สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการหาพิกัดตำแหน่งภายในอาคารโดยใช้คลื่นอัลตร้าโซนิค

T047280

นายปฏิญญา ปันสุข
นายภราดร อัมวัน

เลขหมู่.....
เลขทะเบียน 47280
วัน, เดือน, ปี 27 มิ.ย. 2546

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

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Local Area Positioning System Using Ultrasonic

MR.PATINYA PANSUK

MR.PHARADON AMVAN

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

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

Academic year 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง	ระบบการหาพิกัดตำแหน่งภายในอาคารโดยใช้คลื่นอัลตร้าโซนิค
นักศึกษา	1. นายปฏิญญา ปันสุข
	2. นายภราดร อัมวัน
ภาควิชา	ฟิสิกส์ประยุกต์
สาขาวิชา	ฟิสิกส์ประยุกต์ (โซลิดสเตทอิเล็กทรอนิกส์)
อาจารย์ที่ปรึกษา	รศ.ดร.ปรีชา ยุพาพิน
ภาควิชาฟิสิกส์ประยุกต์	คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	อ.สุรชาติ กมลดิลก	.....
กรรมการ	รศ.ดร.ปรีชา ยุพาพิน	.....
กรรมการ	อ.เบญจพล ตันฮู้	.....
กรรมการ	อ.อัศวิน เรณูสวัสดิ์	.....

.....

(ผู้ช่วยศาสตราจารย์ วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยกต์ คณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง ระบบการหาพิกัดตำแหน่งภายในอาคารโดยใช้คลื่นอัลตร้าโซนิค

นักศึกษา นายปฏิญญา ปันสุข

นายภราดร อัมวัน

ภาควิชา ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**

สาขาวิชา ฟิสิกส์ประยุกต์ (โซลิดสเตทอิเล็กทรอนิกส์)

ปีการศึกษา 2545

อาจารย์ที่ปรึกษา รศ.ดร.ปรีชา ยุพาพิน

บทคัดย่อ

โครงงานพิเศษนี้เป็นการศึกษาเบื้องต้นในการหาพิกัดตำแหน่งภายในอาคาร โดยใช้คลื่นอัลตร้าโซนิค ซึ่งจะหาระยะเวลาการเดินทางของคลื่นอัลตร้าโซนิคจากตัวส่งถึงตัวรับ(Time of Flights) มาคำนวณหาตำแหน่งของผู้ใช้(User) โดยออกแบบเป็นระบบหาพิกัดตำแหน่งที่ใช้ตัวรับสัญญาณติดตั้งบนเพดานทั้งหมด 5 ตัว ที่ทราบตำแหน่งแน่นอน พื้นที่การทำงานจะใช้ห้องทดลองที่มีพื้นที่ 2 x 3.2 เมตร สูง 2.8 เมตรโดยไม่มีสิ่งกีดขวางสัญญาณ ซึ่งการประมวลผลและแสดงตำแหน่งจะใช้คอมพิวเตอร์รับข้อมูลจากตัวรับแต่ละตัวด้วยการสื่อสารแบบอนุกรม RS-232C ส่วนของการจับเวลาในตัวรับจะใช้ไมโครคอนโทรลเลอร์ MCS-51 และการติดต่อสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะใช้การติดต่อแบบระบบมัลติโปรเซสเซอร์ (Multiprocessor System) และใช้โปรแกรมที่พัฒนาขึ้นจากวิชวลเบสิกแสดงตำแหน่งของตัวส่งสัญญาณอัลตร้าโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title Local Area Positioning System Using Ultrasonic

Name Mr.Patinya Pansuk

Mr.Pharadon Amvan

Department Applied Physics

Program Applied Physics (Solid State Electronics)

Academic year 2002

Special Project Advisor Assoc.Prof.Dr.Preecha Yupapin

ABSTRACT

This special project presents an application of a local area positioning system such as home and office use. This work presents the use of the difference time to find the transmitter's position. We work on an indoor solution for a room about 2 x 3.2 m. and no obstructively signal. We tests system by betakes apply preliminary to exhibits the user's position which transmitter is static. And seeks position at most correct. Data from each receiver through series port to computer processing and show the transmitter's position by Visual Basic Program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานพิเศษนี้สำเร็จลุล่วงลงได้ด้วยดี เนื่องจากได้รับความเมตตากรุณาเป็นอย่างยิ่งจาก รศ.ดร.ปรีชา ยุพาพิน อาจารย์ที่ปรึกษาโครงงานพิเศษ ซึ่งให้ความรู้ แนวความคิดแก้ไขปัญหา ให้คำปรึกษาและแนะนำแนวทางในการดำเนินงานเป็นอย่างดี รวมทั้งอนุเคราะห์จัดหาอุปกรณ์ และงบประมาณในการดำเนินงาน ผู้ศึกษาขอกราบขอบพระคุณเป็นอย่างสูง

รศ.วิชิต ศิริโชติ ที่ให้ความรู้ทางด้านไมโครคอนโทรลเลอร์ และเทคนิควิธีการในการทำโครงงานพิเศษนี้

รศ.อนุพงศ์ สรงประภา ที่ให้แนวคิดเริ่มต้นในการทำโครงงานพิเศษนี้

อาจารย์เบญจพล ตันฮู้ ที่ให้คำแนะนำและแนวความคิดในการแก้ไขโครงงานพิเศษ ให้คำปรึกษา อนุเคราะห์สถานที่และอุปกรณ์ในการศึกษาโครงงานพิเศษนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ถ่ายทอดความรู้ให้แก่ผู้ศึกษา

ขอขอบพระคุณบิดามารดา และพี่น้องที่คอยให้กำลังใจ

สุดท้ายนี้ขอขอบพระคุณรุ่นพี่ และเพื่อนๆ รวมทั้งบุคคลที่มีความปรารถนาดีมอบให้แก่ผู้ศึกษาทุกท่าน ที่คอยให้กำลังใจและแนะนำแนวทางที่ดีแก่ผู้ศึกษาตลอดมา

นายปฏิญญา ปันสุข
นายภราดร อัมวัน
คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์ของโครงงานพิเศษ	2
1.3 ขอบเขตของโครงงานพิเศษ	2
1.4 ขั้นตอนการวิจัยและดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 คลื่นอัลตร้าโซนิค	5
2.1.1 ความหมายของคลื่นอัลตร้าโซนิค	5
2.1.2 ชนิดของคลื่นอัลตร้าโซนิค	5
2.1.3 การเกิดอัลตร้าโซนิค	6
2.1.4 ความถี่และความยาวคลื่น (Frequency and Wavelength)	7
2.1.5 ความเร็วของคลื่นอัลตร้าโซนิค	7
2.1.6 ปริมาณพลังงานของคลื่นอัลตร้าโซนิค	8
2.1.7 การลดทอนของคลื่นอัลตร้าโซนิค	8
2.2 ทรานสดิวเซอร์	9
2.2.1 ชนิดของเปียโซอิเลคทริคทรานสดิวเซอร์	9
2.2.2 การทำงานของทรานดิวเซอร์ตัวส่งและตัวรับ	9
2.3 หลักการหาตำแหน่ง	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4 ความรู้พื้นฐานเกี่ยวกับพอร์ตอนุกรม	12
2.4.1 ทำความรู้จักกับพอร์ตอนุกรม	12
2.4.2 การสื่อสารแบบอนุกรม	12
2.4.3 การสื่อสารข้อมูลแบบอะซิงโครนัส	13
2.4.4 มาตรฐานพอร์ตอนุกรมแบบ RS-232	15
2.4.5 คอนเน็กเตอร์สำหรับ RS-232 และการเชื่อมต่อ	17
2.4.6 UART	19
2.4.7 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232	20
2.4.8 การใช้วิชวลเบสิกสื่อสารผ่านพอร์ตอนุกรมด้วย คอนโทรล MSComm	21
2.4.9 การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์	32
2.5 ไมโครคอนโทรลเลอร์	32
บทที่ 3 การออกแบบและดำเนินการวิจัย	35
3.1 ออกแบบระบบ	35
3.2 การออกแบบและสร้างฮาร์ดแวร์	37
3.2.1 การออกแบบภาครับสัญญาณอัลตร้าโซนิค	37
3.2.2 ออกแบบภาคส่งสัญญาณอัลตร้าโซนิค	42
3.2.3 อุปกรณ์แปลงสัญญาณ RS-232 และแหล่งจ่ายไฟกระแสตรง	44
3.3 การสื่อสารในระบบ	46
3.4 การออกแบบทางด้านซอร์ฟแวร์	47
3.4.1 โปรแกรมการทำงานในส่วนของไมโครคอนโทรลเลอร์	47
3.4.2 โปรแกรมประมวลผลและแสดงตำแหน่ง	49
บทที่ 4 การทดลองและอภิปรายผล	55
4.1 การทดสอบอุปกรณ์	55
4.1.1 การทดลองมุมการตอบสนองต่อสัญญาณของตัวรับ	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.1.2 การทดลองวัดระยะทาง	56
4.1.3 การทดลองสื่อสารในระบบ	57
4.2 การลองนำระบบไปใช้งาน	59
บทที่ 5 สรุปและข้อเสนอแนะ	61
5.1 สรุป	61
5.2 ปัญหาที่พบ	61
5.3 ข้อเสนอแนะ	62
5.4 แนวทางในการพัฒนาในอนาคต	62
เอกสารอ้างอิง	63
ภาคผนวก	
ภาคผนวก ก	66
ภาคผนวก ข	91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 1.1 แสดงระยะเวลาในการดำเนินการวิจัย	4
ตารางที่ 2.1 อัตราเร็วของเสียงในตัวกลางต่างๆ	7
ตารางที่ 2.2 แสดงบิตพาริตีของข้อมูล	15
ตารางที่ 4.1 แสดงผลการทดลองมุมการตอบสนองของตัวรับ	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 การขยายและหดตัวของทรานสดิวเซอร์แบบเปียโซอิเล็กทริค 6
รูปที่ 2.2 แสดงทรานสดิวเซอร์ตัวส่งและตัวรับ ชนิดเปียโซอิเลคทริค 10
รูปที่ 2.3 แสดงโครงสร้างภายในของทรานสดิวเซอร์ตัวส่งและตัวรับ 10
รูปที่ 2.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม 13
รูปที่ 2.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 14
รูปที่ 2.6 แสดงรูปแบบของข้อมูลที่ติดต่อสื่อสารแบบมาตราฐาน RS-232C 16
รูปที่ 2.7 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9 17
รูปที่ 2.8 (ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem 18
รูปที่ 2.8 (ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้
สายสัญญาณเพียง 3 เส้น 19
รูปที่ 2.9 ไดอะแกรมโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม 21
รูปที่ 2.10 แผนภาพแสดงการทำงานของไทเมอร์2 ในโหมดแคปเจอร์ของ
ไมโครคอนโทรลเลอร์ 33
รูปที่ 3.1 แสดงภาพรวมของระบบจากการออกแบบในเบื้องต้น 35
รูปที่ 3.2 แสดงแผนภาพการทำงานของระบบ 36
รูปที่ 3.3 แสดงวงจรขยายสัญญาณของตัวรับคลื่นอัลตร้าโซนิค 37
รูปที่ 3.4 แสดงวงจรเรคติไฟร์แบบครึ่งคลื่น 37
รูปที่ 3.5 แสดงวงจรเปรียบเทียบสัญญาณ 38
รูปที่ 3.6 แสดงวงจรควบคุมการเกิดสัญญาณอินเตอร์รัพท์ 39
รูปที่ 3.7 แสดงแผนภาพการทำงานในการจับเวลาการเดินทางของเสียง
จากตัวส่งมาถึงตัวรับ 40
รูปที่ 3.8 แสดงอุปกรณ์ตัวรับสัญญาณอัลตร้าโซนิค 40
รูปที่ 3.9 แสดงวงจรตัวรับสัญญาณอัลตร้าโซนิค 41
รูปที่ 3.10 แสดงวงจรส่งสัญญาณอัลตร้าโซนิค 42
รูปที่ 3.11 แสดงอุปกรณ์ตัวส่งสัญญาณอัลตร้าโซนิค 42
รูปที่ 3.12 แสดงวงจรตัวส่งสัญญาณอัลตร้าโซนิค 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.13 แสดงอุปกรณ์แปลงสัญญาณ RS-232 และแหล่งจ่ายไฟกระแสตรง 44

รูปที่ 3.14 แสดงวงจรแปลงสัญญาณ RS-232 และแหล่งจ่ายไฟกระแสตรง 45

รูปที่ 3.15 แสดงแผนภาพการติดต่อสื่อสารในระบบ 46

รูปที่ 3.16 แผนภาพแสดงให้เห็นแนวการเปลี่ยนแปลงระดับสัญญาณแบบ TTL จาก ไมโครคอนโทรลเลอร์ MCS-51 ไปเป็นระดับสัญญาณอินพุตแบบ RS-232 ไปเป็นระดับสัญญาณแบบ TTL ก่อนที่จะได้เชื่อมต่อกับขาของ ไมโครคอนโทรลเลอร์ MCS-51 47

รูปที่ 3.17 แผนภาพแสดงการทำงานของโปรแกรมในส่วนของไมโครคอนโทรลเลอร์ ที่ทำงานเป็นสเลฟ 48

รูปที่ 3.18 แสดงหน้าตาโปรแกรมหลักของโครงงานนี้ 49

รูปที่ 3.19 แสดงส่วน Setting ของโปรแกรม จากการกดปุ่ม Setting ในส่วนของ Set Properties 51

รูปที่ 3.20 แผนภาพแสดงการทำงานของโปรแกรมวิชวลเบสิก 52

รูปที่ 3.21 แสดงส่วน Single Step ของโปรแกรม จากการกดปุ่ม Single Step ในส่วนของ Option 53

รูปที่ 4.1 แสดงกราฟที่ได้จากผลการทดลองมุมการตอบสนองต่อสัญญาณของตัวรับ 56

รูปที่ 4.2 แสดงกราฟที่ได้จากผลการทดลองวัดระยะทาง โดยในการทดลองนี้ ทำการทดลองที่อุณหภูมิห้อง 28 ^{o} C 57

รูปที่ 4.3 แสดงการสื่อสารข้อมูลในระบบ โดยประกอบด้วยสัญญาณควบคุม จากขา DTRของคอมพิวเตอร์ สัญญาณที่ขา Rx และ Tx ของไมโครคอนโทรลเลอร์ตัวรับ 58

รูปที่ 4.4 แสดงการรับและส่งข้อมูลของไมโครคอนโทรลเลอร์ตัวรับในหนึ่งครั้ง 58

รูปที่ 4.5 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 1 59

รูปที่ 4.6 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 2 60

รูปที่ 4.7 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 3 60

รูปที่ 5.1 รูปแบบการนำไปพัฒนาในอนาคต เป็นระบบควบคุมสื่อสารแบบไร้สาย บนพื้นที่ภายในบ้านหรือที่ทำงาน 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงานพิเศษ

การบอกพิกัดตำแหน่งบนพื้นโลกเป็นศาสตร์ที่ได้รับความสนใจจากมนุษย์มาเป็นเวลานาน ทั้งนี้เพื่อประโยชน์ในการเดินเรือ การกำหนดพรมแดน หรือแม้กระทั่งใช้ในกรณีของการลดข้อพิพาทระหว่างดินแดน ยิ่งไปกว่านั้นในปัจจุบันความต้องการข้อมูลแสดงพิกัดตำแหน่งถือเป็นเรื่องสำคัญยิ่งขึ้น ไม่ว่าจะเป็นเป็นการค้นหาผู้หลงทาง การนำข้อมูลพิกัดไปใช้ในทางธุรกิจ ดังเช่นการประยุกต์ใช้กับเครือข่ายโทรศัพท์เคลื่อนที่ เพื่อค้นหาร้านค้าหรือสถานที่สำคัญที่อยู่ใกล้ตัวของผู้ใช้งานแต่ละคน

ระบบการบอกพิกัดตำแหน่งที่ได้รับความนิยมอย่างมากในปัจจุบัน ได้แก่ ระบบ GPS (Global Positioning System) ซึ่งเป็นระบบที่ใช้กลุ่มดาวเทียมบอกพิกัด 24 ดวง ในการหาตำแหน่งของผู้ใช้ นอกจากนี้แล้วดาวเทียมบอกพิกัดยังมีความสามารถในการวิเคราะห์ความเร็วและเวลามาตรฐาน รวมถึงเวลาท้องถิ่น ณ ตำแหน่งต่างๆ บนโลกโดยไม่จำกัดว่าตำแหน่งเหล่านั้นจะอยู่บนพื้นดิน ทะเล หรือในป่าเขา ทำให้เรียกอีกอย่างว่าเป็นระบบนำร่องกลางแจ้ง (Outdoor Navigation System) แต่ระบบก็ยังมีข้อจำกัดคือ ผู้ใช้งานไม่สามารถใช้ระบบหาตำแหน่งของตัวเองได้ ถ้าอยู่ภายในอาคารหรือในพื้นที่ที่มีการขวางกันสัญญาณจากดาวเทียมบอกพิกัด เช่น ในอุโมงค์ ใจกลางเมืองที่มีตึกสูง เป็นต้น ด้วยเหตุนี้จึงมีระบบหาพิกัดตำแหน่งอีกประเภทหนึ่งที่ใช้กับพื้นที่เฉพาะเหล่านั้นได้ โดยเรียกว่าระบบนำร่องในร่ม (Indoor Navigation System) ซึ่งใช้กับพื้นที่ที่สัญญาณ GPS เข้าไม่ถึง ชนิดของสัญญาณที่นำมาใช้ในการนำร่อง ได้แก่ คลื่นวิทยุ(Radio frequency, RF) แสงอินฟราเรด(Infrared ray, IR) และคลื่นอัลตร้าโซนิค(Ultrasonic wave)

โครงงานพิเศษนี้เป็นการศึกษาระบบนำร่องในร่มเบื้องต้นซึ่งเน้นไปในลักษณะการหาพิกัดตำแหน่งภายในอาคารด้วยคลื่นอัลตร้าโซนิค ซึ่งจะหาระยะเวลาการเดินทางของคลื่นอัลตร้าโซนิคจากตัวส่งถึงตัวรับ(Time of Flights) มาคำนวณหาตำแหน่งของผู้ใช้(User) โดยออกแบบเป็นระบบหาพิกัดตำแหน่งที่ใช้ตัวรับสัญญาณติดตั้งบนเพดานทั้งหมด 5 ตัว ที่ทราบตำแหน่งแน่นอนบนพิกัดพื้นที่ของห้องทดลองขนาด 2x3.2 เมตร ความสูง 2.8 เมตร ส่วนตัวส่งสัญญาณบอกตำแหน่งของผู้ใช้อยู่ที่พื้นห้อง ตัวรับจะทำหน้าที่จับเวลาการเดินทางของคลื่นอัลตร้าโซนิคจากตัวส่ง หลังจากนั้นจะนำเอาเวลาที่ได้จากตัวรับแต่ละตัวมาคำนวณและแสดงผลพิกัดตำแหน่งโดยใช้คอมพิวเตอร์ การทำงานในส่วนของการจับเวลาในตัวรับจะใช้ไมโครคอนโทรลเลอร์ MCS-51 และการติดต่อสื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะใช้การติดต่อแบบระบบมัลติโปรเซสเซอร์ (Multiprocessor System) ผ่านการสื่อสารแบบอนุกรมมาตรฐาน RS-232 เพื่อนำข้อมูลจากไมโครคอนโทรลเลอร์มาคำนวณหาตำแหน่งโดยใช้โปรแกรมที่พัฒนาขึ้นจากวิชวลเบสิก (Visual Basic)

1.2 วัตถุประสงค์ของโครงงานพิเศษ

1. เพื่อศึกษาความเป็นไปได้ในการใช้คลื่นอัลตร้าโซนิค มาใช้ในการนำทางภายในอาคาร
2. เพื่อนำหลักการทางฟิสิกส์มาประยุกต์ใช้ในการออกแบบ และสร้างเครื่องมือที่วัดคุณสมบัติพื้นฐานทางฟิสิกส์ได้
3. เพื่อเป็นแนวทางในการนำไปพัฒนาใช้ในด้านระบบรักษาความปลอดภัยภายในบ้านและที่ทำงาน

1.3 ขอบเขตของโครงงานพิเศษ

โครงงานพิเศษนี้เป็นการศึกษาระบบนำร่องในร่มเบื้องต้น ซึ่งเป็นลักษณะระบบหาพิกัดตำแหน่งพื้นที่ภายในอาคาร โดยใช้คลื่นอัลตร้าโซนิค การออกแบบระบบในเบื้องต้นจะใช้พื้นที่ห้องทดลองขนาด 2x3.2 เมตร ซึ่งไม่มีสิ่งกีดขวางสัญญาณ และใช้ไมโครคอนโทรลเลอร์ AT89C52 เป็นตัวจับเวลาและส่งค่าเวลาที่ได้ไปคำนวณและแสดงผลโดยใช้คอมพิวเตอร์ โปรแกรมที่ทำหน้าที่คำนวณและแสดงผลนี้พัฒนาด้วยวิชวลเบสิก การควบคุมการรับค่าข้อมูลและการสร้างจังหวะพร้อมกันของตัวส่งกับตัวรับจะใช้คอมพิวเตอร์ทำหน้าที่ควบคุม สุดท้ายทำการทดสอบระบบโดยนำไปประยุกต์ใช้เบื้องต้นในการบอกตำแหน่งของผู้ใช้ในแบบหยุดนิ่ง เพื่อหาความสามารถของระบบว่าตำแหน่งที่ได้มีความถูกต้องมากน้อยเพียงใด

1.4 ขั้นตอนการวิจัยและดำเนินงาน

ในการออกแบบและพัฒนาระบบในโครงงานนี้ สามารถสรุปเป็นลำดับขั้นตอนการทำงานได้ 7 ขั้นตอนดังนี้

ขั้นตอนที่ 1 การออกแบบระบบ

ขั้นตอนแรกนี้จะเป็นการศึกษาและทำความเข้าใจในระบบหาตำแหน่งภายในอาคาร ว่าควรจะต้องสร้างหรือนำวงจรอย่างใดบ้างมาประกอบเข้ากันเป็นระบบ การออกแบบวงจรเบื้องต้นในส่วนของส่วนตรวจจับตำแหน่ง ได้แก่ วงจรตัวรับ-ส่งสัญญาณอัลตร้าโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 การออกแบบและสร้างฮาร์ดแวร์

การออกแบบทางด้านฮาร์ดแวร์จะเป็นการออกแบบวงจรต้นแบบซึ่งเป็นวงจรขยายสัญญาณอัลตร้าโซนิค วงจรเปรียบเทียบสัญญาณ วงจรควบคุมการเกิดการอินเตอร์รัพท์ ซึ่งจะใช้ต่อเชื่อมกับไมโครคอนโทรลเลอร์ทั้งในส่วนภาครับและภาคส่งสัญญาณขึ้นมาทดสอบ และปรับปรุงเปลี่ยนแปลงอุปกรณ์ให้มีประสิทธิภาพในการทำงานมากขึ้น โดยใช้แผ่นวงจรทดสอบอเนกประสงค์ (Protoboard) ต่อวงจรขึ้นมาทดสอบก่อนเป็นอันดับแรก หลังจากนั้นจึงต่อวงจรที่สมบูรณ์แล้วในอุปกรณ์จริง

ขั้นตอนที่ 3 การออกแบบซอฟต์แวร์

การดำเนินการในส่วนนี้จะแบ่งเป็นสองส่วนคือ ส่วนของโปรแกรมไมโครคอนโทรลเลอร์และส่วนของโปรแกรมในคอมพิวเตอร์ ในไมโครคอนโทรลเลอร์ตัวรับนั้นจะทำหน้าที่เกี่ยวกับการจับเวลาและส่งค่าข้อมูล ในไมโครคอนโทรลเลอร์ตัวส่งจะทำหน้าที่ในการสร้างสัญญาณความถี่ 40 kHz ส่วนของคอมพิวเตอร์ในโครงงานนี้จะทำหน้าที่ในการควบคุมการรับส่งสัญญาณอัลตร้าโซนิคและข้อมูลจากไมโครคอนโทรลเลอร์นอกจากนี้ยังทำหน้าที่ประมวลผลข้อมูลและแสดงผล

ขั้นตอนที่ 4 การทดสอบปรับปรุงทางฮาร์ดแวร์และซอฟต์แวร์

เป็นการดำเนินการสร้างอุปกรณ์ต้นแบบขึ้นมาทดสอบ ทั้งในภาคส่งและภาครับคลื่นอัลตร้าโซนิค ทำการแก้ไขปรับปรุงทางฮาร์ดแวร์และซอฟต์แวร์ในส่วนของอุปกรณ์ต่อเชื่อมกับไมโครคอนโทรลเลอร์และโปรแกรมของไมโครคอนโทรเลอร์ พัฒนาโปรแกรมในส่วนของภาคประมวลผลและแสดงผลให้มีประสิทธิภาพมากขึ้น

ขั้นตอนที่ 5 การทดลองระบบ

นำเอาอุปกรณ์ทั้งหมดทั้งตัวรับและตัวส่งคลื่นอัลตร้าโซนิคมาต่อเชื่อมเข้าด้วยกันกับคอมพิวเตอร์ โดยใช้การสื่อสารแบบอนุกรมกับพอร์ตคอมพิวเตอร์ ทำการพัฒนาและปรับปรุงโปรแกรมวิชวลเบสิกในส่วนแสดงผลพิกัดตำแหน่ง ให้สามารถบอกตำแหน่งของผู้ใช้ได้ถูกต้องมากที่สุด จากนั้นทดลองหาตำแหน่งของผู้ใช้ที่มีตัวส่งสัญญาณคลื่นอัลตร้าโซนิคติดอยู่

ขั้นตอนที่ 6 สรุปและแนวทางในการพัฒนา

การสรุปผลการทดลองเพื่อหาความแม่นยำของระบบว่าสามารถระบุตำแหน่งของผู้ใช้ได้ในระดับไหน เพื่อทำการประเมินผลสมรรถนะการทำงานของระบบว่าต้องทำการปรับพัฒนาในอนาคตอย่างไรบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่1.1 แสดงระยะเวลาในการดำเนินการวิจัย

ขั้นตอนการดำเนินงาน	ก.ย	ต.ค	พ.ย	ธ.ค	ม.ค	ก.พ	มี.ค
1. การออกแบบระบบ							
2. การออกแบบและสร้างฮาร์ดแวร์							
3. การออกแบบซอฟต์แวร์							
4. การทดสอบปรับปรุงทางฮาร์ดแวร์และซอฟต์แวร์							
5. การทดลองระบบ							
6. สรุปและแนวทางในการพัฒนา							

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถสร้างอุปกรณ์รับส่งคลื่นอัลตร้าโซนิคเพื่อหาตำแหน่งภายในอาคารได้
2. มีความรู้ความชำนาญในการต่อเชื่อมคอมพิวเตอร์กับอุปกรณ์ภายนอกและนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งานมากขึ้น
3. สามารถแก้ปัญหาโดยใช้หลักการทางวิทยาศาสตร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 คลื่นอัลตร้าโซนิค

2.1.1 ความหมายของคลื่นอัลตร้าโซนิค

คลื่นอัลตร้าโซนิค หมายถึงคลื่นที่มีความถี่สูงกว่าที่หูมนุษย์จะได้ยิน โดยทั่วไปแล้วหูของมนุษย์จะได้ยินในย่านความถี่ 20 เฮิร์ต ถึง 20,000 เฮิร์ต คลื่นอัลตร้าโซนิคที่กล่าวถึงโดยทั่วไปจึงหมายถึงคลื่นที่มีความถี่สูงกว่า 20,000 เฮิร์ต ขึ้นไป

2.1.2 ชนิดของคลื่นอัลตร้าโซนิค

คลื่นอัลตร้าโซนิค ที่เดินทางผ่านตัวกลางต่างๆ มีหลายชนิดด้วยกัน แต่ละชนิดจะแตกต่างกันตามการเคลื่อนที่ของอนุภาคในตัวกลางนั้น

1.) คลื่นตามยาว (Longitudinal Wave) คือ คลื่นชนิดหนึ่งซึ่งทุกๆจุดบนคลื่นมีการเคลื่อนที่ในทิศทางเดียวกันกับทิศทางการเคลื่อนที่ คลื่นความยาวนี้สามารถเดินทางผ่านของแข็ง ของเหลว และก๊าซ โดยมากถ้ากล่าวถึงความเร็วของคลื่นแล้วจะหมายถึงความเร็วของคลื่นตามยาว ในการพิจารณาคลื่นตามยาวที่เดินทางผ่านตัวกลางต่างๆ ได้นั้น ตัวกลางจะต้องมีขนาดใหญ่พอเมื่อเทียบกับความยาวคลื่น

2.) คลื่นตามขวาง (Transverse Wave) คือ คลื่นที่ทุกๆจุดบนคลื่นมีการ เคลื่อนที่ในทิศทางที่ตั้งฉากกับทิศทางการเคลื่อน คลื่นชนิดนี้จะเดินทางผ่านตัวกลางที่มี ขนาดของตัวกลางใหญ่กว่าขนาดของความยาวคลื่นและสามารถเดินทางผ่านตัวกลางที่เป็นของแข็ง คลื่นชนิดนี้ไม่สามารถเดินทางผ่านตัวกลางที่เป็นของเหลวและก๊าซ

คลื่นตามขวางมีลักษณะเสมือนการเกิดขั้วลบและขั้วบวก ซึ่งเป็นเหตุผลว่าการเปลี่ยนตำแหน่งของอนุภาคเป็นไปเพียงทิศทางเดียว เช่นในระนาบที่ตั้งฉากกับทิศทางของคลื่นที่เคลื่อนที่ไป ต้นกำเนิดของเคลื่อนตามขวางเป็นพื้นที่หน้าเรียบของระนาบที่เกิดจากการเปลี่ยนแปลงของอนุภาคอันเนื่องจากการแกว่ง

ความหนาแน่นของตัวกลางจะไม่เปลี่ยนแปลงโดยการเคลื่อนที่ของคลื่นแบบตามขวาง ความเร็วของคลื่นชนิดนี้จะน้อยกว่าความเร็วของคลื่นชนิดตามยาว ในขณะที่เดินทางผ่านตัวกลางชนิดเดียวกัน ดังนั้นที่ความถี่เดียวกัน ความยาวคลื่นของคลื่นตามขวางจะน้อยกว่าคลื่นตามยาวเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) คลื่นผิวหน้า (Surface Waves or Rayleigh) คือ คลื่นชนิดหนึ่งซึ่งคล้ายกับคลื่นตามขวาง จะต่างกันตรงที่ว่าการเปลี่ยนตำแหน่งของอนุภาคไม่เป็นเพียงในทิศทางที่ตั้งฉากกับทิศทางการเคลื่อนที่เพียงอย่างเดียว แต่มีการเปลี่ยนแปลงในทิศทางเดียวกับทิศทางของการเคลื่อนที่ด้วย จึงทำให้คลื่นเคลื่อนที่ไปตามระนาบในแนวนอน ด้วยเหตุนี้คลื่นจึงเดินทางผ่านไปเฉพาะบนผิวของตัวกลางเท่านั้น

2.1.3 การเกิดอัลตร้าโซนิค

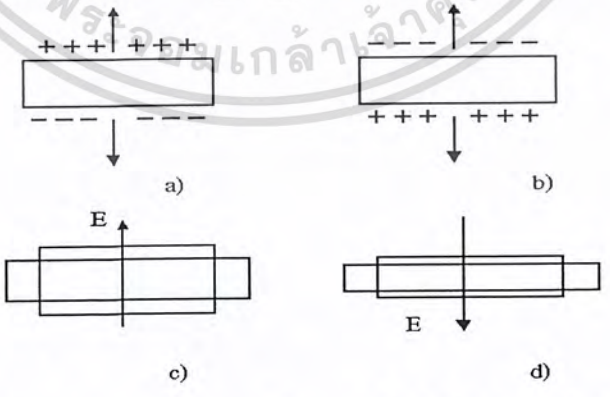
อัลตร้าโซนิคเป็นคลื่นที่เกิดจากการเปลี่ยนแปลงพลังงานในรูปอื่น ให้มาเป็นพลังงานกล โดยการสั่นไปมาเหมือนเกิดจากการเปลี่ยนพลังงานไฟฟ้ากับพลังงานกล ทำให้เกิดคลื่นย่านอัลตร้าโซนิคกระจายออกไปในอากาศดังนั้นจึงถือได้ว่า คลื่นที่เกิดขึ้นเป็นคลื่นกล (Mechanical Wave) คลื่นอัลตร้าโซนิคสามารถสร้างได้ โดยตัวทรานสดิวเซอร์ ซึ่งเป็นอุปกรณ์ที่ใช้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล หลักการสร้างมีหลายวิธีดังนี้

1.) แบบเปียโซอิเลคทริค (Piezo-electric transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้า และพลังงานกล โดยมีความถี่เรโซแนนท์คงที่อยู่ค่าหนึ่ง

2.) แบบแมกนีโตสตริคทีฟ (Magnetrostrictive tranducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่

3.) แบบอิเลคโตรสตริคทีฟ (Electrostrictive transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล

สำหรับแบบเปียโซอิเลคทริคเป็นแบบที่นิยมใช้เพราะมีราคาถูก หาซื้อง่ายรายละเอียดของเปียโซอิเลคทริคจะกล่าวถึงในภาคหลัง



รูปที่ 2.1 การขยายและหดตัวของทรานสดิวเซอร์แบบเปียโซอิเล็กทริค a)เมื่อมีแรงดึง, b) เมื่อมีแรงอัด, c) การขยายตัวเนื่องจากสนามไฟฟ้า E, d) การหดตัวเนื่องจากสนามไฟฟ้า E กลับทิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 ความถี่และความยาวคลื่น (Frequency and Wavelength)

ความถี่ คือจำนวนของการออสซิลเลตที่สมบูรณ์จากแหล่งกำเนิดคลื่นภายในหนึ่งวินาที คลื่นที่ถูกส่งจากแหล่งกำเนิดจะเดินทางด้วยความถี่เดียวกัน เช่น อัตราการสั่นของสายไวโอลินที่มีความถี่ 440 เฮิร์ต มันก็จะมีความถี่เดียวกันกับคลื่นที่ถูกส่งและรับได้จากผู้ฟัง

ความยาวคลื่น คือระยะทางที่คลื่นเดินทางระหว่างแต่ละการสั่นที่สมบูรณ์หรือการเดินทางครบรอบ (1 cycle) สามารถกล่าวได้ว่า ความยาวคลื่นเป็นระยะทางระหว่างการอัดอย่างต่อเนื่อง (Successive Compressions) หรือการเบาบางของอากาศ (Rarefaction) การอัดคือการที่บริเวณนั้นมีความหนาแน่นของโมเลกุลและแรงดันมากกว่าบริเวณรอบๆ ส่วนการเบาบางเป็นบริเวณเฉพาะที่เกิดการลดความหนาแน่นของโมเลกุล และแรงดันสัมพันธ์กับแรงดันของบรรยากาศปกติ

2.1.5 ความเร็วของคลื่นอัลตร้าโซนิค

ตัวกลาง	อัตราเร็วของเสียง (เมตร/วินาที)	
	ที่ 0°C	ที่ 20°C
อากาศ	331.5	343.0
ไฮโดรเจน	1,270.0	1,331.0
น้ำ	1,450.0	1,474.0
เหล็ก	5,100.0	5,130.0

ตารางที่ 2.1 อัตราเร็วของเสียงในตัวกลางต่างๆ

อัตราเร็วของเสียงในอากาศ โดยทั่วไปขึ้นอยู่กับอุณหภูมิในหน่วยของศาสัมบูรณ์ ตามความสัมพันธ์ดังนี้

$$v \alpha \sqrt{T}$$

โดย V = อัตราเร็วเสียง (เมตร/วินาที)

T = อุณหภูมิ (เคลวิน)

การหาอัตราเร็วเสียงในอากาศ ณ อุณหภูมิต่างๆ

จากการวัดอัตราเร็วเสียงที่ 0°C ได้เท่ากับ 331 เมตร/วินาที

ต้องการหาอัตราเร็วเสียงที่ t °C

จาก

$$v \alpha \sqrt{T}$$

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v = K\sqrt{T}$$

ถ้า t = 0°C , T = 273 เคลวิน ได้ ν = 331 m/s

แทนค่าลงในสมการจะได้

$$331 = K\sqrt{273} \quad \text{.....(1)}$$

ถ้า t = t + 273K ให้ $\nu = \nu_t$

แทนค่าลงในสมการจะได้

$$\nu_t = K\sqrt{(t+273)} \quad \text{.....(2)}$$

นำ (2) / (1) จะได้

$$\nu_t = 331\sqrt{(t+273)/273}$$

$$\nu_t = 331\sqrt{(t/273)+1} \quad \text{.....(3)}$$

จากการกระจายเทอม

$$\sqrt{(t/273)+1}$$

จะได้

$$\sqrt{1+(t+273)} = 1+\frac{t}{546}$$

$$\sqrt{1+(t+273)} = 1+\frac{1}{2}\times\frac{t}{273}+\text{.....(4)}$$

แทนค่าในสมการที่ (3) จะได้

$$\nu_t = 331+0.6t$$

หรือ

$$\nu_t = 331\times(1+\frac{t}{546})$$

จากสมการข้างบนจะเห็นได้ว่า ถ้าอุณหภูมิเปลี่ยนไป 1°C อัตราเร็วเสียงจะเปลี่ยนไป 0.6 เมตรต่อวินาที

2.1.6 ปริมาณพลังงานของคลื่นอัลตร้าโซนิค

ปริมาณพลังงานของคลื่นอัลตร้าโซนิคจะถูกวัดในรูปความเข้มของคลื่นอัลตร้าโซนิคซึ่งจะมีหน่วยเป็นวัตต์ ต่อ ตารางเซนติเมตร (W/cm^2) เป็นการไหลของพลังงานผ่านพื้นที่ 1 ตารางเซนติเมตรซึ่งตั้งฉากกับทิศทางการเดินทางของคลื่นใน 1 วินาที

2.1.7 การลดทอนของคลื่นอัลตร้าโซนิค

เมื่อคลื่นเดินทางผ่านตัวกลาง ลำคลื่น (Beam) ของคลื่นอัลตร้าโซนิคจะสูญเสียความเข้ม ซึ่งเกิดจากการลู่ออกของลำคลื่นอัลตร้าโซนิค หรือเกิดจากการกระจายพลังงานของคลื่นออกจากลำคลื่น เนื่องจากความไม่ต่อเนื่องในตัวกลาง และอาจเกิดจากการดูดซับพลังงานส่วนหนึ่งของคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวกลางที่คลื่นเคลื่อนที่ผ่าน พลังงานที่ดูดซับนี้จะเปลี่ยนไปเป็นพลังงานความร้อน การดูดซับพลังงานนี้ขึ้นอยู่กับคุณลักษณะของวัตถุ ความยืดหยุ่นและความหนาแน่น รวมทั้งความถี่ของคลื่นที่ใช้ ยิ่งความถี่สูงพลังงานจะยิ่งถูกดูดซับมาก

2.2 ทรานสดิวเซอร์

ทรานสดิวเซอร์เป็นตัวแปลงสัญญาณรูปหนึ่งไปเป็นอีกรูปหนึ่ง เช่น สัญญาณไฟฟ้าไปเป็นสัญญาณเสียง หรือจากเสียงเป็นไฟฟ้า สำหรับทรานสดิวเซอร์ในการแปลงจากสัญญาณไฟฟ้าเป็นสัญญาณเสียงนั้นได้แก่ ทรานสดิวเซอร์แบบ เปียโซอิเลคทริคทรานสดิวเซอร์ เป็นต้น

2.2.1 ชนิดของเปียโซอิเลคทริคทรานสดิวเซอร์

เปียโซอิเลคทริคทรานสดิวเซอร์สามารถแบ่งออกเป็น 2 ชนิดคือ

1) แบบ Generation-Action transducer ใช้เป็นตัวรับโดยแรงดันไฟฟ้าที่เกิดขึ้นจะหาได้จากแรงดัน และความถี่ที่มากระทำต่อวัสดุเปียโซอิเลคทริค

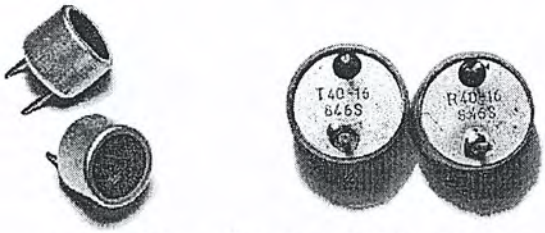
2) แบบ Motor-Action transducer ใช้เป็นตัวส่งโดยการเปลี่ยนแปลงของรูปร่างที่ทำให้เกิดคลื่นอัลตร้าโซนิค จะขึ้นอยู่กับขนาดความสูงและความถี่ของแรงดันไฟฟ้าที่ป้อนให้ในทั้งสองกรณี กรณีค่าแรงดันไฟฟ้าที่เกิดขึ้นจะขึ้นกับขนาดของวัสดุ

2.2.2 การทำงานของทรานสดิวเซอร์ตัวส่งและตัวรับ

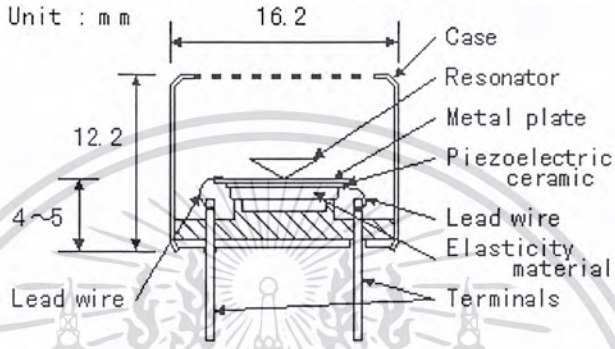
เมื่อเซรามิคได้รับสัญญาณแรงดันมาตกคร่อม จะทำให้ชิ้นสารเซรามิคโก่งงอทำให้เกิดการอัดอากาศโดยรอบเกิดเป็นคลื่นขึ้นมา ดังนั้นถ้าป้อนสัญญาณเป็นห้วงๆ (Electrically Pulse) จากออสซิลเลเตอร์ก็จะทำให้ชิ้นสารโก่งงอมากน้อยหรือทิศทางใดตามขนาดและทิศทางการเปลี่ยนแปลงขนาดของสัญญาณไฟฟ้าจากออสซิลเลเตอร์นั้นออกไป โดยทั่วไปกำลังเอาท์พุทที่ออกมาจะตกลงประมาณ 10 เปอร์เซ็นต์ของกำลังไฟฟ้าที่ป้อนให้ แต่เอาท์พุทจะสูงที่ค่านี้โดยประมาณ ก็ต่อเมื่อความถี่ของสัญญาณออสซิลเลเตอร์ที่ป้อนเข้าชิ้นสารเซรามิคตรงกับความถี่เรโซแนนท์ที่เป็นความถี่ทางกลตามธรรมชาติของชิ้นสารเซรามิคนั้นๆ ส่วนที่ความถี่อื่นๆ กำลังเอาท์พุทจะลดลงกว่านี้

ส่วนการทำงานของทรานสดิวเซอร์ตัวรับ มีการทำงานตรงกันข้ามกับตัวส่งคือเมื่อมีคลื่นเสียงที่มีความถี่ตรงกับความถี่เรโซแนนท์ของชิ้นสารเซรามิคเข้ามาจะทำให้ชิ้นสารโก่งตัวไปมาและเกิดสัญญาณแรงดันไฟฟ้าขึ้นซึ่งมีขนาดเล็ก ขึ้นคร่อมขั้วทั้งสองของตัวมัน คุณสมบัติโดยทั่วไปของคลื่นอุลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคทริคก็คือมีความต้านทานไฟตรงสูงมากอาจสูงถึง 100 เมกกะโอห์ม เรียกว่า ถ้าเอามัลติมิเตอร์ธรรมดามาตั้งสเกลวัดค่าความต้านทานสูงๆ เข็มจะไม่กระดิกเลยแต่ในขณะที่มันทำงานความต้านทานไฟสลับจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงทรานสดิวเซอร์ตัวส่งและตัวรับ ชนิดเปียโซอิเลคทริค



รูปที่ 2.3 แสดงโครงสร้างภายในของทรานสดิวเซอร์ตัวส่งและตัวรับ

ข้อควรรู้ในการใช้งานตัวส่งและตัวรับพอที่จะสรุปเป็นแนวทางใช้งานดังนี้

1. ไม่ควรให้ตัวทรานสดิวเซอร์ได้รับการกระแทกหรือตกจากที่สูงเพื่อป้องกันโครงสร้างภายในมิให้เสียหาย
2. ทรานสดิวเซอร์ที่มีขายกันโดยทั่วไปจะทนแรงดันตกคร่อมตัวมันสูงสุดได้ไม่เกิน 20 Vrms ดังนั้นขนาดของสัญญาณที่จะป้อนให้กับตัวทรานสดิวเซอร์ก็ควรอยู่ภายในขีดจำกัดอันนี้
3. ความถี่เรโซแนนท์ (ความถี่ที่ตัวมันทำงานอย่างมีประสิทธิภาพสูงสุด) ของทรานสดิวเซอร์ 40 กิโลเฮิร์ต ที่มีขายกันโดยทั่วไปจะผิดพลาดไปไม่เกิน +1 กิโลเฮิร์ต และมีแถบความถี่ประมาณ 4.5 กิโลเฮิร์ต (bandwidth) สำหรับตัวส่งและมีแถบความถี่ประมาณ 5.0 กิโลเฮิร์ต สำหรับตัวรับ จะเห็นได้ว่าแถบความถี่ของตัวรับจะกว้างกว่าของตัวส่งเล็กน้อยเพื่อให้แน่ใจว่าตัวรับจะสามารถรับความถี่ทั้งหมดที่ออกมาจากตัวส่งได้
4. อุณหภูมิใช้งานของตัวทรานสดิวเซอร์ควรอยู่ภายในช่วง –20 ถึง 60 องศาเซลเซียส
5. ทั้งตัวส่งและตัวรับจะมีทิศทางคล้ายคลึงกันมากกว่ากล่าวคือ ที่ตำแหน่งเบนจากแนวแกนของตัวส่งไปประมาณ 30 องศา ความแรงของคลื่นเสียงที่ถูกส่งออกไปจะลดลงจากแนวแกนประมาณ 10 เดซิเบล ในทำนองเดียวกันถ้าคลื่นเสียงพุ่งเข้ามาในแนวที่เบี่ยงเบนไปจากแนวแกนของตัวรับไปประมาณ 30 องศา ความไวหรือขนาดแรงดันที่ออกมาก็ลดลงไปประมาณ 10 เดซิเบล ด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการหาตำแหน่ง

ในโครงงานนี้เราจะใช้เทคนิคการประมาณค่าของการหาตำแหน่งแบบ 2 มิติ ซึ่งจะใช้ค่าความแตกต่างของเวลาในการเดินทางของเสียงจากตัวส่งถึงตัวรับ (Time of Flights; TOF) ซึ่งจะใช้ในการหาพิกัดตำแหน่งของตัวส่ง

วิธีการนี้ตัวส่งจะส่งคลื่นอัลตร้าโซนิค ไปยังตัวรับแต่ละตัว โดยที่ตัวรับแต่ละตัวจะติดตั้งอยู่ที่เพดานห้อง (ในโครงงานนี้จะใช้ตัวรับ 5 ตัว) ซึ่งตำแหน่งของตัวรับจะบอกเป็นพิกัด คือ $R1(x_1, y_1, z_1)$, $R2(x_2, y_2, z_2)$, $R3(x_3, y_3, z_3)$, $R4(x_4, y_4, z_4)$ และ $R5(x_5, y_5, z_5)$ โดย ค่า x_i และ y_i (i = 1, 2, 3, 4, 5)คือ ค่าตำแหน่งทางด้านกว้าง และ ยาว ของห้อง ส่วนค่า z_i คือค่าความสูงของห้อง หลักการก็คือ เมื่อตัวส่งได้ส่งสัญญาณอัลตร้าโซนิคออกไป และตัวรับแต่ละตัวจับสัญญาณคลื่นที่ส่งมาได้ก็จะได้ค่า TOF ออกมา ซึ่งค่าเวลาที่ใช้ในการส่งคลื่นอัลตร้าโซนิคออกไปถึงตัวรับแต่ละตัวจะมีค่าแตกต่างกัน ซึ่งเราจะใช้ค่าพิกัดตำแหน่งของตัวรับ และค่า TOF นี้ในการคำนวณหาตำแหน่งของตัวส่งต่อไป

ในการคำนวณเราจะกำหนดให้มีตัวรับอ้างอิง ซึ่งตัวรับอ้างอิงก็คือตัวรับที่อยู่ใกล้ตัวส่งมากที่สุดในเวลานั้นๆ โดยจะให้มีพิกัดเป็น $R1(x_1, y_1, z_1)$ ซึ่งกำหนดให้เป็นพิกัดอ้างอิง ส่วนพิกัดของตัวรับที่เหลือจะเป็น $R2(x_2, y_2, z_2)$, $R3(x_3, y_3, z_3)$, $R4(x_4, y_4, z_4)$ และ $R5(x_5, y_5, z_5)$ ตามลำดับ และเราจะให้ค่า TOF ของตัวรับอ้างอิงนี้เป็นค่าเวลาอ้างอิง T_1 ส่วนค่า TOF ของตัวรับที่เหลือเป็น T_2, T_3, T_4 และ T_5 ตามลำดับ และจะได้ค่าผลต่างของค่าเวลาของตัวรับอ้างอิงกับ ค่าเวลาของตัวรับที่เหลือเป็น

$$\begin{aligned} \Delta T_{12} &= T_2 - T_1 \\ \Delta T_{13} &= T_3 - T_1 \\ \Delta T_{14} &= T_4 - T_1 \\ \Delta T_{15} &= T_5 - T_1 \end{aligned} \tag{1}$$

T คือเวลาในการเดินทางของเสียงจากตัวส่งถึงตัวรับ (Time of Flights; TOF)

และระยะห่างระหว่างตัวรับอ้างอิงกับตัวรับที่เหลือจะเป็น

$$\begin{aligned} (x_1 - x_2)^2 + (y_1 - y_2)^2 &= (d + c\Delta T_{12})^2 \\ (x_1 - x_3)^2 + (y_1 - y_3)^2 &= (d + c\Delta T_{13})^2 \\ (x_1 - x_4)^2 + (y_1 - y_4)^2 &= (d + c\Delta T_{14})^2 \\ (x_1 - x_5)^2 + (y_1 - y_5)^2 &= (d + c\Delta T_{15})^2 \end{aligned} \tag{2}$$

โดยที่ c คือค่าความเร็วของคลื่นอัลตร้าโซนิคในอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจากสมการที่ผ่านมาข้างต้น จะสามารถหาได้ว่า

$$\begin{bmatrix} 2(x_1-x_2) & 2(y_1-y_2) & -2\Delta T_{12} & -2\Delta T_{12}^2 \\ 2(x_1-x_3) & 2(y_1-y_3) & -2\Delta T_{13} & -2\Delta T_{13}^2 \\ 2(x_1-x_4) & 2(y_1-y_4) & -2\Delta T_{14} & -2\Delta T_{14}^2 \\ 2(x_1-x_5) & 2(y_1-y_5) & -2\Delta T_{15} & -2\Delta T_{15}^2 \end{bmatrix} * \begin{bmatrix} u \\ v \\ cd \\ c^2 \end{bmatrix} = \begin{bmatrix} x_1^2+y_1^2-x_2^2-y_2^2 \\ x_1^2+y_1^2-x_3^2-y_3^2 \\ x_1^2+y_1^2-x_4^2-y_4^2 \\ x_1^2+y_1^2-x_5^2-y_5^2 \end{bmatrix} \quad (3)$$

หลังจากแก้สมการเมตริกซ์นี้แล้ว เราจะได้คำตอบออกมาเป็น ค่าพิกัดตำแหน่งของตัวส่งที่เราต้องการหา คือ *(u, v)* และ ค่า *d* คือระยะห่างจากตัวส่งถึงตัวรับอ้างอิง *R1* และการหาพิกัดตำแหน่งด้วยวิธีการนี้ จะทำให้ไม่ต้องนำเอาอุณหภูมิมาคิด

2.4 ความรู้พื้นฐานเกี่ยวกับพอร์ตอนุกรม

2.4.1 ทำความรู้จักกับพอร์ตอนุกรม

มีทางเลือกอยู่ 2 ทาง ในการเคลื่อนย้ายข้อมูลทางคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆ หรือคอมพิวเตอร์ด้วยกัน นั่นคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรม การรับส่งข้อมูลแบบขนาน จะเป็นการรับส่งข้อมูลคราวละ 4 ถึง 8 บิต ในเวลาเดียวกัน ซึ่งจะทำให้การรับส่งข้อมูลทำได้อย่างรวดเร็ว ซึ่งก็หมายความว่าจำนวนของสายที่ใช้ในการการส่งจะต้องมีเท่ากับจำนวนของบิตที่จะใช้ส่ง นอกจากนี้ยังรวมถึงสายที่ที่ใช้สำหรับการควบคุมและการตรวจสอบ การรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายเป็นการจำนวนมากเป็น 2 เท่า ของจำนวนบิตที่จะส่งได้ ซึ่งจะเป็นปัญหาในเรื่องราคาสายที่ใช้ในการเชื่อมต่อแบบขนานมีราคาแพง

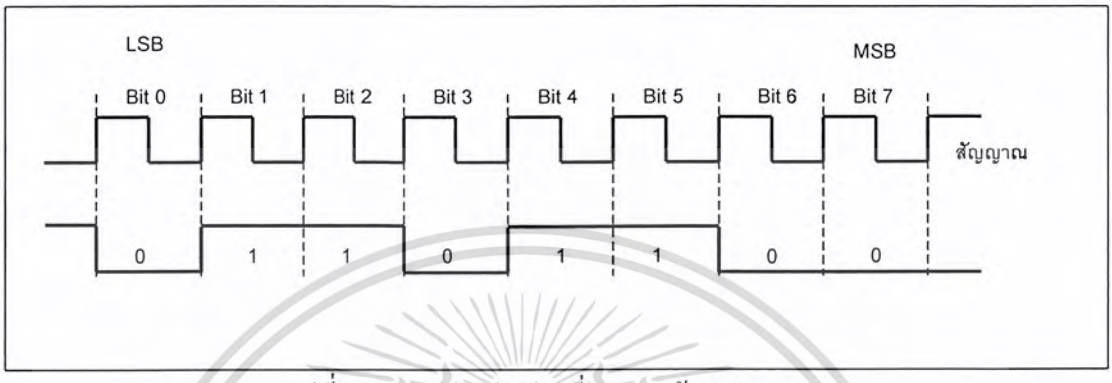
ในขณะการรับส่งข้อมูลแบบอนุกรมเป็นการส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถส่งข้อมูลที่ละหลายๆบิตได้ หากแต่ต้องมีการตกลงกันระหว่างตัวรับกับตัวส่ง จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรับข้อมูลให้ครบทุกบิตเสียก่อนจึงจะทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนของสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายน้อยกว่ามาก อย่างน้อยที่สุดใช้ 2-3 เส้นเท่านั้น แต่อัตราเร็วในการส่งต่ำกว่าแบบขนาน

2.4.2 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมอาจแบ่งได้เป็น 2 แบบคือการสื่อสารแบบอนุกรมซิงโครนัสและการสื่อสารแบบอนุกรมอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับสัญญาณและการส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ คีย์บอร์ดของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ ซึ่งสายส่งหนึ่งจะเป็นสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายส่งข้อมูล ดั้งนั้น การติดต่อกันแบบซิงโครนัสจึงต้องใช้สายต่อกันอย่างน้อย 3 เส้น คือ สัญญาณนาฬิกา ข้อมูล และกราวด์ รูปที่ 2.4 แสดงให้เห็นไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

2.4.3 การสื่อสารข้อมูลแบบอะซิงโครนัส

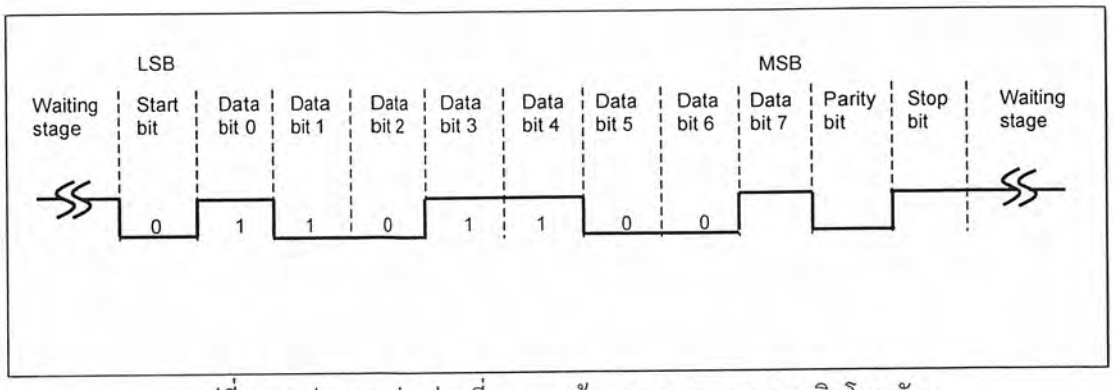
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้กำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็นบิตต่อวินาที (bit per second : bps)

รูปแบบข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพารีตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต

รูปที่ 2.5 แสดงรูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส ขา DATA จะมีสถานะ ลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นก็ตามด้วยบิตพาริตี เพื่อใช้ตรวจสอบความผิดพลาดในการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขา DATA มีสถานะเป็น ลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต 1.5 บิต หรือ 2 บิต ซึ่งเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือค่าบอดเรต ซึ่งก็คือจำนวนบิตต่อวินาทีที่ใช้ในการรับส่งข้อมูลบอดเรตมาตรฐาน ที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่ามากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรต คือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลแบบอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตีความเร็วในการรับส่งจะเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (Odd) แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนรวมของบิตที่เป็น ลอจิก “1” ภายในข้อมูลที่ส่งไป 1 ไบต์ มีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตีเข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวน ลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ค่าในบิตพาริตี จะต้องมี ลอจิกเป็น “0” แต่ถ้าพาริตีเป็นคี่ ค่าพาริตีจะเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตีที่มีจำนวนบิตที่เป็นลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่ ตารางที่ 2.2 แสดงตัวอย่างของบิตพาริตีในการรับส่งข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

ตารางที่ 2.2 แสดงบิตพาริตีของข้อมูล

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี ให้ตรงกันว่าจะตรวจสอบพาริตีคู่หรือพาริตีคี่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้เป็นเลขคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีข้อผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UART เบอร์ 8250 UART ชิพ เหล่านี้ที่ระดับเป็นแบบทีทีแอล (0 และ +5 V) แต่เพื่อให้มีแรงดันตามมาตรฐาน RS - 232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดย ลอจิก "0" มีระดับแรงดัน +3 V ถึง +12 V ในขณะที่ ลอจิก "1" มีระดับแรงดัน –3 V จนถึง –12 V

2.4.4 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนั้นสื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์ อีกชุดซึ่งอยู่ไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตร

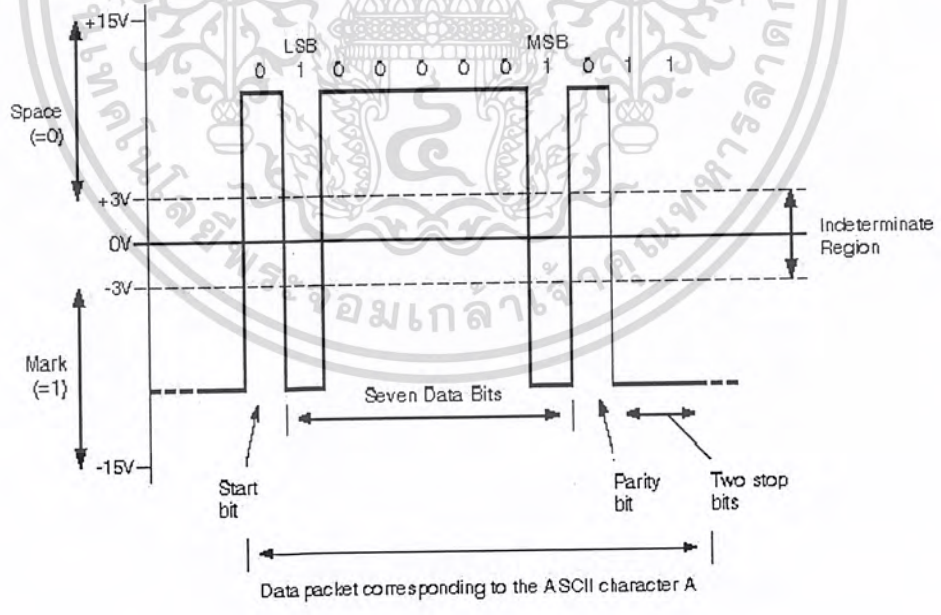
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในระยะแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีสัญญาณตั้งแต่ –3 ถึง –12 V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรม ส่วนประกอบ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสอง จะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และ อุปกรณ์ DCE อย่างหนึ่งที่เห็นชัดก็คือ คอนเน็กเตอร์ของ DTE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็ม หรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร



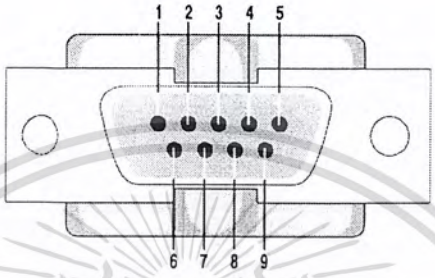
รูปที่ 2.6 แสดงรูปแบบของข้อมูลที่ติดต่อสื่อสารแบบมาตราฐาน RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.4.5 คอนเน็กเตอร์สำหรับ RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กแบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้ในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป รูปร่างและตำแหน่งขาแสดงดังรูปที่ 2.7

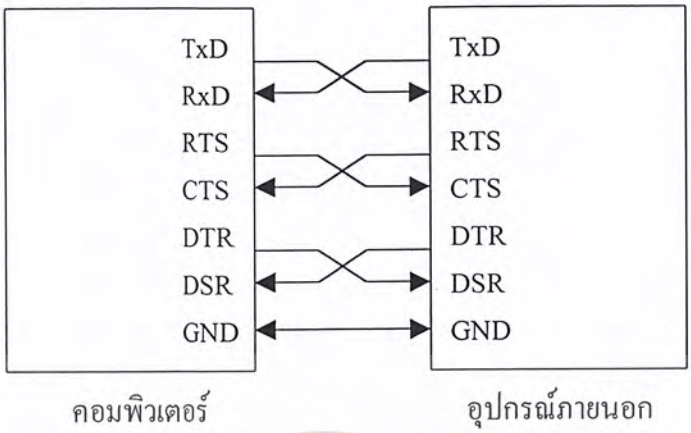


Pin	Signal	Type of Signal
1	Data Carrier Detect : DCD	Input
2	Received Data : RxD	Input
3	Transmitted Data : TxD	Output
4	Data Terminal Ready : DTR	Output
5	Signal Ground : GND	-
6	Data Set Ready : DSR	Input
7	Request to Send : RTS	Output
8	Clear to Send : CTS	Input
9	Ring Indicator : RI	Input

รูปที่ 2.7 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9

สำหรับการเชื่อมคอมพิวเตอร์กับอุปกรณ์ภายนอก แสดงในรูปที่ 2.8 ลูกศรในรูปแสดงทิศทางของข้อมูล ในรูปที่ 2.8 (ก) เป็นการเชื่อมต่อแบบ Null Modem หรือการเชื่อมต่อโดยตรงไม่ต้องผ่านโมเด็ม มีการตรวจสอบหรือแฮนด์เช็คเต็มรูปแบบ ส่วนในรูปที่ 2.8 (ข) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับการส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดการทำงาน ในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
47280



รูปที่ 2.8 (ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem

Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD สายนี้จะแอกทีฟ เมื่อมีการส่งสัญญาณพาหะจากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็มสำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานนัก

Receiver Data : RD หรือ ***RxD*** สายนี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

Transmitted Data : TD หรือ ***TxD*** ขานี้เพื่อใช้ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ ต้องการติดต่อกับขา DTR นี้จะต้องเชื่อมกับขา DSR ของอุปกรณ์ที่ปลายทางและขา DTR ของอุปกรณ์ปลายทางต้องต่อกับขา DSR ของคอมพิวเตอร์ถ้าใช้เชื่อมค่าเป็นแบบ Null Modem ซึ่งใช้สายในการต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีการสื่อสารที่ใช่มีการตรวจจับสัญญาณพาหะ

Signal Ground : SNG ขากราวด์ของระบบ

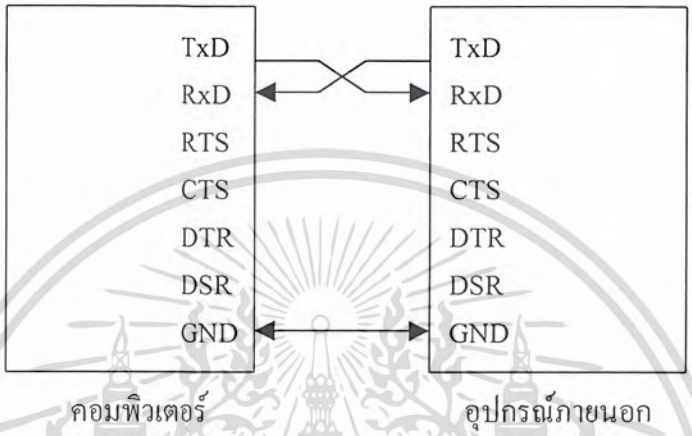
Data Set Ready : DSR ขานี้จะใช้คู่กับ DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกที่ถูกส่งมาจากขา DTR

Request To Send : RTS เป็นขาสำหรับสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้เชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและการส่งข้อมูลเกิดขึ้นได้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่รับข้อมูลหรือไม่

Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น



รูปที่ 2.8 (ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

2.4.6 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้,เฟรมข้อมูล,โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1 – 65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบ ฮาล์ฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

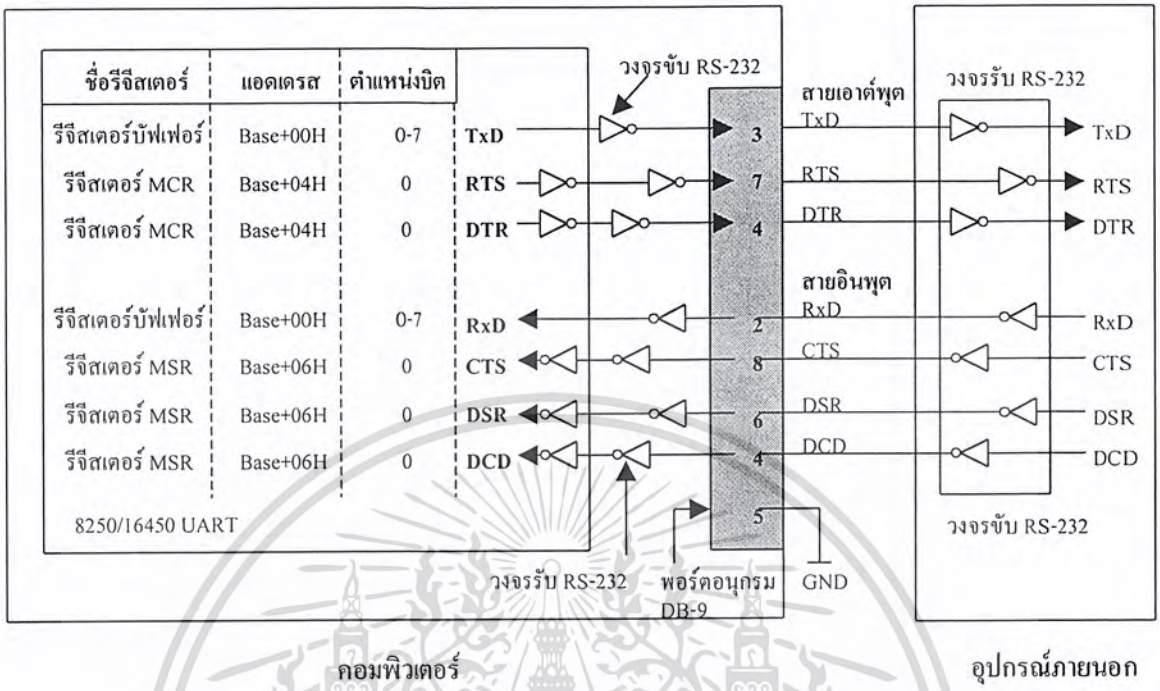
UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5 V และ +3 V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาที เมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.4.7 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาครับและส่งจะไม่ถูกกลับสถานะ UART จะได้รับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จะต้องส่งเข้าสู่วงจรเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกันเพื่อให้ได้ระดับสัญญาณเดียวกันแต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ ดังแสดงเป็นบล็อกไดอะแกรมในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ไดอะแกรมโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

2.4.8 การใช้วิชวลเบสิกสื่อสารผ่านพอร์ตอนุกรมด้วยคอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.VBX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ *การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication)* เป็นรูปแบบการใช้งานที่มีประสิทธิภาพสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรม หรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send (RST) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็น*การคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent* หลังจากให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปสามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยากดังนี้

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

object.CommPort [= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

object.Settings [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตี้, D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600, N, 8, 1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

1,200 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2,400 บิตต่อวินาที

9,600 บิตต่อวินาที (ค่าปกติ)

14,400 บิตต่อวินาที

19,200 บิตต่อวินาที

28,800 บิตต่อวินาที

38,400 บิตต่อวินาที (สงวน)

56,000 บิตต่อวินาที (สงวน)

128,000 บิตต่อวินาที (สงวน)

256,000 บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	*พาริตี้คู่ (Even)*
M	*ลอจิก "1" (MARK)*
N	*ไม่ใช้ (ค่าปกติ)*
O	*พาริตี้คี่ (Odd)*
S	*ลอจิก "0" (Space)*

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ *4, 5, 6, 7 และ 8 (เป็นค่าปกติ)*

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ *1 (เป็นค่าปกติ), 1.5 และ2*

ตัวอย่างการใช้งานคำสั่ง Setting โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSComm1.Settings = "9600, N, 8, 1"

<u>หมายเหตุ</u> สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด "" เนื่องจาก ค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

object.PortOpen [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Settings = "9600, n, 8, 1"
MSComm1.CommPort = 1
MSComm1.PortOpen = True
```

Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีในการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click()
Dim InString as String
    MSComm1.InputLen = 0                    ' Retrieve all available data.
    If MSComm1.InBufferCount Then           ' Check for Data.
    InString = MSComm1.Input                ' Read data.
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

InBufferCount

ส่งจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

object.InBufferCount [= value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ BufferCount มีค่าเป็น 0

<u>หมายเหตุ</u> อย่าสับสนระหว่างคำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSize นั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วนเป็นไบต์

รูปแบบการใช้งานคำสั่ง

object.InBufferSize [= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ ค่าเริ่มต้นกำหนดไว้ที่ 1,024 ไบต์

<u>หมายเหตุ</u> กำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วนความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่ามากขึ้น

InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

object.InputLen [= value]

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้ คำสั่ง Input ของ MSComm อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่าง (“”) กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InBufferCount โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

ตัวอย่างโปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```
Private Command_Click( )
Dim CommData as String
MSComm1.InputLen = 10              ' Specify a 10 character block of data.
CommData = MSComm1.Input           ' Read data.
End Sub
```

InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

object.InputMode [= value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

ComInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

ComInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบต์จากพอร์ตอนุกรมและเก็บข้อมูลไว้ในตัวแปรแบบอาเรย์ ชนิดข้อมูลเป็นแบบไบต์

```
Private Sub Command_Click ( )
Dim Buffer as Variant
Dim Arr( ) as Byte
MSComm1.Commport = 1               ' Set and open port
MSComm1.PortOpen = True
MSComm1.InputMode = comInputModeBinary   ' Set InputMode to read
Binary data
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Do Until MSComm1.InBufferCount < 10  ' Wait until 10 byte are in the input buffer
        DoEvents
Loop
Buffer  =  MSComm1.Input                ' Store binary data in buffer
Arr  =  Buffer                          ' Assign to byte array for processing
End  Sub
```

Output

ใช้ในการส่งขบวนข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

object.Output [= value]

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output

```
Private Sub Form_KeyPress  (KeyAscuu  As  Integer)
        Dim Buffer as Variant
        MSComm1.CommPort  =  1                  ' Use COM1
        MSComm1.PortOpen  =  True               ' Open port
        Buffer  =  Chr$ (KeyAscii)
        MSComm1.Output  =  Buffer               ' Send DATA
End  Sub
```

OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

object.OutBufferCount [= value]

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>หมายเหตุ</u> ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดของตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

object.OutBufferSize [= value]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

<u>หมายเหตุ</u> การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้ มีหน่วยความจำเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการใช้ แฮนด์เช็ค วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512 ไบต์ดูก่อน ถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี้

รูปแบบการใช้งานคำสั่ง

object.ParityReplace [= value]

บิตพาริตี้ เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี้ คอนโทรล MSCOM จะทำการบวกบิตทุกบิตที่มีค่าลอจิก “1” ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก “1” เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตี้ไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่า เริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตี้ผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง (“”) จะเป็นการยกเลิกการใช้งาน ParityReplace และไม่มีการป้อนข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งจะสามารถใช้ค่าใดๆ ก็ได้ที่เป็น ANSI มีค่าอยู่ระหว่าง 0-255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งาน

object.DTREnable [= value]

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น "0" หรือ "1" โดย

True หมายถึง ให้ขา DTR มีลอจิก "1"

False หมายถึง ให้ขา DTR มีลอจิก "0" (เป็นค่าปกติ)

หมายเหตุ เมื่อขา DTR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะลอจิด "1" เมื่อทำการเปิดพอร์ตและจะมีสถานะเป็น "0" เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ที่ขา DTR จะมีสถานะลอจิก "0" ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

สำหรับการใช้งานกับโมเด็ม การทำให้ขา DTR เป็นลอจิก "0" จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Request To Send (RTS) โดยขา RTS จะเป็นสัญญาณที่ส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

object.RTSEnable [= value]

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก "0" หรือ "1" ให้ขา RTS โดย

True หมายถึง ให้ขา RTS มีลอจิก "1"

False หมายถึง ให้ขา RTS มีลอจิก "0" (เป็นค่าปกติ)

หมายเหตุ เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก "1" เมื่อเปิดพอร์ตและมีสถานะลอจิก "0" เมื่อปิดพอร์ต และเมื่อมีการกลับสถานะของขา RTS ขา TxD จะมีสถานะลอจิกเป็น "1" แต่ที่คุณสมบัติ Break ยังคงเป็นค่าเดิม

EOFEnable

เป็นการกำหนดให้ MSComm รอสัญลักษณ์แสดงส่วนท้ายสุดของไฟล์ (End of file : EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งาน

object.EOFEnable [= value]

โดย value เป็นค่าสถานะ True หรือ False เพื่อเอ็นเอเบิลหรือดิสเอเบิลการทำงานของเหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EOF โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงานด้วย EOF

False หมายถึง เหตุการณ์ OnCOmm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF (เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มีการตรวจสอบสัญลักษณ์ EOF

CTSHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send (CTS) ได้ว่ามีสถานลอจิก "0" หรือ "1" โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CTSHolding เป็น True ขา CTS จะมีสถานะลอจิกเป็น "1" ถ้าค่า CTSHolding เป็น False ขา CTS จะมีสถานะลอจิกเป็น "0"

รูปแบบการใช้งาน

object.CTSHolding

เมื่อขา CTS เป็นลอจิก "0" (CTSHolding = False) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

CDHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect (DCD) ได้ว่ามีสถานะลอจิกเป็น "1" หรือ "0" โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CDHolding เป็น True ขา DCD จะมีสถานะลอจิก "1" ถ้าค่า CDHolding เป็น False ขา DCD จะมีสถานะลอจิก

รูปแบบการใช้งาน

object.CDHolding

เมื่อขา DCD มีลอจิก "1" (CDHolding = True) และเกิดไทม์เอาต์คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCDTO (Carrier Detect Timeout Error) และกระตุ้นให้เกิดเหตุการณ์ OnComm

DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR (DSR) ได้ว่ามีสถานะลอจิก "1" หรือ "0" โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า DSRHolding เป็น True ขา DSR จะมีสถานะลอจิก "1" ถ้าค่า DSRHolding เป็น False ขา DSR จะมีสถานะลอจิก "0"

รูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

object.DSRHolding

เมื่อขา DSR เป็นลอจิก "1" (DSRHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

Handshaking

กำหนดคุณสมบัติและคืนค่ารูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งานคำสั่ง

object.Handshaking [= value]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบด้วยกันคือ

1. comNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)
2. comXOnXOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF
3. comRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request To Send/Clear To Send)
4. comRTSXOnXOff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request To Send และ XON/XOFF

คุณสมบัติ Handshanking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายใน ระหว่างที่ข้อมูลถูกส่งไปยังบัฟเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัฟเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัฟเฟอร์ภาครับ โปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงของข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ handshaking ช่วยให้ผู้ใช้งานแน่ใจว่าข้อมูลที่ได้รับมานั้นไม่มีการสูญหายเมื่อบัฟเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือโอเวอร์โฟลว (overflow) โดยใช้วิธีการตรวจสอบความพร้อมของบัฟเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

object.Break [= value]

โดย Value เป็นค่าบูลีน ถ้า Value = True หมายถึง การส่งสัญญาณ Break ออกไป (ขา TxD เป็นลอจิก "1") ถ้า Value = False หมายถึงการเคลียร์สัญญาณ Break (ขา TxD เป็นลอจิก "0")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

ตัวอย่าง เป็นวิธีการส่งสัญญาณ Break ออกไปเป็นช่วงเวลาสั้นๆ ที่ 1/10 ของวินาที

```
MSComm.Break = True              ' Set the Break condition.
Duration! = Timer + .1           ' Set duration to 1/10 second.
Do Until Timer > Duration!               ' Wait for the duration to pass.
Loop
MSComm1.Break = False            ' Clear the Break condition.
```

2.4.9 การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์

จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังขาสถานะและขาควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายดายมาก โดยใช้คำสั่งเหล่านี้

DTREnable	สำหรับสั่งให้ขา DTR มีลอจิก “0” หรือ “1”
RTSEnable	สำหรับสั่งให้ขา RTS มีลอจิก “0” หรือ “1”
CTSHolding	สำหรับอ่านค่าสถานะจากขา CTS ว่ามีลอจิก “0” หรือ “1”
CDHolding	สำหรับอ่านค่าสถานะจากขา DCD ว่ามีลอจิก “0” หรือ “1”
DSRHolding	สำหรับอ่านค่าสถานะจากขา DSR ว่ามีลอจิก “0” หรือ “1”
Break	สำหรับการสั่งให้ขา TxD มีลอจิก “0” หรือ “1”

2.5 ไมโครคอนโทรลเลอร์

ตัวรับรับและตัวส่งในโครงงานนี้ ได้เลือกใช้ไมโครคอนโทรลเลอร์ 8 บิต ตระกูล MCS-51 โดยใช้เบอร์ AT89C52 มีคุณสมบัติดังนี้

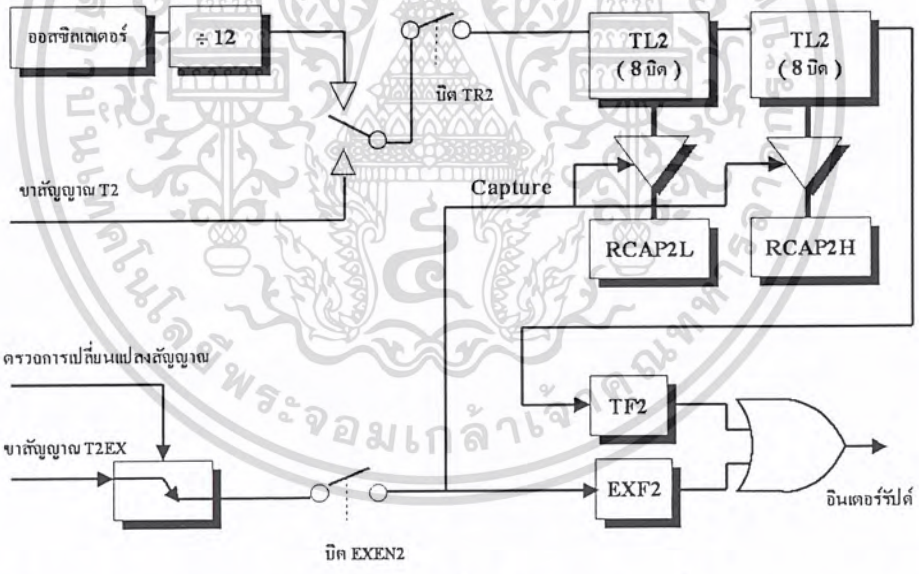
- เป็นไมโครคอนโทรเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำเป็นแบบแฟลสสามารถลบและเขียนใหม่ได้ 1000 ครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นแบบแรม
- ขาพอร์ตเป็นแบบ 2ทิศ ทางสามารถใช้ได้ทั้งอินพุตและเอาท์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูง 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิฟ

ไมโครคอนโทรเลอร์ในโครงงานนี้ จะใช้การทำงานไทเมอร์ 2 (Timer2)ในโหมดแคปเจอร์ (Capture mode)เป็นตัวตรวจจับสัญญาณอัลตร้าโซนิค การทำงานเป็นแบบ 16 บิต ไทเมอร์ 0 (Timer0) จะใช้ในการตรวจสอบสัญญาณเริ่มนับจากคอมพิวเตอร์ ไทเมอร์ 1(Timer 1) ถูกใช้ในการกำเนิดอัตราเร็วในการสื่อสารแบบอนุกรมหรืออัตราบอด (baud rate generater)

การทำงานในโหมดแคปเจอร์หรือตรวจจับสัญญาณในไทเมอร์ 2 นี้จะต้องกำหนดบิต EXEN2 ในรีจิสเตอร์ควบคุม T2CON ให้มีค่าเป็น "1" อินพุตในการตรวจจับสัญญาณคือขาอินพุต T2EX ซึ่งตรงกับพอร์ตขา P1.1 ที่ขา T2EX จะมีวงจรตรวจจับการเปลี่ยนแปลงของระดับลอจิก "1" หรือ "0" ริจิสเตอร์ TL2 และ TH2 จะทำการนับเพิ่มขึ้นเรื่อยๆจนกว่าที่ขา T2EX จะตรวจจับการเปลี่ยนแปลงได้ เมื่อตรวจจับได้ ค่าของรีจิสเตอร์ TL2 และ TH2 จะส่งต่อไปยังรีจีสเตอร์ RCAP2L และRCAP2H พร้อมกันนั้นยังทำการการเซตบิต EXF2 ให้เป็น "1" เพื่อสร้างสัญญาณอินเตอร์รัพท์ให้เกิดขึ้น



รูปที่ 2.10 แผนภาพแสดงการทำงานของไทเมอร์2 ในโหมดแคปเจอร์ของไมโครคอนโทรลเลอร์

การทำการนับ 1 ครั้งจะใช้สัญญาณนาฬิกา 12 ถ้าใช้สัญญาณนาฬิกา 12 MHz การนับใน 1 ครั้งของไทเมอร์จะใช้เวลาเท่ากับ 1ไมโครวินาที ไทเมอร์ 2 ในโหมดแคปเจอร์ 16 บิต นี้สามารถนับได้สูงสุด 65535 ดังนั้นจะใช้เวลาเท่ากับ 65.535 มิลลิวินาที

ความเร็วในการเดินทางของคลื่นเสียงในอากาศ มีค่าเท่ากับ 346 เมตรต่อวินาที ที่อุณหภูมิ 25 องศาเซนเซียส ในระบบออกแบบมาให้ระยะไกลสุดของการเดินทางของเสียงจากตัวส่งมาถึงตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับเท่ากับ 5 เมตร เวลาในการเดินทางของเสียงจะกับ 5 (เมตร) / 346 (เมตร/วินาที) = 0.014450 วินาที (14.45 มิลิวินาที) ซึ่งยังอยู่ในช่วงที่ไมโครคอนโทรลเลอร์สามารถวัดได้

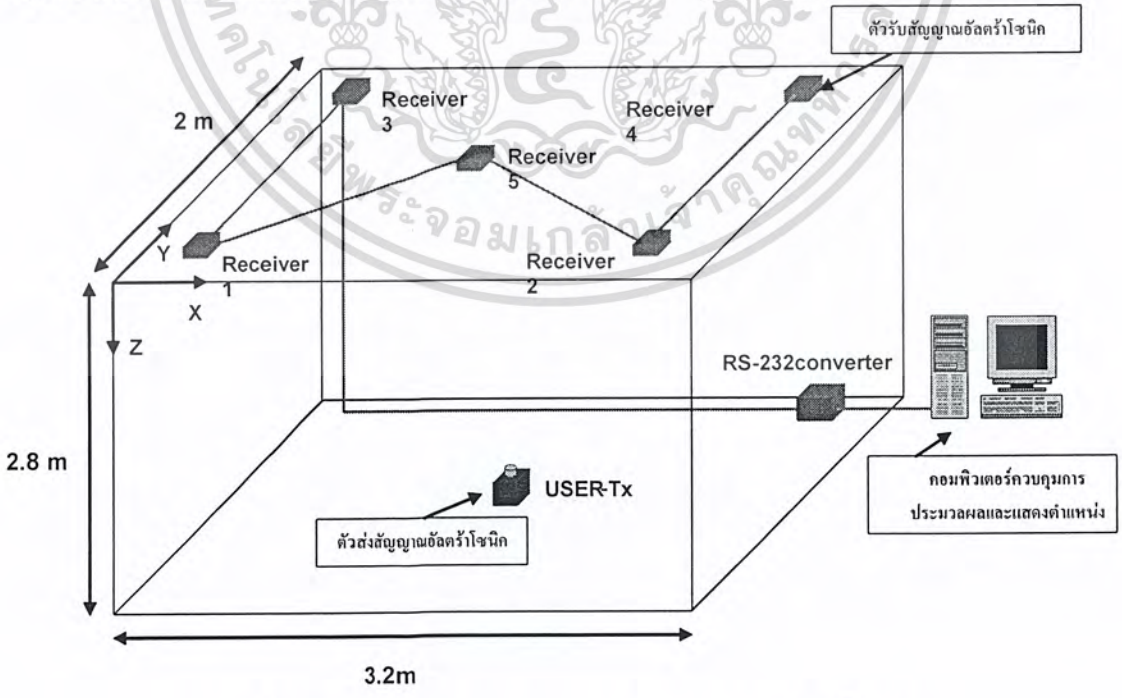
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและดำเนินการวิจัย

การดำเนินงานวิจัยขั้นแรกเริ่มจากการออกแบบระบบก่อน จากนั้นจึงเป็นการออกแบบและสร้างฮาร์ดแวร์ซึ่งได้แก่ ตัวรับส่งคลื่นสัญญาณอัลตร้าโซนิค ตัวแปลงระดับสัญญาณ RS-232C และแหล่งจ่ายไฟกระแสตรง การสื่อสารในระบบ สุดท้ายเป็นการออกแบบทางด้านซอฟท์แวร์ซึ่งมีสองส่วนคือ โปรแกรมไมโครคอนโทรลเลอร์กับโปรแกรมประมวลผลและแสดงผลโดยใช้คอมพิวเตอร์

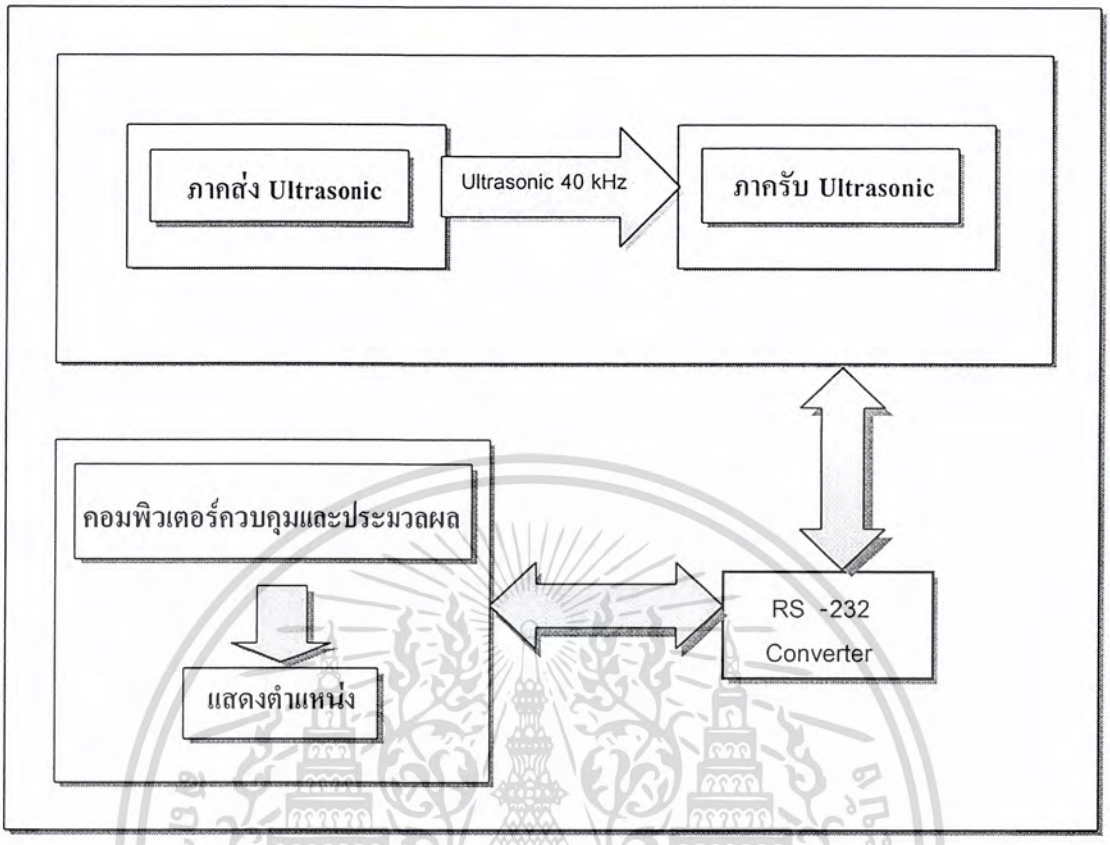
3.1 ออกแบบระบบ

การออกแบบระบบในโครงงานนี้จะใช้ตัวรับอยู่บนเพดานและตัวส่งอยู่บนพื้นห้อง การออกแบบในรูปแบบนี้มีความง่ายในการใช้คลื่นอัลตร้าโซนิคเป็นสัญญาณนำทางเพราะใช้ตัวส่งเพียงตัวเดียวความถี่ของคลื่นที่ส่งมีค่าเดียวจึงทำให้ตัวรับทั้งหมด 5 ตัวจับเวลาการเดินของคลื่นอัลตร้าโซนิคพร้อมกันได้ ซึ่งจะแตกต่างจากระบบที่ตัวส่งอยู่บนเพดานแล้วตัวรับอยู่บนพื้นห้องซึ่งจะต้องใช้การรับส่งคลื่นสัญญาณ 1 ครั้งต่อตัวส่งที่อยู่บนเพดานหนึ่งตัว ทำให้ต้องสูญเสียเวลาในการบอกตำแหน่งไปมากกว่า และการที่ใช้ตัวส่งจำนวนมากจะทำให้เกิดปัญหาของคลื่นสัญญาณสะท้อนที่จะส่งผลต่อการบอกตำแหน่งอย่างมากอีกด้วย



รูปที่ 3.1 แสดงภาพรวมของระบบจากการออกแบบในเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงแผนภาพการทำงานของระบบ

การทำงานของระบบโดยรวมจะเริ่มต้นจากภาคส่งอัลตร้าโซนิคส่งคลื่นสัญญาณอัลตร้าโซนิคออกไปพร้อมกันนั้นภาครับสัญญาณที่ประกอบด้วยตัวรับทั้งหมดที่อยู่บนเพดานจะเริ่มการจับเวลาการเดินทางของคลื่นอัลตร้าโซนิคและเมื่อตัวรับตัวใดได้รับสัญญาณก็จะหยุดจับเวลา เวลาที่จับได้ในตัวรับแต่ละตัวจะถูกส่งแบบการสื่อสารอนุกรมผ่านตัวแปลงระดับสัญญาณ RS-232C ไปคำนวณและแสดงตำแหน่งบนคอมพิวเตอร์ ซึ่งจะแสดงให้เห็นพิกัดตำแหน่งของผู้ใช้ ณ ตำแหน่งพิกัดที่สร้างขึ้นบนพื้นห้องขนาด 2 x 3.2 เมตร การควบคุมจังหวะการรับส่งข้อมูลและควบคุมการการจับเวลาจะใช้คอมพิวเตอร์ที่มีโปรแกรมที่พัฒนาขึ้นด้วยวิชวลเบสิก ในโครงงานนี้การควบคุมจังหวะการจับเวลาที่ตัวรับกับการส่งสัญญาณที่ตัวส่งให้เกิดขึ้นพร้อมกันนั้นจะใช้สายสัญญาณขา DTR ของพอร์ตอนุกรมของคอมพิวเตอร์เป็นสายสัญญาณควบคุม ทำให้มีความง่ายในการปรับเวลาในการอัพเดทตำแหน่งในแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

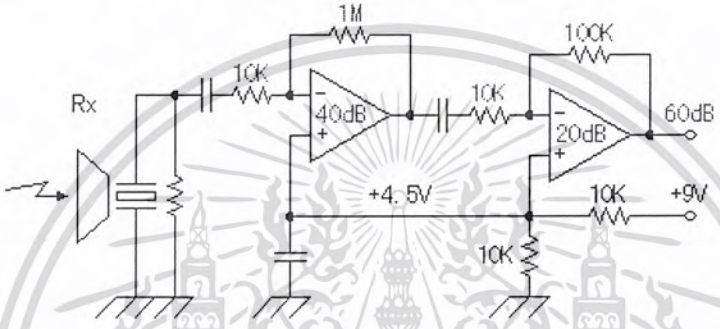
3.2 การออกแบบและสร้างฮาร์ดแวร์

3.2.1 การออกแบบภาครับสัญญาณอัลตร้าโซนิค

อุปกรณ์ตัวรับสัญญาณอัลตร้าโซนิคถูกติดตั้งไว้บนเพดานซึ่งมีทั้งหมด 5 ตัว วงจรที่สำคัญในตัวรับ ได้แก่ วงจรขยายสัญญาณ วงจรตรวจจับสัญญาณ วงจรควบคุมการเกิดสัญญาณอินเตอร์รัพท์

ภาครับสัญญาณอัลตร้าโซนิคแบ่งวงจรออกเป็น

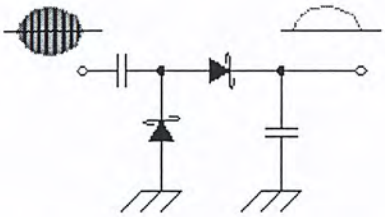
1. วงจรขยายสัญญาณ (Signal amplification circuit)



รูปที่ 3.3 แสดงวงจรขยายสัญญาณของตัวรับคลื่นอัลตร้าโซนิค

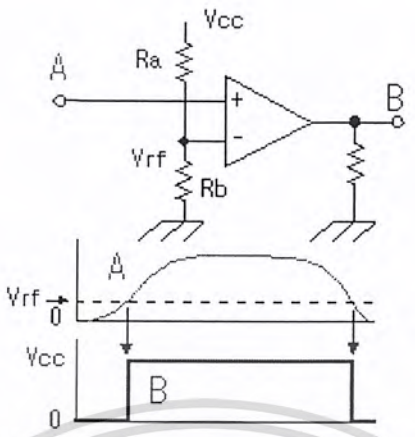
สัญญาณอัลตร้าโซนิคที่รับมาจากตัวเซนเซอร์แล้วผ่านการขยาย 1000 เท่า(60 dB) จากออปแอมป์ 2 ตัว ตัวแรกขยาย 100 เท่า(40 dB) และตัวที่สองขยาย 10 เท่า(20dB) การทำงานของของวงจรจะใช้แหล่งจ่ายไฟด้านบวก +9 โวลต์ ด้วยเหตุนี้ที่อินพุตด้านบวกของออปแอมป์จึงทำงานเพียงครึ่งหนึ่งจากแรงดันที่ให้เข้าไป เมื่อสัญญาณที่ใช้เป็นกระแสสลับการขยายจะมีค่าเท่ากับ 4.5 โวลต์ จากแนวกึ่งกลางถึงค่าสูงสุดในด้านบวก การใช้ออปแอมป์ทำงานแบบป้อนกลับแบบลบจะทำให้แรงดันที่ขยายทางด้านบวกและด้านลบมีค่าเท่ากัน เทคนิคนี้ทำให้ออปแอมป์ที่ต้องการไฟบวกลบสามารถทำงานด้วยแหล่งจ่ายไฟเพียงด้านเดียวได้ ในวงส่วนขยายนี้ออปแอมป์เบอร์ TL082 ในการขยายสัญญาณ

2. วงจรส่วนตรวจจับสัญญาณ



รูปที่ 3.4 แสดงวงจรเรคติไฟร์แบบครึ่งคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



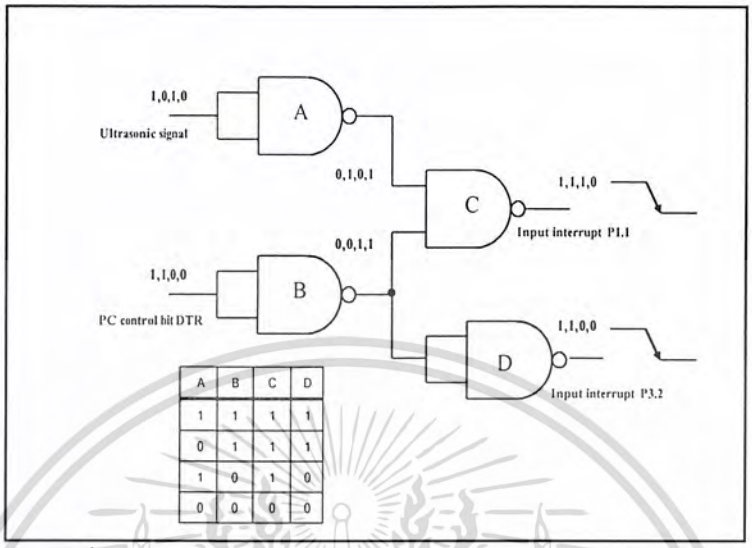
รูปที่ 3.5 แสดงวงจรเปรียบเทียบสัญญาณ

เมื่อสัญญาณผ่านการขยายแล้วจะเข้าวงจรเรคติไฟร์แบบครึ่งคลื่น(half-wave retification circuit) โดยใช้ช็อคกี้ไดโอด ระดับของสัญญาณจะเป็นตามสัญญาณที่เข้ามา การใช้ช็อคกี้ไดโอดเพราะสามารถตอบสนองที่สัญญาณความถี่สูงได้ดี สัญญาณที่เอาท์พุตของวงจรเรคติไฟร์แบบครึ่งคลื่นจะเข้าสู่วงจรเปรียบเทียบสัญญาณ(Comparator) ซึ่งเป็นวงจรที่ใช้ออปแอมป์เบอร์ LM358 ทำหน้าที่เป็นตัวเปรียบเทียบสัญญาณ โดยใช้ความแตกต่างเอาท์พุตทางด้านบวกและลบของออปแอมป์

การทำงานของวงจรนี้ไม่ต้องใช้การทำงานป้อนกลับแบบลบ (Negative feedback) ซึ่งในกรณีนี้สัญญาณเข้ามาที่ด้านบวก ถ้ามีค่าแรงดันมากกว่าสัญญาณที่ด้านลบเพียงเล็กน้อยเอาท์พุตจะเข้าสู่สภาวะอิ่มตัว (saturation state) และมีค่าเท่ากับแรงดันไฟเลี้ยง แต่ถ้าสัญญาณเข้ามาที่ด้านบวกถ้ามีค่าแรงดันน้อยกว่าสัญญาณที่ด้านลบเพียงเล็กน้อยเอาท์พุตของวงจรจะมีค่าแรงดันไฟเท่ากับ 0 โวลต์ แสดงในรูปที่ 3.5 ในวงจรขาด้านบวกของออปแอมป์จะเป็นอินพุตของสัญญาณและด้านลบจะต่อกับแรงดันที่มีค่าคงที่ ซึ่งเป็นแรงดันอ้างอิง (Vrf) จะเป็นตัวบอกช่วงตอบสนองของสัญญาณ เมื่อมีสัญญาณอัลตร้าโซนิคเข้ามามีค่าสูงกว่า 0.4 V ที่เอาท์พุตของวงจรตรวจจับจะมีค่าเท่ากับแรงดันไฟเลี้ยง(+9 V) สัญญาณที่ผ่านวงจรนี้แล้วจะเข้าสู่วงจรกลับสัญญาณให้มีค่าตรงกันข้ามกับสัญญาณที่เข้ามา โดยปรกติเมื่อยังไม่มีสัญญาณอินพุตใดๆเข้ามาสัญญาณจะมีระดับเท่ากับไฟเลี้ยง แต่ถ้ามีอินพุตเข้ามาจะเปลี่ยนเป็น 0 โวลต์ โดยทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

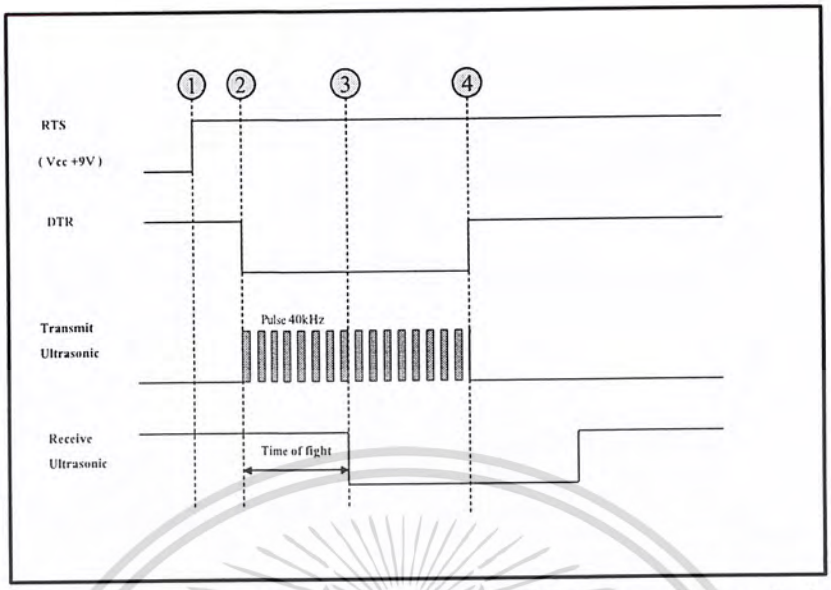
3. วงจรควบคุมการเกิดสัญญาณอินเตอร์รัพท์



รูปที่3.6 แสดงวงจรควบคุมการเกิดสัญญาณอินเตอร์รัพท์

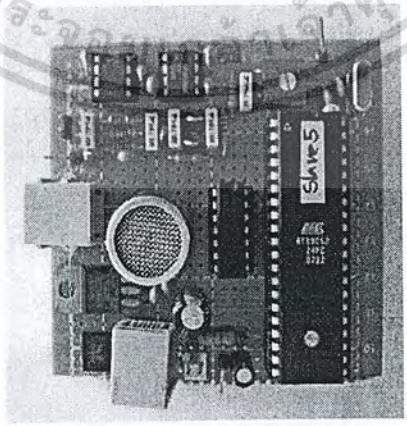
สัญญาณที่ผ่านออปแอมป์ที่เป็นวงจรเปรียบเทียบสัญญาณมาแล้วนั้นสัญญาณที่ได้จะมีระดับแรงดันที่สูงอยู่ไม่สามารถต่อเข้าไมโครคอนโทรลเลอร์ตรงๆได้ จึงต้องนำมาผ่านวงจรซึ่งทั้งทำหน้าที่ในการปรับระดับสัญญาณให้เป็น TTL และควบคุมการจับเวลาของไมโครคอนโทรลเลอร์ โดยได้ออกแบบวงจรออร์เกตที่สร้างขึ้นจากไอซีแนนเกตเบอร์ 4011 ดังรูปที่ 3.6 จากรูป A ,B, C คือวงจรออร์เกต ที่ขาอินพุต A จะรับสัญญาณจากวงจรส่วนขยายสัญญาณอัลตร้าโซนิค ที่ขาอินพุต B จะรับสัญญาณมาจากคอมพิวเตอร์ขา DTR ของพอร์ตอนุกรมซึ่งทำหน้าที่ให้สัญญาณเริ่มนับพร้อมกับและควบคุมการรับสัญญาณอัลตร้าโซนิค ซึ่งถ้าอินพุตที่ขา B นี้เป็นระดับลอจิกเป็น “ 1” อินพุตใดๆที่เข้ามาที่ A จะไม่ทำให้เกิดการอินเตอร์รัพท์เกิดขึ้นที่ขา P1.1 ของไมโครคอนโทรลเลอร์จนกว่าอินพุตที่ขา B จะมีระดับรอจิกเป็น “ 0 ” จึงจะยอมให้เกิดกาอินเตอร์รัพท์ที่ P1.1 ได้ และการที่อินพุตที่ B เป็น “ 0 ” ยังทำให้เกิดการอินเตอร์รัพท์ที่ P3.2 อีกด้วย จากวงจรได้มีการต่อแนนเกตเพิ่มอีกหนึ่งตัวคือ D เพื่อทำหน้าที่กลับลอจิกที่เอาท์พุตของ B ให้เกิดการอินเตอร์รัพท์ที่ระดับสัญญาณขาลงได้ การทำงานของวงจรนี้สำคัญมากถ้าเกิดมีสัญญาณรบกวนเกิดขึ้นที่อินพุตทั้ง A และ B ระยะทางที่คำนวณได้จะเกิดความคาดเคลื่อนไปอย่างมาก และที่ขาไฟเลี้ยงของ 4011 นี้ต้องต่อตัวเก็บประจุ 0.1 uF เข้าไปด้วยเพื่อแก้สัญญาณสไปร์ที่เกิดจากการดึงกระแสในไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



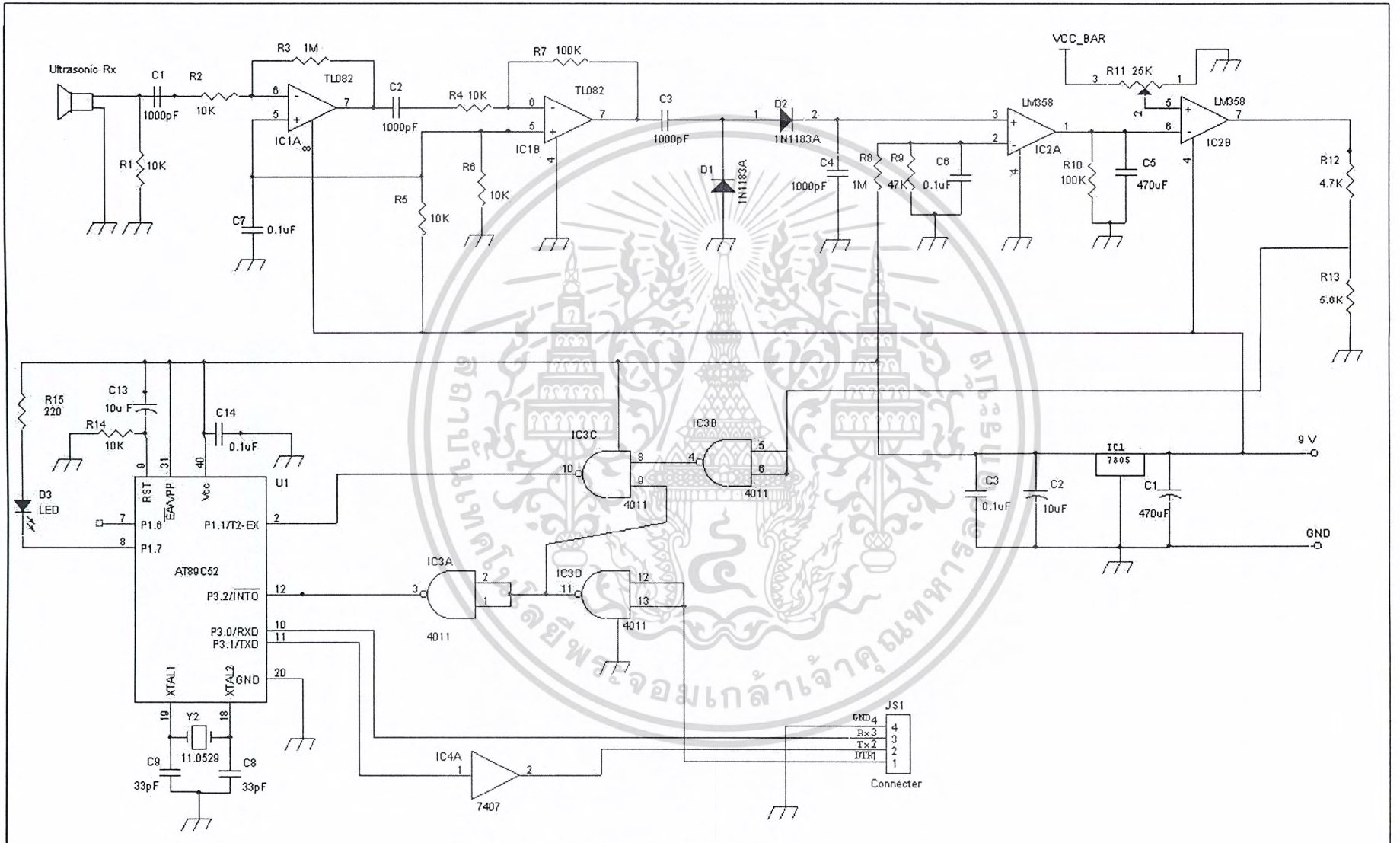
รูปที่ 3.7 แสดงแผนภาพการทำงานในการจับเวลาการเดินทางของเสียงจากตัวส่งมาถึงตัวรับ

การทำงานของภาครับในการจับเวลาการเดินทางของคลื่นอัลตร้าโซนิคจากตัวส่งเริ่มต้นจากสถานะที่ 1 ขา RTS ของพอร์ตอนุกรมของคอมพิวเตอร์มีค่าระดับสัญญาณลอจิกเป็น “1” จะเป็นการจ่ายไฟเลี้ยงให้กับอุปกรณ์ทั้งในภาครับและส่ง สถานะที่ 2 ขาสัญญาณ DTR ระดับสัญญาณจากลอจิก “1” ไปเป็น “0” จะส่งผลให้ตัวส่งอัลตร้าโซนิคส่งคลื่นสัญญาณความถี่ 40 kHz ออกมาพร้อมตัวรับทั้งหมดเริ่มจับเวลาพร้อมกัน สถานะที่ 3 เมื่อตัวรับได้รับคลื่นสัญญาณที่ส่งมาก็จะหยุดจับเวลา ทำให้ได้คาบเวลาของการเดินทางของคลื่นสัญญาณจากตัวส่งเดินทางถึงตัวรับหรือเรียกว่า Time of Flights สถานะที่ 4 ระดับลอจิกของขาสัญญาณ DTR กลับไปเป็น “1” ทำให้ตัวส่งคลื่นสัญญาณ อัลตร้าโซนิคหยุดการส่งคลื่นสัญญาณ



รูปที่ 3.8 แสดงอุปกรณ์ตัวรับสัญญาณอัลตร้าโซนิค

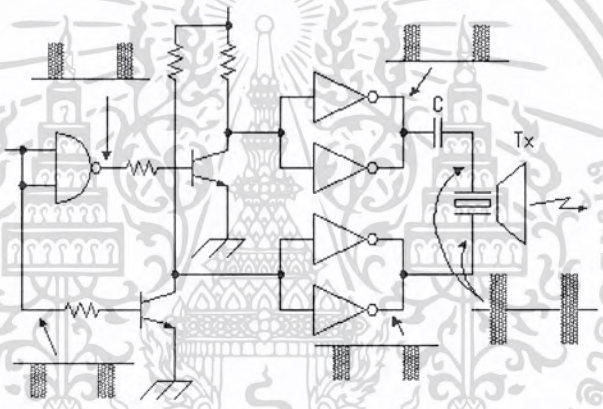
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



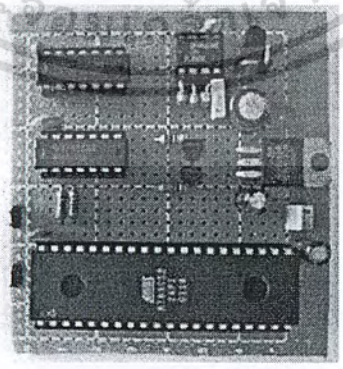
รูปที่ 3.9 แสดงวงจรตัวรับสัญญาณอัลตร้าโซนิค

3.2.2 ออกแบบภาคส่งสัญญาณอัลตร้าโซนิค

ในวงจรจะใช้อินเวอร์เตอร์ขับตัวทรานสดิวเซอร์อัลตร้าโซนิค โดยใช้ตัวอินเวอร์เตอร์สองตัวต่อขนานกันเพื่อเพิ่มกำลังในการส่ง เฟสของแรงดันที่ให้เข้าไปทางด้านบวกและด้านลบของตัวทรานสดิวเซอร์จะมีค่าแตกต่างกัน 180 องศา กระแสถูกตัดโดยตรงจากตัวเก็บประจุและค่าแรงดันที่ให้เข้าไปในตัวทรานสดิวเซอร์ทั้งสองด้าน แรงดันไฟเลี้ยงของวงจรขับมีค่าเท่ากับ +9 โวลต์ ซึ่งจะถูกกลับแรงดันจากตัวทรานซิสเตอร์ที่ความคุมการทำงานมาจากไมโครคอนโทรลเลอร์ซึ่งจะสร้างสัญญาณความถี่ 40 kHz และความสามารถของตัวอินเวอร์เตอร์ซึ่งใช้ไอซีเบอร์ 4069 ซึ่งเป็น CMOS สามารถทำงานเป็น On/Off ที่ความถี่สูงได้

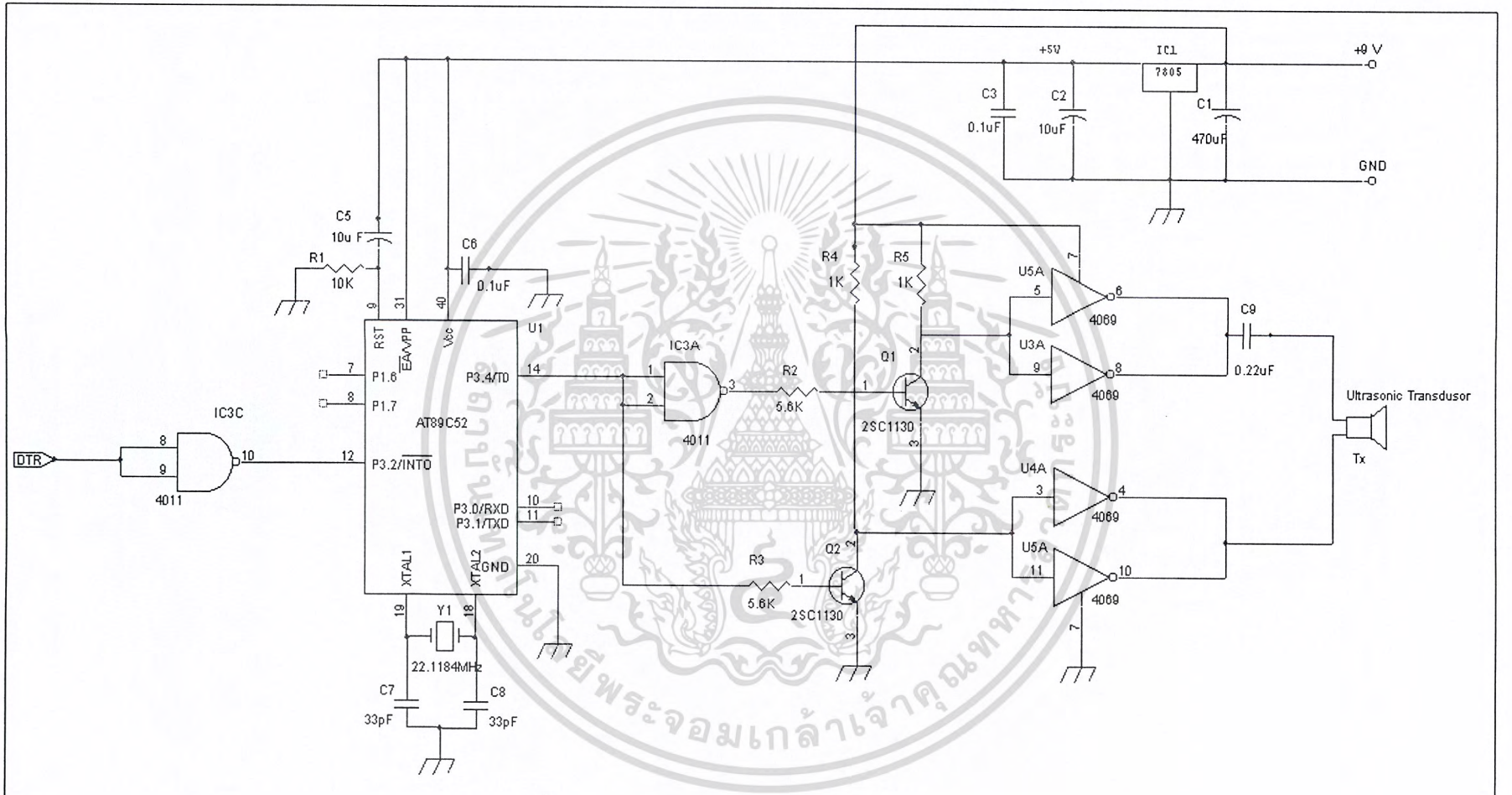


รูปที่ 3.10 แสดงวงจรส่งสัญญาณอัลตร้าโซนิค



รูปที่ 3.11 แสดงอุปกรณ์ตัวส่งสัญญาณอัลตร้าโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



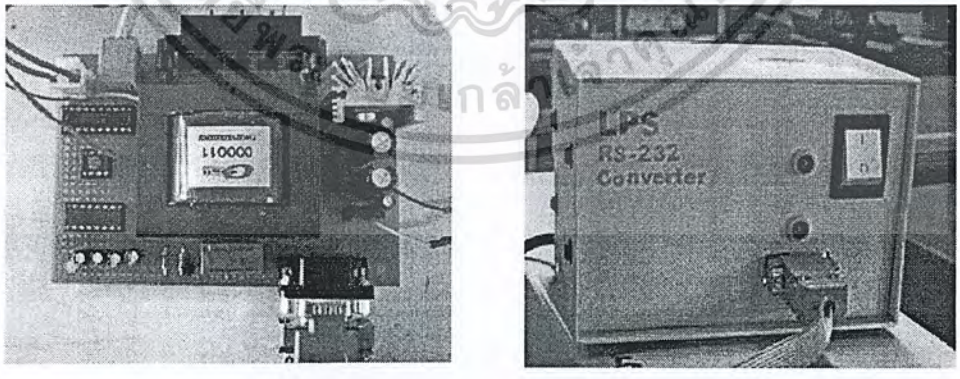
รูปที่ 3.12 แสดงวงจรตัวส่งสัญญาณอัลตร้าโซนิค

3.2.3 อุปกรณ์แปลงสัญญาณ RS-232 และแหล่งจ่ายไฟกระแสตรง

เริ่มต้นจากคอมพิวเตอร์เชื่อมต่อกับกล่องแปลงสัญญาณผ่านทางคอนเน็กเตอร์แบบ DB-9 สัญญาณจากคอมพิวเตอร์นี้จะมีระดับแรงดันตามมาตรฐาน RS-232 กล่าวคือ มีระดับแรงดัน ±3V ถึง ±12V เพื่อให้สามารถทำงานเข้ากันได้กับวงจรของไมโครคอนโทรลเลอร์ ซึ่งระดับสัญญาณเป็นแบบ TTLจึงต้องต่อผ่าน DS275 เพื่อแปลงสัญญาณให้อยู่ในระดับทีทีแอลเสียก่อน เนื่องจากสายสัญญาณที่ใช้มีมากถึง 4 สายจึงต้องเพิ่มไอซี MAX232 เพิ่มเข้ามาอีกหนึ่งตัว คือสายสัญญาณจากเอาต์พุตของขา DTR และ RTS ของพอร์ตอนุกรมของคอมพิวเตอร์

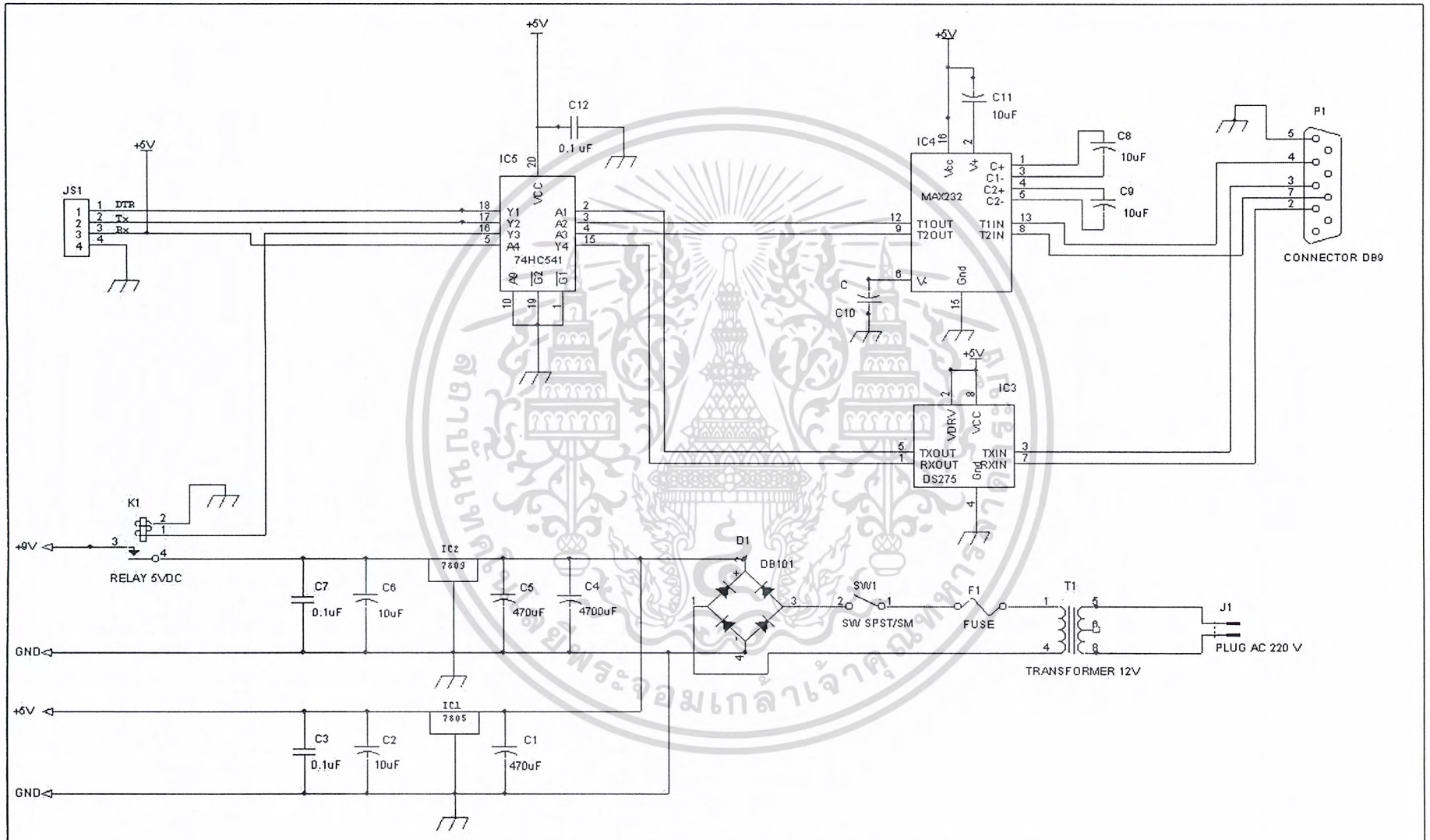
สัญญาณที่ผ่านไอซีแปลงสัญญาณแล้วจะถูกส่งต่อไปยังไอซีบัพเฟอร์เบอร์ 74HC541 ทำหน้าที่ขยายกระแสให้กับสายสัญญาณทั้งหมดพร้อมทั้งยังช่วยป้องกันความเสียหายที่อาจเกิดกับคอมพิวเตอร์กรณีมีความผิดพลาดจากการต่อวงจร สายสัญญาณทั้งหมดจากพอร์ตอนุกรมจะถูกส่งต่อไปที่ตัวรับแต่ละตัว โดยใช้ขาสัญญาณเอาต์พุตของพอร์ตอนุกรมซึ่งประกอบด้วยขา TxD ,ขา RxD,ขา DTR ,ขาRTS และขาสัญญาณอินพุต RxD นอกจากนี้ยังมีสายไฟเลี้ยง +9V และกราวด์ต่อไปยังตัวรับทุกตัวด้วย

ในวงจรยังมีภาคแปลงไฟกระแสตรง +5V และ +9V ซึ่งไฟกระแสตรง +5V จะใช้เป็นไฟเลี้ยงไอซีในวงจรและรีเลย์ ส่วน+9V จะใช้เป็นแหล่งจ่ายไฟให้กับตัวรับสัญญาณทั้งหมด การควบคุมไฟเลี้ยงที่จ่ายให้กับตัวรับนี้จะผ่านตัวรีเลย์ซึ่งควบคุมจากคอมพิวเตอร์ผ่านขาสัญญาณ RTS เพื่อสามารถทำการรีเซตไมโครคอนโทรลเลอร์ได้เมื่อเกิดปัญญาหาหยุดการทำงาน



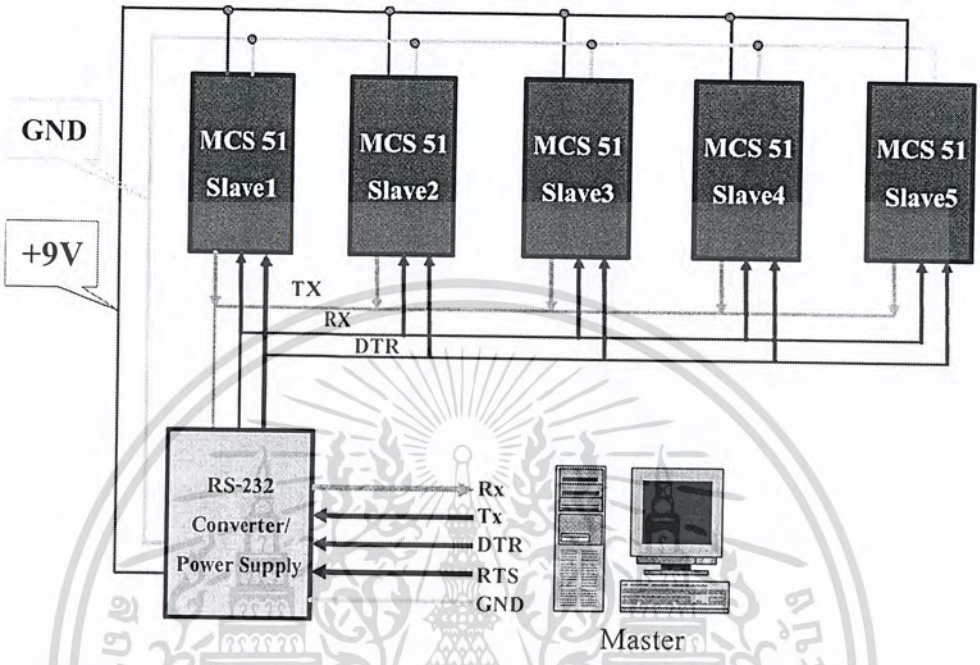
รูปที่ 3.13 แสดงอุปกรณ์แปลงสัญญาณ RS-232 และแหล่งจ่ายไฟกระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงวงจรแปลงระดับสัญญาณ RS-232Cและแหล่งจ่ายไฟกระแสตรง

3.3 การสื่อสารในระบบ



รูปที่ 3.15 แสดงแผนภาพการติดต่อสื่อสารในระบบ

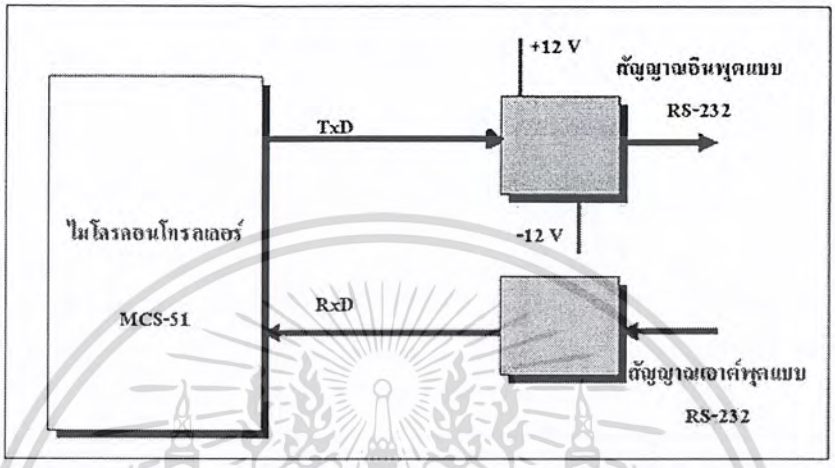
การสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะใช้การสื่อสารแบบอนุกรมในระบบมัลติโปรเซสเซอร์(Multiprocessor System) ซึ่งเป็นการติดต่อสื่อสารกันระหว่างไมโครคอนโทรลเลอร์หลายตัว โดยใช้คอมพิวเตอร์ทำหน้าที่เป็นตัวหลักในระบบ ซึ่งจะเรียกว่า มาสเตอร์ (Master) และในส่วนของไมโครคอนโทรลเลอร์จะเรียกว่า สเลฟ(Slave) โดยมีจุดประสงค์ของการสื่อสารเพื่อเก็บข้อมูลจากสเลฟแต่ละตัว เป็นลำดับไป การเชื่อมต่อไมโครคอนโทรลเลอร์กับคอมพิวเตอร์จะใช้ตัวแปลงสัญญาณ RS-232 ติดต่อกับพอร์ตอนุกรม สามารถส่งข้อมูลไปได้ระยะทางไกลสุด 50 ฟุต และขึ้นอยู่กับอัตราบอดที่ใช้ด้วย

การสื่อสารในระบบมัลติโปรเซสเซอร์(Multiprocessor System)จะเริ่มต้นโดยการส่งข้อมูลค่าของแอดเดรสจากมาสเตอร์ไปยังสเลฟที่ต้องการติดต่อออกไปก่อนเป็นอันดับแรกก่อนการส่งข้อมูลจริง ไมโครคอนโทรลเลอร์ทุกตัวจะได้รับข้อมูลที่ส่งออกนั้นมาเปรียบเทียบว่าตรงกับแอดเดรสของตนเองหรือไม่ หากสเลฟตัวใดมีแอดเดรสตรงจะรอรับข้อมูลจริงที่จะส่งออกมาอีกครั้ง และการสิ้นสุดการรับข้อมูลจะจบด้วยรหัสสิ้นสุดการส่งข้อมูลจากมาสเตอร์

ในโครงงานนี้การสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะเป็นลักษณะแบบฮาฟดูเพล็กซ์ซึ่งเป็นการสื่อสารแบบสองทางที่ไม่ได้เกิดขึ้นในเวลาเดียวกัน โดยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งข้อมูลแอดเดรสออกไปก่อนและจะรอรับค่าข้อมูลตอบกลับจากไมโครคอนโทรลเลอร์ตัวที่ติดต่อนั้น ดังนั้นเวลาในการรับส่งจะเกิดขึ้นไม่พร้อมกัน เนื่องจากการเชื่อมต่อกันของทั้งสองอุปกรณ์มีระดับสัญญาณไม่เท่ากันจึงต้องใช้ตัวแปลงระดับสัญญาณจากระดับสัญญาณ TTL ในด้านของไมโครคอนโทรลเลอร์ให้เป็นระดับสัญญาณ RS-232 เพื่อติดต่อกับคอมพิวเตอร์ได้



รูปที่ 3.16 แผนภาพแสดงให้เห็นถึงแนวการเปลี่ยนแปลงระดับสัญญาณแบบ TTL จากไมโครคอนโทรลเลอร์ MCS-51 ไปเป็นระดับสัญญาณอินพุตแบบ RS-232 ไปเป็นระดับสัญญาณแบบ TTL ก่อนที่จะได้เชื่อมต่อกับขาของไมโครคอนโทรลเลอร์ MCS-51

ในระบบยังได้ออกแบบให้ใช้ขาสัญญาณในDB9 ของพอร์ตอนุกรมของคอมพิวเตอร์อีก 2 ขา นอกเหนือจากขา Tx ใช้ในการส่งข้อมูลและขา Rx ใช้ในการรับข้อมูล คือขา DTR และขา RTS โดยขา DTR ทำหน้าที่ควบคุมไมโครคอนโทรลเลอร์ในการเริ่มจับเวลาให้มีจังหวะพร้อมกันการอธิบายในส่วนนี้จะอยู่ในหัวข้อส่วนของการออกแบบซอฟท์แวร์ และขา RTS จะทำหน้าที่รีเซตไมโครคอนโทรลเลอร์เมื่อเกิดปัญหาให้หยุดการทำงาน

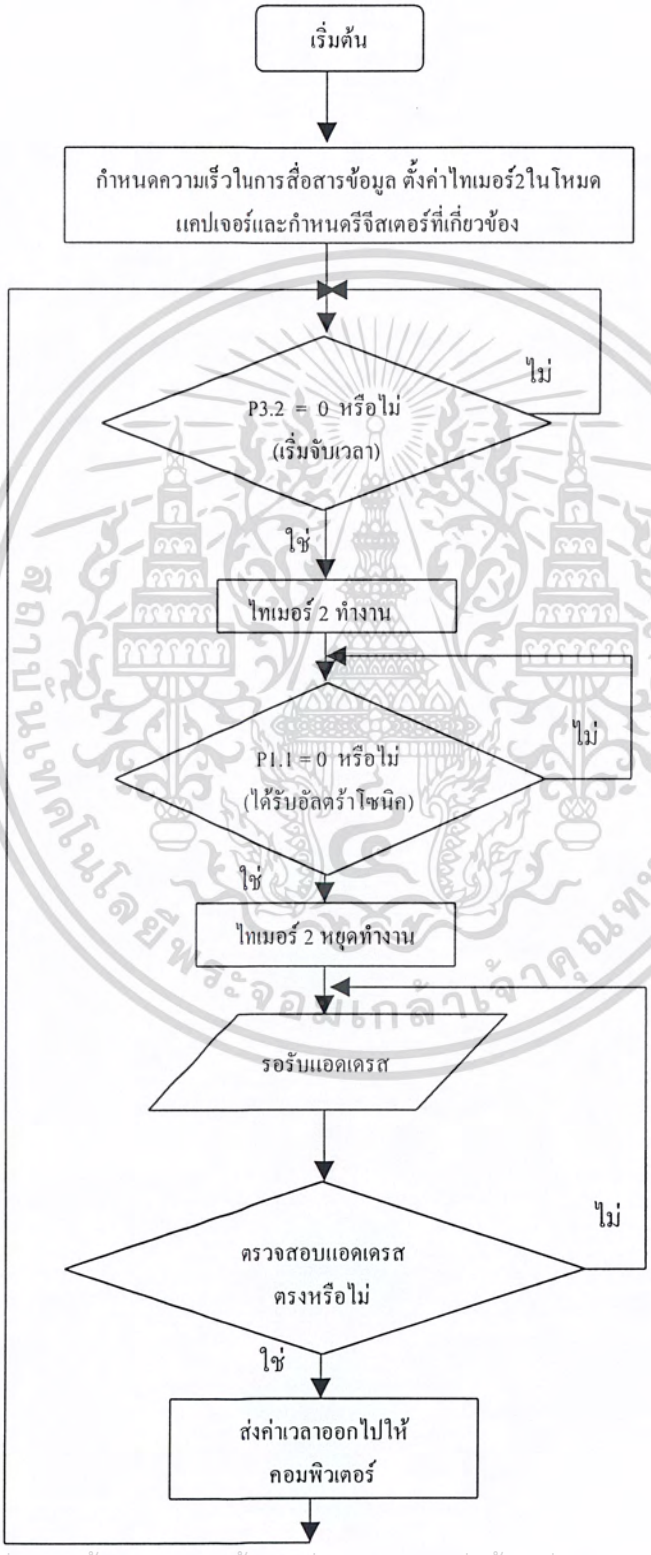
3.4 การออกแบบทางด้านซอร์ฟแวร์

3.4.1 โปรแกรมการทำงานในส่วนของไมโครคอนโทรลเลอร์

ในส่วนของโปรแกรมของไมโครคอนโทรลเลอร์เขียนด้วยภาษาซี การทำงานจะตั้งค่ารีจีสเตอร์ไทเมอร์ 0 รอรับสัญญาณอินเทอร์รัพท์จากภายนอกและ ตั้งค่าไทเมอร์ 2 ให้อยู่ในโหมดแคปเจอร์ เมื่อมีสัญญาณสั่งเริ่มนับเข้ามาสั่งจากสายสัญญาณ DTR จะทำให้ไทเมอร์ 2 ทำงานจับเวลาและเมื่อตรวจจับสัญญาณอัลตร้าโซนิคได้ ไทเมอร์ 2 จะหยุดทำงาน และรอรับค่าแอดเดรสจากพอร์ตอนุกรมเมื่อมีค่าแอดเดรสที่ตรงเข้าไมโครคอนโทรเลอร์ก็จะส่งค่าเวลาที่จับได้ออกไปให้คอมพิวเตอร์ประมวลผลต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.17 แผนภาพแสดงการทำงานของโปรแกรมในส่วนของไมโครคอนโทรลเลอร์ที่ทำงานเป็นสเลฟ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

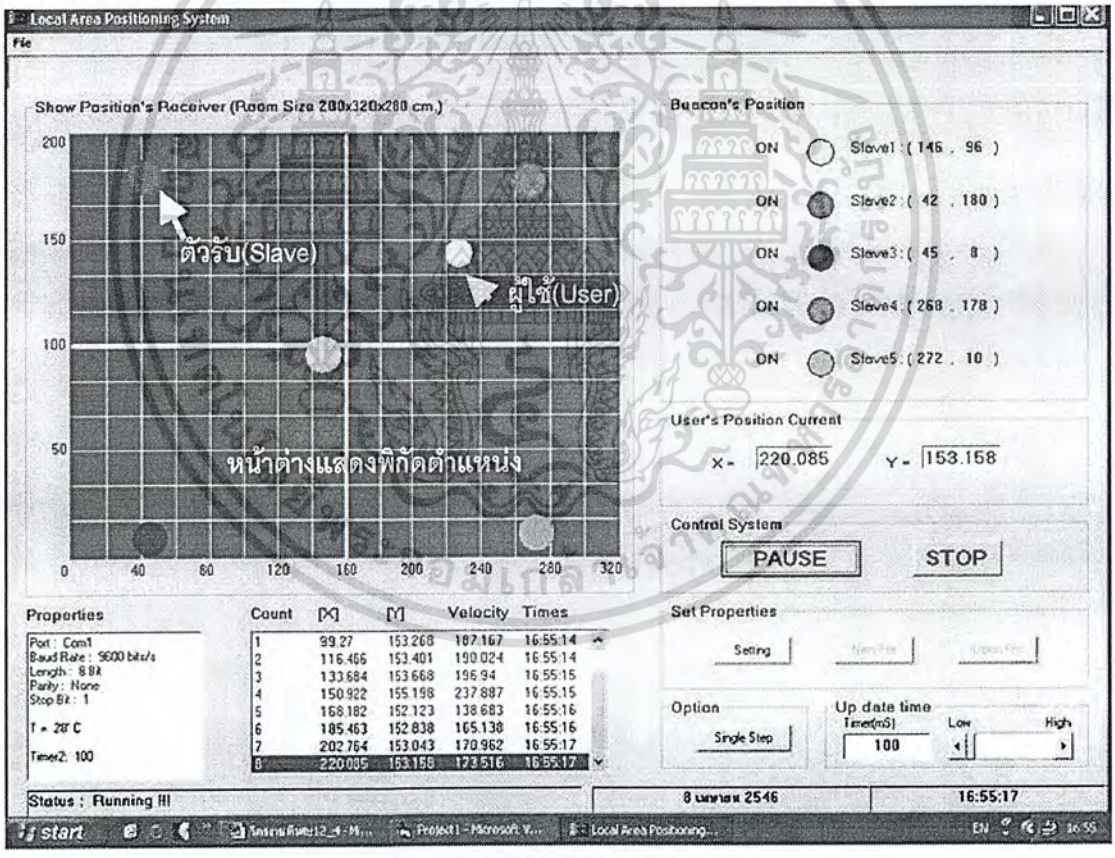
3.4.2 โปรแกรมประมวลผลและแสดงตำแหน่ง

ประกอบด้วย 3 ส่วนด้วยกันคือ

1. ส่วนของโปรแกรมหลัก

ในส่วนของโปรแกรมหลักนี้ ขั้นแรกเราต้องสร้างไฟล์ขึ้นมาก่อน (เพื่อที่จะเก็บค่าข้อมูลที่ได้จากการทำงานของโปรแกรม) เมื่อกดปุ่ม Start โปรแกรมจะเริ่มทำงานและจะแสดงข้อมูลของตัวส่งในเวลาต่างๆ ที่ Position's Data ซึ่งข้อมูลในส่วนนี้จะถูกเก็บไว้ในไฟล์ที่เราสร้างไว้ตอนแรก โดยในส่วนนี้เราสามารถปรับเปลี่ยนค่าเวลาให้เร็วขึ้นหรือช้าลงได้จาก Timer Control

โปรแกรมหลักของโครงงานนี้ใช้โปรแกรม Visual Basic6 ในการเขียน ซึ่งโปรแกรมนี้ประกอบด้วยส่วนต่างๆ ดังนี้



รูปที่ 3.18 แสดงหน้าตาโปรแกรมหลักของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Show Position เป็นส่วนแสดงตำแหน่งของตัวส่ง และตัวรับ

Properties เป็นส่วนแสดงค่า Port, Baudrate, Data Length, Parity, Stop Bit, T และ Timer2 จากที่ตั้งไว้

Position's Data เป็นส่วนแสดงข้อมูลตำแหน่งขณะที่ตัวส่งเคลื่อนที่ไปตามเวลา

Receiver's Position เป็นส่วนแสดงค่าพิกัดของตัวรับแต่ละตัว (สามารถปรับเปลี่ยนค่าได้โดยแก้ไขที่ Setting)

User's Position เป็นส่วนแสดงค่าพิกัดตำแหน่งของตัวส่ง ตามพิกัด (x, y)

Control System เป็นส่วนควบคุมการทำงาน โดยจะเริ่มทำงานเมื่อกดปุ่ม Start และหยุดทำงานเมื่อกดปุ่ม Stop

Set Properties ประกอบด้วย

Setting เป็นส่วนตั้งค่าของ Properties ต่างๆ

New File เป็นส่วนเก็บค่าข้อมูล ซึ่งต้องทำการเปิดทุกครั้ง ก่อนที่จะเริ่มให้โปรแกรมทำงาน

Open File เป็นส่วนเปิดข้อมูลที่ได้จากการบันทึกจาก New File

Option ประกอบด้วย

Single Step เป็นส่วนทดลองการติดต่อกับคอมพอร์ต โดยการส่งสัญญาณ US ไปยังตัวรับทีละตัว

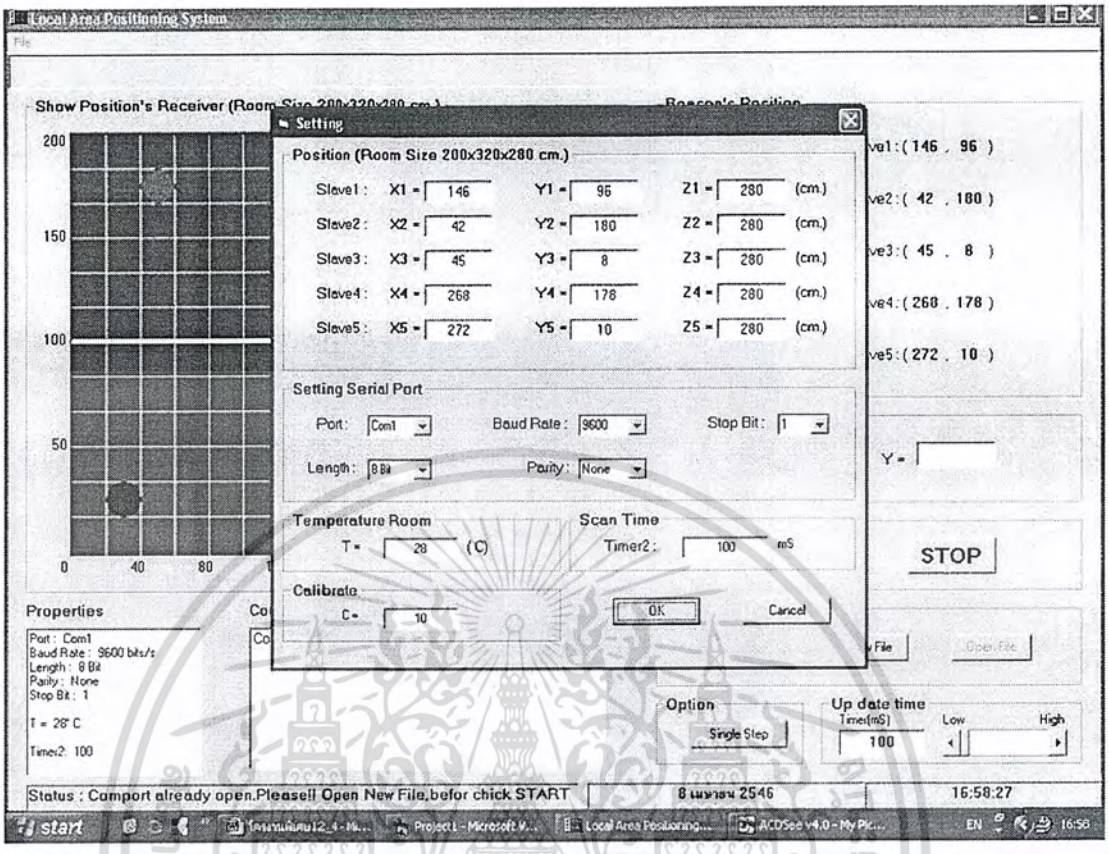
Timer Control เป็นส่วนควบคุมค่าเวลา โดยจะมีค่าอยู่ระหว่าง 100 – 1000 ms

2. ส่วนของ Setting

ในส่วนของ Setting นี้เราสามารถที่จะปรับเปลี่ยนค่าในส่วนต่างๆ ได้ ซึ่งผลที่ได้ก็จะเปลี่ยนแปลงไปตามค่าที่ปรับเปลี่ยนไป

ส่วนของ Setting ประกอบด้วยส่วนต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงส่วน Setting ของโปรแกรม จากการกดปุ่ม Setting ในส่วนของ Set Properties

Position เป็นส่วนของการตั้งค่าของตำแหน่งของตัว Receiver แต่ละตัว โดยจะใช้ในการตั้งค่า x และ y ส่วนค่า z เป็นค่าคงที่(ค่าความสูงของห้อง)

Setting Serial Port เป็นส่วนของการตั้งค่าของ Serial Port ซึ่งประกอบด้วย ค่า Port, Data Length, Baud Rate, Parity และ Stop Bit

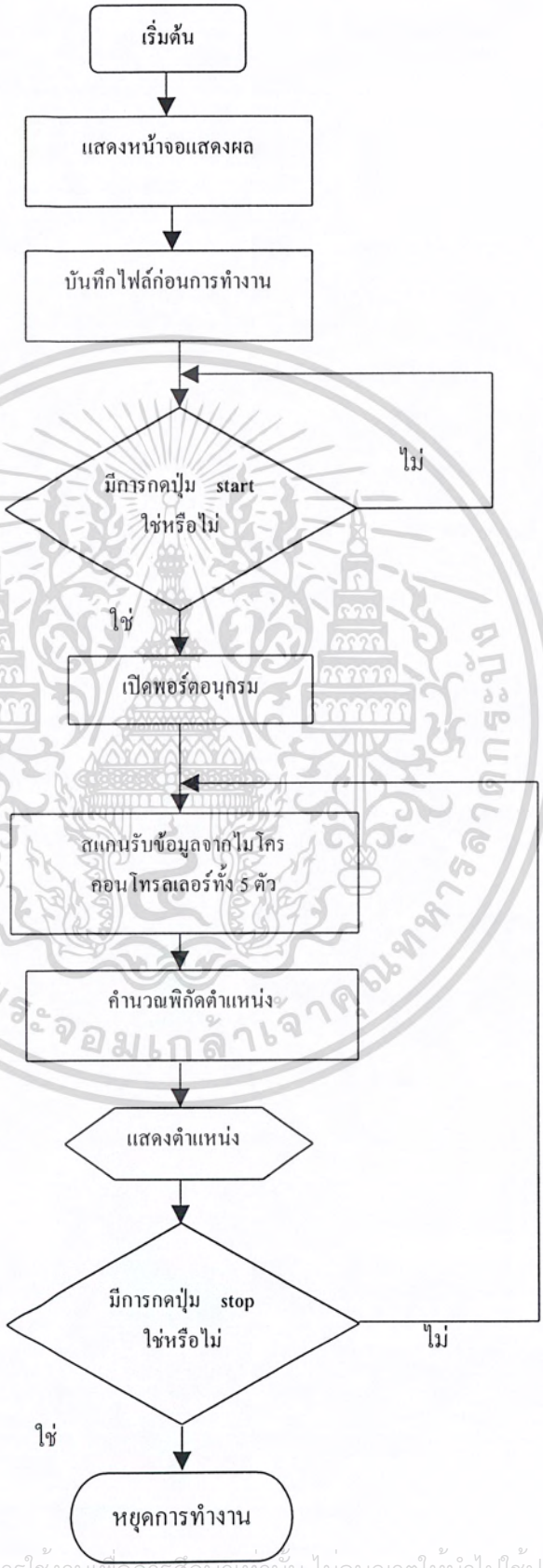
Temperature Room เป็นส่วนแสดงค่าของ อุณหภูมิห้อง ในโครงงานนี้จะกำหนดให้เป็น 25 °C

Scan Time เป็นส่วนที่กำหนดค่าของเวลา Timer2

Calibrate เป็นส่วนปรับค่าของค่าเวลา ที่อาจเกิดเนื่องจากความล่าช้าในการส่งสัญญาณ US

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 แผนภาพแสดงการทำงานของโปรแกรมวิชวลเบสิก

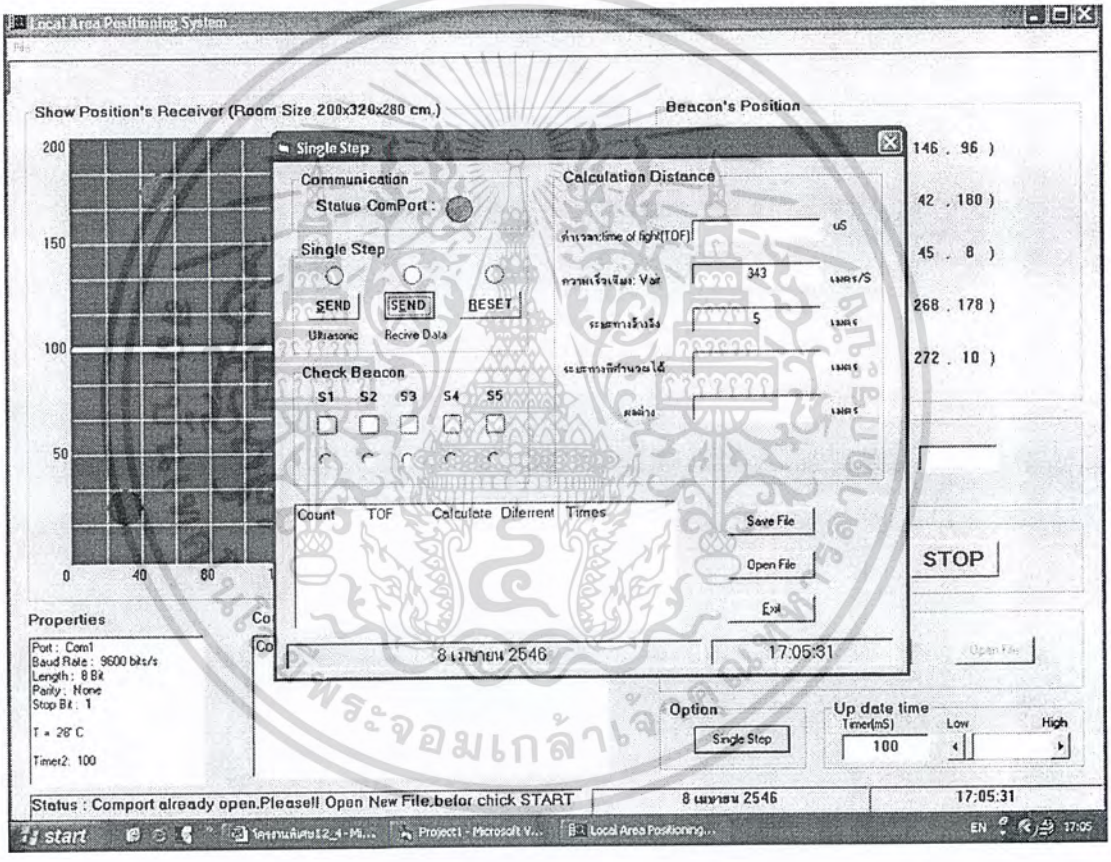


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนของ Single Step

ในส่วนของ Single Step นี้ เราต้องเลือกก่อนว่าจะใช้ตัวรับตัวใด แล้วจึงทำการกดปุ่มส่งสัญญาณอัลตร้าโซนิคก่อน และทำการกดปุ่มส่งข้อมูลของตัวรับ เพื่อที่จะให้ได้ค่า TOF ออกมา ซึ่งค่านี้จะไปแสดงในส่วนของค่าเวลา TOF ในส่วนของ Calculate Distance และจะได้ค่าผลต่างออกมา ซึ่งในส่วนนี้เราจะใช้ในการทดสอบการทำงานของตัวรับแต่ละตัวว่าสามารถแสดงค่าเวลา TOF จากตัวส่งสัญญาณอัลตร้าโซนิค มายังตัวรับที่จะทดสอบนั้น มีค่าถูกต้องหรือไม่

ในส่วนของ Single Step ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้



รูปที่ 3.21 แสดงส่วน Single Step ของโปรแกรม จากการกดปุ่ม Single Step ในส่วนของ Option

Communication เป็นส่วนแสดงสถานะของ ComPort

Single Step ประกอบด้วย

- *ปุ่มส่งสัญญาณ Us*
- *ปุ่มส่งค่าข้อมูล*
- *ปุ่มรีเซ็ต*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Check Receiver เป็นส่วนเลือกว่าจะทดลองใช้ตัวรับตัวไหน เป็นตัวรับสัญญาณ

Calculate Distance เป็นส่วนของการคำนวณ ประกอบด้วย

- *ค่าเวลา TOF* เป็นค่าที่ได้มาจากการกดปุ่มส่งค่าข้อมูล ในส่วนของ Single Step
- *ค่าความเร็วเสียง* เป็นค่าความเร็วเสียงในอากาศ มีค่าประมาณ 343 m/s
- *ระยะทางอ้างอิง* เป็นระยะทางจริงๆ ที่กำหนดไว้
- *ระยะทางที่คำนวณได้* เป็นระยะทางที่ได้จากการคำนวณจากค่าเวลาและค่าคามเร็วเสียง
- *ผลต่าง* เป็นค่าความแตกต่างของระยะทางจริงๆ กับระยะทางที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและอภิปรายผล

การทดลองในโครงงานนี้แบ่งเป็นสองตอนคือ ตอนที่ 1 เป็นการทดลองเพื่อทดสอบอุปกรณ์และปรับเทียบให้มีความถูกต้องในการหาพิกัดตำแหน่ง ตอนที่ 2 เป็นการทดสอบการนำไปใช้งานหาตำแหน่ง เพื่อหาเปอร์เซ็นต์ความคลาดเคลื่อนของระบบและหาผลกระทบต่างๆเมื่อนำไปใช้งานจริง

<u>ตอนที่ 1</u>

4.1 การทดสอบอุปกรณ์

4.1.1 การทดลองมุมการตอบสนองต่อสัญญาณของตัวรับ

การทดลองนี้เป็นการทดสอบมุมของตัวรับในการตอบสนองสัญญาณกับตัวส่ง ว่ามีผลอย่างไรกับมุมที่เปลี่ยนไป โดยวัดค่าเอาท์พุตจากวงจรขยาย

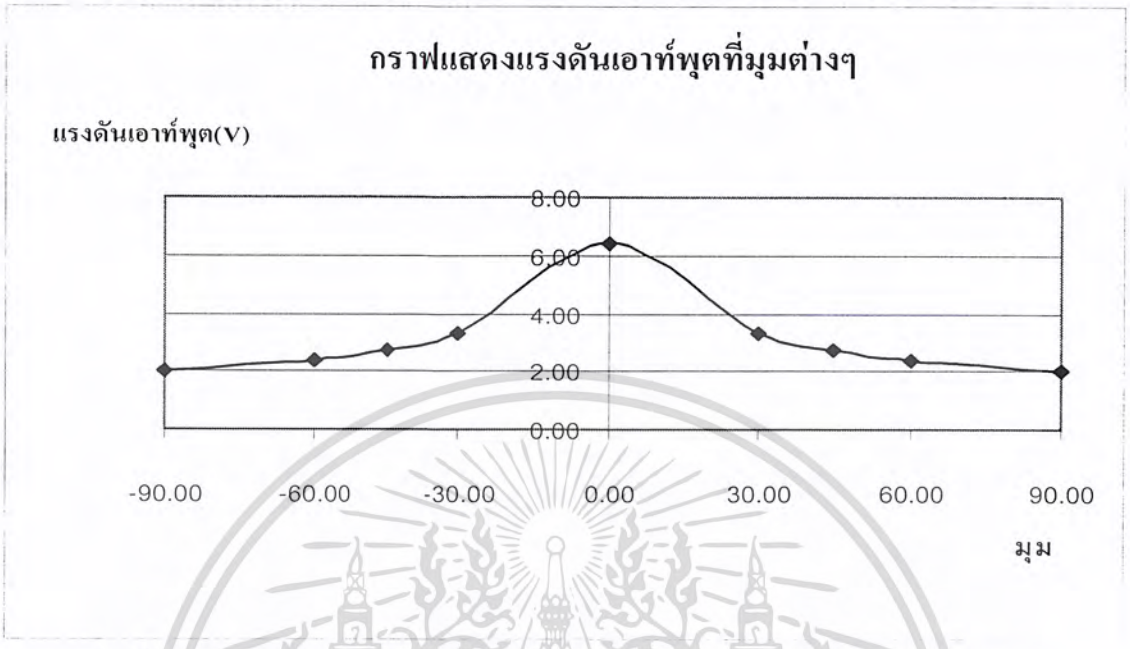
อุปกรณ์ที่ใช้ ได้แก่ ตัวรับและตัวส่งคลื่นอัลตร้าโซนิค ตลับเมตร ไม้บรรทัดวัดมุม

การดำเนินการทดลองนี้จะกำหนดระยะห่างของตัวรับกับตัวส่งไว้ที่ 5 เมตร ซึ่งกำหนดเป็นระยะที่ตัวรับและตัวส่งสามารถส่งคลื่นสัญญาณได้อย่างไม่มีปัญหา การวัดค่าแรงดันเอาท์พุตนั้นจะใช้ออสซิลโลสโคปทำการวัดจากเอาท์พุตของวงจรขยาย

ตารางที่ 4.1 แสดงผลการทดลองมุมการตอบสนองของตัวรับ

มุมการรับ (องศา)	ค่าเอาท์พุตของวงจรขยาย
-90	2.00
-60	2.40
-45	2.72
-30	3.28
0	6.40
30	3.28
45	2.72
60	2.40
90	2.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงกราฟที่ได้จากผลการทดลองมุมการตอบสนองต่อสัญญาณของตัวรับ

วิจารณ์ผลการทดลอง

ในการทดลองนี้จะเห็นว่าที่มุม 0 องศา (ตัวส่งกับตัวรับอยู่ตรงกัน) การรับส่งสัญญาณจะทำได้ดีที่สุด และค่าแรงดันเอาท์พุตของตัวรับจะลดลงไปตามมุมที่เปลี่ยนแปลงไป แต่วงจรตรวจจับของตัวรับยังสามารถตอบสนองได้ เนื่องจากยังมีค่ามากกว่า Vrf ของวงจรComparator ของวงจรตัวรับในบทที่ 2 ซึ่งออกแบบไว้มีค่าต่ำสุดอยู่ที่ 0.4 โวลต์ ทำให้แม้ว่าจะเอียงตัวรับทำมุม ± 90 องศา ตัวรับก็ยังสามารถตอบสนองต่อสัญญาณจากตัวส่งได้ ซึ่งทำให้มั่นใจได้ว่าตัวรับทั้ง 5 ตัวจะสามารถรับสัญญาณจากตัวส่งได้ทุกตัว

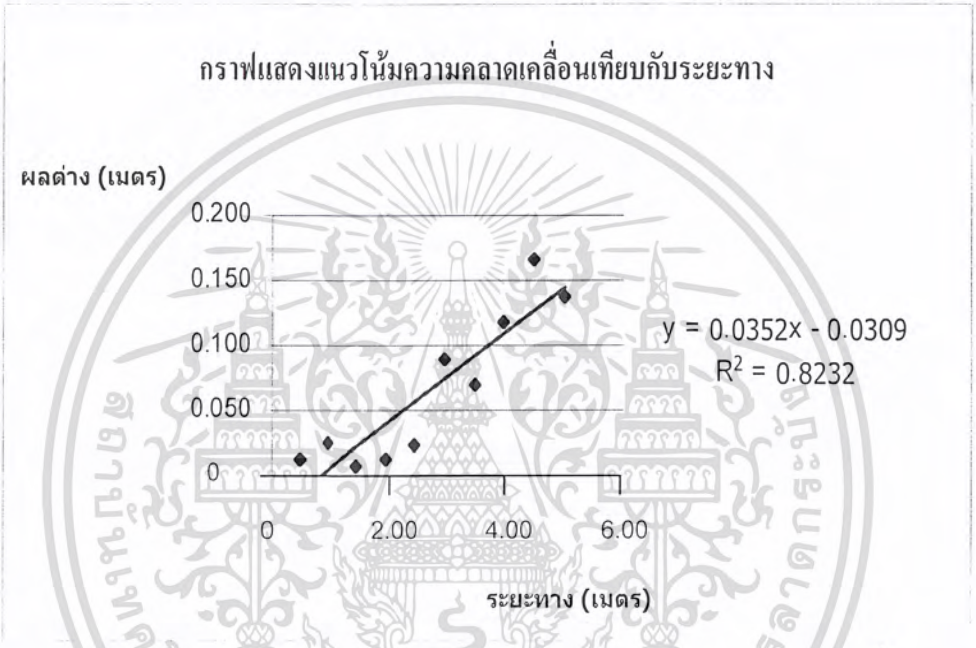
4.1.2 การทดลองวัดระยะทาง

การทดลองนี้ทำขึ้นเพื่อทำการทดสอบความแม่นยำในการหาตำแหน่ง โดยการทดสอบจะใช้การกำหนดระยะทางที่แน่นอนขึ้นมาและทำการทดลองวัดระยะทางนั้นว่าถูกต้องหรือแตกต่างจากระยะทางอ้างอิงไปเท่าใด เพื่อนำมาหาค่าแก้ไขให้การวัดระยะทางจากตัวส่งถึงตัวรับให้ถูกต้องมากขึ้น

อุปกรณ์ที่ใช้ในการทดลอง ได้แก่ ตัวรับและตัวส่งคลื่นอัลตร้าโซนิค ตลับเมตร ออสซิลโลสโคป มัลติมิเตอร์ ตัวแปลงระดับสัญญาณ RS 232C แหล่งจ่ายไฟกระแสตรง และคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดำเนินงานทดลองจะเริ่มจากติดตั้งอุปกรณ์ตัวรับต่อเชื่อมกับคอมพิวเตอร์ผ่านตัวแปลงระดับสัญญาณ RS-232C จากนั้นใช้โปรแกรม Single Step เก็บบันทึกผลการทดลอง จัดตัวรับกับตัวส่งให้อยู่ห่างกันเป็นระยะ 0.5 เมตรซึ่งเป็นระยะสั้นสุดและเพิ่มขึ้นทุกๆ 0.5 เมตร จนถึงระยะ 5 เมตร ซึ่งการทดลองเก็บค่า 30 ค่าในแต่ละระยะ จากนั้นนำผลการทดลองแต่ระยะที่ได้มาหาค่าเฉลี่ยและผลต่างจากระยะอ้างอิง เพื่อนำมาเขียนกราฟแสดงแนวโน้มความคลาดเคลื่อน



รูปที่ 4.2 แสดงกราฟที่ได้จากผลการทดลองวัดระยะทาง โดยในการทดลองนี้ทำการทดลองที่อุณหภูมิห้อง 28 $^{\circ}$ C

วิจารณ์ผลการทดลอง

ในการทดลองนี้ที่ระยะที่ยิ่งไกลมากขึ้นความคลาดเคลื่อนก็จะเพิ่มมากขึ้น ดังนั้นในระบบจึงออกแบบให้ระยะทางไกลสุดที่ระบบสามารถทำงานไม่เกินระยะ 5 เมตร ซึ่งความคลาดเคลื่อนที่ยอมรับได้ ไม่เกิน 20 เซนติเมตร แต่เมื่อทำการปรับแก้แล้วความคลาดเคลื่อนจะลดน้อยลง

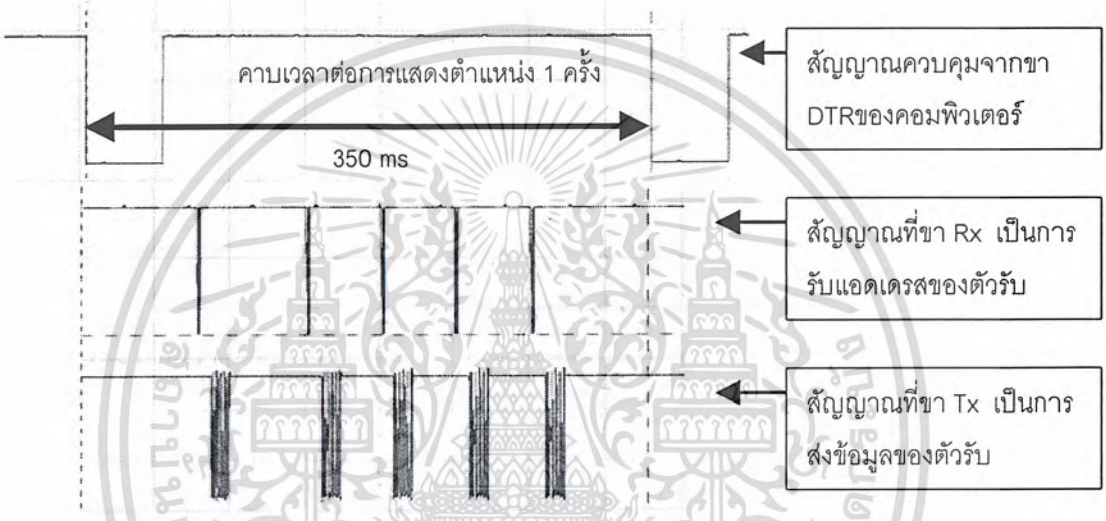
4.1.3 การทดลองสื่อสารในระบบ

ในการทดลองนี้เป็นการนำเอาตัวรับทั้งหมด 5 ตัว มาเชื่อมต่อเข้าด้วยกันและทดลองรับส่งค่าข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ เพื่อทำการปรับปรุงโปรแกรมหลักในส่วนควบคุมและหาเวลาน้อยที่สุดที่สามารถทำได้โดยที่การรับส่งข้อมูลไม่เกิดความผิดพลาด

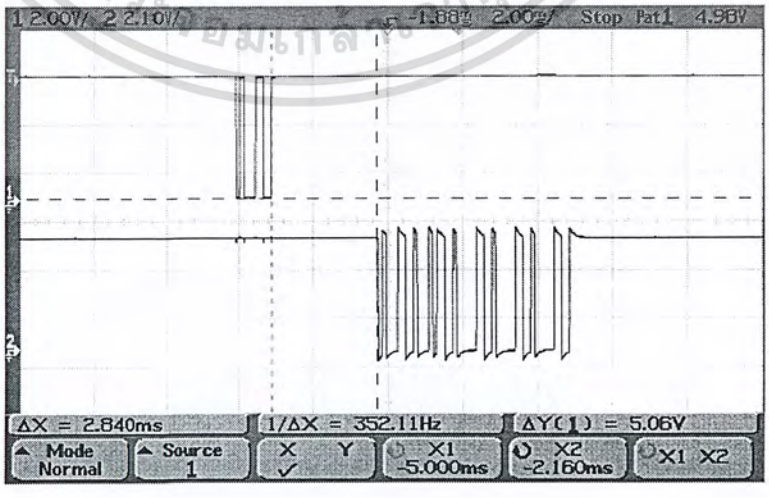
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้ในการทดลอง ได้แก่ ตัวรับและตัวส่งคลื่นอัลตร้าโซนิค ตลับเมตร ออสซิลโลสโคป มัลติมิเตอร์ ตัวแปลงระดับสัญญาณ RS-232C และแหล่งจ่ายไฟกระแสตรง และคอมพิวเตอร์

ในการทดลองนี้จะใช้ออสซิโลสโคปจับสัญญาณ 3 เส้นที่เข้าไมโครคอนโทรลเลอร์ตัวรับ ซึ่งได้แก่ สายสัญญาณจากขา DTR ของคอมพิวเตอร์ สายสัญญาณจาก Rx และ Tx ของไมโครคอนโทรลเลอร์



รูปที่ 4.3 แสดงการสื่อสารข้อมูลในระบบ โดยประกอบด้วยสัญญาณควบคุมจากขา DTR ของคอมพิวเตอร์ สัญญาณที่ขา Rx และ Tx ของไมโครคอนโทรลเลอร์ตัวรับ



รูปที่ 4.4 แสดงการรับและส่งข้อมูลของไมโครคอนโทรลเลอร์ตัวรับในหนึ่งครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

การทดลองนี้ใช้อัลตราบอดเรดที่ 9600 บิต/วินาที ซึ่งจากการทดลองพบว่าจะต้องปรับ Timer ที่ควบคุมการรับส่งข้อมูลของตัวรับให้มีคาบเวลาเร็วสุดได้ที่ 50 ms ซึ่งการรับและส่งข้อมูลทั้งหมด 5 ครั้ง จะใช้เวลาเท่ากับ 250 ms และรวมคาบเวลาในการควบคุมของขาสัญญาณ DTR ด้วยอีก 100 ms ดังนั้นเวลาที่ระบบสามารถแสดงพิกัดตำแหน่งได้เร็วสุดต่อครั้งเท่า 350 mS

วิจารณ์ผลการทดลอง

การทดลองนี้ทำการพบว่าไม่สามารถปรับค่า Timer ในส่วนของการรับส่งข้อมูลในโปรแกรมหลักของคอมพิวเตอร์ให้มีคาบเวลาต่ำกว่า 50 ms ได้เนื่องจากต้องเผื่อเวลาในการรอรับการส่งคลื่นสัญญาณมาจากตัวส่ง ซึ่งกำหนดไว้มีค่าไม่ต่ำกว่า 14 ms และต้องเผื่อเวลาในการเดินทางของข้อมูลในสายสัญญาณและการทำงานของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ดังนั้นจึงทำให้ 1 วินาทีระบบสามารถแสดงตำแหน่งได้ 3 ครั้ง หรือ 350 mS ต่อการแสดงผล 1 ครั้ง

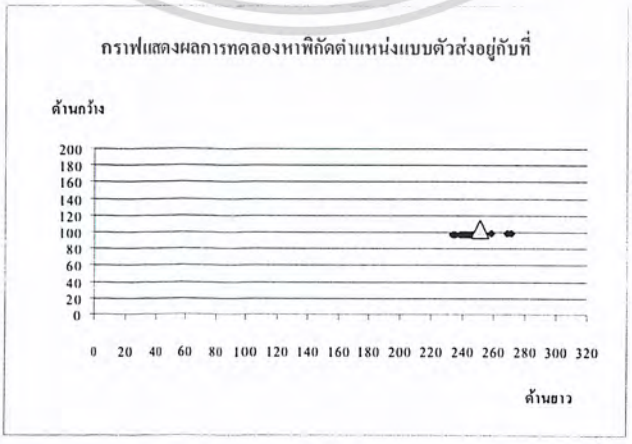
ตอนที่ 2

4.2 การลองนำระบบไปใช้งาน

การทดลองหาตำแหน่งแบบหยุดนิ่ง (Static position)

การทดลองนี้เป็นการนำตัวรับทั้ง 5 ตัวมารวมกันเป็นระบบ โดยเชื่อมต่อกับคอมพิวเตอร์แล้วทำการใช้โปรแกรมที่เขียนขึ้นจากวิชวลเบสิก (Visual Basic) มาเป็นโปรแกรมรับค่าจากตัวรับทั้ง 5 ตัวและประมวลผลออกมาเป็นพิกัดตำแหน่ง ของตัวส่ง หรือตำแหน่งของผู้ใช้ (User) การทดลองนี้ได้ออกแบบขนาดของพื้นที่ห้องให้เหมาะสม โดยมีขนาดของห้องเป็น 2 x 3.2 เมตร และมีความสูง 2.8 เมตร

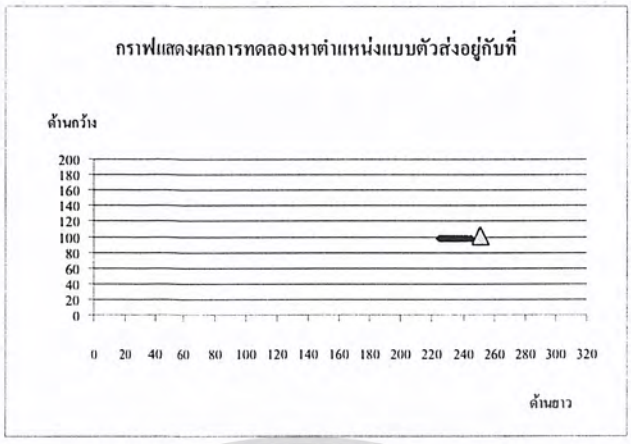
ผลการทดลอง



พิกัดเฉลี่ย
(245.46, 97.52)
คลาดเคลื่อน
x = 1.82 %
y = 2.48 %

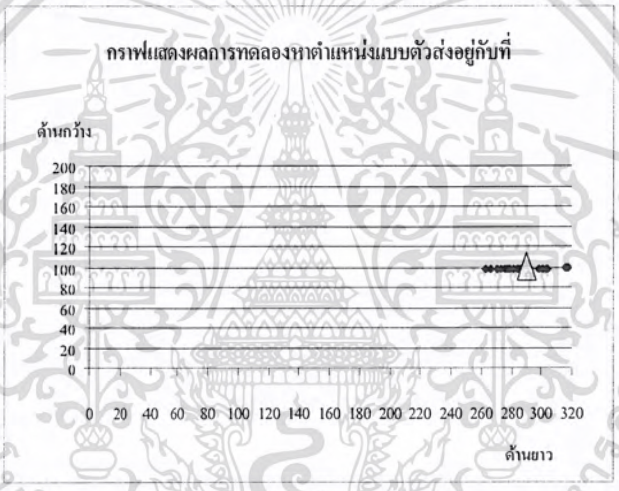
รูปที่4.5 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



พิกัดเฉลี่ย
(233.84, 97.32)
คลาดเคลื่อน
x = 6.46 %
y = 2.68 %

รูปที่4.6 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 2



พิกัดเฉลี่ย
(287.43, 98.25)
คลาดเคลื่อน
x = 14.97 %
y = 1.75 %

รูปที่4.7 แสดงผลการทดลองหาพิกัดตำแหน่งแบบตัวส่งอยู่กับที่ ครั้งที่ 3

*หมายเหตุ △ คือพิกัดตำแหน่งของตัวส่งสัญญาณ : กำหนดอยู่ที่พิกัด (250, 100)

วิจารณ์ผลการทดลอง

จากทดลองระบบสามารถบอกพิกัดตำแหน่งของตัวส่งค่อนข้างแม่นยำอยู่ในระดับความคลาดเคลื่อน 5-10 cm จากการทดลองทั้งสามครั้งพบว่าในแนวแกน x มีความคลาดเคลื่อนค่อนข้างมาก สาเหตุของความคลาดเคลื่อนในการทดลองนี้ส่วนใหญ่เกิดจากอุปกรณ์ในส่วนของภาครับ ซึ่งมีปัญหาในการสื่อสารและมีสัญญาณรบกวน และในการทดลองในส่วนนี้ได้พยายามทดลองให้ระบบแสดงตำแหน่งในแบบเคลื่อนที่ ซึ่งทดลองโดยใช้รถขนาดเล็กเคลื่อนที่ไปตามรางที่สร้างขึ้น ผลการทดลองในส่วนนี้พบว่าไม่สามารถทำได้ เพราะเกิดปัญหาในการรับส่งคลื่นอัลตร้าโซนิค ทำให้พิกัดตำแหน่งที่ได้มีความคลาดเคลื่อนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุป

โครงงานพิเศษนี้เป็นการศึกษาการนำเอาคลื่นอัลตร้าโซนิคมาใช้ในการหาพิกัดตำแหน่งภายในอาคาร โดยใช้ตัวรับทั้งหมด 5 ตัวติดตั้งไว้บนเพดานและตัวส่งอยู่บนพื้นห้อง และแสดงพิกัดตำแหน่งบนพื้นที่ของห้องทดลองขนาด 2x3.2 เมตร ความสูงของห้อง 2.8 เมตร และแสดงตำแหน่งบนคอมพิวเตอร์ด้วยโปรแกรมที่เขียนขึ้นจากวิชวลเบสิก

จากการทดลองนำเอาระบบที่ได้ออกแบบและสร้างขึ้นนี้ไปใช้งานจริงพบว่าระบบสามารถบอกตำแหน่งได้ค่อนข้างแม่นยำในระดับความคลาดเคลื่อนอยู่ระหว่าง 5-10 เซนติเมตร ดังนั้นถ้าพัฒนาด้านเทคนิคและปรับปรุงอุปกรณ์รับส่งสัญญาณให้ดีมากกว่านี้คาดว่าการทำงานของระบบในโครงงานนี้จะสามารถนำไปเป็นระบบนำทางภายในอาคาร และใช้ในพื้นที่กว้างมากกว่านี้ได้

5.2 ปัญหาที่พบ

เนื่องจากโครงงานนี้เป็นการศึกษาในเบื้องต้น และมีเวลาค่อนข้างจำกัดในการศึกษาพัฒนาด้านเทคนิคและอุปกรณ์ จึงทำให้โครงงานนี้มีปัญหาเกิดขึ้นอย่างมาก

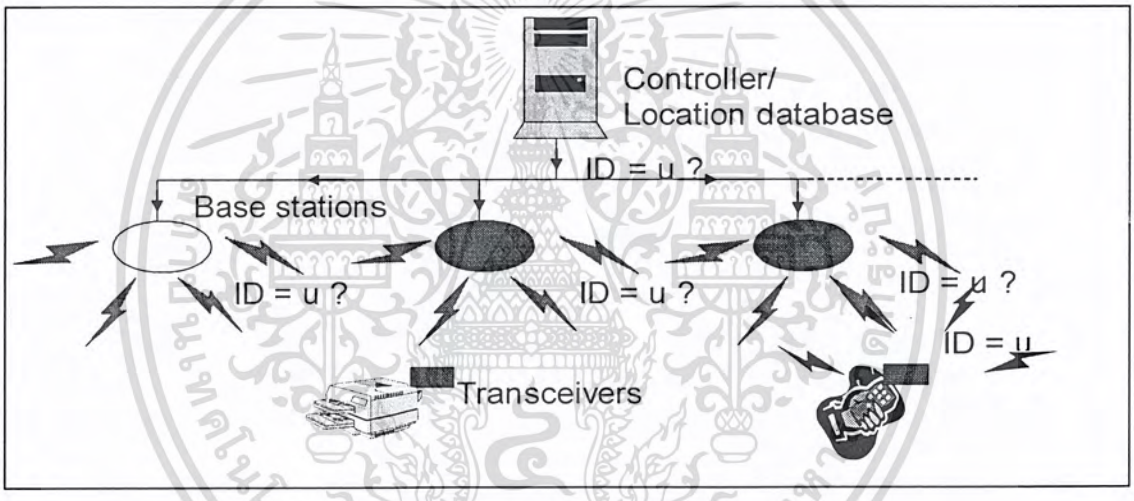
- ปัญหาด้านอุปกรณ์ตัวรับตัวส่ง ที่มีสัญญาณรบกวนค่อนข้างมากเนื่องจากความไม่เสถียรของวงจร ทำให้การทำงานหาพิกัดตำแหน่งของระบบเกิดความคลาดเคลื่อน
- การสื่อสารกันระหว่างไมโครคอนโทรลเลอร์ตัวรับกับคอมพิวเตอร์ ยังมีปัญหาค่อนข้างมากในด้านการรับข้อมูลของคอมพิวเตอร์ เนื่องจากเป็นการเชื่อมต่อกันกับไมโครคอนโทรลเลอร์หลายตัวทำให้เกิดปัญหาในด้านการรับส่งข้อมูลผิดพลาดขึ้นได้ ซึ่งต้องหาเทคนิคและวิธีการที่ดีกว่านี้
- ปัญหาด้านสภาพแวดล้อมของห้องทดลองซึ่งทำให้เกิดสัญญาณสะท้อนรบกวนการรับสัญญาณของตัวรับ ทำให้ตำแหน่งที่ได้เกิดความคลาดเคลื่อนอย่างมาก
- อุณหภูมิห้องทดลองต้องคงที่ไม่เช่นนั้นจะทำให้การหาตำแหน่งเกิดความคลาดเคลื่อนขึ้นได้
- การติดตั้งตัวรับทั้ง 5 ตัว จะต้องให้ระยะห่างของตัวรับแต่ละตัวแตกต่างกันพอสมควร ไม่ควรที่จะให้ระยะห่างเท่ากันเนื่องจากจะมีปัญหาในการคำนวณหาพิกัดตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

- อุปกรณ์ที่ทำขึ้นในโครงงานนี้ ทำขึ้นเป็นต้นแบบซึ่งการเชื่อมวงจรใช้สายไฟเชื่อมต่อด้วยการบัดกรี ถ้าทำเป็น PCB คาดว่าจะลดปัญหาของสัญญาณรบกวนได้
- การสื่อสารในระบบใช้ระดับสัญญาณมาตราฐาน RS-232C ซึ่งส่งไปได้ไกลสุดเพียง 15 เมตร และมีปัญหาค่อนข้างมากในการเชื่อมต่อกับไมโครคอนโทรลเลอร์หลายตัว ถ้าเปลี่ยนเป็นมาตราฐาน 485 คาดว่าจะส่งได้ไกลและลดปัญหาในส่วนของการเชื่อมต่อกับไมโครคอนโทรลเลอร์นี้ได้อย่างมาก

5.4 แนวทางในการพัฒนาในอนาคต



รูปที่ 5.1 รูปแบบการนำไปพัฒนาในอนาคต เป็นระบบควบคุมสื่อสารแบบไร้สายบนพื้นที่ภายในบ้านหรือที่ทำงาน

ระบบที่ออกแบบขึ้นมานี้ยังไม่สามารถที่นำไปใช้งานได้อย่างมีประสิทธิภาพนัก ต้องทำการปรับปรุงอุปกรณ์ตัวรับ-ตัวส่งและรูปแบบการติดต่อสื่อสารข้อมูล เพื่อให้สามารถใช้งานในพื้นที่กว้างขวางมากขึ้น แนวทางในการพัฒนาในอนาคตคือ สามารถนำไปพัฒนาเป็นระบบรักษาความปลอดภัยในบ้านหรือที่ทำงาน นำไปเป็นระบบนำทางของหุ่นยนต์อัตโนมัติ และสามารถพัฒนาไปเป็นระบบควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Reinhard Braunstingl. 1996. **Improving Reactive Navigation Using Perception Tracking.** Robotics And Manufacturing, Recent Trends in Research and Applications. (Vol.6): 25-30. New York.

[2] Jesús Savage, Fernando Lepe-Casillas, Yukihiro Minami, Carlos Rivera, Bohumil Psenicka. **Robot Localization Using Ultrasonic Beacons.** Facultad de Ingeniería DIE. Departamento de Eléctrica University of Mexico. Mexico.

[3] R.N. Aguilar, G.C.M. Meijer. **3-D Position Sensing Using The Difference in The Time-Of-Flights from The Use of Multiples Ultrasonic Sources and To One Receiver.** Electronic Instrumentation Laboratory. Delft University of Technology. Netherlands.

[4] Allen Ka Lun Miu. 2002. **Design and Implementation of an Indoor Mobile Navigation System.** Master of Science. Department of Electrical Engineering and Computer Science. Massachusetts Institue of Tecchnology.

[5] Changdon Kee, Doohee Yun, Haeyoung Jun, Bradford Parkinson, Sam Pullen, Tom Lagenstein. 2001. **Centimeter-Accuracy Indoor Navigation Using GPS-Like Pseudolites.** GPS World. (November 1).

[6] Ajay Mahajan, Fernando Figueroa. 1997. **Intelligent seam tracking using ultrasonic sensors for robotic welding.** Robotica. (volume15): 275-281. United Kingdom.

[7] Ajay Mahajan, Maurice Walworth. 1997. **3D Position Sensing Using the Differences in the Time-of-Flights From a Wave Source to Various Receivers.** United Kingdom.

[8] Probir Kumar Ray, Ajay Mahajan. **A Genetic Algorithm Based Approach to Calculate the Optimal Configuration of Ultrasonic Sensors in a 3D Position Estimation System.** Department of Mechanical Engineering and Energy Processes. Southern Illinois University. Carbondale.

[9] Probir Kumar Ray, Nishant Unnikrishanan, Ajay Mahajan. **Accuracy Considerations in a 3D Ultrasonic Positioning System based on the Difference in Time of Flights.** Department of Mechanical Engineering and Energy Processes. Southern Illinois University. Carbondale.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

[10] Johann Borenstein, David Wehe, Liqiang Feng, Yoram Koren. 1995. **Mobile Robot Navigation in Narrow Aisles with Ultrasonic Sensors.** ANS 6th Topical Meeting on Robotics and Remote Systems. Monterey. California. February 5-10.

[11] Shraga Shoval, Johann Borenstein. 2001. **Using coded Signals to Benefit from Ultrasonic Sensor Crosstalk in Mobile Robot Obstacle Avoidance.** IEEE International Conference on Robotics and Automation. (May 21-26): 2879-2884. Korea,

[12] P. Hoppenot, E. Colle and C. Barat. 2000. **Off line localisation of a mobile robot using ultrasonic measures.** Robotica. (Vol 18): 315-323. France.

[13] Andy Ward, Alan Jones, Andy Hopper. 1997. **A New Location Technique for the Active Office.** IEEE Personal Communications. (October). Cambridge. United Kingdom.

[14] Nissanka Priyantha, Allen Miu, Hari Balakrishnan, Seth Teller. **The Cricket Compass for Context-Aware Mobile Applications.** MIT Laboratory for Computer Science. (http://nms.lcs.mit.edu/)

[15] รศ.วิชิต ศิริโชติ. **การทดลอง MMT-1 (ภาษา C).** พิมพ์ครั้งที่ 1. บริษัท แอนนาดิจิท กรุ๊ป จำกัด. 2544.

[16] สุนทร วิทูสุรพจน์. **การใช้งานไมโครคอนโทรลเลอร์ ตระกูล 8051.** บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2537.

[17] วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฉบับ AT89C5x ของ Atmel.** บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[18] อรรถพล บุญยะโภคา, วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่าน พอร์ตอนุกรม.** บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[19] PIC Circuit Gallery. **Ultrasonic Range Meter.** (http://www.interq.or.jp/japan/se-inoue/e_pic6_6.htm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

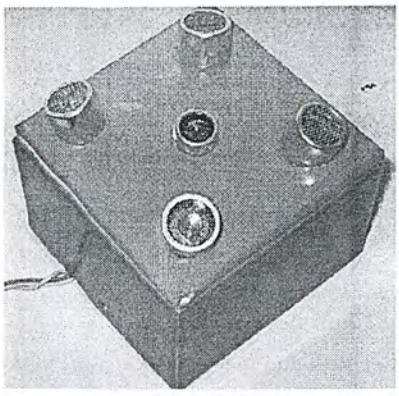
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

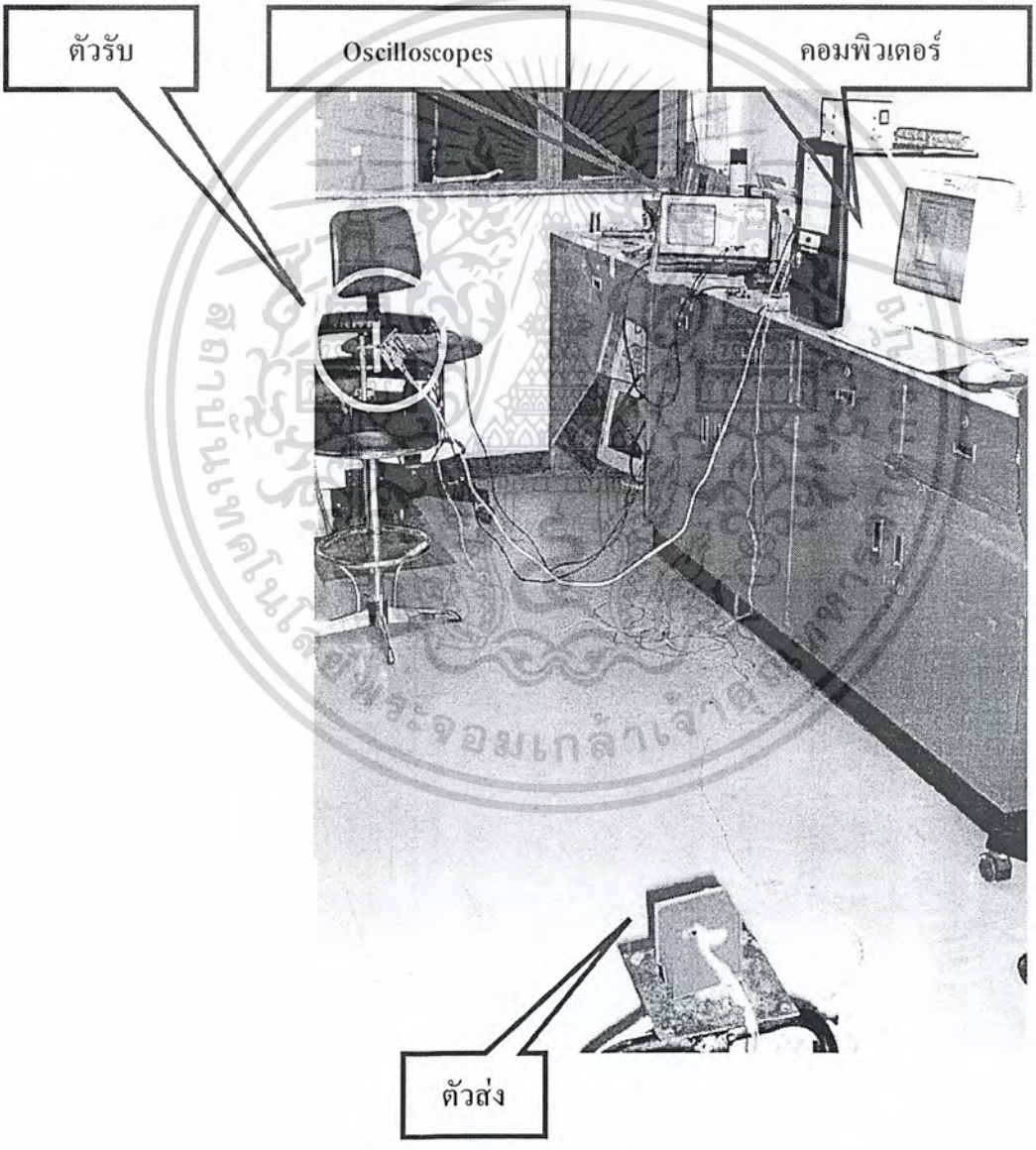
ภาคผนวก ก

- รูปแสดงการทดลอง และข้อมูลผลการทดลอง
- โปรแกรมไมโครคอนโทรลเลอร์ และวิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

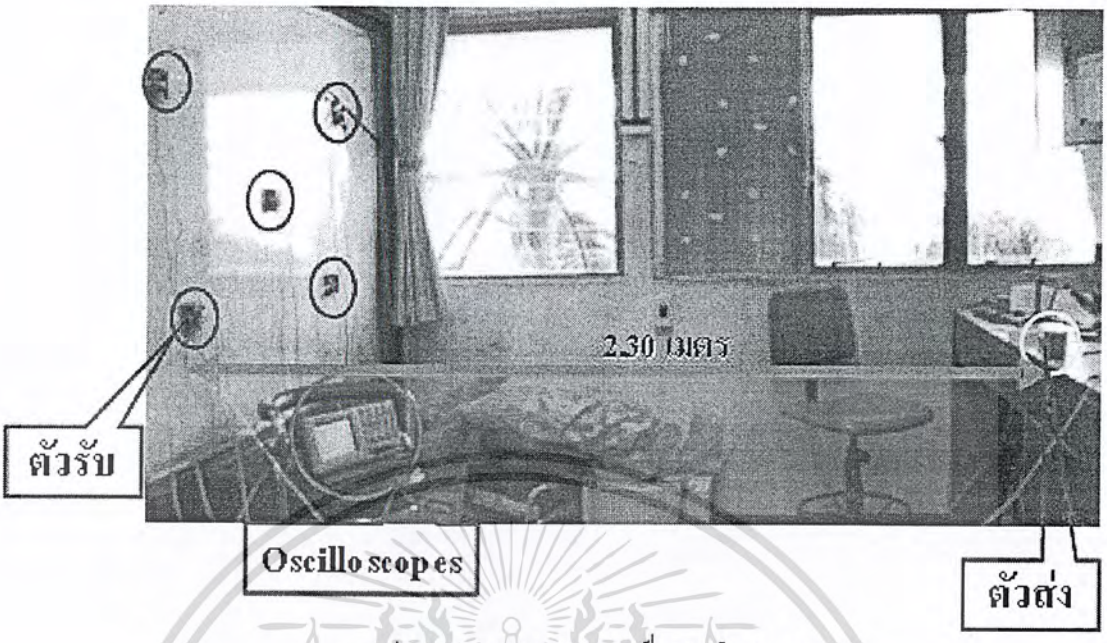


รูปแสดงตัวส่งคลื่นสัญญาณอัลตร้าโซนิค

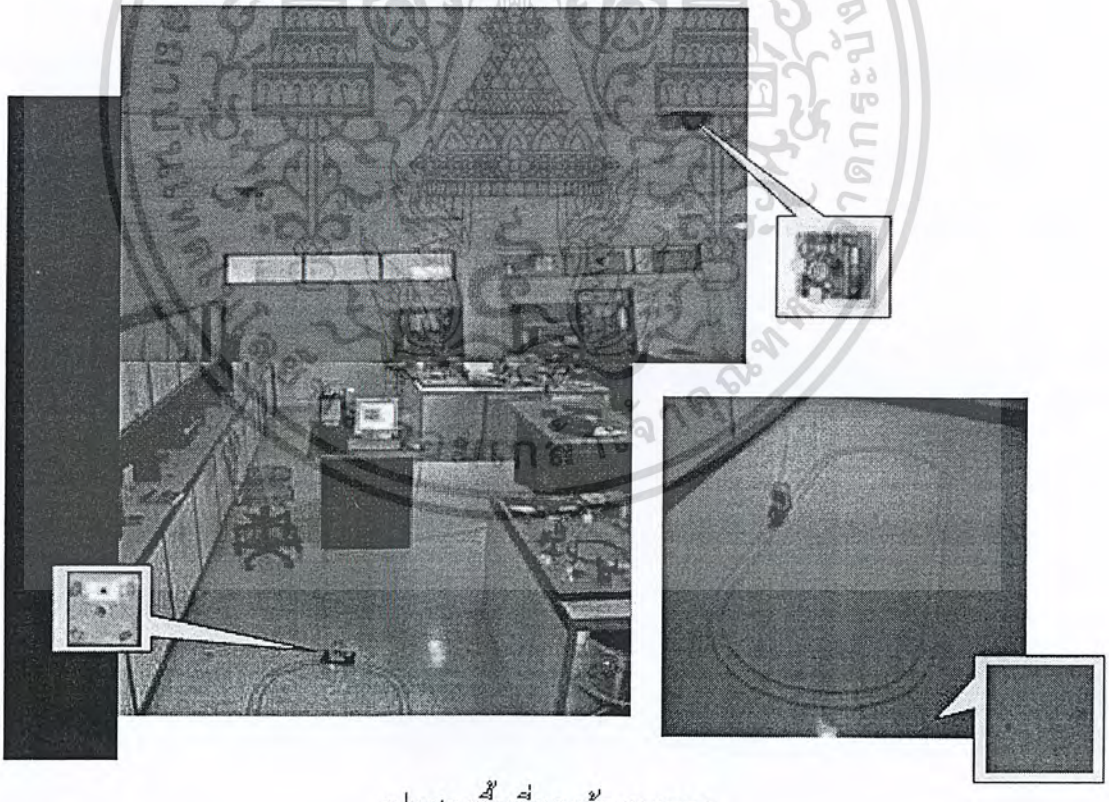


รูปแสดงการทดลองวัดระยะทางที่ค่าระยะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการทดลองการสื่อสารในระบบ



รูปแสดงพื้นที่ของห้องทดลอง

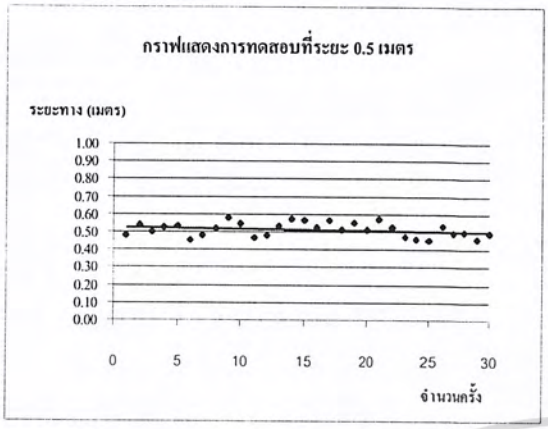
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองหาระยะทาง

	ระยะทาง (เมตร)									
	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
1	0.482	1.041	1.512	2.024	2.499	2.857	3.431	3.869	4.317	4.871
2	0.542	0.983	1.546	1.969	2.425	2.948	3.459	3.924	4.313	4.910
3	0.502	0.998	1.456	2.010	2.483	2.978	3.454	3.843	4.351	4.820
4	0.526	1.061	1.524	1.962	2.525	2.867	3.413	3.863	4.322	4.873
5	0.536	1.010	1.487	1.981	2.451	2.852	3.461	3.918	4.320	4.899
6	0.453	1.028	1.531	2.029	2.508	2.849	3.396	3.936	4.310	4.857
7	0.480	0.998	1.507	1.972	2.454	2.861	3.446	3.874	4.355	4.863
8	0.520	1.034	1.459	1.961	2.469	2.902	3.399	3.877	4.330	4.812
9	0.576	1.033	1.527	2.003	2.475	2.939	3.440	3.876	4.381	4.871
10	0.549	1.015	1.438	1.968	2.529	2.966	3.475	3.923	4.324	4.871
11	0.466	1.000	1.442	2.000	2.460	2.929	3.407	3.833	4.328	4.786
12	0.479	1.026	1.491	1.996	2.506	2.926	3.409	3.895	4.293	4.868
13	0.530	1.010	1.551	1.970	2.476	2.913	3.475	3.888	4.315	4.886
14	0.571	1.074	1.489	1.957	2.468	2.918	3.421	3.927	4.370	4.908
15	0.565	1.057	1.506	2.019	2.441	2.976	3.408	3.892	4.308	4.898
16	0.529	1.013	1.479	2.055	2.456	2.928	3.417	3.867	4.326	4.866
17	0.566	1.023	1.519	1.964	2.521	2.899	3.410	3.911	4.348	4.927
18	0.516	1.045	1.437	2.000	2.458	2.888	3.450	3.895	4.356	4.837
19	0.554	1.064	1.471	1.960	2.488	2.925	3.415	3.945	4.308	4.824
20	0.514	0.972	1.497	2.023	2.466	2.898	3.433	3.871	4.299	4.834
21	0.574	1.019	1.470	1.985	2.526	2.933	3.391	3.906	4.337	4.863
22	0.526	0.972	1.511	1.979	2.487	2.898	3.474	3.886	4.359	4.877
23	0.472	1.000	1.511	2.021	2.512	2.938	3.446	3.917	4.284	4.831
24	0.461	1.018	1.453	1.956	2.459	2.862	3.467	3.860	4.383	4.875
25	0.455	1.082	1.470	2.022	2.469	2.935	3.431	3.874	4.308	4.835
26	0.530	1.020	1.526	2.005	2.487	2.885	3.428	3.916	4.312	4.875
27	0.492	1.060	1.512	2.007	2.445	2.921	3.489	3.859	4.343	4.874
28	0.500	0.998	1.506	1.991	2.456	2.899	3.475	3.917	4.375	4.899
29	0.459	1.045	1.527	1.930	2.446	2.896	3.453	3.861	4.326	4.835
30	0.495	1.033	1.460	1.964	2.459	2.879	3.397	3.854	4.347	4.909

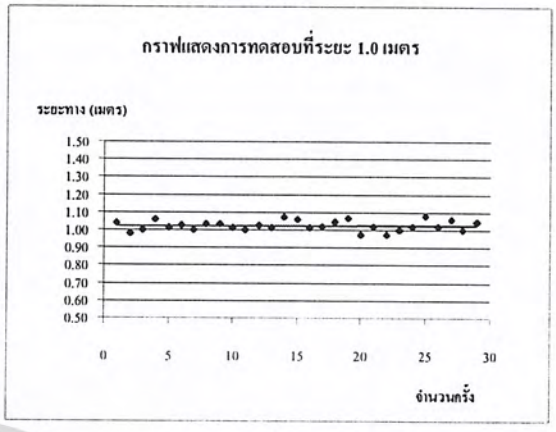
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงกราฟการทดสอบวัดระยะทาง



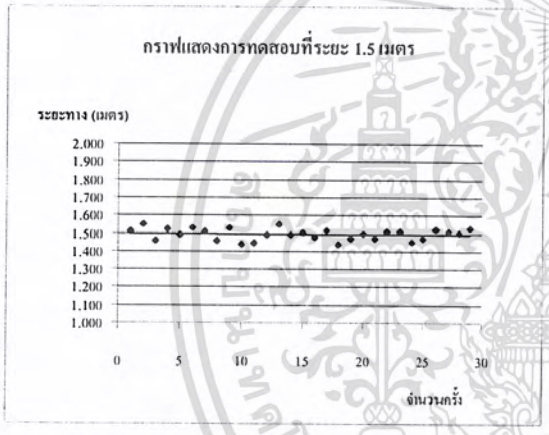
ระยะทางเฉลี่ย = 0.512 เมตร

ผลต่าง = 0.012 เมตร



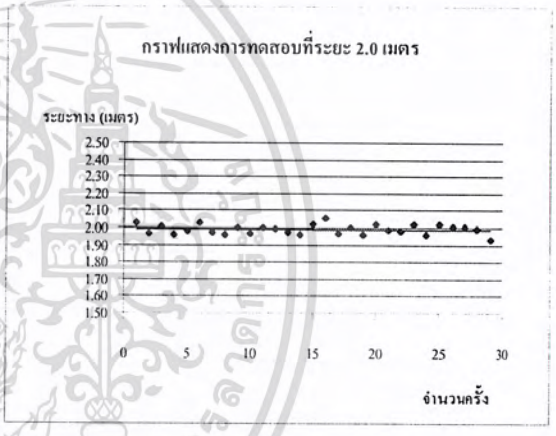
ระยะทางเฉลี่ย = 1.025 เมตร

ผลต่าง = 0.025 เมตร



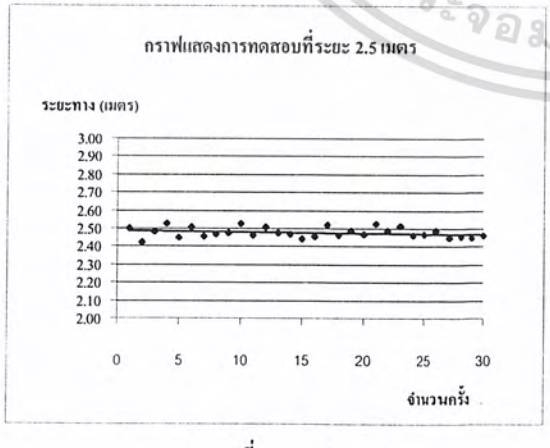
ระยะทางเฉลี่ย = 1.493 เมตร

ผลต่าง = 0.007 เมตร



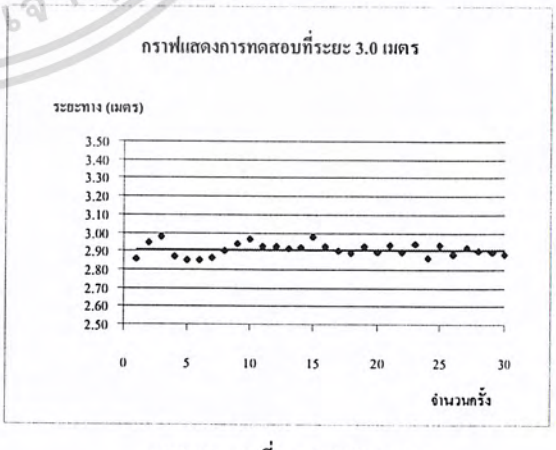
ระยะทางเฉลี่ย = 1.988 เมตร

ผลต่าง = 0.012 เมตร



ระยะทางเฉลี่ย = 2.476 เมตร

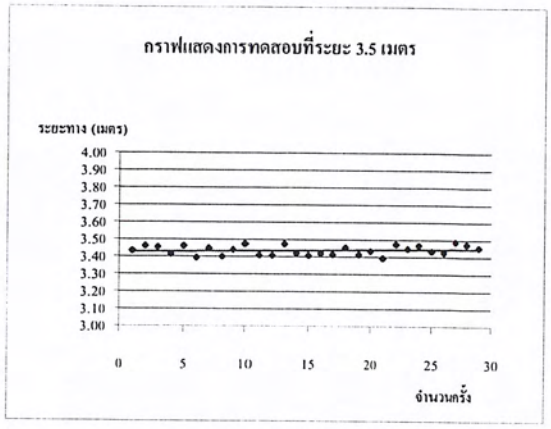
ผลต่าง = 0.024 เมตร



ระยะทางเฉลี่ย = 2.910 เมตร

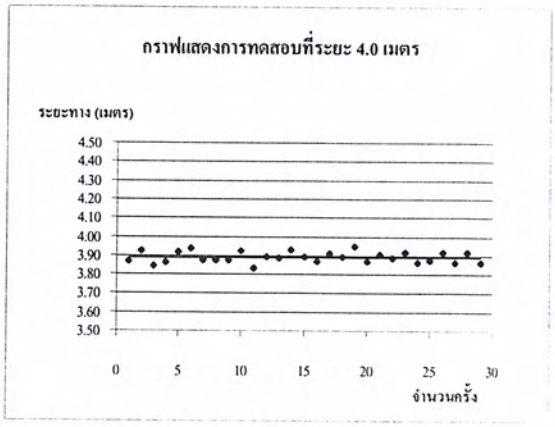
ผลต่าง = 0.090 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



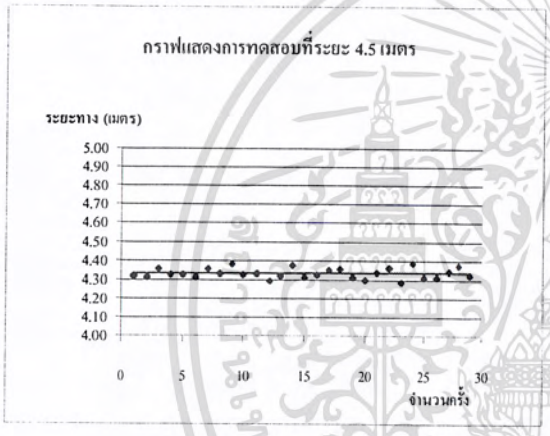
ระยะทางเฉลี่ย = 3.431 เมตร

ผลต่าง = 0.069 เมตร



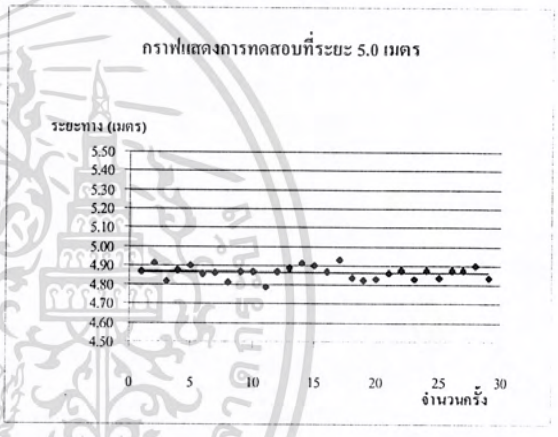
ระยะทางเฉลี่ย = 3.883 เมตร

ผลต่าง = 0.117 เมตร



ระยะทางเฉลี่ย = 4.334 เมตร

ผลต่าง = 0.166 เมตร



ระยะทางเฉลี่ย = 4.862 เมตร

ผลต่าง = 0.138 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมไมโครคอนโทรลเลอร์

```
#include <8051io.h>
#include <8051reg.h>
#include <8051int.h>
#include <8051bit.h>
#include <8051pin.h>
#define  LED P1.7
register  unsigned char key;
register  unsigned int  count;
INTERRUPT(_IE0_) external0_interrupt()
  {
        setbit(T2CON.2)  /* start timer2 */
        setbit(LED)
  }
INTERRUPT(_EXF2_) timer2_interrupt()
  {
   clrbit(T2CON.2)    /*stop timer2 */
        clrbit(T2CON.6)   /* enable EXF2*/
        count = RCAP2H;
        count <<= 8;
        count += RCAP2L;
  }
main()
  {
  serinit(9600);                                    // T2CON=0x09;
   setbit(T2CON.0)   /*select capture mode */
   clrbit(T2CON.1)   /*Internal timer(OSC/12) */
   setbit(T2CON.3)   /*enable EXEN2 */
   clrbit(T2CON.4)   /*TCLK = 0*/
   clrbit(T2CON.5)   /*RCLK = 0*/
        clrbit(T2CON.7)   /* RF2=0*/              //TCON= 0x01;
        clrbit(TCON.0)    /*falling edge*/        // IE = 0xA1;*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    setbit(IE.0)    /*set bit EX0 enable external0 interrupt */
    setbit(IE.5)    /*set bit ET2 enable external2 interrupt */
    setbit(IE.7)    /*EA enable interrup*/
    clrbit(LED)
          SendCount();
  }
SendCount()
          {
          while(1)
                {
                key = getchr();
        if (key =='1')
                {       clrbit(LED)
                   printf("%d",count);
                    count = 0;
                    TH2=0x00;
                    TL0=0x00;
                    RCAP2L = 0x00;
                    RCAP2H = 0x00;
                }
        }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมตัวส่งสัญญาณอัลตร้าโซนิค

```
$mod52
;-------------------------------------------------------------------------
; Define Port&Pin Name
;-------------------------------------------------------------------------
T0_PIN                  BIT             P3.4            ; Timer/Counter 0 PIN
;-------------------------------------------------------------------------
; Main Program.
;-------------------------------------------------------------------------
                ORG     0000H                   ; Reset Vector
                AJMP    MAIN                    ; Jump to Main

                ORG     000BH                   ; TF0 Vector
                MOV     TH0,#0FFH               ; 12.5 us Interval Time
                MOV     TL0,#0F1H               ; Reload Counter
                  CPL                   T0_PIN          ;
                RETI                            ; Return interrupt
MAIN:           ORL     TMOD,#10011001B ; T0 16Bit
                MOV     TH0,#0FFH               ; 12.5 us Interval Time
                MOV     TL0,#0F1H               ;
                  MOV           IE,#10000010B   ; En. EA,ET0
                  SETB          TR0                     ; Start Timer 0
                  AJMP          $                       ; Jump to Main loop
                  End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมวิชวลเบสิก

โปรแกรมหลัก (Form1)

```
Option Explicit
Dim X1 As Variant
Dim X2 As Variant
Dim X3 As Variant
Dim X4 As Variant
Dim X5 As Variant
Dim Y1 As Variant
Dim Y2 As Variant
Dim Y3 As Variant
Dim Y4 As Variant
Dim Y5 As Variant
Dim Xr1 As Variant
Dim Xr2 As Variant
Dim Xr3 As Variant
Dim Xr4 As Variant
Dim Xr5 As Variant
Dim Yr1 As Variant
Dim Yr2 As Variant
Dim Yr3 As Variant
Dim Yr4 As Variant
Dim Yr5 As Variant
Dim Xus As Variant
Dim Yus As Variant
Dim Xshp1 As Variant
Dim Yshp1 As Variant
Dim Dis1 As Variant
Dim Dis2 As Variant
Dim Dis3 As Variant
Dim Dis4 As Variant
Dim Dis5 As Variant
Dim D1 As Variant
Dim D2 As Variant
Dim D3 As Variant
Dim D4 As Variant
Dim D5 As Variant
Dim Ds1 As Variant
Dim Ds2 As Variant
Dim Ds3 As Variant
Dim Ds4 As Variant
Dim Ds5 As Variant
Dim Response As Variant
Dim Response2 As Variant
Dim T(5) As Variant
Dim D(5) As Variant
Dim X(5) As Variant
Dim Y(5) As Variant
Dim Dis(5) As Variant
Dim XDis(5) As Variant
Dim YDis(5) As Variant
Dim i As Variant
Dim DMin As Variant
Dim DisX As Variant
Dim DisY As Variant
Dim XP As Variant
Dim YP As Variant
Dim Xpos1 As Variant
Dim Xpos2 As Variant
Dim Xpos3 As Variant
Dim Xpos4 As Variant
Dim Xpos5 As Variant
Dim Ypos1 As Variant
Dim Ypos2 As Variant
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Ypos3 As Variant
Dim Ypos4 As Variant
Dim Ypos5 As Variant
Dim Xp1 As Variant
Dim Xp2 As Variant
Dim Xp3 As Variant
Dim Xp4 As Variant
Dim Xp5 As Variant
Dim Yp1 As Variant
Dim Yp2 As Variant
Dim Yp3 As Variant
Dim Yp4 As Variant
Dim Yp5 As Variant
Dim T1 As Variant
Dim T2 As Variant
Dim T3 As Variant
Dim T4 As Variant
Dim T5 As Variant
Dim T12 As Variant
Dim T13 As Variant
Dim T14 As Variant
Dim T15 As Variant
Dim a11 As Variant
Dim a12 As Variant
Dim a13 As Variant
Dim a14 As Variant
Dim a21 As Variant
Dim a22 As Variant
Dim a23 As Variant
Dim a24 As Variant
Dim a31 As Variant
Dim a32 As Variant
Dim a33 As Variant
Dim a34 As Variant
Dim a41 As Variant
Dim a42 As Variant
Dim a43 As Variant
Dim a44 As Variant
Dim ac11 As Variant
Dim ac12 As Variant
Dim ac13 As Variant
Dim ac14 As Variant
Dim ac21 As Variant
Dim ac22 As Variant
Dim ac23 As Variant
Dim ac24 As Variant
Dim ac31 As Variant
Dim ac32 As Variant
Dim ac33 As Variant
Dim ac34 As Variant
Dim ac41 As Variant
Dim ac42 As Variant
Dim ac43 As Variant
Dim ac44 As Variant
Dim b11 As Variant
Dim b21 As Variant
Dim b31 As Variant
Dim b41 As Variant
Dim detA As Variant
Dim acc11 As Variant
Dim acc12 As Variant
Dim acc13 As Variant
Dim acc14 As Variant
Dim acc21 As Variant
Dim acc22 As Variant
Dim acc23 As Variant
Dim acc24 As Variant
Dim acc31 As Variant
```

เอกสารนี้สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim acc32 As Variant
Dim acc33 As Variant
Dim acc34 As Variant
Dim acc41 As Variant
Dim acc42 As Variant
Dim acc43 As Variant
Dim acc44 As Variant
Dim Tt12 As Variant
Dim Tt13 As Variant
Dim Tt14 As Variant
Dim Tt15 As Variant
Dim U As Variant
Dim V As Variant
Dim cd As Variant
Dim c2 As Variant
Dim str As Variant
Dim Data1, File1 As String
Dim Count1 As Integer
Dim Xs As Variant
Dim Ys As Variant
Dim R(1) As Variant
Dim S(1) As Variant
Dim steps As Variant
Dim steps2 As Variant
Dim Px1 As Variant
Dim Px2 As Variant
Dim Py1 As Variant
Dim Py2 As Variant
Dim Vs As Variant
Dim Ca As Variant
Dim C As Variant
Dim MCT1 As Variant
Dim MCT2 As Variant
Dim MCT3 As Variant
Dim MCT4 As Variant
Dim MCT5 As Variant
Dim counter As Variant
Dim Xuser As Variant
Dim Xuser1 As Variant
Dim Xuser2 As Variant
Dim Yuser As Variant
Dim Yuser1 As Variant
Dim Yuser2 As Variant

Private Sub Form_Load()
   'สถานะยังไม่เริ่มทำงาน
    Call Open_comport
    ' lbl2.Caption = "Status : Comport already Open.
Please!! Open New File"
    Call DTRLow
    str = str & "Port :  " & Form2.cmb1.Text & vbCrLf
    str = str & "Baud Rate :  " & Form2.cmb2.Text &
" bits/s" & vbCrLf
    str = str & "Length :  " & Form2.cmb3.Text &
vbCrLf
    str = str & "Parity :  " & Form2.cmb4.Text &
vbCrLf
    str = str & "Stop Bit :  " & Form2.cmb5.Text &
vbCrLf
   str = str & vbCrLf
   str = str & "T =  " & Form2.txtTemp.Text & " C " &
vbCrLf
   str = str & vbCrLf
   str = str & "Timer2:  " & Form2.txtTime2.Text &
vbCrLf
   txt1.Text = str
   lblPX1.Caption = Form2.txtX1.Text
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lblPX2.Caption = Form2.txtX2.Text
    lblPX3.Caption = Form2.txtX3.Text
    lblPX4.Caption = Form2.txtX4.Text
    lblPX5.Caption = Form2.txtX5.Text
    lblPY1.Caption = Form2.txtY1.Text
    lblPY2.Caption = Form2.txtY2.Text
    lblPY3.Caption = Form2.txtY3.Text
    lblPY4.Caption = Form2.txtY4.Text
    lblPY5.Caption = Form2.txtY5.Text
    lblOn1.Visible = True
    lblOn2.Visible = True
    lblOn3.Visible = True
    lblOn4.Visible = True
    lblOn5.Visible = True
    Call Show_Receiver
    shpUser.Visible = False
    mnuOpen.Enabled = False
    cmdOpen.Enabled = False
    CmdStart.Enabled = False
    ' เพิ่มเติมในส่วน list  file
    Data1 = "Count1" & Chr(9) & " X " & _
    Chr(9) & " Y " & Chr(9) & "Velocity" &
Chr(9) & "Times"
    List1.AddItem Data1
    Lbl2.Caption = "Status : Comport
already open.Please!! Open New
File,before click START"
    End Sub
```

'โปรแกรมย่อมเริ่มการทำงาน

```
Private Sub CmdStart_Click()
    Timer2.Enabled = Not Timer2.Enabled
      If Timer2.Enabled = True Then
        CmdStart.Caption = "PAUSE"
        Lbl2.Caption = "Status :  Running !!!"
        Else
        CmdStart.Caption = "CONTINUE"
        Lbl2.Caption = "Status :  Pause!!!"
      End If
End Sub
Private Sub CmdStop_Click()
    Close #1
    List1.Clear
    File1 = ""
    Data1 = "Count1" & Chr(9) & " X " & _
    Chr(9) & " Y " & Chr(9) & "Velocity" & Chr(9) &
"Times"
    List1.AddItem Data1
    CmdStart.Caption = "START"
    CmdStart.Enabled = False
    Timer2.Enabled = False
    If MSComm1.DTREnable = False Then
    MSComm1.DTREnable = True
    End If
    Timer2.Enabled = False
    shpUser.Visible = False
    mnuNew.Enabled = True
    cmdSave.Enabled = True
    mnuOpen.Enabled = True
    cmdOpen.Enabled = True
    Lbl2.Caption = "Status : Stop Running"
End Sub
Private Sub HScroll1_Change()

    Timer2.Interval = HScroll1.Value
    txtShowTime.Text = HScroll1.Value
End Sub
Private Sub mnuNew_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  If Not (File1 = " ") And
(MSComm1.PortOpen = False) Then
    Response = MsgBox("Please! Click Stop",
vbOKOnly, "Error!")
    Else
    On Error Resume Next
    Dlg1.CancelError = True
    Dlg1.Flags = cdlOFNHideReadOnly
    Dlg1.Filter = "All Files (*.*)|*.*|Microsoft
Excel Files (*.xls)|*.xls"
    Dlg1.FilterIndex = 2
    Dlg1.Action = 2
    If Err = 32755 Then Exit Sub
    File1 = Dlg1.FileName
    End If
    Open File1 For Output As #1
    mnuNew.Enabled = False
    cmdSave.Enabled = False
    mnuOpen.Enabled = False
    cmdOpen.Enabled = False
    List1.Clear
    Data1 = "Count1" & Chr(9) & " X " & _
    Chr(9) & " Y " & Chr(9) & "Velocity" &
Chr(9) & "Times"
    List1.AddItem Data1
End Sub
Private Sub mnuOpen_Click()
Dim CLine As String
Dim Filenum As Integer
    If Not (File1 = " ") And
(MSComm1.PortOpen = False) Then
    Response = MsgBox("Please! Click Stop",
vbOKOnly, "Error!")
    Else
    On Error Resume Next
    Dlg1.Flags = cdlOFNHideReadOnly
    Dlg1.CancelError = True
    Dlg1.FileName = ""
    Dlg1.Filter = "All Files (*.*)|*.*|Microsoft
Excel Files (*.xls)|*.xls"
    Dlg1.FilterIndex = 2
    Dlg1.Action = 1
    If Err = 32755 Then Exit Sub
    Form4.Text1.Text = ""
    Filenum = FreeFile
    Open Dlg1.FileName For Input As #Filenum
    Form4.Text1.Text = Input$(LOF(Filenum),
#Filenum)
    Form1.Caption = Dlg1.FileName
    Form4.Show vbModal
    End If
End Sub
Private Sub Timer1_Timer()
 Lbl3.Caption = Format(Date, "Long date")
 Lbl4.Caption = Time()
End Sub
Private Sub Timer2_Timer()  'ส่วนโปรแกรมย่อยควบ
คุมการส่งและรับค่า
MSComm1.InBufferCount = 0
counter = counter + 1
If counter = 1 Then
MSComm1.DTREnable = False
ElseIf counter = 2 Then
MSComm1.Output = "5"
ElseIf counter = 3 Then
 If MSComm1.InBufferCount > 0 Then
 MCT1 = MSComm1.Input
 MSComm1.InBufferCount = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  End If
ElseIf counter = 4 Then
MSComm1.Output = "2"
ElseIf counter = 5 Then
If MSComm1.InBufferCount > 0 Then
 MCT2 = MSComm1.Input
 MSComm1.InBufferCount = 0
 End If
ElseIf counter = 6 Then
MSComm1.Output = "1"
ElseIf counter = 7 Then
If MSComm1.InBufferCount > 0 Then
 MCT3 = MSComm1.Input
 MSComm1.InBufferCount = 0
 End If
ElseIf counter = 8 Then
MSComm1.Output = "3"
ElseIf counter = 9 Then
If MSComm1.InBufferCount > 0 Then
MCT4 = MSComm1.Input
 MSComm1.InBufferCount = 0
 End If
 ElseIf counter = 10 Then
MSComm1.Output = "4"
 ElseIf counter = 11 Then
If MSComm1.InBufferCount > 0 Then
 MCT5 = MSComm1.Input
 MSComm1.InBufferCount = 0
 End If
 ElseIf counter = 12 Then
     Call DisMin
     Call Matrix
     Call Show_User
  Data1 = Count1 & Chr(9) & Xs & _
    Chr(9) & Ys & Chr(9) & Round(Vs, 3) & Chr(9) &
Format(Time, "h:mm:ss")
   List1.AddItem Data1
   List1.ListIndex = List1.ListCount - 1
   Count1 = Count1 + 1
   Print #1, List1.List(List1.ListCount - 1)
 ElseIf counter = 14 Then
 counter = 0
 MSComm1.DTREnable = True
 End If
End Sub
Private Sub Open_comport()
 If Not MSComm1.PortOpen Then
  MSComm1.PortOpen = True
  Else
  MsgBox ("Port Is Use By Another Program"), ,
"ComPort Error"
 End If
 End Sub
Private Sub DisMin()
    D(1) = D1
    D(2) = D2
    D(3) = D3
    D(4) = D4
    D(5) = D5
    X(1) = lblPX1.Caption
    X(2) = lblPX2.Caption
    X(3) = lblPX3.Caption
    X(4) = lblPX4.Caption
    X(5) = lblPX5.Caption
    Y(1) = lblPY1.Caption
    Y(2) = lblPY2.Caption
    Y(3) = lblPY3.Caption
    Y(4) = lblPY4.Caption
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Y(5) = lblPY5.Caption
    T(1) = MCT1
    T(2) = MCT2
    T(3) = MCT3
    T(4) = MCT4
    T(5) = MCT5
    For i = 1 To 5
      Dis(i) = D(i)
      XDis(i) = X(i)
      YDis(i) = Y(i)
    Next i
    DMin = Dis(1)
    DisX = XDis(1)
    DisY = YDis(1)
    i = 1
    Do
      If Dis(i) < DMin Then
        DMin = Dis(i)
        DisX = XDis(i)
        DisY = YDis(i)
      End If
      i = i + 1
    Loop While i <= 5
End Sub
Private Sub Matrix()
    XP = DisX
    YP = DisY
    Xpos1 = lblPX1.Caption
    Xpos2 = lblPX2.Caption
    Xpos3 = lblPX3.Caption
    Xpos4 = lblPX4.Caption
    Xpos5 = lblPX5.Caption
    Ypos1 = lblPY1.Caption
    Ypos2 = lblPY2.Caption
    Ypos3 = lblPY3.Caption
    Ypos4 = lblPY4.Caption
    Ypos5 = lblPY5.Caption
    Ca = txtC.Text
    C = Val(Ca)
If XP - Xpos1 = 0 Then
    Xp2 = Xpos2
    Xp3 = Xpos3
    Xp4 = Xpos4
    Xp5 = Xpos5
    Yp2 = Ypos2
    Yp3 = Ypos3
    Yp4 = Ypos4
    Yp5 = Ypos5
    T1 = MCT1 * (C) / 1000000
    T2 = MCT2 * (C) / 1000000
    T3 = MCT3 * (C) / 1000000
    T4 = MCT4 * (C) / 1000000
    T5 = MCT5 * (C) / 1000000
    T12 = T2 - T1
    T13 = T3 - T1
    T14 = T4 - T1
    T15 = T5 - T1
    Call Calculate
ElseIf XP - Xpos2 = 0 Then
    Xp2 = Xpos1
    Xp3 = Xpos3
    Xp4 = Xpos4
    Xp5 = Xpos5
    Yp2 = Ypos1
    Yp3 = Ypos3
    Yp4 = Ypos4
    Yp5 = Ypos5
    T1 = MCT2 * (C) / 1000000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T2 = MCT1 * (C) / 1000000
T3 = MCT3 * (C) / 1000000
T4 = MCT4 * (C) / 1000000
T5 = MCT5 * (C) / 1000000
  T12 = T2 - T1
  T13 = T3 - T1
  T14 = T4 - T1
  T15 = T5 - T1
  Call Calculate
ElseIf XP - Xpos3 = 0 Then
  Xp2 = Xpos1
  Xp3 = Xpos2
  Xp4 = Xpos4
  Xp5 = Xpos5
  Yp2 = Ypos1
  Yp3 = Ypos2
  Yp4 = Ypos4
  Yp5 = Ypos5
T1 = MCT3 * (C) / 1000000
T2 = MCT1 * (C) / 1000000
T3 = MCT2 * (C) / 1000000
T4 = MCT4 * (C) / 1000000
T5 = MCT5 * (C) / 100000
  T12 = T2 - T1
  T13 = T3 - T1
  T14 = T4 - T1
  T15 = T5 - T1
  Call Calculate
ElseIf XP - Xpos4 = 0 Then
  Xp2 = Xpos1
  Xp3 = Xpos2
  Xp4 = Xpos3
  Xp5 = Xpos5
  Yp2 = Ypos1
  Yp3 = Ypos2
  Yp4 = Ypos3
  Yp5 = Ypos5
T1 = MCT4 * (C) / 1000000
T2 = MCT1 * (C) / 1000000
T3 = MCT2 * (C) / 1000000
T4 = MCT3 * (C) / 1000000
T5 = MCT5 * (C) / 1000000
  T12 = T2 - T1
  T13 = T3 - T1
  T14 = T4 - T1
  T15 = T5 - T1
  Call Calculate
ElseIf XP - Xpos5 = 0 Then
  Xp2 = Xpos1
  Xp3 = Xpos2
  Xp4 = Xpos3
  Xp5 = Xpos4
  Yp2 = Ypos1
  Yp3 = Ypos2
  Yp4 = Ypos3
  Yp5 = Ypos4
T1 = MCT5 * (C) / 1000000
T2 = MCT1 * (C) / 1000000
T3 = MCT2 * (C) / 1000000
T4 = MCT3 * (C) / 1000000
T5 = MCT4 * (C) / 1000000
  T12 = T2 - T1
  T13 = T3 - T1
  T14 = T4 - T1
  T15 = T5 - T1
  Call Calculate
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Calculate()
    a11 = 2 * (XP - Xp2)
    a12 = 2 * (YP - Yp2)
    a13 = -(2 * T12)
    a14 = -(2 * T12 ^ 2)
    a21 = 2 * (XP - Xp3)
    a22 = 2 * (YP - Yp3)
    a23 = -(2 * T13)
    a24 = -(2 * T13 ^ 2)
    a31 = 2 * (XP - Xp4)
    a32 = 2 * (YP - Yp4)
    a33 = -(2 * T14)
    a34 = -(2 * T14 ^ 2)
    a41 = 2 * (XP - Xp5)
    a42 = 2 * (YP - Yp5)
    a43 = -(2 * T15)
    a44 = -(2 * T15 ^ 2)
    b11 = XP ^ 2 + YP ^ 2 - Xp2 ^ 2 - Yp2 ^ 2
    b21 = XP ^ 2 + YP ^ 2 - Xp3 ^ 2 - Yp3 ^ 2
    b31 = XP ^ 2 + YP ^ 2 - Xp4 ^ 2 - Yp4 ^ 2
    b41 = XP ^ 2 + YP ^ 2 - Xp5 ^ 2 - Yp5 ^ 2
    ac11 = ((a22 * ((a33 * a44) - (a43 * a34))) -
(a23 * ((a32 * a44) - (a42 * a34))) + (a24 *
((a32 * a43) - (a42 * a33))))
    ac12 = -((a21 * ((a33 * a44) - (a43 * a34)))
- (a23 * ((a31 * a44) - (a41 * a34))) + (a24 *
((a31 * a43) - (a41 * a33))))
    ac13 = ((a21 * ((a32 * a44) - (a42 * a34))) -
(a22 * ((a31 * a44) - (a41 * a34))) + (a24 *
((a31 * a42) - (a41 * a32))))
    ac14 = -((a21 * ((a32 * a43) - (a42 * a33)))
- (a22 * ((a31 * a43) - (a41 * a33))) + (a23 *
((a31 * a42) - (a41 * a32))))
    ac21 = -((a12 * ((a33 * a44) - (a43 * a34))) - (a13 *
((a32 * a44) - (a42 * a34))) + (a14 * ((a32 * a43) - (a42
* a33))))
    ac22 = ((a11 * ((a33 * a44) - (a43 * a34))) - (a13 *
((a31 * a44) - (a41 * a34))) + (a14 * ((a31 * a43) - (a41
* a33))))
    ac23 = -((a11 * ((a32 * a44) - (a42 * a34))) - (a12 *
((a31 * a44) - (a41 * a34))) + (a14 * ((a31 * a42) - (a41
* a32))))
    ac24 = ((a11 * ((a32 * a43) - (a42 * a33))) - (a12 *
((a31 * a43) - (a41 * a33))) + (a13 * ((a31 * a42) - (a41
* a32))))
    ac31 = ((a12 * ((a23 * a44) - (a43 * a24))) - (a13 *
((a22 * a44) - (a42 * a24))) + (a14 * ((a22 * a43) - (a42
* a23))))
    ac32 = -((a11 * ((a23 * a44) - (a43 * a24))) - (a13 *
((a21 * a44) - (a41 * a24))) + (a14 * ((a21 * a43) - (a41
* a23))))
    ac33 = ((a11 * ((a22 * a44) - (a42 * a24))) - (a12 *
((a21 * a44) - (a41 * a24))) + (a14 * ((a21 * a42) - (a41
* a22))))
    ac34 = -((a11 * ((a22 * a43) - (a42 * a23))) - (a12 *
((a21 * a43) - (a41 * a23))) + (a13 * ((a21 * a42) - (a41
* a22))))
    ac41 = -((a12 * ((a23 * a34) - (a33 * a24))) - (a13 *
((a22 * a34) - (a32 * a24))) + (a14 * ((a22 * a33) - (a32
* a23))))
    ac42 = ((a11 * ((a23 * a34) - (a33 * a24))) - (a13 *
((a21 * a34) - (a31 * a24))) + (a14 * ((a21 * a33) - (a31
* a23))))
    ac43 = -((a11 * ((a22 * a34) - (a32 * a24))) - (a12 *
((a21 * a34) - (a31 * a24))) + (a14 * ((a21 * a32) - (a31
* a22))))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ac44 = ((a11 * ((a22 * a33) - (a32 * a23))) -
(a12 * ((a21 * a33) - (a31 * a23))) + (a13 *
((a21 * a32) - (a31 * a22))))
    detA = (a11 * ((a22 * ((a33 * a44) - (a43 *
a34))) - (a23 * ((a32 * a44) - (a42 * a34))) +
(a24 * ((a32 * a43) - (a42 * a33))))) - (a12 *
((a21 * ((a33 * a44) - (a43 * a34))) - (a23 *
((a31 * a44) - (a41 * a34))) + (a24 * ((a31 *
a43) - (a41 * a33))))) + (a13 * ((a21 * ((a32 *
a44) - (a42 * a34))) - (a22 * ((a31 * a44) -
(a41 * a34))) + (a24 * ((a31 * a42) - (a41 *
a32))))) - (a14 * ((a21 * ((a32 * a43) - (a42 *
a33))) - (a22 * ((a31 * a43) - (a41 * a33))) +
(a23 * ((a31 * a42) - (a41 * a32)))))
    acc11 = ac11 / detA
    acc12 = ac12 / detA
    acc13 = ac13 / detA
    acc14 = ac14 / detA
    acc21 = ac21 / detA
    acc22 = ac22 / detA
    acc23 = ac23 / detA
    acc24 = ac24 / detA
    acc31 = ac31 / detA
    acc32 = ac32 / detA
    acc33 = ac33 / detA
    acc34 = ac34 / detA
    acc41 = ac41 / detA
    acc42 = ac42 / detA
    acc43 = ac43 / detA
    acc44 = ac44 / detA
    U = (b11 * acc11) + (b21 * acc21) + (b31 *
acc31) + (b41 * acc41)
    V = (b11 * acc12) + (b21 * acc22) + (b31 *
acc32) + (b41 * acc42)
    cd = (b11 * acc13) + (b21 * acc23) + (b31 * acc33) +
(b41 * acc43)
    c2 = (b11 * acc14) + (b21 * acc24) + (b31 * acc34) +
(b41 * acc44)
    If c2 >= 0 Then
    Xs = Round(U, 3)
    Ys = Round(V, 3)
    lblX.Caption = Round(U, 3)
    lblY.Caption = Round(V, 3)
    Vs = Sqr(c2) / 100
    Call User_Position
    Else
    c2 = Abs(c2)
    End If
End Sub
Private Sub R1_Position()
    Xr1 = shp1.Left
    Yr1 = shp1.Top
    X1 = (Xr1 / 480) * 25
    Y1 = (12 - (Yr1 / 484.84)) * 25
End Sub
Private Sub R2_Position()
    Xr2 = shp2.Left
    Yr2 = shp2.Top
    X2 = (Xr2 / 480) * 25
    Y2 = (12 - (Yr2 / 484.84)) * 25
End Sub
Private Sub R3_Position()
    Xr3 = shp3.Left
    Yr3 = shp3.Top
    X3 = (Xr3 / 480) * 25
    Y3 = (12 - (Yr3 / 484.84)) * 25
End Sub
Private Sub R4_Position()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Xr4 = shp4.Left
    Yr4 = shp4.Top
    X4 = (Xr4 / 480) * 25
    Y4 = (12 - (Yr4 / 484.84)) * 25
End Sub
Private Sub R5_Position()
    Xr5 = shp5.Left
    Yr5 = shp5.Top
    X5 = (Xr5 / 480) * 25
    Y5 = (12 - (Yr5 / 484.84)) * 25
End Sub
Private Sub User_Position()
  Xuser2 = lblX.Caption
  Yuser2 = lblY.Caption
  Xuser1 = (Val(Xuser2) * 5) / 100
  Yuser1 = (13 - (Val(Yuser2) * 6) / 100)
  Xuser = (480 * (Xuser1)) - 240
  Yuser = (484.84 * (Yuser1 - 1)) - 242.42
  shpUser.Visible = True
  shpUser.Move Xuser, Yuser
End Sub
Private Sub R1_User()
    Dis1 = ((Xus - X1) ^ 2 + (Yus - Y1) ^ 2)
    Ds1 = Sqr(Dis1)
    D1 = Sqr(350 ^ 2 + Ds1 ^ 2)
    lblDis1.Caption = Round(D1, 3)
End Sub
Private Sub R2_User()
    Dis2 = ((Xus - X2) ^ 2 + (Yus - Y2) ^ 2)
    Ds2 = Sqr(Dis2)
    D2 = Sqr(350 ^ 2 + Ds2 ^ 2)
    lblDis2.Caption = Round(D2, 3)
End Sub
Private Sub R3_User()
    Dis3 = ((Xus - X3) ^ 2 + (Yus - Y3) ^ 2)
    Ds3 = Sqr(Dis3)
    D3 = Sqr(350 ^ 2 + Ds3 ^ 2)
    lblDis3.Caption = Round(D3, 3)
End Sub
Private Sub R4_User()
    Dis4 = ((Xus - X4) ^ 2 + (Yus - Y4) ^ 2)
    Ds4 = Sqr(Dis4)
    D4 = Sqr(350 ^ 2 + Ds4 ^ 2)
    lblDis4.Caption = Round(D4, 3)
End Sub
Private Sub R5_User()
    Dis5 = ((Xus - X5) ^ 2 + (Yus - Y5) ^ 2)
    Ds5 = Sqr(Dis5)
    D5 = Sqr(350 ^ 2 + Ds5 ^ 2)
    lblDis5.Caption = Round(D5, 3)
End Sub
Private Sub DTRHigh()
 Form1.MSComm1.DTREnable = False
 End Sub
Private Sub DTRLow()
 Form1.MSComm1.DTREnable = True
 End Sub
Private Sub Reset()
Form1.MSComm1.RTSEnable = True
Delay
Form1.MSComm1.RTSEnable = False
 DTRLow
End Sub
Private Sub Delay()
Dim a As Single
a = Timer + 0.0001
Do While Timer < a
Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub Show_Shp1()
Dim Xl1 As Variant
Dim Yl1 As Variant
Dim Xv1 As Variant
Dim Yv1 As Variant
Dim X1 As Variant
Dim Y1 As Variant
   Xl1 = Form2.txtX1.Text
   Yl1 = Form2.txtY1.Text
   Xv1 = (Val(Xl1) * 5) / 100
   Yv1 = (13 - (Val(Yl1) * 6) / 100)
   X1 = (480 * (Xv1)) - 240
   Y1 = (484.84 * (Yv1 - 1)) - 242.42
   shp1.Move X1, Y1
End Sub
Private Sub Show_Shp2()
Dim Xl2 As Variant
Dim Yl2 As Variant
Dim Xv2 As Variant
Dim Yv2 As Variant
Dim X2 As Variant
Dim Y2 As Variant
   Xl2 = Form2.txtX2.Text
   Yl2 = Form2.txtY2.Text
   Xv2 = (Val(Xl2) * 5) / 100
   Yv2 = (13 - (Val(Yl2) * 6) / 100)
   X2 = (480 * (Xv2)) - 240
   Y2 = (484.84 * (Yv2 - 1)) - 242.42
   shp2.Move X2, Y2
End Sub
Private Sub Show_Shp3()
Dim Xl3 As Variant
Dim Yl3 As Variant
Dim Xv3 As Variant
Dim Yv3 As Variant
Dim X3 As Variant
Dim Y3 As Variant
   Xl3 = Form2.txtX3.Text
   Yl3 = Form2.txtY3.Text
   Xv3 = (Val(Xl3) * 5) / 100
   Yv3 = (13 - (Val(Yl3) * 6) / 100)
   X3 = (480 * (Xv3)) - 240
   Y3 = (484.84 * (Yv3 - 1)) - 242.42
   shp3.Move X3, Y3
End Sub
Private Sub Show_Shp4()
Dim Xl4 As Variant
Dim Yl4 As Variant
Dim Xv4 As Variant
Dim Yv4 As Variant
Dim X4 As Variant
Dim Y4 As Variant
   Xl4 = Form2.txtX4.Text
   Yl4 = Form2.txtY4.Text
   Xv4 = (Val(Xl4) * 5) / 100
   Yv4 = (13 - (Val(Yl4) * 6) / 100)
   X4 = (480 * (Xv4)) - 240
   Y4 = (484.84 * (Yv4 - 1)) - 242.42
   shp4.Move X4, Y4
End Sub
Private Sub Show_Shp5()
Dim Xl5 As Variant
Dim Yl5 As Variant
Dim Xv5 As Variant
Dim Yv5 As Variant
Dim X5 As Variant
Dim Y5 As Variant
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Xl5 = Form2.txtX5.Text
    Yl5 = Form2.txtY5.Text
    Xv5 = (Val(Xl5) * 5) / 100
    Yv5 = (13 - (Val(Yl5) * 6) / 100)
    X5 = (480 * (Xv5)) - 240
    Y5 = (484.84 * (Yv5 - 1)) - 242.42
    shp5.Move X5, Y5
End Sub
```

โปรแกรมย่อย Single Step

```
**********
Option Explicit
Dim Data1 As Variant
Dim Count1 As Variant
Dim X As Variant
Dim Y As Variant
Dim Result As Variant
Dim Distance As Variant
Dim Choose As Variant
Dim Diff As Variant
Dim File2 As Variant
Dim Response As Variant
Dim Response2 As Variant
Dim Response3 As Variant
Dim add As Variant
Dim Data As Variant
Private Sub cmdOpen_Click()
   Dlg1.Flags = cdlOFNHideReadOnly
   Dlg1.Filter = "All Files (*.*)|*.*|Text Files
(*.txt)|*.txt"
   Dlg1.FilterIndex = 2
   Dlg1.ShowOpen
   Close #2
End Sub
Private Sub CmdReset_Click()
   Close #2
   File2 = ""
   opt1.Value = False
   shp4.FillColor = &HFFFFFF
   opt2.Value = False
   shp5.FillColor = &HFFFFFF
   opt3.Value = False
   shp6.FillColor = &HFFFFFF
   opt4.Value = False
   shp7.FillColor = &HFFFFFF
   opt5.Value = False
   shp8.FillColor = &HFFFFFF
   shp9.FillColor = &H80FF&
   shp2.FillColor = &HFFFFFF
   shp3.FillColor = &HFFFFFF
   txtTime.Text = ""
   txtSub.Text = ""
   txtCal.Text = ""
   Call RTSHigh
   Delay
   Call RTSLow
End Sub
Private Sub cmdSave_Click()
   Dlg1.Flags = cdlOFNHideReadOnly
   Dlg1.Filter = "All Files (*.*)|*.*|Text Files
(*.txt)|*.txt"
   Dlg1.FilterIndex = 2
   Dlg1.ShowSave
   File2 = Dlg1.FileName
   If File2 = "" Then
   Exit Sub
   End If
   Open File2 For Output As #2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub cmdSendUs_Click()
   shp2.FillColor = &H80FF&
   shp3.FillColor = &HFFFFFF
   shp9.FillColor = &HFFFFFF
   Call DTRHigh 'ส่งสัญญาณ ir
End Sub

Private Sub Calculate()
If txtTime.Text = "" Then
   Response3 = MsgBox("Check Comport
or Beacon not connect", vbOKOnly,
"Error!")
Else
   shp3.FillColor = &H80FF&
   Data = txtTime.Text
   Y = txtSoundV.Text
   Result = (Data * Y) / 1000000
   txtCal.Text = Result
   Diff = txtDistance.Text - Result
  txtSub.Text = Diff
   Data1 = Count1 & Chr(9) & Round
(Data, 3) & _
   Chr(9) & Round(Result, 3) & Chr(9) &
Round(Diff, 3) & Chr(9) & Format(Time,
"h:mm:ss")
   List1.AddItem Data1
   List1.ListIndex = List1.ListCount - 1
   Count1 = Count1 + 1
   Print #2, List1.List(List1.ListCount - 1)
End If
End Sub
Private Sub cmdSendRd_Click()
Dim Chosen As Variant
   shp2.FillColor = &HFFFFFF
   shp9.FillColor = &HFFFFFF
   Call DTRLow
   If File2 = "" Then
   Response = MsgBox("Please! Save File, first",
vbOKOnly, "Error!")
   ElseIf (opt1.Value = False) And (opt2.Value =
False) And (opt3.Value = False) And (opt4.Value =
False) And (opt5.Value = False) Then
   Response2 = MsgBox("Please! chick select
Beacon", vbOKOnly, "Error!")
   Else
   Select Case Chosen
     Case Chosen = opt1.Value
      txtTime.Text = "0"
      txtSub.Text = "0"
      add = "1"
      shp3.FillColor = &H80FF&
      Call Scan_data
      Call Calculate
    Case Chosen = opt2.Value
      txtTime.Text = "0"
      txtSub.Text = "0"
      add = "2"
      shp3.FillColor = &H80FF&
      Call Scan_data
      Call Calculate
   Case Chosen = opt3.Value
      txtTime.Text = "0"
       txtSub.Text = "0"
      add = "3"
      shp3.FillColor = &H80FF&
      Call Scan_data
      Call Calculate
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Case Chosen = opt4.Value
            txtTime.Text = "0"
             txtSub.Text = "0"
            add = "4"
            shp3.FillColor = &H80FF&
            Call Scan_data
            Call Calculate
        Case Chosen = opt5.Value
            txtTime.Text = "0"
             txtSub.Text = "0"
            add = "5"
            shp3.FillColor = &H80FF&
            Call Scan_data
            Call Calculate
    End Select
    End If
End Sub
Private Sub Form_Load()
    Form1.MSComm1.InputMode =
comInputModeText
    Data1 = "Count" & Chr(9) & " TOF " & _
    Chr(9) & " Calculate " & Chr(9) & "
Diferrent" & Chr(9) & "Times"
    List1.AddItem Data1
    shp1.FillColor = &H80FF&
End Sub
Private Sub opt1_Click()
    If (opt2.Value = False) And (opt3.Value
= False) And (opt4.Value = False) And
(opt5.Value = False) Then
    shp5.FillColor = &HFFFFFF
    shp6.FillColor = &HFFFFFF
    shp7.FillColor = &HFFFFFF
    shp8.FillColor = &HFFFFFF
    shp9.FillColor = &HFFFFFF
    End If
    shp4.FillColor = &H80FF&
End Sub
Private Sub opt2_Click()
    If (opt1.Value = False) And (opt3.Value = False)
And (opt4.Value = False) And (opt5.Value = False)
Then
    shp4.FillColor = &HFFFFFF
    shp6.FillColor = &HFFFFFF
    shp7.FillColor = &HFFFFFF
    shp8.FillColor = &HFFFFFF
    shp9.FillColor = &HFFFFFF
    End If
    shp5.FillColor = &H80FF&
End Sub
Private Sub opt3_Click()
    If (opt1.Value = False) And (opt2.Value = False)
And (opt4.Value = False) And (opt5.Value = False)
Then
    shp4.FillColor = &HFFFFFF
    shp5.FillColor = &HFFFFFF
    shp7.FillColor = &HFFFFFF
    shp8.FillColor = &HFFFFFF
    shp9.FillColor = &HFFFFFF
    End If
    shp6.FillColor = &H80FF&
End Sub
Private Sub opt4_Click()
    If (opt1.Value = False) And (opt2.Value = False)
And (opt3.Value = False) And (opt5.Value = False)
Then
    shp4.FillColor = &HFFFFFF
    shp5.FillColor = &HFFFFFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    shp6.FillColor = &HFFFFFF
    shp8.FillColor = &HFFFFFF
    shp9.FillColor = &HFFFFFF
    End If
    shp7.FillColor = &H80FF&
End Sub
Private Sub opt5_Click()
    If (opt1.Value = False) And (opt2.Value
= False) And (opt3.Value = False) And
(opt4.Value = False) Then
    shp4.FillColor = &HFFFFFF
    shp5.FillColor = &HFFFFFF
    shp6.FillColor = &HFFFFFF
    shp7.FillColor = &HFFFFFF
    shp9.FillColor = &HFFFFFF
    End If
    shp8.FillColor = &H80FF&
End Sub
Private Sub Scan_data()
    Form1.MSComm1.InputLen = 0
    Form1.MSComm1.Output = add
    Form1.MSComm1.OutBufferCount = 0
    Delay
    If Form1.MSComm1.InBufferCount
Then
    Data = Form1.MSComm1.Input
    txtTime.Text = Data
    Form1.MSComm1.InBufferCount = 0
    End If
End Sub
Private Sub Timer1_Timer()
    lblDate.Caption = Format(Date, "Long
Date")
    lblTime.Caption = Time()
End Sub
Private Sub DTRHigh()  'DTR เป็น 1 ส่งสัญญาณ IR
และเริ่มจับเวลาการเดินทางของอัลตร้าโซนิค
Form1.MSComm1.DTREnable = True
End Sub
Private Sub DTRLow() 'DTR เป็น 0
Form1.MSComm1.DTREnable = False
End Sub
Private Sub RTSHigh()
Form1.MSComm1.RTSEnable = False
End Sub
Private Sub RTSLow()
Form1.MSComm1.RTSEnable = True
End Sub
Private Sub Delay()
Dim a As Single
 'A = 100
  a = Timer + 0.01
  Do While Timer < a
    'DoEvents
  Loop
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

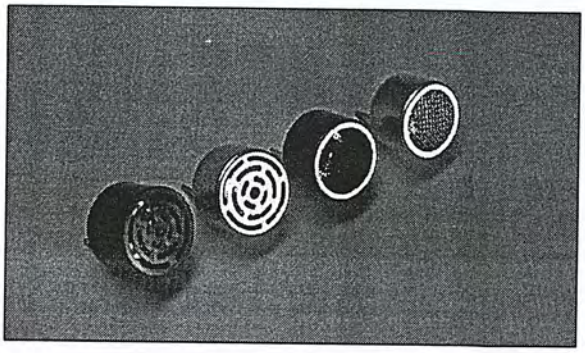
ภาคผนวก ข

ข้อมูลอุปกรณ์

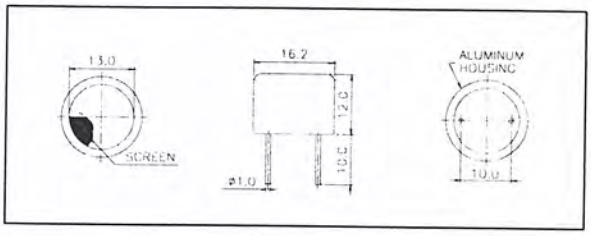
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Air Ultrasonic Ceramic Transducers

400ST/R160



Dimensions: dimensions are in mm



Impedance/Phase Angle vs. Frequency

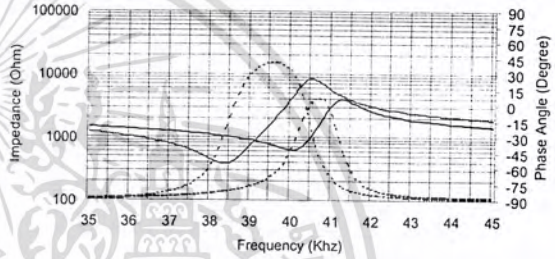
Tested under 1Vrms Oscillation Level

400SR160 Impedance ——————

400SR160 Phase ——————

400ST160 Impedance ·····

400ST160 Phase ·····

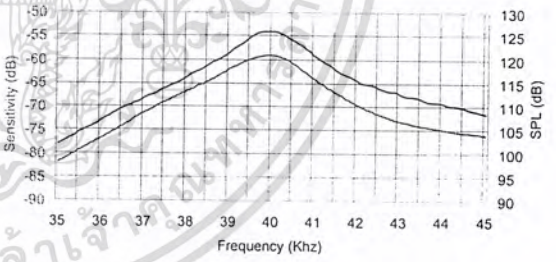


Specification

400ST160		Transmitter
400SR160		Receiver
Center Frequency		40.0±1.0Khz
Bandwidth (-6dB)	400ST160	2.0Khz
	400SR160	2.5Khz
Transmitting Sound Pressure Level at 40.0Khz; 0dB re 0.0002μbar per 10Vrms at 30cm		120dB min.
Receiving Sensitivity at 40.0Khz 0dB = 1 volt/μbar		-65dB min.
Capacitance at 1Khz	±20%	2400 pF
Max. Driving Voltage (cont.)		20Vrms
Total Beam Angle	-6dB	55° typical
Operation Temperature		-40 to 85°C
Storage Temperature		-40 to 85°C

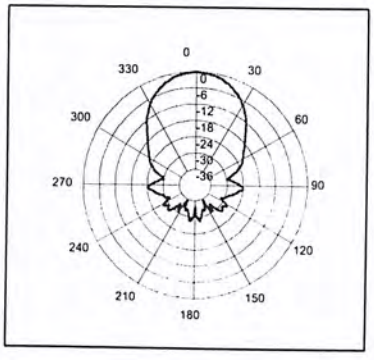
Sensitivity/Sound Pressure Level

Tested under 10Vrms @30cm



All specification taken typical at 25°C
Closer frequency tolerance can be supplied upon request.

Beam Angle: Tested at 40.0Khz frequency



Models available:

1	400ST/R160	Aluminum Housing
2	400ST/R16B	Black Al. Housing
2	400ST/R10P	Plastic Housing
3	400ST/R16F	Al. Housing w/Solid Grid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปเผยแพร่ต่อ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

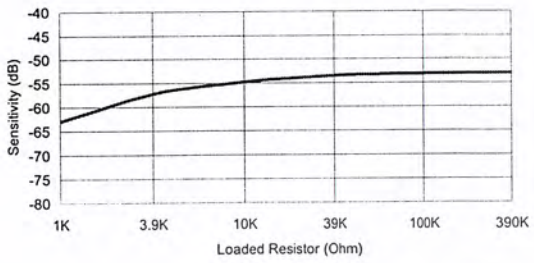
Products List | Home

Air Ultrasonic Ceramic Transducers

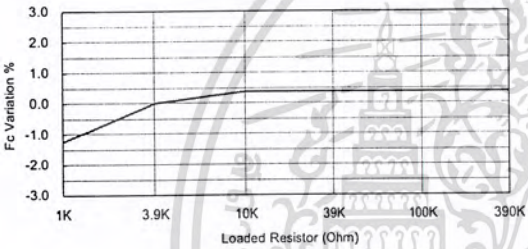
400ST/R160

400SR160 Receiver

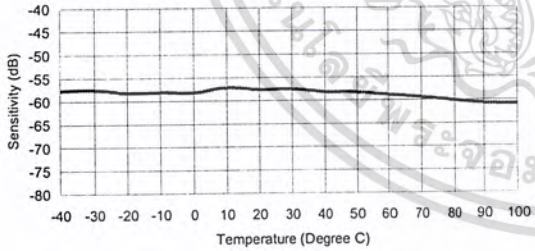
Sensitivity Variation vs. Loaded Resistor



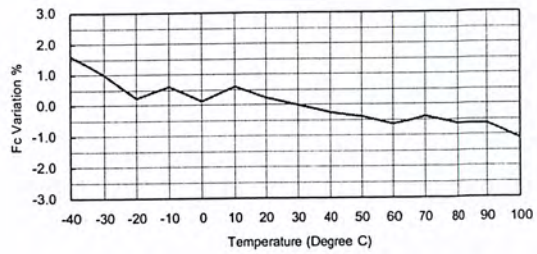
Center Frequency Shift vs. Loaded Resistor



Sensitivity Variation vs. Temperature

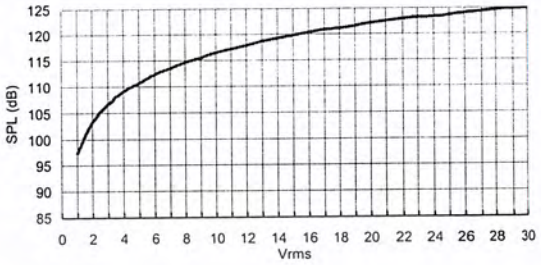


Center Frequency Shift vs. Temperature

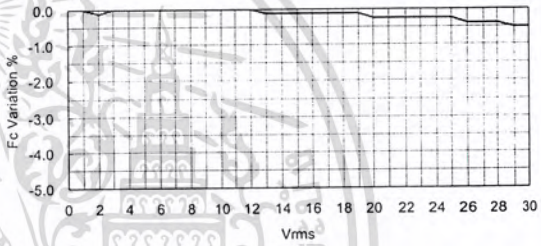


400ST160 Transmitter

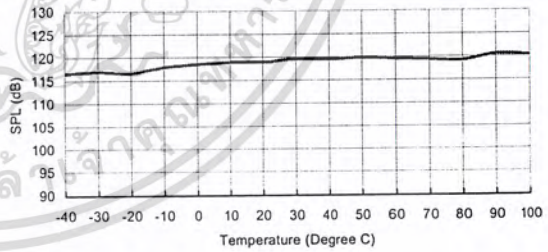
SPL Variation vs. Driving Voltage



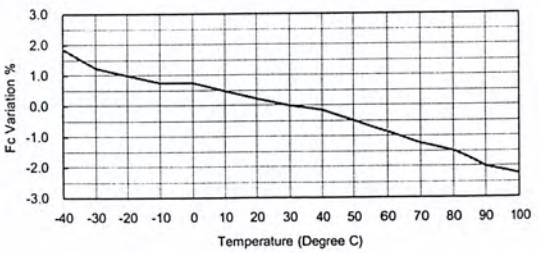
Center Frequency Shift vs. Driving Voltage



SPL Variation vs. Temperature



Center Frequency Shift vs. Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Products List Home

National Semiconductor

August 2000

TL082
Wide Bandwidth Dual JFET Input Operational Amplifier

General Description

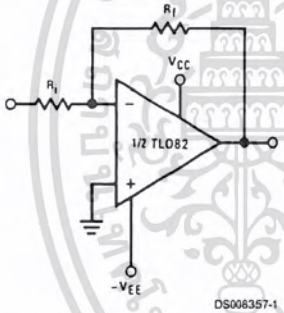
These devices are low cost, high speed, dual JFET input operational amplifiers with an internally trimmed input offset voltage (BI-FET II™ technology). They require low supply current yet maintain a large gain bandwidth product and fast slew rate. In addition, well matched high voltage JFET input devices provide very low input bias and offset currents. The TL082 is pin compatible with the standard LM1558 allowing designers to immediately upgrade the overall performance of existing LM1558 and most LM358 designs.

These amplifiers may be used in applications such as high speed integrators, fast D/A converters, sample and hold circuits and many other circuits requiring low input offset voltage, low input bias current, high input impedance, high slew rate and wide bandwidth. The devices also exhibit low noise and offset voltage drift.

Features

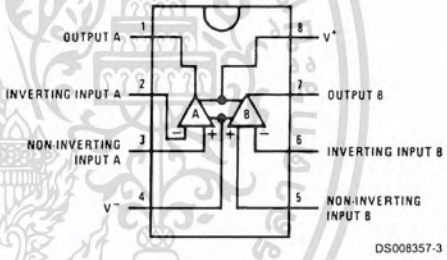
■ Internally trimmed offset voltage:	15 mV
■ Low input bias current:	50 pA
■ Low input noise voltage:	16nV/√Hz
■ Low input noise current:	0.01 pA/√Hz
■ Wide gain bandwidth:	4 MHz
■ High slew rate:	13 V/μs
■ Low supply current:	3.6 mA
■ High input impedance:	$10^{12}\Omega$
■ Low total harmonic distortion:	≤0.02%
■ Low 1/f noise corner:	50 Hz
■ Fast settling time to 0.01%:	2 μs

Typical Connection



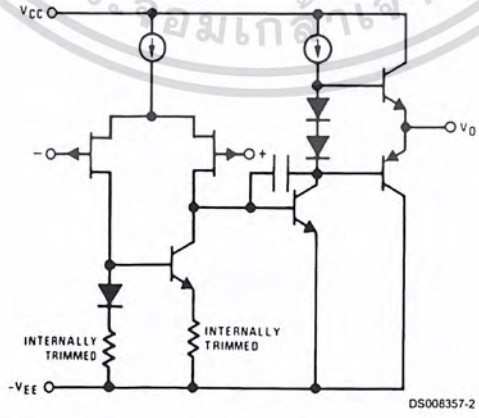
Connection Diagram

DIP/SO Package (Top View)



Order Number TL082CM or TL082CP
See NS Package Number M08A or N08E

Simplified Schematic



BI-FET II™ is a trademark of National Semiconductor Corp.

© 2000 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	±18V
Power Dissipation	(Note 2)
Operating Temperature Range	0°C to +70°C
$T_{j(MAX)}$	150°C
Differential Input Voltage	±30V
Input Voltage Range (Note 3)	±15V
Output Short Circuit Duration	Continuous
Storage Temperature Range	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	260°C

ESD rating to be determined.

Note 1: "Absolute Maximum Ratings" indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits.

DC Electrical Characteristics (Note 5)

Symbol	Parameter	Conditions	TL082C			Units
			Min	Typ	Max	
V_{OS}	Input Offset Voltage	R_S = 10 kΩ, T_A = 25°C		5	15	mV
		Over Temperature			20	mV
$\Delta V_{OS}/\Delta T$	Average TC of Input Offset Voltage	R_S = 10 kΩ		10		µV/°C
I_{OS}	Input Offset Current	T_j = 25°C, (Notes 5, 6)		25	200	pA
		$T_j \leq$ 70°C			4	nA
I_B	Input Bias Current	T_j = 25°C, (Notes 5, 6)		50	400	pA
		$T_j \leq$ 70°C			8	nA
R_{IN}	Input Resistance	T_j = 25°C		10^{12}		Ω
A_{VOL}	Large Signal Voltage Gain	V_S = ±15V, T_A = 25°C	25	100		V/mV
		V_O = ±10V, R_L = 2 kΩ				
		Over Temperature	15			V/mV
V_O	Output Voltage Swing	V_S = ±15V, R_L = 10 kΩ	±12	±13.5		V
V_{CM}	Input Common-Mode Voltage Range	V_S = ±15V	±11	+15		V
				−12		V
CMRR	Common-Mode Rejection Ratio	$R_S \leq$ 10 kΩ	70	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 7)	70	100		dB
I_S	Supply Current			3.6	5.6	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Note 5)

Symbol	Parameter	Conditions	TL082C			Units
			Min	Typ	Max	
	Amplifier to Amplifier Coupling	T_A = 25˚C, f = 1Hz-20 kHz (Input Referred)		−120		dB
SR	Slew Rate	V_S = ±15V, T_A = 25˚C	8	13		V/µs
GBW	Gain Bandwidth Product	V_S = ±15V, T_A = 25˚C		4		MHz
e_n	Equivalent Input Noise Voltage	T_A = 25˚C, R_S = 100Ω, f = 1000 Hz		25		nV/√Hz
i_n	Equivalent Input Noise Current	T_j = 25˚C, f = 1000 Hz		0.01		pA/√Hz
THD	Total Harmonic Distortion	A_V = +10, R_L = 10k, V_O = 20 Vp − p, BW = 20 Hz−20 kHz		<0.02		%

Note 2: For operating at elevated temperature, the device must be derated based on a thermal resistance of 115˚C/W junction to ambient for the N package.

Note 3: Unless otherwise specified the absolute maximum negative input voltage is equal to the negative power supply voltage.

Note 4: The power dissipation limit, however, cannot be exceeded.

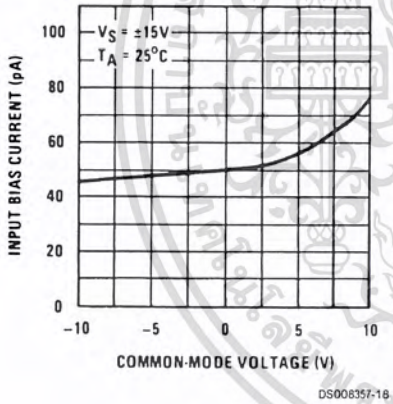
Note 5: These specifications apply for V_S = ±15V and 0˚C ≤T_A ≤ +70˚C. V_{OS}, I_B and I_{OS} are measured at V_{CM} = 0.

Note 6: The input bias currents are junction leakage currents which approximately double for every 10˚C increase in the junction temperature, T_j. Due to the limited production test time, the input bias currents measured are correlated to junction temperature. In normal operation the junction temperature rises above the ambient temperature as a result of internal power dissipation, P_D. $T_j = T_A + \theta_{jA} P_D$ where θ_{jA} is the thermal resistance from junction to ambient. Use of a heat sink is recommended if input bias current is to be kept to a minimum.

Note 7: Supply voltage rejection ratio is measured for both supply magnitudes increasing or decreasing simultaneously in accordance with common practice. V_S = ±6V to ±15V.

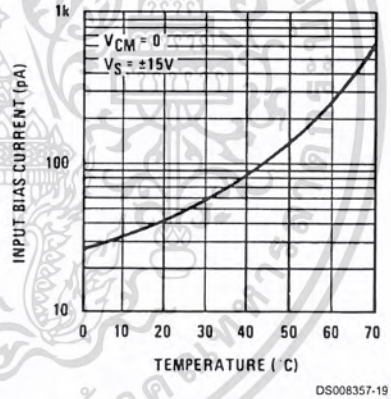
Typical Performance Characteristics

Input Bias Current



DS008357-16

Input Bias Current



DS008357-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

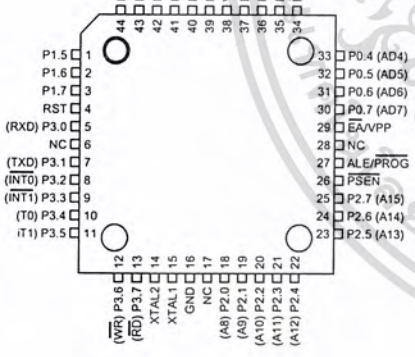
Description

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

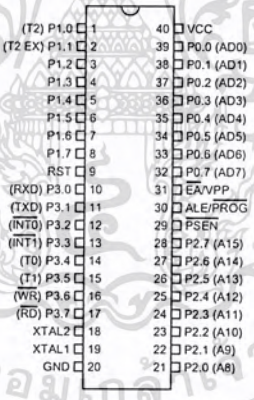
(continued)

Pin Configurations

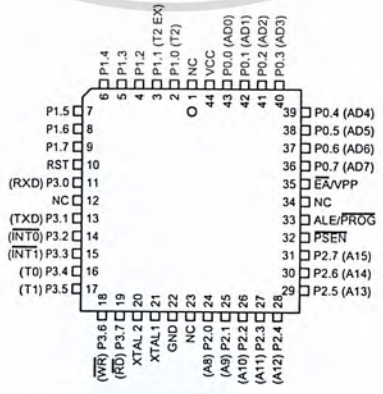
PQFP/TQFP



PDIP



PLCC



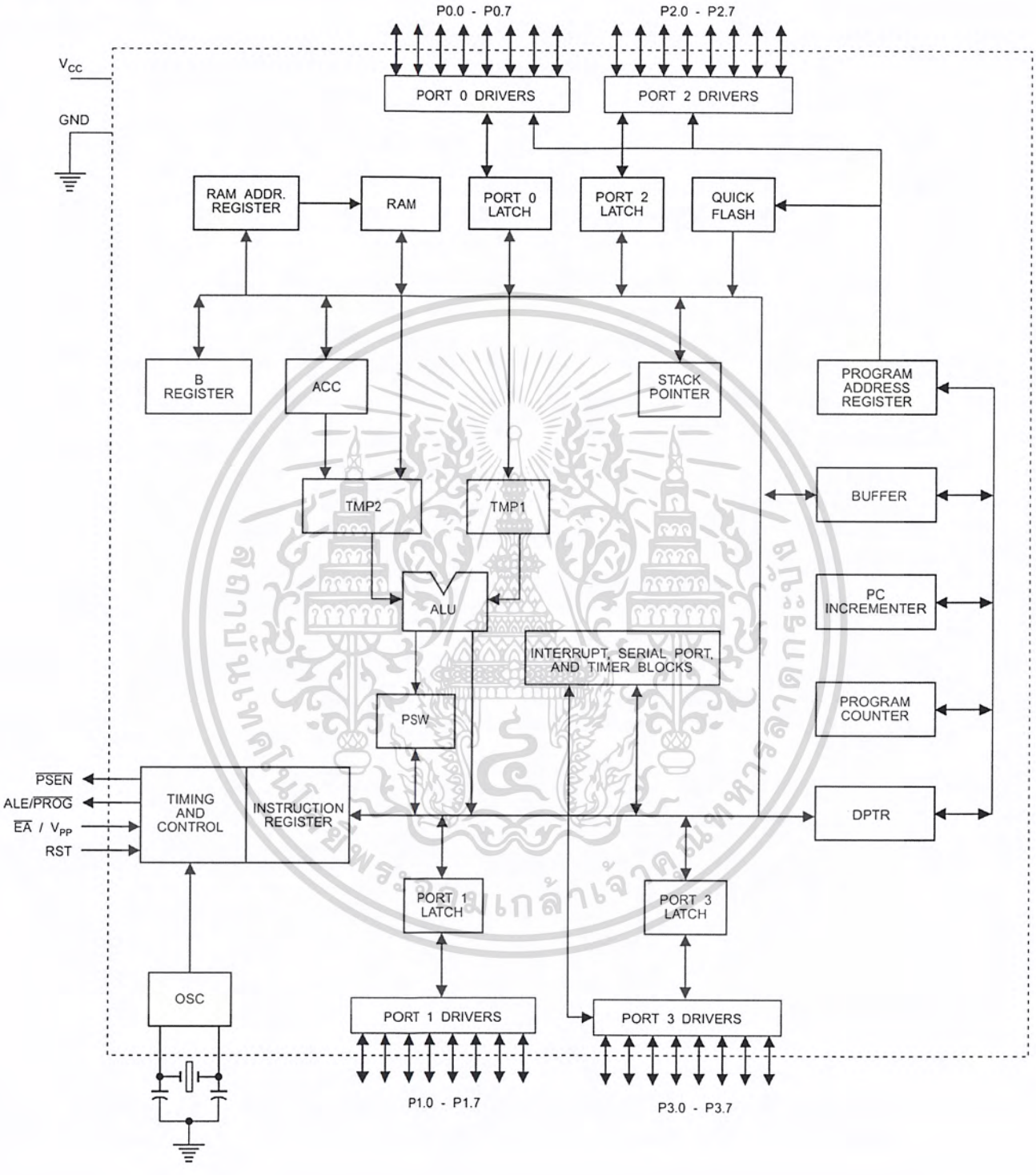
8-bit Microcontroller with 8K Bytes Flash

AT89C52

Rev. 0313G–04/99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.
Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$
Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/V_{PP}

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000 87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Data Memory

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

RCLK +TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

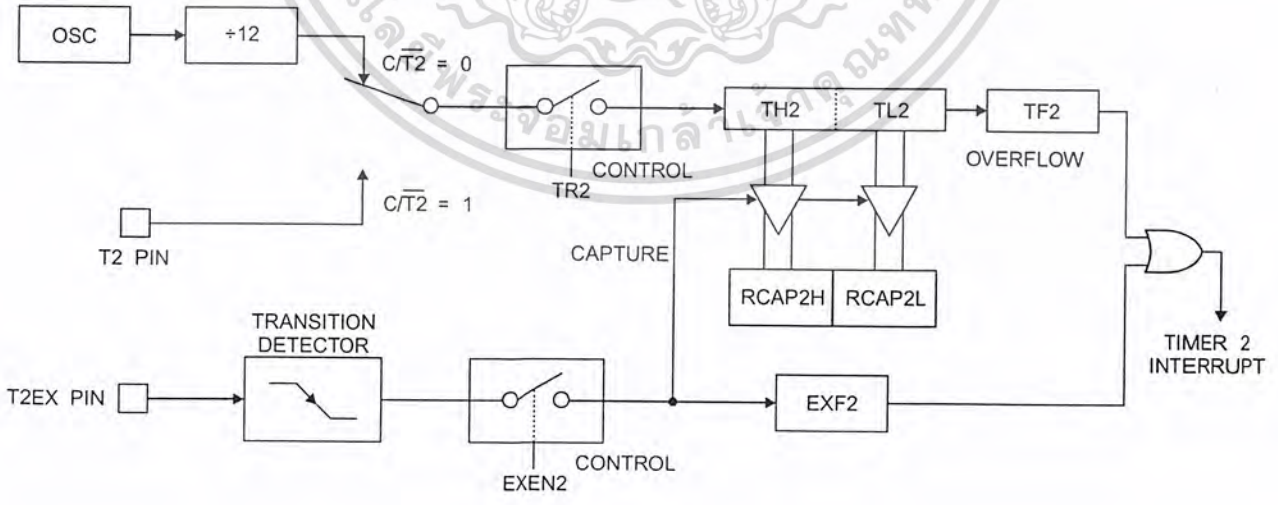
Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 1. Timer in Capture Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

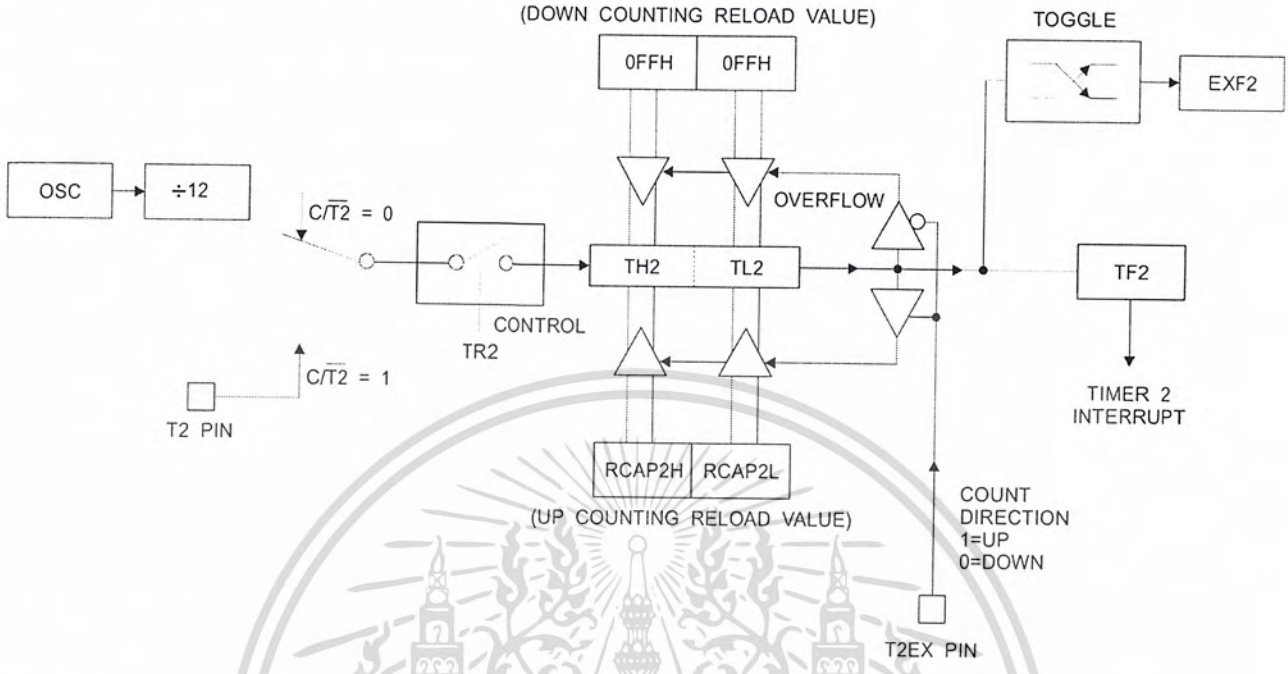
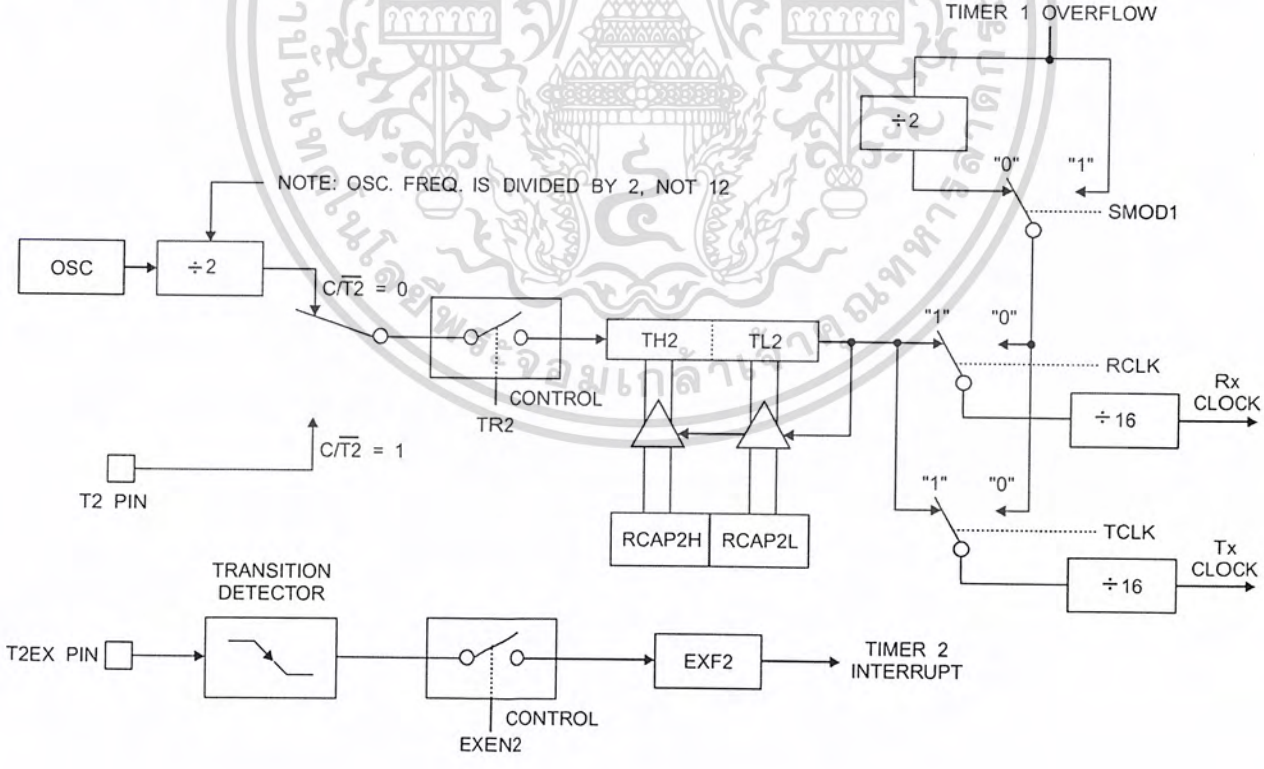


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

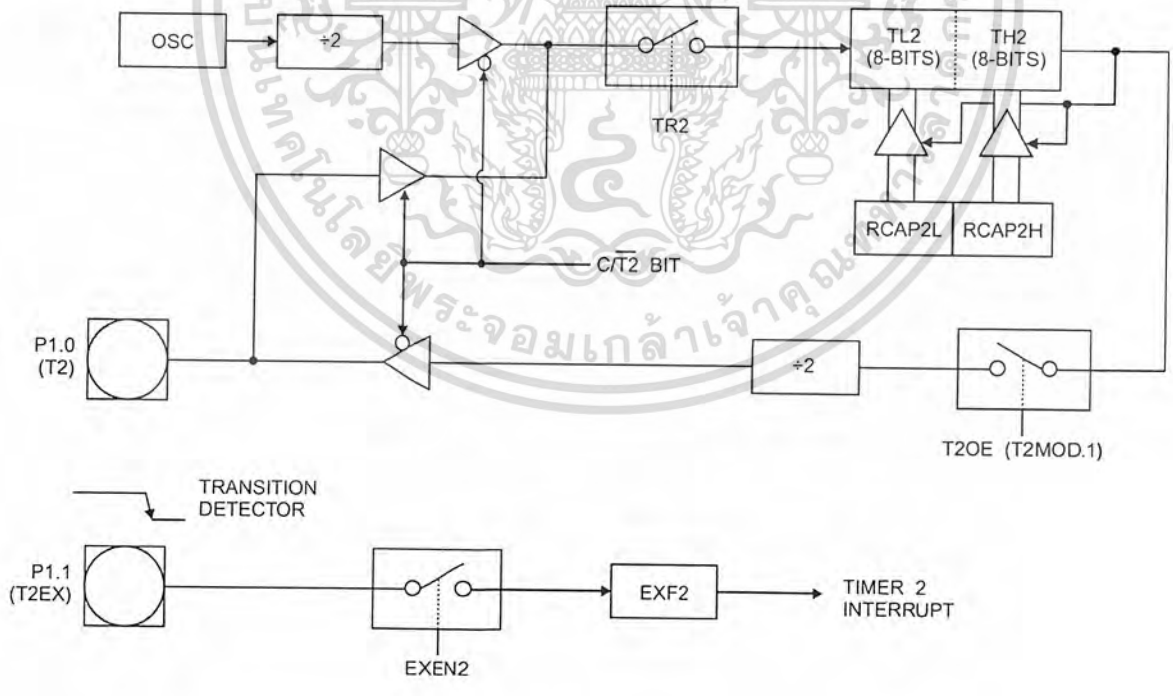
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-out Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Fequency}}{4 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

UART

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

Interrupts

The AT89C52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

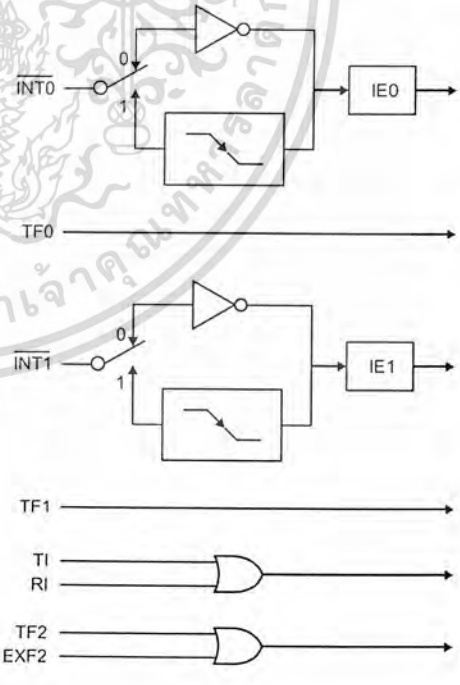
(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Figure 6. Interrupt Sources



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

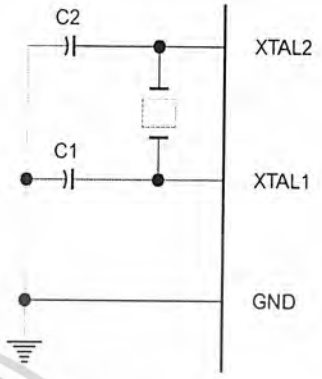
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power Down Mode

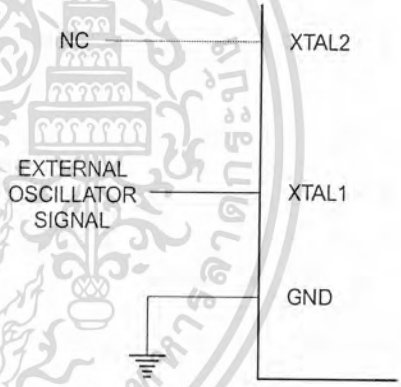
In the power down mode, the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 7. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 8. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash

The AT89C52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third- party Flash or EPROM programmers.

The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-side Mark	AT89C52 xxxx yyww	AT89C52 xxxx - 5 yyww
Signature	(030H) = 1EH (031H) = 52H (032H) = FFH	(030H) = 1EH (031H) = 52H (032H) = 05H

The AT89C52 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figures 9 and 10. To program the AT89C52, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling The AT89C52 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$ The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chip Erase The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

Reading the Signature Bytes The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 52H indicates 89C52
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9. Programming the Flash Memory

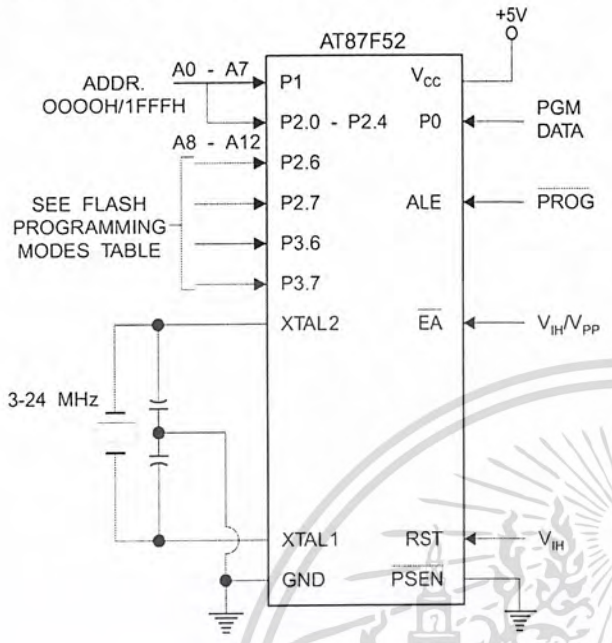
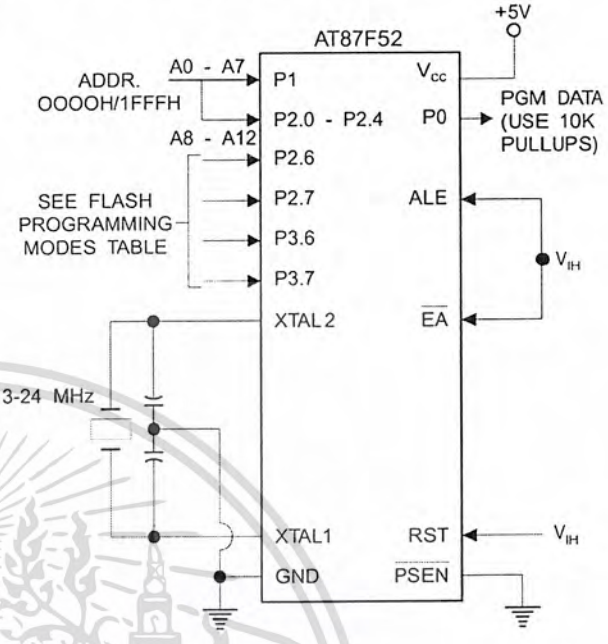


Figure 10. Verifying the Flash Memory



Flash Programming and Verification Characteristics

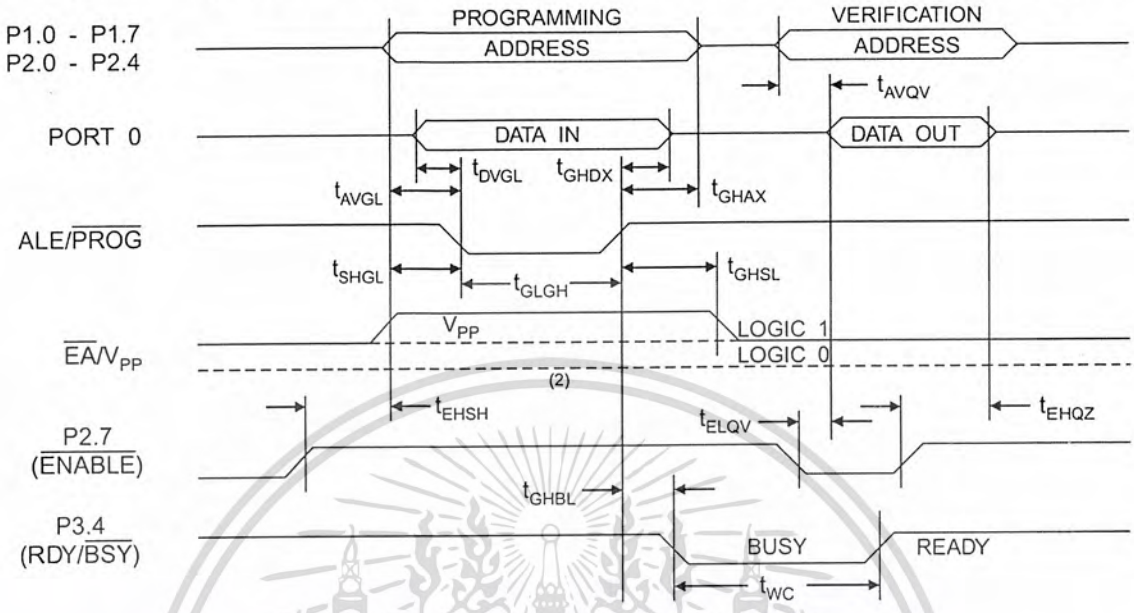
T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}[1]	Programming Enable Voltage	11.5	12.5	V
I_{PP}[1]	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}[1]	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

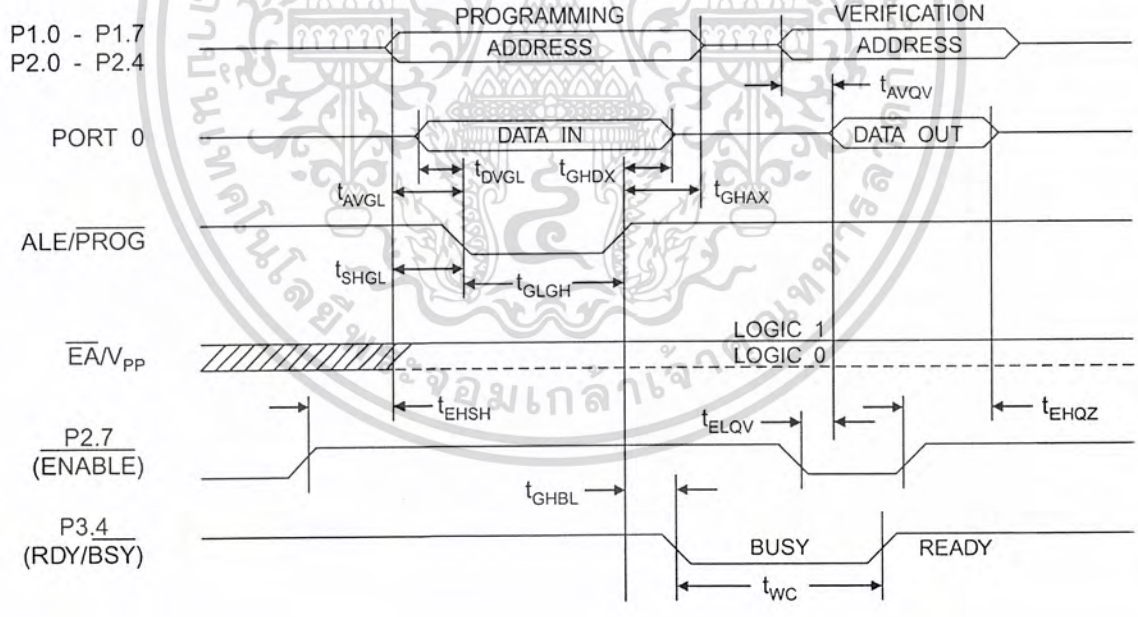
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode (V_{PP}=12V)



Flash Programming and Verification Waveforms - Low Voltage Mode (V_{PP}=5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้