0 และเคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งบนซ้ายสุดของจอแสดงผล

ตารางที่ 2.7 คำสั่ง Cursor at Home

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

คำสั่ง Cursor at Home หรือ Return Home เป็นคำสั่งที่ใช้ในการเลื่อนตำแหน่งของเคอร์เซอร์ ไปอยู่ที่ตำแหน่งบนซ้ายสุดของจอแสดงผลโดยข้อมูลที่อยู่ใน DDRAM หรือที่หน้าจอแสดงผลจะไม่เปลี่ยนแปลง

ตารางที่ 2.8 คำสั่งโหมดในการป้อนข้อมูล (Entry Mode Set)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

คำสั่งโหมดในการป้อนข้อมูล (Entry Mode Set) ใช้สำหรับกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งของแอดเดรสของ DDRAM ดังนี้

-I/D เป็นบิตที่ใช้ในการกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ว่าจะให้เพิ่มหรือลดลงเมื่อเขียนหรืออ่านข้อมูลแล้ว

บิต I/D = 0 แอดเดรสของ DDRAM จะลดลง

บิต I/O = 1 แอดเดรสของ DDRAM จะเพิ่มขึ้นส่วนเคอร์เซอร์จะเลื่อนตามตำแหน่งแอดเดรสของ DDRAM

-S เป็นบิตที่ใช้กำหนดลักษณะของการแสดงผลเมื่อมีการเขียนข้อมูล

บิต S = 1 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะอยู่กับที่แต่ตัวอักษรข้อมูลเดิมจะถูกผลักไปทางซ้าย

บิต S = 0 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

ตารางที่ 2.9 คำสั่งควบคุมการแสดงผล (Display ON/OFF)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

คำสั่งควบคุมการแสดงผล เป็นคำสั่งที่ใช้ในการเปิดปิดจอแสดงผลและเคอร์เซอร์มีลักษณะดังนี้

D = 0 กำหนดให้ปิดจอแสดงผล (Display OFF)

D = 1 กำหนดให้เปิดจอแสดงผล (Display ON)

C = 0 กำหนดให้ปิดเคอร์เซอร์ (Cursor OFF)

C = 1 กำหนดให้เปิดเคอร์เซอร์ (Cursor ON)

B = 0 กำหนดให้ไม่มีการกระพริบที่เคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่เคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยมทึบ)

ตารางที่ 2.10 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร (Display Shift)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร เป็นการควบคุมเลื่อนเคอร์เซอร์และตัวอักษร บนจอแสดงผล โดยขึ้นอยู่กับกำหนัดบิต S/C และ R/L โดยมีลักษณะดังนี้

S/C	R/L	ลักษณะการเลื่อน
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย
0	0	เลื่อนเคอร์เซอร์ไปทางขวา
1	0	เลื่อนอักษรตัวใหม่ไปทางซ้าย
1	1	เลื่อนอักษรตัวใหม่ไปทางขวา

ตารางที่ 2.11 คำสั่งการกำหนดฟังก์ชันการทำงาน (Function Set)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้ติดต่อกับ LCD Module เป็นแบบ 4 บิต

DL = 1 กำหนดให้ติดต่อกับ LCD Module เป็นแบบ 8 บิต

N = 0 กำหนดการแสดงผลแบบ 1 บรรทัด

N = 1 กำหนดการแสดงผลตั้งแต่ 2 บรรทัดขึ้นไป

F = 0 กำหนดความละเอียดของการแสดงผลเป็น 5 x 8 Dot.

F = 1 กำหนดความละเอียดของการแสดงผลเป็น 5 x 10 Dot.

ตารางที่ 2.12 คำสั่งเลือกแอดเดรสของ CGRAM (Set CGRAM Address)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

คำสั่งนี้ใช้สำหรับกำหนดตำแหน่ง Address ของ Character Generator หรือ CGRAM โดยจะต้องกำหนดค่านี้ทุกครั้งที่เขียนหรืออ่านข้อมูลกับ CGRAM ซึ่งกำหนดที่ (DB0 – DB5) ส่วน DB6 ต้องเป็น “1” และ DB7 ต้องเป็น “0” (01XX XXXX B) ซึ่งก็คือ (40H – 7FH)

ตารางที่ 2.13 คำสั่งเลือกแอดเดรสของ DDRAM (Set DDRAM Address)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

แอดเดรสของ LCD 16 อักขร 4 บรรทัด

บรรทัดที่ 1 1000XXXXB = (80 – 8F)H

บรรทัดที่ 2 1001XXXXB = (90 – 9F)H

บรรทัดที่ 3 1100XXXXB = (C0 – CF)H

บรรทัดที่ 4 1101XXXXB = (D0 – DF)H

คำสั่งนี้ใช้สำหรับกำหนดตำแหน่งของ Address ของ Display Data RAM หรือ DDRAM หรือตำแหน่งของเคอร์เซอร์สำหรับการแสดงผลทางหน้าจอ LCD ซึ่งเมื่อมีการอ่านหรือเขียนค่าตัวอักษรให้กับ LCD ในแต่ละครั้งนั้นค่าตำแหน่งของ DDRAM Address จะเพิ่มขึ้นหรือลดลง 1 ตำแหน่งโดยอัตโนมัติเสมอ ซึ่งจะเพิ่มหรือลดนั้นกำหนดได้จากบิต I/D ใน Entry Mode SET

ตารางที่ 2.14 คำสั่งอ่านแฟล็ก Busy และ Address (Read Busy Flag & Address)

RS	R/ W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	BF	CGRAM/DDRAM ADDRESS						

คำสั่งนี้ใช้สำหรับการอ่านค่าของ Busy Flag (BF) ซึ่งบอกถึงความพร้อมของ LCD ในการรับข้อมูล

ถ้า BF = 0 หมายถึง LCD พร้อมที่จะรับข้อมูลต่อไปได้

ถ้า BF = 1 หมายถึงว่ายังไม่พร้อมที่จะรับข้อมูล

นอกจากนี้แล้วทุกครั้งที่อ่านค่าแฟล็ก BF เข้ามาแล้วก็จะได้ตำแหน่งของ CGRAM หรือ DDRAM ณ. ในตำแหน่งของ (DR0 – DB6)

สำหรับตัวอักษรที่ใช้ในการแสดงผลของ LCD Module นั้นจะเป็นรหัส ASCII ที่ใช้กันทั่วไป ดังแสดงในรูปที่ 2.36 ตัวอย่างเช่นต้องการให้แสดงเป็นตัวอักษร “A” จะต้องส่งค่า 41H ออกไปให้ที่ตัว LCD หรือ “a” ก็ต้องส่งค่า 61H ออกไป ส่วนค่า 00H จนถึง 07H หรือ 08H จนถึง 0FH (บิต D₃ นั้นจะเป็น “0” หรือ “1” ก็ได้ 0000XYYY) YYY คือตำแหน่งได้ทั้งหมด 8 ตำแหน่งเพื่อนำตัวอักษรที่ไหลตนเองใน CGRAM ออกไปแสดงผล

ตัวอักษรที่ใช้กับ LCD Module ดังแสดงในรูป 2.36

Higher 4bit Lower 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxx>0000			๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒
xxx>0001			๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓
xxx>0010			๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔
xxx>0011			๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕
xxx>0100			๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖
xxx>0101			๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗
xxx>0110			๘	๙	๐	๑	๒	๓	๔	๕	๖	๗	๘
xxx>0111			๙	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙
xxx>1000			๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๐
xxx>1001			๑	๒	๓	๔	๕	๖	๗	๘	๙	๐	๑
xxx>1010			๒	๓	๔	๕	๖	๗	๘	๙	๐	๑	๒
xxx>1011			๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓
xxx>1100			๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔
xxx>1101			๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕
xxx>1110			๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖
xxx>1111			๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗

รูปที่ 2.36 แสดงรหัสตัวอักษรที่ใช้กับ LCD Module

2.11 สเต็ปป์มอเตอร์ (Stepping Motor)

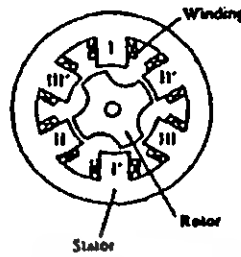
สเต็ปป์มอเตอร์เป็นมอเตอร์ที่หมุนทีละสเต็ป โดยแต่ละสเต็ปมอเตอร์จะหมุนด้วยความเร็วคงที่ค่าหนึ่ง ซึ่งในการควบคุมการหมุนของสเต็ปป์มอเตอร์นั้นจะอาศัยวงจรควบคุมทางดิจิทัล โดยที่วงจรดิจิทัลนี้จะทำหน้าที่ในการจัดลำดับการกระตุ้นในแต่ละเฟสของสเต็ปป์มอเตอร์ ซึ่งจะสามารถทำให้สามารถกำหนดทิศทางการหมุน ความเร็วในการหมุน และตำแหน่งที่ต้องการจะเคลื่อนไปของสเต็ปป์มอเตอร์ได้อย่างถูกต้องแม่นยำ

เนื่องจากวงจรทางดิจิทัลที่ใช้ในการควบคุมการหมุนของสเต็ปป์มอเตอร์สามารถกำหนดความเร็วในการหมุนและตำแหน่งที่ต้องการจะเคลื่อนไปของสเต็ปป์มอเตอร์ได้อย่างแม่นยำ ดังนั้นจึงไม่จำเป็นต้องมีการป้อนกลับ (Feedback Control) เพื่อควบคุมความเร็วและตำแหน่งในการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

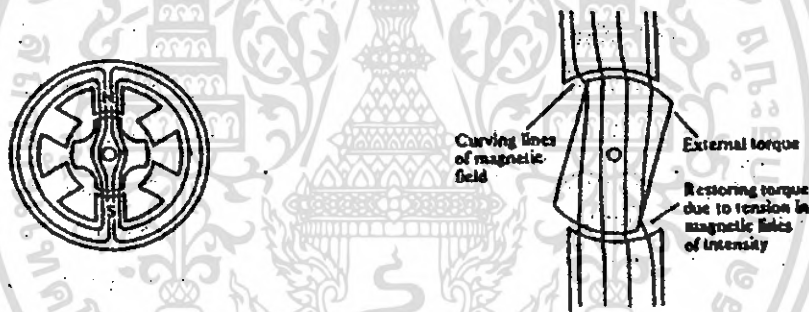
2.11.1 โครงสร้างและหลักการทำงานของสเต็ปปีงมอเตอร์

ภายในสเต็ปปีงมอเตอร์ประกอบด้วย สเตเตอร์ (Stator), โรเตอร์ (rotor) และขดลวด (coil) ประกอบเข้าด้วยกันดังรูปที่ 2.37

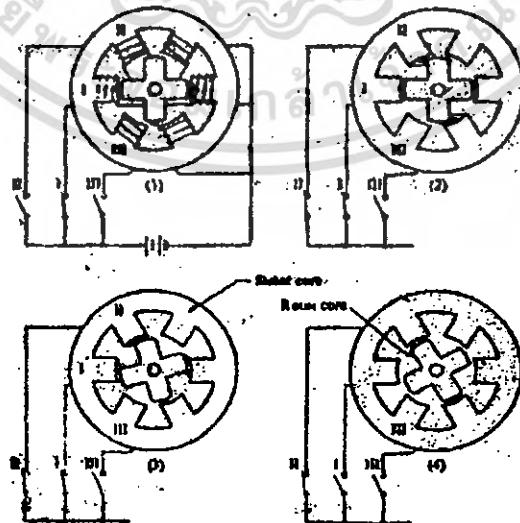


รูปที่ 2.37 โครงสร้างของสเต็ปปีงมอเตอร์

เนื่องจากโรเตอร์เป็นเหล็กอ่อน ซึ่งมีคุณสมบัติพยายามปรับตัวเองให้อยู่ในแนวเส้นแรงแม่เหล็กผ่านมากที่สุด ดังรูปที่ 2.38 เมื่อเกิดเส้นแรงแม่เหล็กที่ตัวสเตเตอร์ผ่าน โรเตอร์ตัวโรเตอร์ก็จะพยายามปรับตัวเองให้แรงแม่เหล็กผ่านมากที่สุดโดยหมุนตัวเองทำให้เคลื่อนที่เกิดการหมุนและจะหยุดหมุนเมื่อเส้นแรงแม่เหล็กที่ตัดผ่านตัวมันถึงจุดมากที่สุด



รูปที่ 2.38 เส้นแรงแม่เหล็กที่ทำให้เกิดแรงบิด



รูปที่ 2.39 แสดงการทำงานของสเต็ปปีงมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำให้สเต็ปป์มอเตอร์หมุนนั้นสามารถทำได้โดยอาศัยหลักการนี้แต่ต้องให้เส้นแรงแม่เหล็กเกิดขึ้นโดยรับช่วงต่อกันไปเรื่อยๆ ดังรูปที่ 2.39 เป็นสเต็ปป์มอเตอร์ที่แกนเหล็กสเตเตอร์ (Stator Core) มีซี่ฟัน (Tech) 6 ซึ่งขณะที่โรเตอร์มีฟัน 4 ซี่ ที่โรเตอร์ (Rotor) และสเตเตอร์เป็นเหล็กอ่อน ขดลวด (Core) 3 ชุด ถูกต่ออยู่ดังรูปที่ 2.39 แต่ละชุดขดลวดมี 2 ขดลวดต่อกันเรียกแต่ละชุดว่าเฟส (Phase) และผลจากการต่อแบบนี้เรียกว่า มอเตอร์ 3 เฟส (3 Phase Motor) กระแสถูกจ่ายไปยังขดลวดแต่ละขดลวดผ่านสวิตช์ I, II และ III ในสภาวะ (1) ขดลวดของเฟส 1 ได้รับกระแสไฟโดยผ่านสวิตช์ I หรือเฟส I ถูกกระตุ้น เส้นแรงแม่เหล็กที่เกิดขึ้นไซช่องอากาศ (Air Gap) เกิดขึ้นเนื่องจากการกระตุ้น ซึ่งแสดงด้วยลูกศร ในสภาวะที่สเตเตอร์ 2 ขั้วของเฟส 1 จะอยู่ในแนวเดียวกับ 2 ซี่ที่อยู่ตรงข้ามกันของโรเตอร์นี้เป็นสภาวะสมดุลซึ่งอยู่ในเทอมของไดนามิก (Dynamics) เมื่อสวิตช์ II ปิดเพื่อกระตุ้นเฟส 2 กับเฟส 1 เส้นแรงแม่เหล็กจะถูกสร้างขึ้นที่ขั้วของสเตเตอร์ของเฟส 2 ในลักษณะซึ่งแสดงในสภาวะ (2) ทอร์ก (Torque) ทิศทางทวนเข็มนาฬิกาจะถูกสร้างมาจากความเครียด (Tension) ในฟลักซ์แม่เหล็กเอียงไปยังแกนมอเตอร์ที่อยู่ใกล้ หลังจากนั้นมอเตอร์จะมาอยู่ในสภาวะ (3)

ดังนั้นโรเตอร์จะหมุนไปด้วยมุมการเปลี่ยนแปลงคงที่ซึ่งเรียกว่ามุมสเต็ป (Step Angle) ในที่นี้คือ 15 องศา ขณะที่สวิตช์มีการเปลี่ยนแปลงอีกครั้งหนึ่งคือ สวิตช์ I จะถูกปิดเพื่อลดพลังงานในเฟสโดยโรเตอร์จะหมุนไป 15 องศา อยู่ในสภาวะที่ (4) ตำแหน่งมุมโรเตอร์จะถูกควบคุมโดยการ เปิด-ปิดสวิตช์ ถ้าสวิตช์ถูกเปิด-ปิดเป็นลำดับ โรเตอร์จะหมุนในลักษณะที่เป็นสเต็ป ความเร็วเฉลี่ยจะสามารถควบคุมได้ด้วยการเปิด-ปิดสวิตช์ดังรูปที่ 2.39

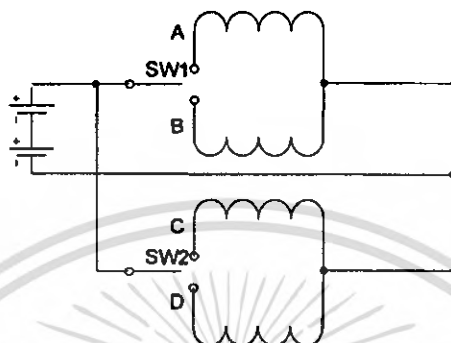
จากที่กล่าวมาพอที่จะกล่าวถึงคุณสมบัติสเต็ปป์มอเตอร์ได้ว่า

1. การหมุนของมอเตอร์จะเป็นสเต็ปเป็นขั้นๆ (สเต็ปละกี่องศาขึ้นอยู่กับชนิดของมอเตอร์)
2. ความเร็วในการหมุนขึ้นกับสัญญาณพัลส์ที่ให้เข้ามาทางอินพุตของมอเตอร์ (ความเร็ว)
3. ความผิดพลาดในการหมุน 1 สเต็ปมีค่าน้อยมาก แต่ต้องจำกัดอยู่ในความเร็วที่พอดีด้วย
4. คุณสมบัติการตอบสนองสัญญาณที่มอเตอร์เริ่มทำงาน และหยุดการทำงานดีมาก
5. เนื่องจากไม่มีแปรงถ่าน (Commutator) เหมือนมอเตอร์ในระบบ DC ดังนั้นจึงมีความแน่นอนในการทำงานสูง
6. แหล่งจ่ายแรงดันไฟที่ใช้ในการขับมอเตอร์มีค่าไม่มาก
7. ไม่ทำให้เกิดเสียงรบกวนและการสั่นได้ง่าย
8. ทำงานแบบโอเพ่นลูป (Open Loop)

2.11.2 การทำงานพื้นฐาน

สเต็ปป์มอเตอร์เป็นมอเตอร์ที่ถูกใช้งานโดยสัญญาณอินพุตที่เป็นพัลส์ต่างๆ สัญญาณพัลส์ที่เข้ามาจะทำให้การเปลี่ยนแปลงสถานะสนามแม่เหล็กไฟฟ้าที่เกิดขึ้นและให้การหมุนของมอเตอร์เป็นมุมที่คงที่ ซึ่งจะแตกต่างกับการหมุนของมอเตอร์แบบธรรมดาในระบบควบคุมที่ใช้สัญญาณดิจิทัล เช่นในการส่งข้อมูลการควบคุมข้อมูลโดยตรงได้เป็นอย่างดี การทำงานพื้นฐานของสเต็ปป์มอเตอร์แสดงในรูปที่ 2.40 ซึ่งแสดงบล็อกรวมของวงจรขับทำให้กระแสไฟดีซีเรียงลำดับเข้าไปตามเฟสต่างๆ ของมอเตอร์หมุนไปตามองศาที่กำหนดไว้ การที่ทำให้กระแสดีซีเรียงลำดับเข้าไปที่มอเตอร์ได้ทำได้โดยการ

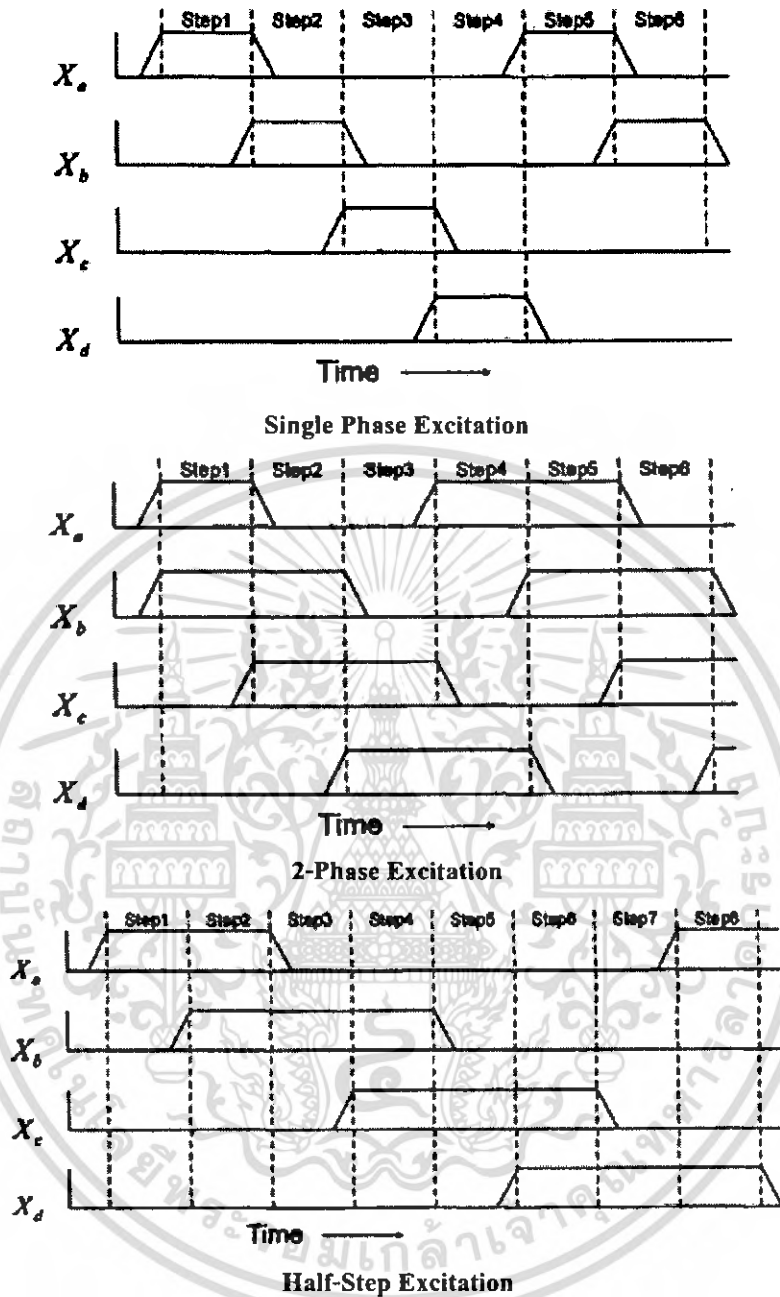
ใช้งานของสวิตช์และเพื่อให้มอเตอร์หมุนจำเป็นต้องมีวงจรขั้วสแต๊ปปิ้งมอเตอร์ที่จะควบคุมการไหลของกระแสไฟ คือสวิตช์ 1 สวิตช์ 2 ถ้าให้การทำงานของสวิตช์ตามตารางการทำงานที่แสดงไว้ในรูปที่ 2.40 ถ้าให้การทำงานโดยใช้รีเลย์ (Relay) หรือมือโยกสแต๊ปปิ้งมอเตอร์ก็จะหมุนเหมือนกันแต่หมุนช้าเนื่องจากรีเลย์หรือมือโยกให้การเปลี่ยนแปลงของสวิตช์ช้า ซึ่งจะทำให้เร็วเท่ากับความเร็วสูงสุดของสัญญาณพัลส์ที่มอเตอร์สามารถทำงานได้ (เป็นจำนวนพัลส์ต่อวินาที)



รูปที่ 2.40 การทำงานพื้นฐานของสแต๊ปปิ้งมอเตอร์

วงจรขั้วมอเตอร์มีหลายชนิด ขึ้นอยู่กับชนิดของสแต๊ปปิ้งมอเตอร์ที่มีอยู่ 3 ระบบ คือ

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



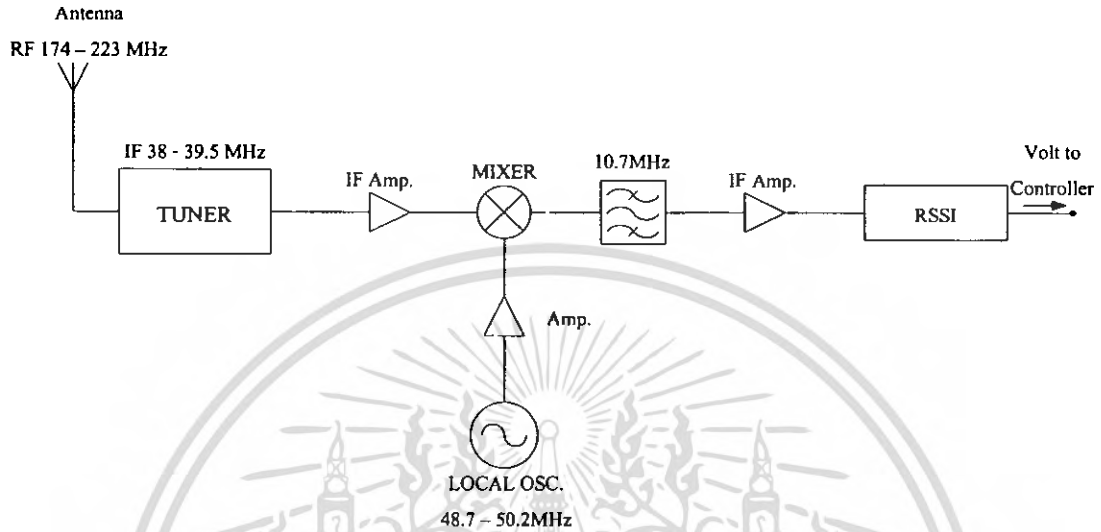
รูปที่ 2.41 ชนิดการขับของสเต็ปปีงมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

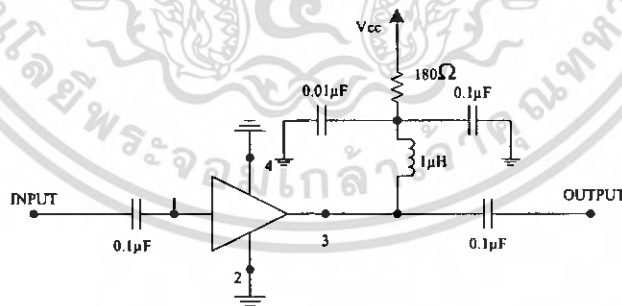
3.1 บล็อกไดอะแกรมของระบบตรวจวัดและบันทึกความแรงสัญญาณวิทยุ



รูปที่ 3.1 บล็อกไดอะแกรมของวงจรส่วนรับและตรวจจับสัญญาณ

3.2 วงจรขยายสัญญาณ IF ของจูนเนอร์

ในบางทิศทางที่สายอากาศรับคลื่นได้นั้นสัญญาณที่รับเข้ามาจะอ่อนมากทำให้ในการวัดความแรงนั้นทำได้ยาก เพื่อให้ระบบสามารถตรวจจับสัญญาณได้ดีจึงต้องนำสัญญาณ IF ที่ออกจากจูนเนอร์นั้นมาทำการขยายโดยใช้ไอซี NLB-310 โดยมีวงจรดังรูปที่ 3.2

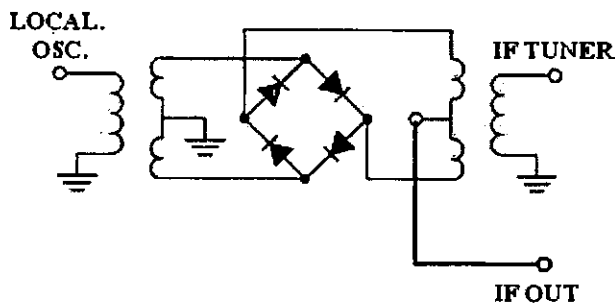


รูปที่ 3.2 วงจรขยาย

3.3 วงจรมิกเซอร์

วงจรมิกเซอร์จะรับสัญญาณความถี่จากวงจรขยายสัญญาณที่มีระดับสัญญาณรบกวนต่ำ เป็นสัญญาณความถี่ IF จากจูนเนอร์และรับสัญญาณโลคอลออสซิลเลเตอร์ โดยวงจรมิกเซอร์นี้ใช้ไอซีเบอร์ JMS-11x ต่อวงจรตามรูปที่ 3.3

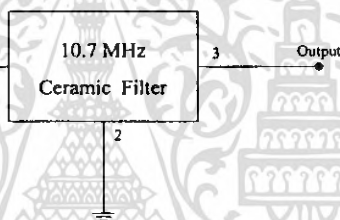
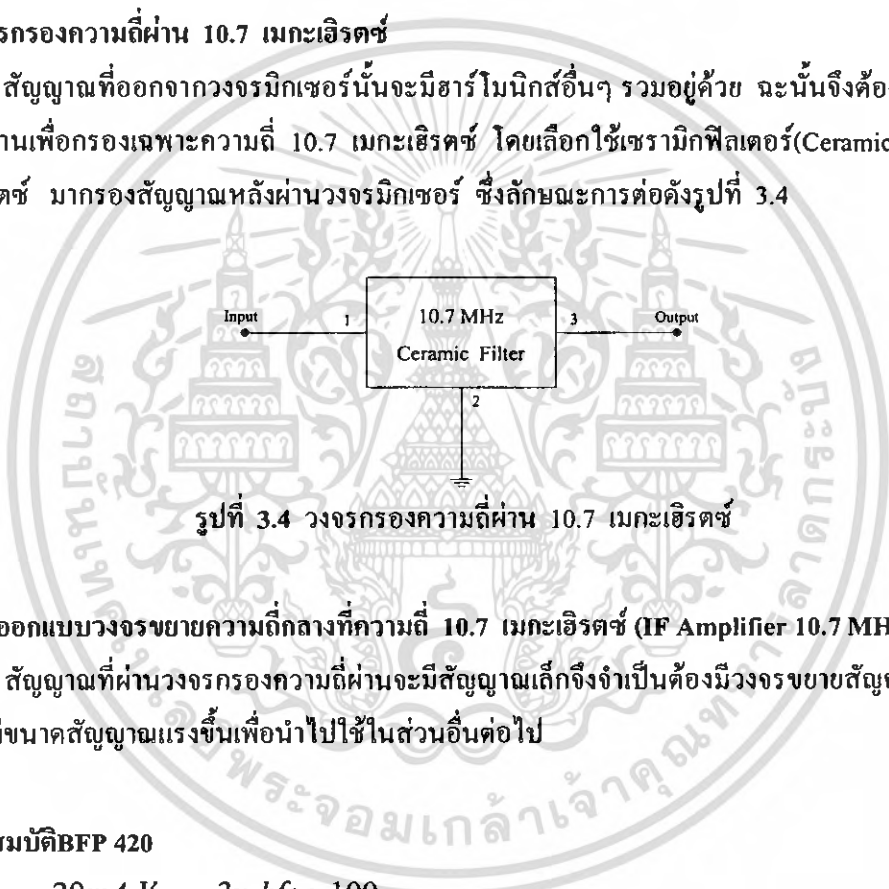
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรสมมูล

3.4 วงจรกรองความถี่ผ่าน 10.7 เมกะเฮิร์ตซ์

สัญญาณที่ออกจากวงจรมิกเซอร์นั้นจะมีฮาร์โมนิกส์อื่นๆ รวมอยู่ด้วย ฉะนั้นจึงต้องมีการกรองความถี่ผ่านเพื่อกรองเฉพาะความถี่ 10.7 เมกะเฮิร์ตซ์ โดยเลือกใช้เซรามิกฟิลเตอร์(Ceramic Filter) 10.7 เมกะเฮิร์ตซ์ มากรองสัญญาณหลังผ่านวงจรมิกเซอร์ ซึ่งลักษณะการต่อดังรูปที่ 3.4



รูปที่ 3.4 วงจรกรองความถี่ผ่าน 10.7 เมกะเฮิร์ตซ์

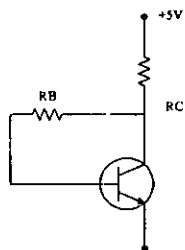
3.5 การออกแบบวงจรขยายความถี่กลางที่ความถี่ 10.7 เมกะเฮิร์ตซ์ (IF Amplifier 10.7 MHz)

สัญญาณที่ผ่านวงจรกรองความถี่ผ่านจะมีสัญญาณเล็กจึงจำเป็นต้องมีวงจรขยายสัญญาณในส่วนนี้ทำให้มีขนาดสัญญาณแรงขึ้นเพื่อนำไปใช้ในส่วนอื่นต่อไป

จากคุณสมบัติ BFP 420

$$I_C = 20mA, V_{CE} = 3v, h_{fe} \approx 100$$

$$I_B = \frac{I_C}{h_{fe}} = \frac{20 \times 10^{-3}}{100} = 200 \mu A$$



รูปที่ 3.5 วงจรสมมูลทาง DC ของวงจรคอมมอนอีมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_C = \frac{V_{CC} - V_{CE}}{I_C} = \frac{5 - 3}{20mA} = 200\Omega$$

$$R_B = \frac{V_{CC} - (V_{RC} + V_{BE})}{I_B} = \frac{5 - 2.7}{200 \times 10^{-6}} = 11.5k\Omega$$

การคำนวณทาง AC

ใส่ RFC ค่า $\mu 10H$ ที่ขา C $X_L = 2\pi fL = 2\pi \times 10.7 \times 10^6 \times 10 \times 10^{-6} = 671.96\Omega$

ใส่ C feed back ค่า $10000pF$ $X_C = \frac{1}{2\pi fC} = \frac{1}{2\pi \times 10.7 \times 10^6 \times 10000 \times 10^{-12}} = 1.488\Omega$

และจากคุณสมบัติของ BFP 420

$$I_C = 20mA, V_{CE} = 3v, f \approx 10.7MHz$$

$$S_{11} = 0.4471 \angle -1.7^\circ = 0.446 - j0.013$$

$$S_{22} = 0.9515 \angle -0.5^\circ = 0.9515 - j0.0083$$

$$S_{21} = 37.643 \angle 177.9^\circ = -37.617 + j1.37$$

$$S_{12} = 0.0008 \angle 70.8^\circ = 0.00026 + j0.000755$$

1. Delta factor (D_S)

$$\begin{aligned} D_S &= S_{11}S_{22} - S_{12}S_{21} \\ &= (0.4471 \angle -1.7^\circ \times 0.9515 \angle -0.5^\circ) - (0.0008 \angle 70.8^\circ \times 37.643 \angle 177.9^\circ) \\ &= 0.4352 \angle 1.57^\circ \end{aligned}$$

2. stability factor (K)

$$\begin{aligned} K &= \frac{1 + |D_S|^2 - |S_{11}|^2 - |S_{22}|^2}{2 \cdot |S_{21}| \cdot |S_{12}|} \\ &= \frac{1 + |0.435|^2 - |0.4471|^2 - |0.9515|^2}{2 \cdot |37.643| \times |0.0008|} \\ &= 1.4128 \end{aligned}$$

- เนื่องจากค่า $K > 1$ และค่า delta factor < 1 วงจรจะอยู่ในสถานะ unconditionally stable

3. เป็นการหาค่าสัมประสิทธิ์การสะท้อนของโหลดชอร์ตของวงจรขยายสัญญาณ เพื่อใช้ในการแมชชิงอิมพีแดนซ์กับวงจรอื่น

3.1 หาคจุดศูนย์กลางของวงกลมเสถียรภาพของอินพุตจากสมการ

$$\begin{aligned}
 C_s &= \frac{S_{11}^* - D_s^* \cdot S_{22}}{|S_{11}|^2 - |D_s|^2} \\
 &= \frac{0.4471 \angle 1.7^\circ - (0.435 \angle -1.57^\circ \times 0.9515 \angle -0.5^\circ)}{|0.4471|^2 - |0.435|^2} \\
 &= 4.41 \angle 41.09^\circ
 \end{aligned}$$

3.2 หารัศมีของวงกลมเสถียรภาพของอินพุตจากสมการ

$$\begin{aligned}
 r_s &= \left| \frac{S_{12} \cdot S_{21}}{|S_{11}|^2 - |D_s|^2} \right| \\
 &= \left| \frac{0.0008 \times 37.643}{|0.4471|^2 - |0.435|^2} \right| \\
 &= 3.1146
 \end{aligned}$$

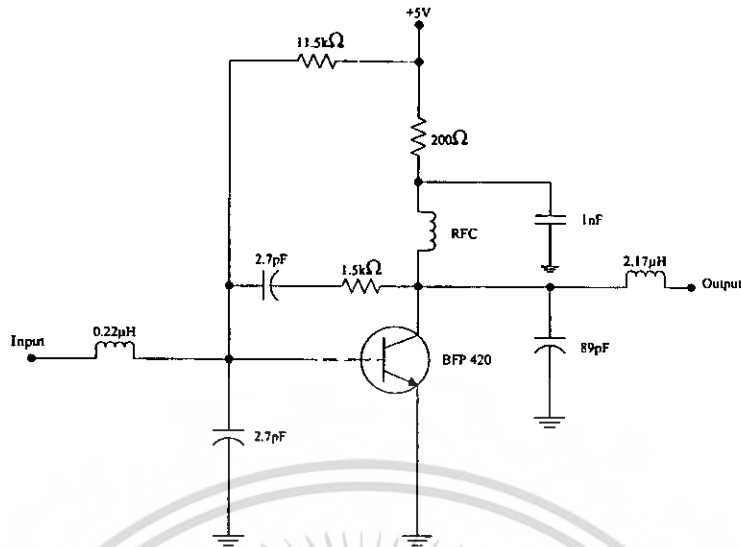
3.3 หาคจุดศูนย์กลางของวงกลมเสถียรภาพของเอาต์พุตจากสมการ

$$\begin{aligned}
 C_l &= \frac{S_{22}^* - D_s^* S_{11}}{|S_{22}|^2 - |D_s|^2} \\
 &= \frac{0.9515 \angle 0.5^\circ - (0.435 \angle -1.57^\circ \times 0.4471 \angle -1.7^\circ)}{|0.9515|^2 - |0.435|^2} \\
 &= 1.058 \angle 1.46^\circ
 \end{aligned}$$

3.4 หารัศมีของวงกลมเสถียรภาพของเอาต์พุตจากสมการ

$$\begin{aligned}
 r_l &= \left| \frac{S_{12} \cdot S_{21}}{|S_{22}|^2 - |D_s|^2} \right| \\
 &= \left| \frac{0.0008 \times 37.643}{|0.9515|^2 - |0.435|^2} \right| \\
 &= 0.042
 \end{aligned}$$

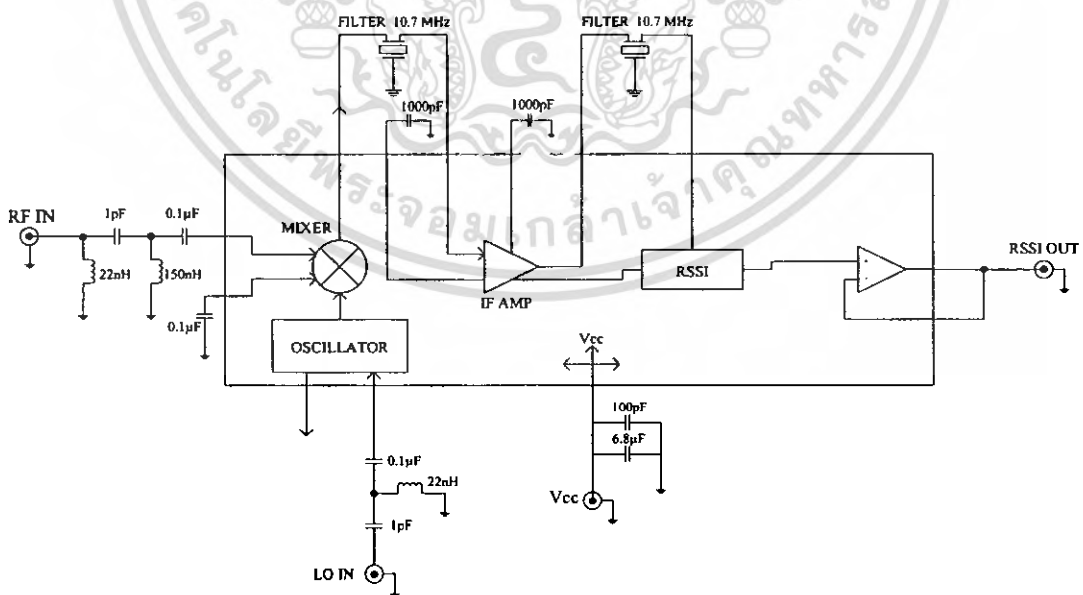
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรขยาย 10.7 เมกะเฮิร์ตซ์

3.6 การออกแบบวงจรวัดความแรงของสัญญาณ

สัญญาณ ไอเอฟที่รับความถี่มาจากจูนเนอร์นั้น จะถูกนำมาวัดความแรงของสัญญาณ โดยผ่าน วงจรที่ใช้ไอซี SA636 และใช้เซรามิกฟิลเตอร์ 10.7 เมกะเฮิร์ตซ์ กรองให้เป็นความถี่กลางก่อนที่จะเข้า ส่วนของ RSSI (Received Signal Strength Indicator) เพื่อนำความแรงของสัญญาณที่ได้นั้นมาเปลี่ยนเป็น ค่าแรงดันไฟกระแสตรง แล้วส่งเข้าคอนโทรลเลอร์แปลงสัญญาณจากอนาลอกไปเป็นดิจิทัลเข้าสู่ระบบ การประมวลผลและแสดงผลต่อไปซึ่งได้แสดงดังรูปที่ 3.7

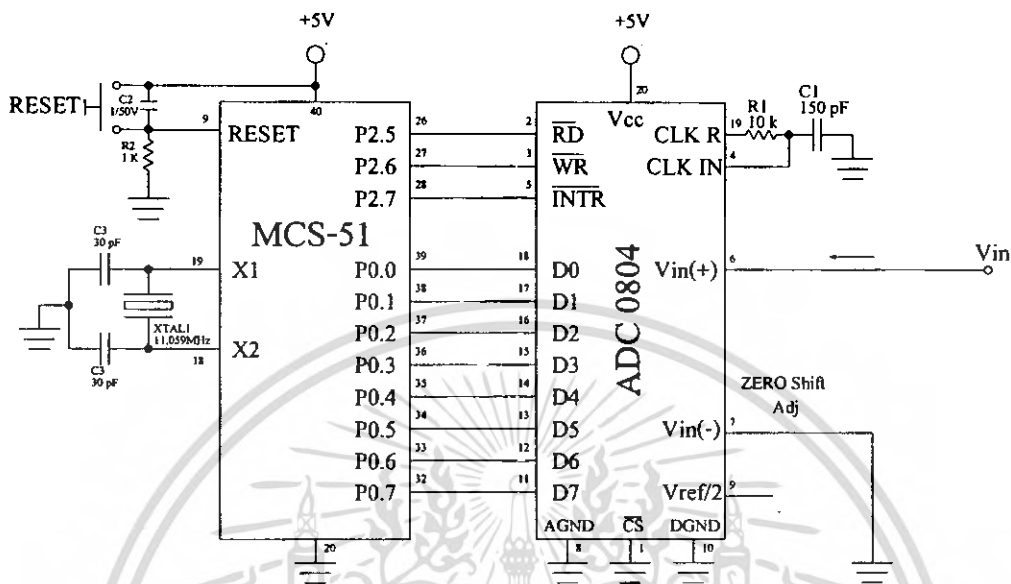


รูปที่ 3.7 วงจรวัดความแรงของสัญญาณความถี่กลาง 10.7 เมกะเฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การแปลงสัญญาณอนาลอกเป็นดิจิทัล

ในโครงการนี้จะใช้อิซซีที่ทำงานเป็นตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลคือ ADC 0804 เป็น A/D ขนาด 8 บิตโดยมีรายละเอียดการเชื่อมต่อกับ MCS - 51 ดังรูปที่ 3.8



รูปที่ 3.8 การเชื่อมต่อ ADC 0804 เข้ากับไมโครคอนโทรลเลอร์

การเชื่อมต่อ ADC 0804 เข้ากับ MCS 51 โดยกำหนดให้

- ขา DATA (D0-D7) ต่อเข้ากับขา Port 0 (P0)
- ขา \overline{RD} ต่อเข้ากับขา P2.5
- ขา \overline{WR} ต่อเข้ากับขา P2.6
- ขา \overline{INTR} ต่อเข้ากับขา P2.7
- ขา $V_{in(+)}$ และ $V_{in(-)}$ 2 ขานี้เป็น Differential Analog Input โดยที่ $V_{in} = V_{in(+)} - V_{in(-)}$
โดยต่อ $V_{in(-)}$ ลงกราวด์ และต่อ $V_{in(+)}$ กับสัญญาณภายนอก
- CLK R และ CLK IN 2 ขานี้จะต่อเข้ากับ R และ C เพื่อให้เกิดการอออสซิลเลตภายใน
ค่าความถี่โดยคำนวณจาก $f = (1/1.1)RC$ ใช้ค่า $R = 10k\Omega$, $C = 150pF$
 $f = (1/1.1)(10 \times 10^3 * 150 \times 10^{-12})$
 $f = 606 \text{ kHz}$

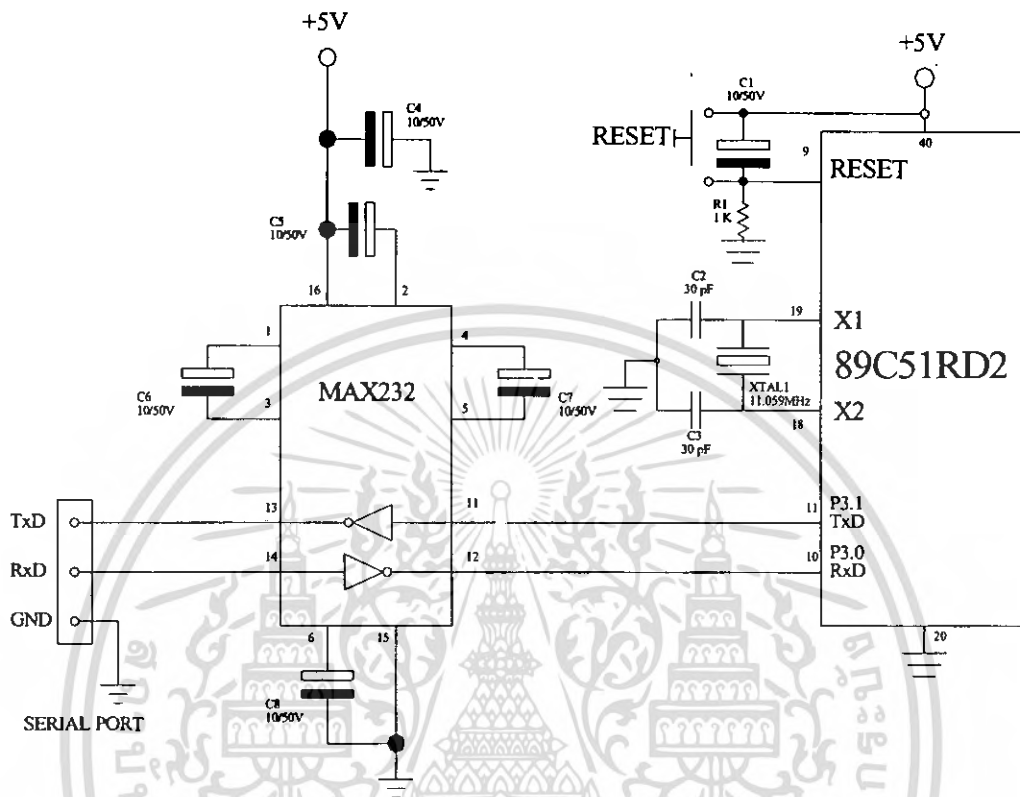
จะได้ค่า Conversion Time (T) = 110 μ S

- ขา \overline{WR} ใช้สแตร์ท A/D เพื่อเริ่มการ Conversion ขานี้ต้องกระตุ้นด้วย Low
- ขา \overline{RD} ใช้อ่านข้อมูลจาก A/D ไปเก็บโดยต้องป้อนด้วย Low
- ขา Vref/2 ใช้กำหนดช่วงกว้างแรงดันอินพุตที่เข้า A/D ในที่นี้จะปล่อยให้ลอยไว้เพื่อรับอินพุต
ช่วงกว้างได้ 5V, $V_{in(-)} = 0 \text{ v}$, Input Range (V) = 0 to 5, Step Size (mV) = $5/256 = 19.53 \text{ mV}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบวงจรรับ-ส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ของไมโครคอนโทรลเลอร์ MCS - 51

ลักษณะของวงจรรับ-ส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ของไมโครคอนโทรลเลอร์ในโครงการนี้มีลักษณะดังรูปที่ 3.9

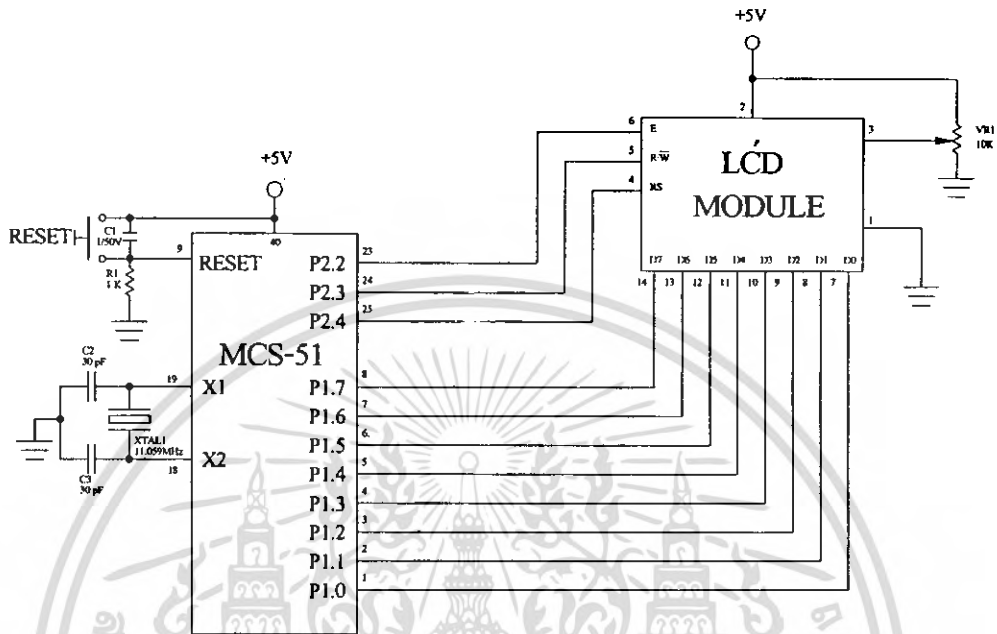


รูปที่ 3.9 วงจรการสื่อสารข้อมูลผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

จากรูปเป็นวงจรรับ-ส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ผ่านทางพอร์ต COM1 โดยใช้ไอซี MAX 232 แปลงระดับแรงดันตามมาตรฐาน RS-232 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS - 51 สามารถสื่อสารข้อมูลได้ทั้งแบบรับและส่งในเวลาเดียวกัน (Full Duplex) โดยจะใช้พอร์ต P3.0 สำหรับรับข้อมูล และพอร์ต P3.1 สำหรับส่งข้อมูล ซึ่งภายในจะมีจิสเตอร์สำหรับรับละส่งข้อมูลแยกอีกต่างหากคือ SBUF โดยก่อนการใช้งานเราต้องกำหนดโหมดการทำงานของพอร์ตอนุกรมและอัตราบอด (Baud Rate) ก่อนเสมอ

3.9 การแสดงข้อมูลบนโมดูล LCD เข้ากับไมโครคอนโทรลเลอร์

โครงการนี้เป็นารเชื่อมต่อโมดูล LCD 16 ตัวอักษร 1 บรรทัดโดยตรงเข้ากับไมโครคอนโทรลเลอร์ MCS 51 ดังแสดงในรูปที่ 3.10

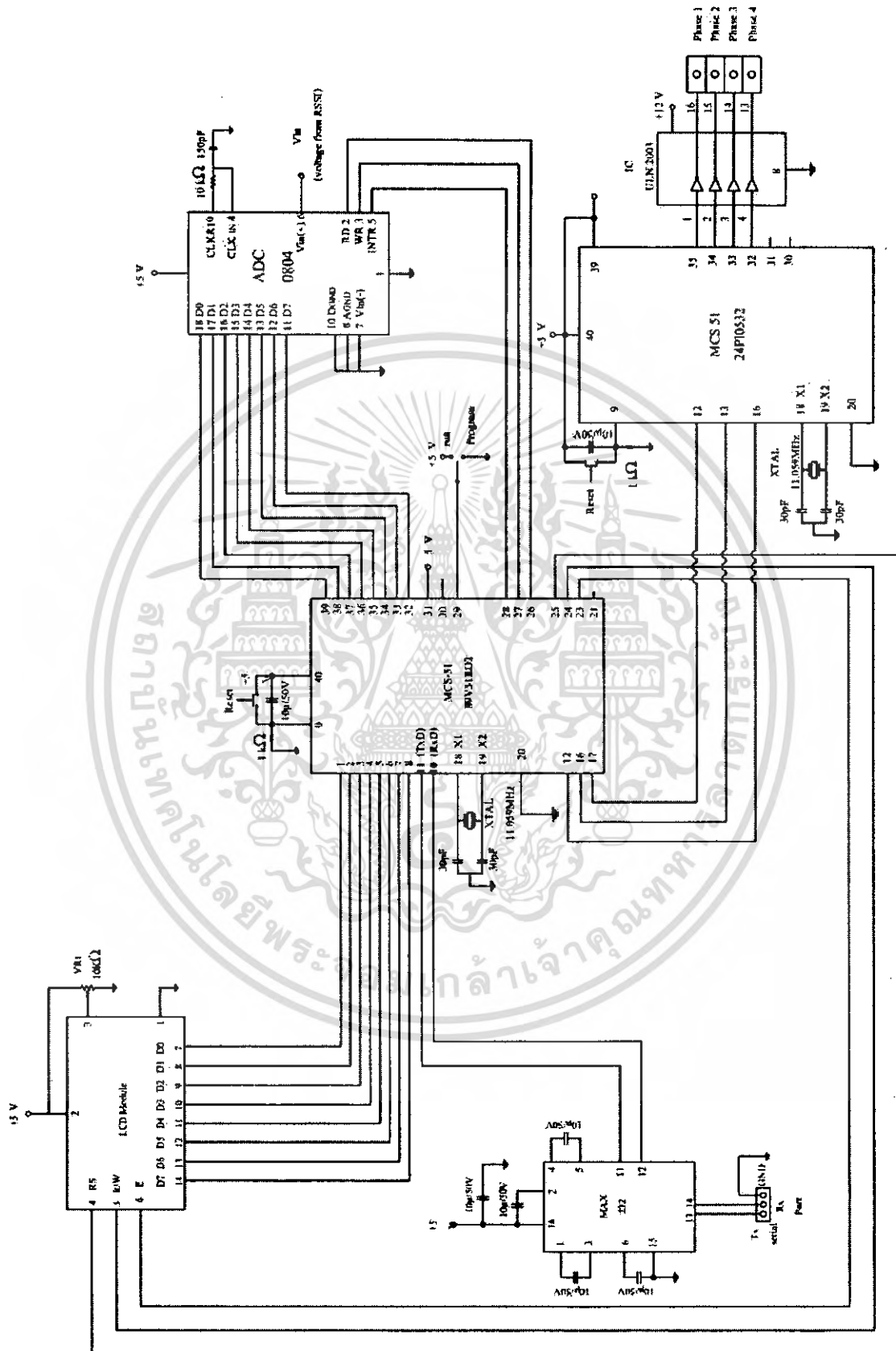


รูปที่ 3.10 แสดงการเชื่อมต่อโมดูล LCD 16 ตัวอักษร 1 บรรทัดเข้ากับไมโครคอนโทรลเลอร์ MCS 51

ในการติดต่อโมดูล LCD จะต้องมีกรหน่วงเวลาหลังจากที่การส่งรหัสคำสั่งหรือข้อมูลเพื่อให้คอนโทรลเลอร์ภายในโมดูล LCD แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป สำหรับรายละเอียดการทำงานมีดังนี้

- RS (ขา4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล
- R/ \bar{W} (ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" เป็นการอ่านข้อมูล
- E (ขา 6) : เป็นขาอินพุต LCD ให้ทำงาน
- D0-D7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

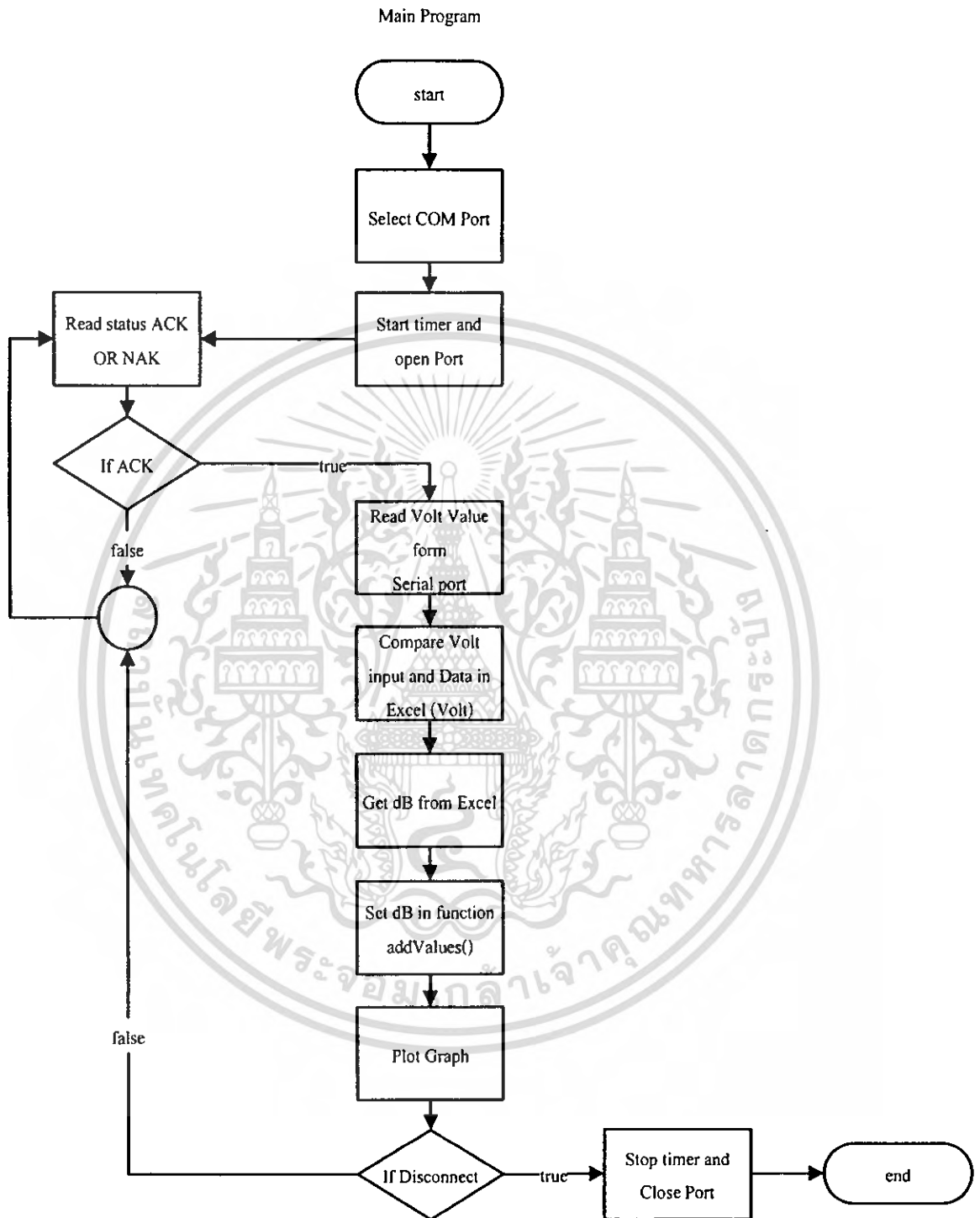
3.10 การออกแบบวงจรควบคุมสเตปปีงมอเตอร์และ รับ-ส่งข้อมูลผ่านคอนโทรลเลอร์



รูปที่ 3.11 วงจรการควบคุมสเตปปีงมอเตอร์และรับส่งข้อมูลเข้าคอมพิวเตอร์

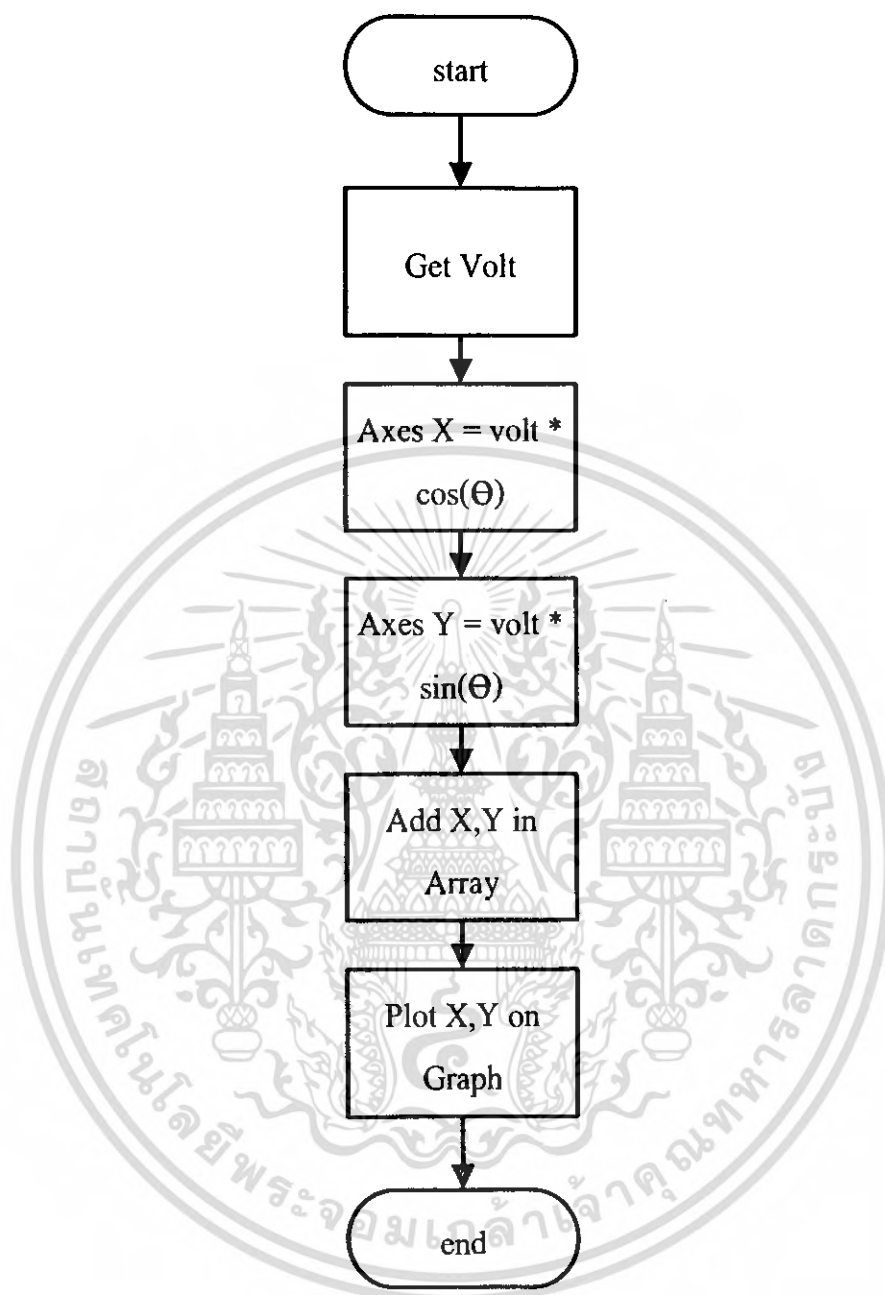
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 โปรแกรมการตรวจวัดสัญญาณและการบันทึกผลบนคอมพิวเตอร์



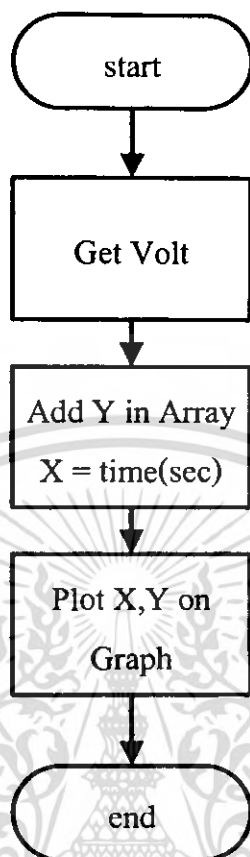
รูปที่ 3.12 โค้ดโปรแกรมการทำงานโปรแกรมหลักของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ไคอะแกรมการทำงานของโปรแกรมส่วนของการพล็อตกราฟวงกลม

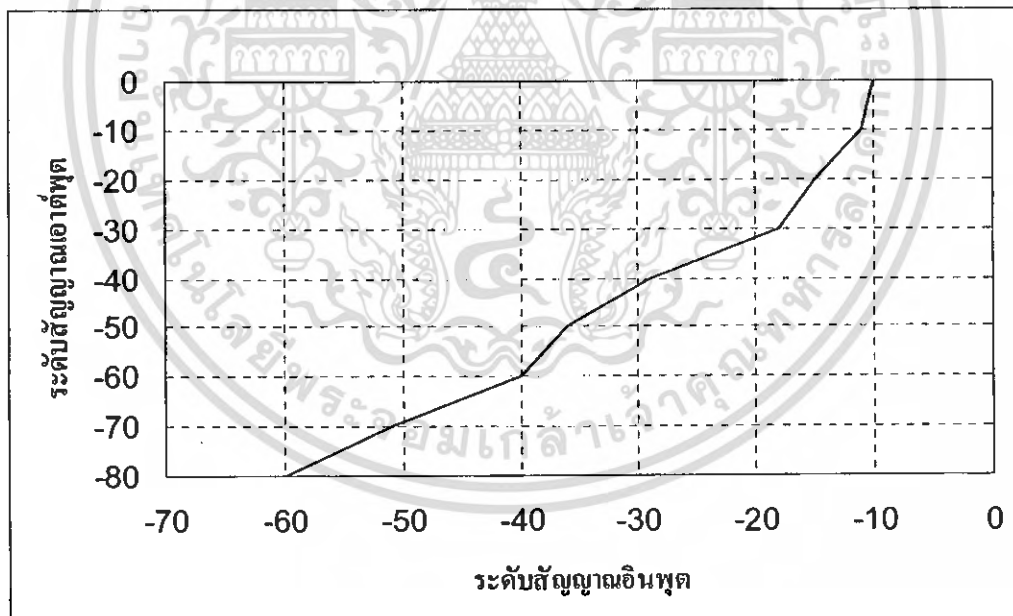
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ไคอะแกรมการทำงานของโปรแกรมส่วนของการพล็อตกราฟเส้นและกราฟบาร์

ตารางที่ 4.1 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับระดับสัญญาณเอาต์พุตที่ความถี่ 38 เมกะเฮิร์ตซ์

ระดับสัญญาณอินพุต (dBm)	ระดับสัญญาณเอาต์พุต (dBm)
0	-10
-10	-11
-20	-15
-30	-18
-40	-29
-50	-36
-60	-40
-70	-51
-80	-60

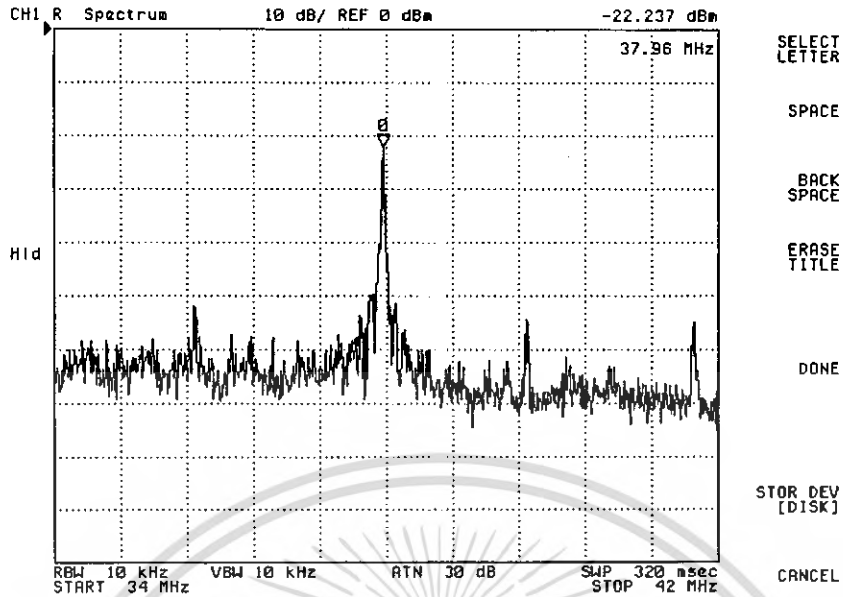


รูปที่ 4.3 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุตของวงจรขยาย

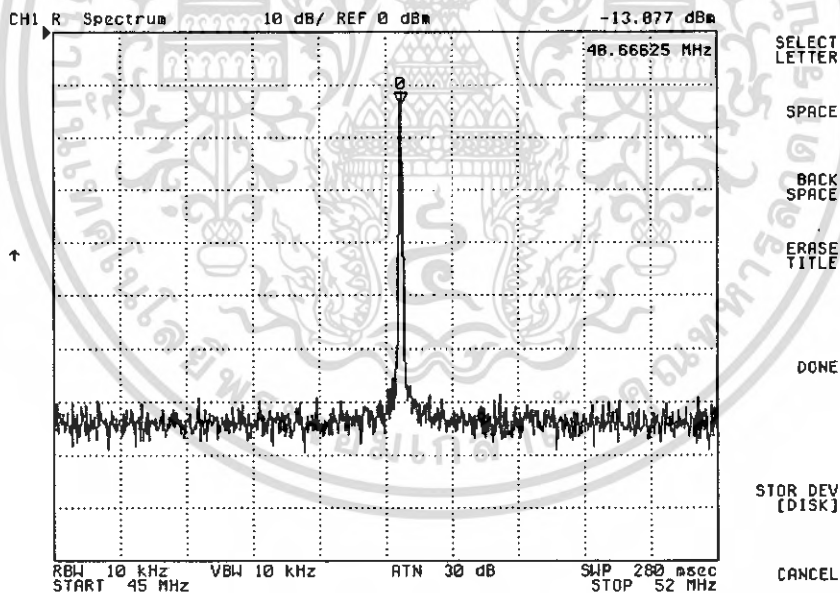
4.2 ผลการทดลองวงจรมิกเซอร์ (Mixer)

วงจรมิกเซอร์ทำหน้าที่ลดความถี่ที่รับเข้ามาจากสัญญาณภาพ (IF ภาพ) และจากเครื่องส่งเพราะที่ความถี่ (Signal Generator) เพื่อนำมาผสมกันให้เป็นความถี่กลางที่ 10.7 เมกะเฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



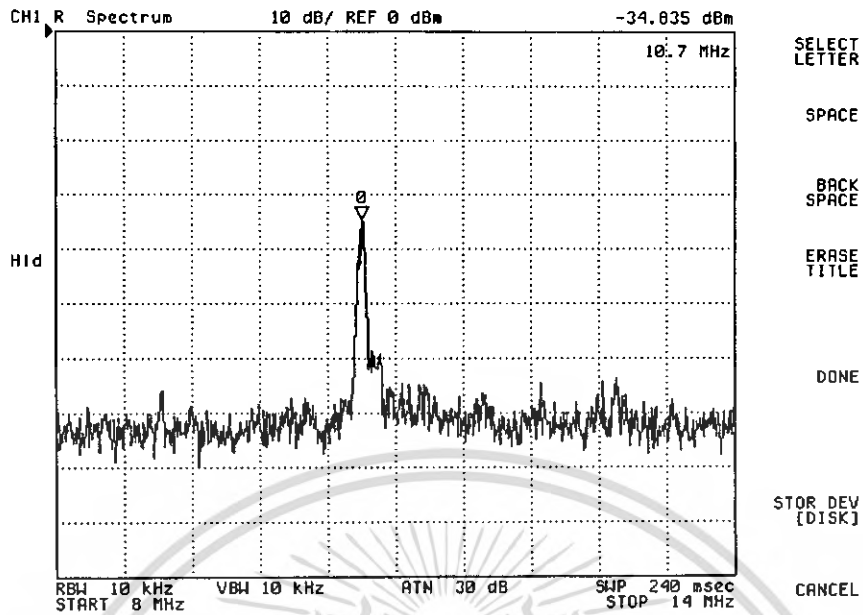
รูปที่ 4.4 สเปกตรัมของสัญญาณภาพที่ออกจากจูนเนอร์ (IF Tuner) 37.96 เมกะเฮิร์ตซ์



รูปที่ 4.5 สเปกตรัมของสัญญาณจากเครื่องส่งเคราะห์ความถี่ (Signal Generator) 48.66 เมกะเฮิร์ตซ์

เมื่อได้ความถี่ของสัญญาณภาพและความถี่จากเครื่องส่งเคราะห์ความถี่แล้วนำมาผสมกันในวงจรมิกเซอร์จะได้สัญญาณเอาต์พุตคิงรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 สเปกตรัมของสัญญาณเอาต์พุตที่ได้จากวงจรมิกเซอร์

4.3 ผลการทดลองวงจรกรองแถบความถี่ 10.7 เมกะเฮิร์ตซ์

ผลการวัดการตอบสนองความถี่ของวงจรกรองแถบความถี่ (Ceramic Filter) 10.7 เมกะเฮิร์ตซ์

ได้ผลการทดลองดังรูปที่ 4.7

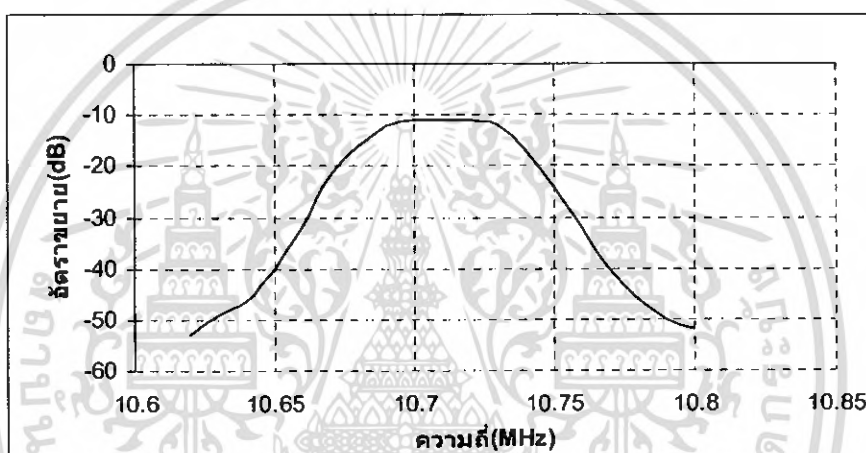
ตารางที่ 4.2 ตารางผลตอบสนองความถี่ของวงจรกรองแถบความถี่ 10.7 เมกะเฮิร์ตซ์

ความถี่ (MHz)	อัตราขยาย (dB)
10.62	-53
10.63	-49
10.64	-46
10.65	-40
10.66	-32
10.67	-22
10.68	-16
10.69	-12
10.70	-11
10.71	-11
10.72	-11
10.73	-12
10.74	-17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ)

ความถี่ (MHz)	อัตราขยาย (dB)
10.75	-24
10.76	-32
10.77	-40
10.78	-46
10.79	-50
10.80	-52



รูปที่ 4.7 กราฟผลตอบสนองความถี่ของวงจรความถี่ 10.7 เมกะเฮิรตซ์

4.4 ผลการทดลองวงจรมายสัญญาณความถี่กลางที่ 10.7 เมกะเฮิรตซ์

ทำการเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุตที่ความถี่ 10.7 เมกะเฮิรตซ์

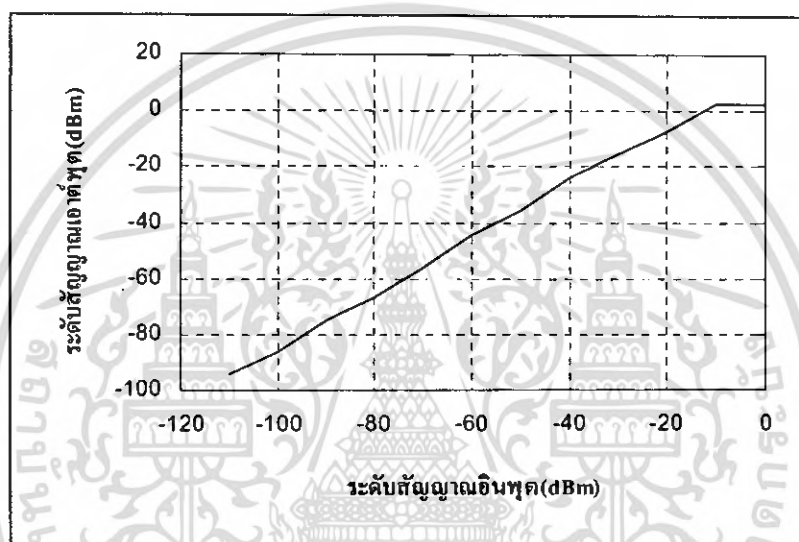
ตารางที่ 4.3 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับสัญญาณเอาต์พุต

ระดับสัญญาณอินพุต (dBm)	ระดับสัญญาณเอาต์พุต (dBm)
0	3
-10	3
-20	-7
-30	-15
-40	-24
-50	-36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 (ต่อ)

ระดับสัญญาณอินพุต (dBm)	ระดับสัญญาณเอาต์พุต (dBm)
-60	-44
-70	-56
-80	-67
-90	-75
-100	-86
-110	-94



รูปที่ 4.8 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุตของวงจรขยายสัญญาณ ความถี่กลาง 10.7 เมกะเฮิร์ตซ์

4.5 การทดลองวงจรเอจีสซีของจูนเนอร์

เมื่อค่าความแรงของสัญญาณมีค่าน้อยลงจะทำให้แรงดันเอจีสซีปรับค่าเพิ่มขึ้นเพื่อไปควบคุมระดับสัญญาณทางเอาต์พุตให้ได้มาตรฐาน

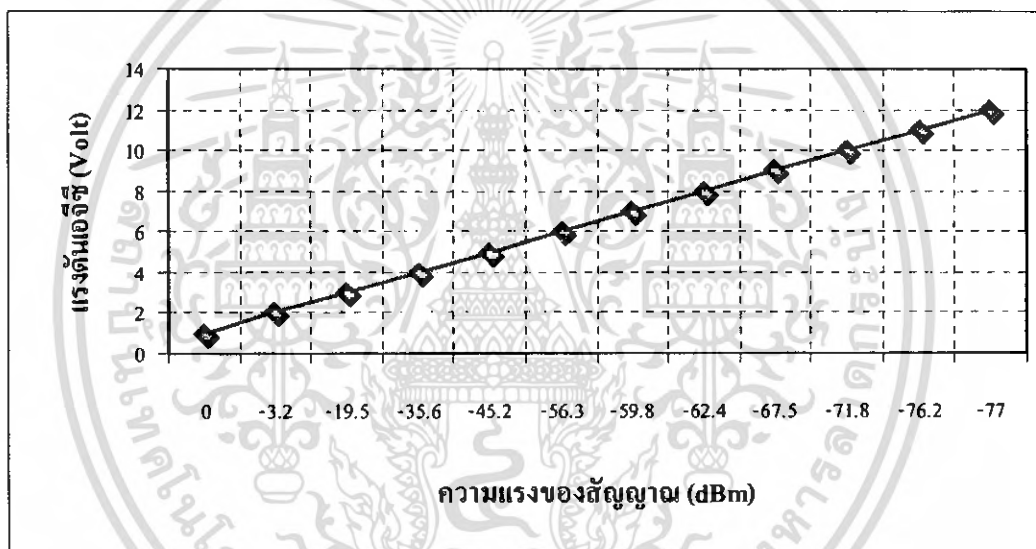
ตารางที่ 4.4 การเปรียบเทียบระหว่างแรงดันเอจีสซีของจูนเนอร์และความแรงของสัญญาณ

แรงดันเอจีสซี (Volt)	ความแรงของสัญญาณ (dBm)
1	0
2	-3.2
3	-19.5
4	-35.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 (ต่อ)

แรงดันเอจซี (Volt)	ความแรงของสัญญาณ (dBm)
5	-45.2
6	-56.3
7	-59.8
8	-62.4
9	-67.5
10	-71.8
11	-76.2
12	-77



รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างแรงดันเอจซีกับความแรงของสัญญาณ

4.6 ผลการทดลองวงจรวัดความแรงสัญญาณ

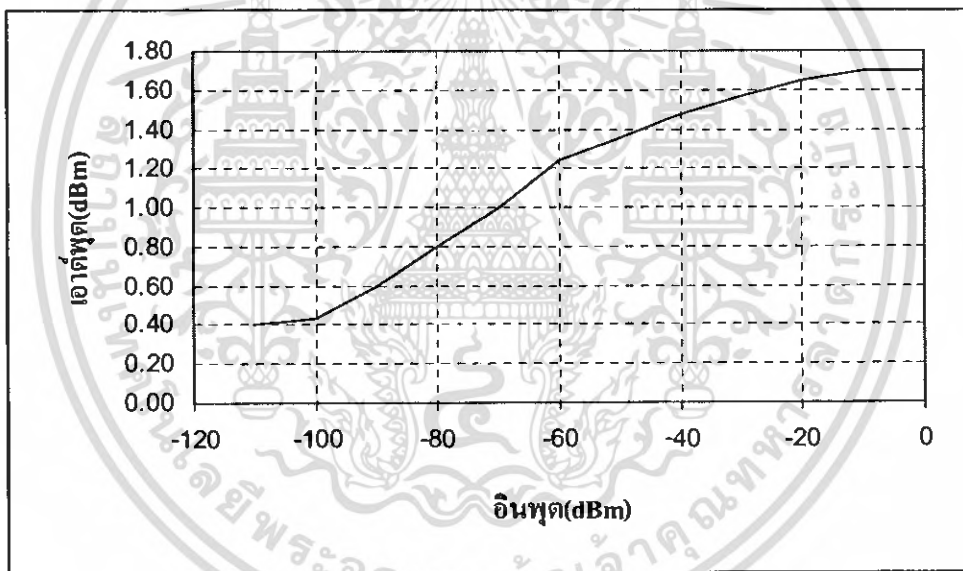
หลังจากที่สัญญาณผ่านวงจรขยายความถี่กลาง 10.7 เมกะเฮิร์ตซ์แล้ว สัญญาณจะถูกส่งมาที่วงจรวัดความแรงสัญญาณ โดยใช้ไอซี SA 636 และจะแสดงผลออกมาในรูปของสัญญาณไฟตรงได้ผลการทดลองดังนี้

ตารางที่ 4.5 ตารางเปรียบเทียบระหว่างระดับของสัญญาณอินพุตกับระดับสัญญาณเอาต์พุต

อินพุต (dBm)	เอาต์พุต (Volt)
0	1.70
-10	1.70
-20	1.65

ตารางที่ 4.5 (ต่อ)

อินพุต (dBm)	เอาต์พุต (Volt)
-30	1.57
-40	1.48
-50	1.36
-60	1.24
-70	1.00
-80	0.80
-90	0.60
-100	0.43
-110	0.40



รูปที่ 4.10 กราฟเปรียบเทียบระหว่างระดับสัญญาณอินพุตกับสัญญาณเอาต์พุตของวงจรวัดความแรงสัญญาณ

4.7 ผลการทดลองวงจรแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอล (Analog to Digital Converter) แล้วแสดงผลที่หน้าจอ LCD

ตารางที่ 4.6 ตารางเปรียบเทียบอินพุตที่มีค่าเป็นแรงดันกับเอาต์พุตที่ได้ค่าเป็นดิจิตอล

ระดับแรงดัน (0-5V)	จอ LCD แสดงผล							
0	0	0	0	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

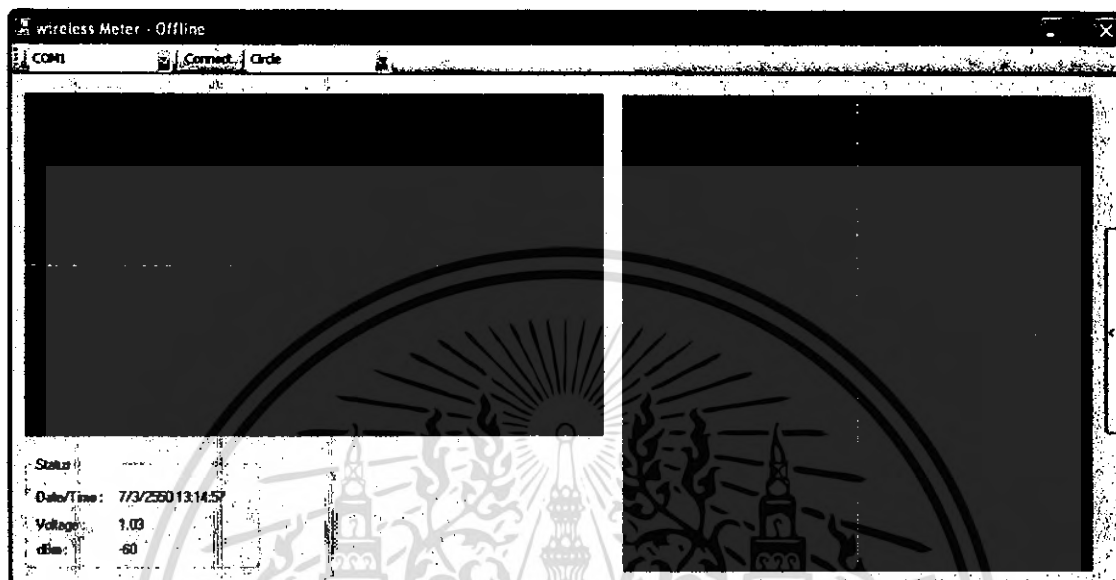
ตารางที่ 4.6 (ต่อ)

ระดับแรงดัน (0-5V)	จอ LCD แสดงผล							
0.2	0	0	0	0	1	0	0	0
0.4	0	0	0	1	0	0	1	1
0.6	0	0	0	1	1	1	0	0
0.8	0	0	1	0	1	0	0	0
1.0	0	0	1	1	0	0	1	0
1.2	0	0	1	1	1	1	0	0
1.4	0	1	0	0	0	1	1	1
1.6	0	1	0	1	0	0	1	0
1.8	0	1	0	1	1	1	0	0
2.0	0	1	1	0	0	1	1	0
2.2	0	1	1	1	0	0	0	1
2.4	0	1	1	1	1	0	1	1
2.6	1	0	0	0	0	1	0	0
2.8	1	0	0	1	0	0	0	0
3.0	1	0	0	1	1	0	1	0
3.2	1	0	1	0	0	1	0	1
3.4	1	0	1	0	1	1	1	1
3.6	1	0	1	1	1	0	1	0
3.8	1	1	0	0	0	1	0	0
4.0	1	1	0	0	1	1	1	0
4.2	1	1	0	1	1	0	0	0
4.4	1	1	1	0	0	0	1	0
4.6	1	1	1	0	1	1	0	1
4.8	1	1	1	1	1	0	0	0
5.0	1	1	1	1	1	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 แสดงผลของโปรแกรมการหาทิศทางของสัญญาณ เมื่อทำการรับค่าจากไมโครคอนโทรลเลอร์

จากรูปจะเห็นได้ว่าโปรแกรมถูกแบ่งออกเป็น 2 ส่วนคือกราฟวงกลมจะใช้ในการหาทิศทางที่แรงที่สุดของสัญญาณแต่ละช่องความถี่ที่ทำการวัดและกราฟแสดงระดับของสัญญาณเมื่อมีการเปลี่ยนแปลงพร้อมทั้งการบอกสถานะของสัญญาณว่ามีค่าเท่าไร วัน เวลาไหนที่ทำการบันทึก



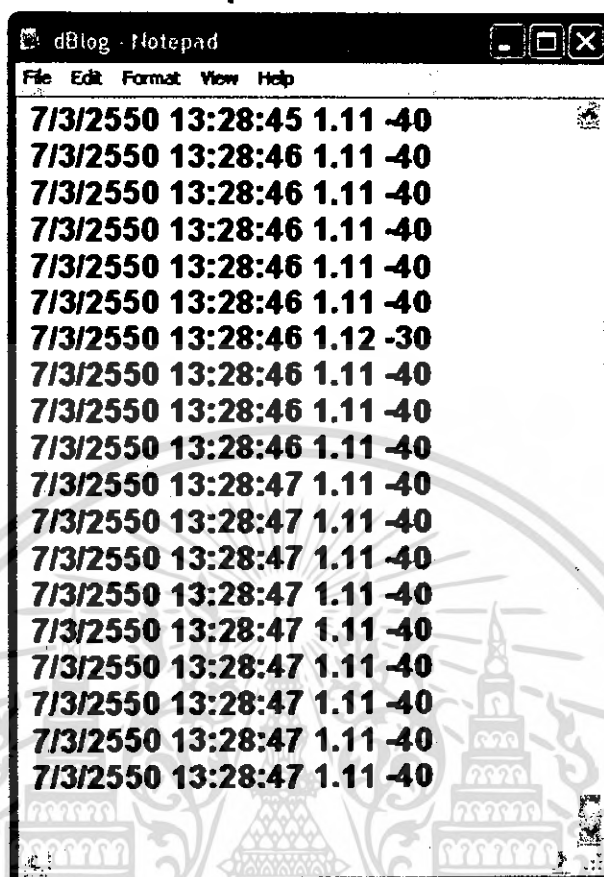
รูปที่ 4.11 โปรแกรมการหาทิศทางและการแสดงสถานะของค่าที่ทำการวัด



รูปที่ 4.12 กราฟเส้นแสดงการเปลี่ยนแปลงของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 แสดงผลของการเก็บบันทึกค่าสถานะต่างๆขณะที่ทำการวัด



```

dBlog - Notepad
File Edit Format View Help
7/3/2550 13:28:45 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.12 -30
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:46 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40
7/3/2550 13:28:47 1.11 -40

```

รูปที่ 4.13 รูปของการบันทึกค่าสถานะต่างๆที่ทำการตรวจวัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทสรุปและวิจารณ์

จากที่ผ่านมาเราได้ทำการออกแบบและสร้างวงจรโดยอาศัยทฤษฎีในบทที่ 2 ในการวิเคราะห์วงจร จากนั้นจึงเก็บผลที่ได้ทำการสร้างขึ้นพบว่า

วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัลจะแปลงระดับโวลต์ตรงที่ป้อนให้จาก 0 ถึง 5 โวลต์ ให้เปลี่ยนเป็นรหัสไบนารีจาก 00000000 ถึง 11111111 จากนั้นได้เขียนโปรแกรมภาษาแอสเซมบลีให้เป็นรหัสไบนารีเป็นระดับเลขฐานสิบซึ่งมีค่า 0 ถึง 255 อีกที

วงจรขั้วมอเตอร์สามารถหมุนตามสเต็ปที่ต้องการได้แสดงว่าโปรแกรมควบคุมการทำงานด้วยภาษาแอสเซมบลีนั้นทำงานได้อย่างถูกต้อง และวงจรขั้วสเต็ปปีงมอเตอร์สามารถขับกระแสให้กับมอเตอร์ได้อย่างเพียงพอ

วงจรวัดความแรงของสัญญาณความถี่กลาง 10.7 เมกะเฮิรตซ์ มาป้อนให้กับวงจร RSSI โดยใช้ไอซี SA 636 ซึ่งจะได้ค่าออกมาในรูปของแรงดันไฟตรงโดยข้อมูลที่ได้อาจจะถูกส่งต่อไปยังคอมพิวเตอร์เพื่อทำการเขียนกราฟวงกลมในการหาทิศทางความแรงของสัญญาณในทิศทางที่แรงที่สุดและเขียนกราฟเส้นเพื่อดูผลกระทบจากสิ่งแวดล้อมที่เกิดขึ้นในทิศทางที่ตั้งสายอากาศรับสัญญาณ

ปัญหาที่พบ

- วงจรกรองความถี่สามารถกรองความถี่ได้ตามที่ต้องการแต่เกิดการลดทอนและมีแบนด์วิดท์ไม่ตรงตามการคำนวณ

- เกิดการคลาดเคลื่อนระหว่างเฟืองทดกับเฟืองของสเต็ปมอเตอร์

- อุปกรณ์ที่ต้องใช้กับ IC เบอร์ SA 636 ในวงจร RSSI(Received Signal Strength Indicator) นั้นค่อนข้างหายาก

- วงจรมิกเซอร์ยังเกิดปัญหาในเรื่องของสัญญาณจากจูนเนอร์ที่จะนำมาผสมนั้นเป็นสัญญาณภาพที่มีการมอดูเลตแบบ AM ทำให้สัญญาณไม่หยุดนิ่ง

แนวทางการแก้ไขปัญหา

- ในส่วนของวงจรกรองนั้นจากปัญหาข้างต้นการแก้ไขจะทำโดยพยายามให้อุปกรณ์ที่มีคุณสมบัติใกล้เคียงกับค่าที่คำนวณมากที่สุด

- ทำให้เกิดความคลาดเคลื่อนระหว่างเฟืองทดกับเฟืองของสเต็ปมอเตอร์ให้เกิดการคลาดเคลื่อนน้อยที่สุด

- ส่วนของวงจรมิกเซอร์เมื่อสัญญาณภาพไม่นิ่งทำให้ต้องปรับความถี่จากเครื่องส่งเพราะหาค่าความถี่ให้ผสมกันได้พอดีอยู่ตลอดเวลาเมื่อทำการทดลอง

หนังสืออ้างอิง

- บุญชาติ เนติศักดิ์. 2541. ทฤษฎีและปฏิบัติเครื่องรับโทรทัศน์ เล่ม 1. กรุงเทพฯ : ซีเอ็ดบุ๊คเซ็น.
- สมยศ จุณณะปิยะ. 2546. การใช้งานไมโครคอลโทรลเลอร์ตระกูล MCS-51. คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- Samuel, Y. L., 1987. **Microwave circuit Analysis and Amplifier Design**. United States :
Prentice -Hall International INC.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรมคอนโทรลเลอร์ในการแสดงผลที่จอ LCD ส่งออกพอร์ตอนุกรม

```

ORG            0000H

MOV            SCON, #0F0H    ; SERIAL (MODE 2), REN=0
MOV            PCON, #00H    ; SMOD =0
MOV            TMOD, #20H    ; TIMER1(MODE 2)
MOV            TH1, #0FDH    ; 9600 BAUD
SETB          TR1            ; (1)
SETB          TB8            ; TB8=1      (1)

MOV            R2, #04H
ACALL         DELAY

MOV            A, #00000110B  ; ENTRY MODE SET  I/D=1
ACALL         LCD_INS
MOV            A, #00000001B  ; CLEAR DISPLAY
ACALL         LCD_INS
MOV            R2, #01H
ACALL         DELAY
MOV            A, #00111100B  ; FUNCTION SET DL=1, N=1,
F=0(5*8)
ACALL         LCD_INS
MOV            A, #00001100B  ; DISPLAY=ON, CURSOR=OFF,
BLINK=OFF
ACALL         LCD_INS

MAIN:
MOV            A, #80H
ACALL         LCD_INS

WAIT:
SETB          P2.7
JB            P2.7, WAIT

ACALL         GET_DATA
MOV            R2, #8
LOOP:
ACALL         CONV
DJNZ         R2, LOOP

MOV            R1, #0DH
ACALL         SERIAL

MOV            R1, #0AH
ACALL         SERIAL

SJMP         MAIN

;+++++++ CONVERT DATA ++++++++
CONV:
MOV            A, R0
RLC
MOV            R0, A
JC            ZERO
MOV            A, #31H
MOV            R1, A
ACALL         LCD_DATA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ACALL SERIAL
                RET

ZERO:           MOV          A, #30H
                MOV          R1, A
                ACALL LCD_DATA
                ACALL SERIAL
                RET

;+++++++ LCD WRITE INSTRUCTION CODE ++++++
LCD_INS:       MOV          P1, A

                CLR          P2.4
                CLR          P2.3
                CLR          P2.2

                SETB        P2.2
                NOP
                NOP
                CLR          P2.2

LCDINS:        MOV          A, #0
                DEC          A
                JNZ          LCDINS
                RET

;+++++++ LCD WRITE DATA ++++++
LCD_DATA:      MOV          P1, A

                SETB        P2.4
                CLR          P2.3
                CLR          P2.2

                SETB        P2.2
                NOP
                NOP
                CLR          P2.2

LCDDDATA:      MOV          A, #0
                DEC          A
                JNZ          LCDDDATA
                RET

;+++++++ GET DATA FORM ADC ++++++

GET_DATA:      MOV          P0, #0FFH
                SETB        P2.5
                SETB        P2.6

                CLR          P2.6
                NOP
                SETB        P2.6

FINISH:        JB           P2.7, FINISH
                CLR          P2.5
                MOV          A, P0
                MOV          R0, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETB P2.5
RET

;+++++++ DELAY ++++++
DELAY:      MOV      R3,#00H
DELAY1:    MOV      R4,#00H
           DJNZ    R4,$
           DJNZ    R3,DELAY1
           DJNZ    R2,DELAY
           RET

;+++++++ SERIAL ++++++
SERIAL:    MOV      A,R1
           MOV      SBUF,A      ; SEND DATA FORM P1
TO SERIAL          (2)
WAIT2:    JNB     TI,WAIT2      ; CHECK TO FINISH      (2)
           CLR     TI
           RET
END

```

2. โปรแกรมการแสดงผลที่คอมพิวเตอร์

ส่วน Interface

```

cmd Cmd = new cmd();
string read_data;

private string B2D(string bin)
{
    try
    {
        return Convert.ToString(Convert.ToInt32(bin, 2), 10);
    }
    catch
    {
        return "0";
    }
}

private void Connect()
{
    toolStripButton1.Text = "Disconnect";
    Cmd.ClearTable();
    try
    {
        toolStripComboBox1.Enabled = false;
        serialPort1.PortName = toolStripComboBox1.Text;
        serialPort1.Open();
        timer2.Start();
        timer1.Start();
        if (serialPort1.IsOpen == true)
            this.Text = "Online";
    }
    catch
    {
        Disconnect();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MessageBox.Show(this, @"เครื่องของท่านไม่มี Port " +
        toolStripComboBox1.Text, @"เกิดข้อผิดพลาด", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

```

```

private void Disconnect()
{
    toolStripComboBox1.Enabled = true;
    toolStripButton1.Text = "Connect";
    serialPort1.Close();
}

```

ส่วนของ Command

```

DataTable table = new DataTable("Data_Logger");

public void CreateTable()
{
    DataColumn column;

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "_datetime";
    table.Columns.Add(column);

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "Volt";
    table.Columns.Add(column);

    column = new DataColumn();
    column.DataType = Type.GetType("System.String");
    column.ColumnName = "dB";
    table.Columns.Add(column);
}

public void insert_logger(string _dtype, string volt,
string dB)
{
    DataRow row;
    row = table.NewRow();
    row["_datetime"] = _dtype;
    row["Volt"] = volt;
    row["dB"] = dB;
    table.Rows.Add(row);
}

public void ClearTable()
{
    table.Clear();
}

public DataTable databind()
{
    return table;
}

public void save_File() //Method save data logger ที่ตั้งอยู่ใน
Memory ในกรณีที่มี array เติม
{
    string path = Application.StartupPath + @"\dBLog.log";
    //กำหนดชื่อของ File ที่ต้องการ save
    if (!File.Exists(path)) //check ว่ามี File ที่กำหนดว่ามีอยู่แล้วหรือยัง
    {
        //ถ้ายังไม่มีให้ Create File ขึ้นมาใหม่
        using (StreamWriter sw = File.CreateText(path))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        for (int i = 0; i < table.Rows.Count; i++)
            sw.WriteLine(
                table.Rows[i]["datetime"] + " " +
                table.Rows[i]["Volt"] + " " +
                table.Rows[i]["dB"]
            );
    }
}
else
{
    //กรณีที่มีแถวจะทำการเขียน Data ต่อจากเดิม
    using (StreamWriter sw = File.AppendText(path))
    {
        for (int i = 0; i < table.Rows.Count; i++)
            sw.WriteLine(
                table.Rows[i]["datetime"] + " " +
                table.Rows[i]["Volt"] + " " +
                table.Rows[i]["dB"]
            );
    }
}
}

-> dB
public float ConvertV2dB(double Volt) //Method แปลง Voltage
{
    //การคำนวณ จะ ref ที่ 1 Volt 0 volt จะไม่นิยามถ้าเป็น 0 เข้ามาจะสมมติค่าให้มีค่า=1
    float dB;
    if (Volt == 0)
        dB = (float)Math.Round(-20 * Math.Log(1), 3);
    else
        dB = (float)Math.Round(-20 * Math.Log(Volt), 3);
    return dB; //เมื่อคำนวณเสร็จจะทำการส่งค่าออกไปให้ ส่วนที่เรียกใช้ Method นี้
}

ส่วนของ Chart
/* Public Properties */
public float Maximum
{
    get
    {
        return m_Maximum;
    }
}

public float Minimum
{
    get
    {
        return m_Minimum;
    }
}

public float[] values
{
    set
    {
        Values = value;
    }
}

/* Public Vars */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public int LineWidth;
public int PixelsPer1; //Per One Gridbox
public int LineDifference;
public float ValueMultiplier; //in order to make values
larger if you need to improve scale
public Color AboveColor, UnderColor, GridColor,
ChartBackColor, AxesColor;
public ChartControlOpenType OpenType;

/* Private Vars */

private Graphics g;
private float[] Values;
private float m_Maximum, m_Minimum;
private int CurrentYGridStart;
private int CurrentNumberOfValues;
private Size CurrentSize = new Size(0, 0);

public enum ChartControlOpenType
{
    Bar,
    Graph
};

public Chart()
{
    InitializeComponent();

    OpenType = ChartControlOpenType.Graph; //Default Value
    LoadDefaultValues();
    InitChart();
}

public Chart(ChartControlOpenType NOpenType)
{
    // This call is required by the Windows.Forms Form
    Designer.
    InitializeComponent();

    OpenType = NOpenType;
    LoadDefaultValues();
    InitChart();
}

private void LoadDefaultValues()
{
    /* Loading Default Values */

    g = ChartPanel.CreateGraphics();
    PixelsPer1 = 10;
    ChartBackColor = Color.Black;
    GridColor = Color.Green;
    AboveColor = Color.Chartreuse;
    UnderColor = Color.Red;
    AxesColor = Color.White;
    CurrentYGridStart = 0;
    ValueMultiplier = 1;
    m_Maximum = ChartPanel.Size.Height / 2;
    m_Minimum = (-1) * (ChartPanel.Size.Height / 2);
    LineDifference = 1;

    /* Initializing Value Array (Size is width since every
    Line width is 1 pixel) */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (OpenType == ChartControlOpenType.Bar) //WIDTH
because there's no space between lines
            Values = new float[ChartPanel.Size.Width];
        else
            Values = new float[ChartPanel.Size.Width / 2];
//WIDTH/2 because there's a 2 pixel space between dots

        for (int i = 0; i < Values.Length; i++)
            Values[i] = 0;

        CurrentNumberOfValues = 0;
    }

    public void RefreshControl()
    {
        PostInitChart();
        DrawChart();
    }

    public void AddValue(float val)
    {
        /* Adding value at end of array */
        if ((Minimum != 0) && (Maximum != 0)) /* Prevent first
draw errors */
            if ((val * ValueMultiplier > Maximum) || (val *
ValueMultiplier < Minimum))
                return; //Value is too high or too low for
display.
            for (int i = 0; i < Values.Length - 1; i++)
                Values[i] = Values[i + 1];
            Values[Values.Length - 1] = val * ValueMultiplier;
            if (CurrentNumberOfValues < Values.Length)
                CurrentNumberOfValues++;
1) if (CurrentYGridStart < PixelsPer1 * LineDifference -
        {
            if (OpenType == ChartControlOpenType.Bar)
                CurrentYGridStart++;
            else
                CurrentYGridStart += 2;
        }
        else
        {
            CurrentYGridStart = 0;
        }

        /* Redrawing chart */
        DrawChart();
    }

    public void LoadFromValues(ArrayList NewValues)
    {
        Values = new float[NewValues.Count];

        for (int i = 0; i < NewValues.Count; i++)
            Values[i] = ValueMultiplier *
(float)Convert.ToDouble(NewValues[i]);

        PostInitChart();
    }

    public void LoadFromValues(float[] NewValues)
    {
        for (int i = 0; i < NewValues.Length; i++)
            Values[i] = NewValues[i] * ValueMultiplier;
        PostInitChart();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void InitChart()
{
    /* First Time Chart Init */
    CurrentYGridStart = 0;
    PostInitChart();
}

int IntCmp(float num1, float num2)
{
    /* Compares 2 floats,
    * if num1 > 0 and num2 < 0, returns 1,
    * \if both are > 0 or both are < 0 returns 0,
    * if num2 > 0 and num1 < 0 returns -1
    */

    if ((num1 >= 0) && (num2 >= 0))
        return 0;
    if ((num1 < 0) && (num2 < 0))
        return 0;
    if ((num1 >= 0) && (num2 < 0))
        return 1;
    if ((num1 < 0) && (num2 >= 0))
        return -1;

    return 0;
}

public void ChangeInnerChartControlSize(Size NewSize)
{
    this.Size = NewSize;
    if (ChartPanel != null)
        PostInitChart();
}

public void PostInitChart()
{
    /* Refresh of Component, Clear and redraw of grid,
    which also redraws values */

    if ((ChartPanel.Height != 0) && (ChartPanel.Width !=
0)) //To avoid drawing a 0 width or 0 size rectangle
    {
        g.Clear(ChartBackColor);

        DrawGrid();
    }
}

private void DrawGrid()
{
    /* Drawing X Grid */

    for (int i = (ChartPanel.Size.Height / 2) + PixelsPer1
* LineDifference; i < ChartPanel.Size.Height; i += PixelsPer1 *
LineDifference)
        g.DrawLine(new Pen(GridColor), 0, i,
ChartPanel.Size.Width, i);
    for (int i = (ChartPanel.Size.Height / 2) - PixelsPer1
* LineDifference; i > 0; i -= PixelsPer1 * LineDifference)
        g.DrawLine(new Pen(GridColor), 0, i,
ChartPanel.Size.Width, i);

    /* Drawing Y Grid */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (int i = CurrentYGridStart; i <
ChartPanel.Size.Width; i += PixelsPer1 * LineDifference)
            g.DrawLine(new Pen(GridColor), i, 0, i,
ChartPanel.Size.Height);

        /* Drawing Axes */

        g.DrawLine(new Pen(AxesColor), 0,
(int)(ChartPanel.Size.Height / 2), ChartPanel.Size.Width,
(int)(ChartPanel.Size.Height / 2));
    }

private void DrawChart()
{
    PostInitChart(); //Refresh of the chart

    Pen AbovePen = new Pen(AboveColor);
    Pen UnderPen = new Pen(UnderColor);

    if (OpenType == ChartControlOpenType.Bar) //Drawing
bar
    {
        for (int i = Values.Length -
CurrentNumberOfValues; i < Values.Length; i++)
        {
            if (Values[i] > 0) // More than 0, AboveColor
is used
            {
                g.DrawLine(AbovePen, Values.Length - i -
1, (int)(ChartPanel.Size.Height / 2) - 1, Values.Length - i - 1,
(int)(ChartPanel.Size.Height / 2) - Values[i]);
            }
            if (Values[i] < 0) // Less than 0, UnderColor
is used
            {
                g.DrawLine(UnderPen, Values.Length - i -
1, (int)(ChartPanel.Size.Height / 2) + 1, Values.Length - i - 1,
(int)(ChartPanel.Size.Height / 2) - Values[i]);
            }
        }
    }
    else if (OpenType == ChartControlOpenType.Graph)
//Drawing Graph
    {
        for (int i = Values.Length -
CurrentNumberOfValues; i < Values.Length; i++)
        {
            if (Values[i] >= 0) // More than 0, AboveColor
is used
            {
                if (IntCmp(Values[i], Values[i - 1])) > 0)
//if it goes bellow 0
                {
                    g.DrawLine(UnderPen, (Values.Length -
i) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i - 1],
(Values.Length - i) * 2 - 1, (int)(ChartPanel.Size.Height / 2));
                    g.DrawLine(AbovePen, (Values.Length -
i) * 2 - 1, (int)(ChartPanel.Size.Height / 2), (Values.Length - i
- 1) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i]);
                }
                else //if it stays above 0
                {
                    g.DrawLine(AbovePen, (Values.Length -
i) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i - 1],
(Values.Length - i - 1) * 2, (int)(ChartPanel.Size.Height / 2) -
Values[i]);
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    }
    if (Values[i] < 0) // Less than 0, UnderColor
is used
    {
        if (IntCmp(Values[i], Values[i - 1]) < 0)
//if it goes above 0
        {
            g.DrawLine(AbovePen, (Values.Length -
i) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i - 1],
(Values.Length - i) * 2 - 1, (int)(ChartPanel.Size.Height / 2));
            g.DrawLine(UnderPen, (Values.Length -
i) * 2 - 1, (int)(ChartPanel.Size.Height / 2), (Values.Length - i
- 1) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i]);
        }
        else //if it stays bellow 0
        {
            g.DrawLine(UnderPen, (Values.Length -
i) * 2, (int)(ChartPanel.Size.Height / 2) - Values[i - 1],
(Values.Length - i - 1) * 2, (int)(ChartPanel.Size.Height / 2) -
Values[i]);
        }
    }
}
}
//Dont forget to dispose variables...
UnderPen.Dispose();
AbovePen.Dispose();
}
private void RecalculateSize()
{
    if ((CurrentSize.Height != 0) && (CurrentSize.Width !=
0)) //avoid divide by 0
    {
        m_Maximum = ChartPanel.Size.Height / 2;
        m_Minimum = (-1) * (ChartPanel.Size.Height / 2);

        float SizeChange = ((float)Size.Height /
(float)CurrentSize.Height);

        if (Size.Height != 0)
            ValueMultiplier *= SizeChange;

        int i, j;

        float[] NewValues = new float[Size.Width];

        for (i = Values.Length - 1, j = NewValues.Length -
1; ((i >= 0) && (j >= 0)); i--, j--)
        {
            if (SizeChange != 0)
                NewValues[j] = Values[i] * SizeChange;
        }

        Values = NewValues;

        g.Dispose();
        g = ChartPanel.CreateGraphics();

        DrawChart();
    }
}
protected override void OnResize(EventArgs e)
{
    base.OnResize(e);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (ChartPanel != null)
{
    if ((Size.Height == 0) || (Size.Width == 0))
        return;

    if ((CurrentSize.Height == 0) &&
(CurrentSize.Width == 0))
    {
        CurrentSize = Size;
        return;
    }

    RecalculateSize();
    CurrentSize = Size;
}
}

private void ChartPanel_Paint(object sender,
PaintEventArgs e)
{
    if (ChartPanel != null)
        OnResize(new EventArgs());
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้