

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ตัวส่งสัญญาณนอกผ่านคลื่นวิทยุขนาดเล็ก

ร/พ.
๒๑๑๖๓
๒๕๓๙

นายบดินทร์ ดำรงค์ดี
นางสาวราลี ไตวณะบุตร

เลขหมู่.....
เลขทะเบียน.....
วันเดือนปี.....

๖๑๒๕๑๖๗๔๕

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกส์ประยุกต์

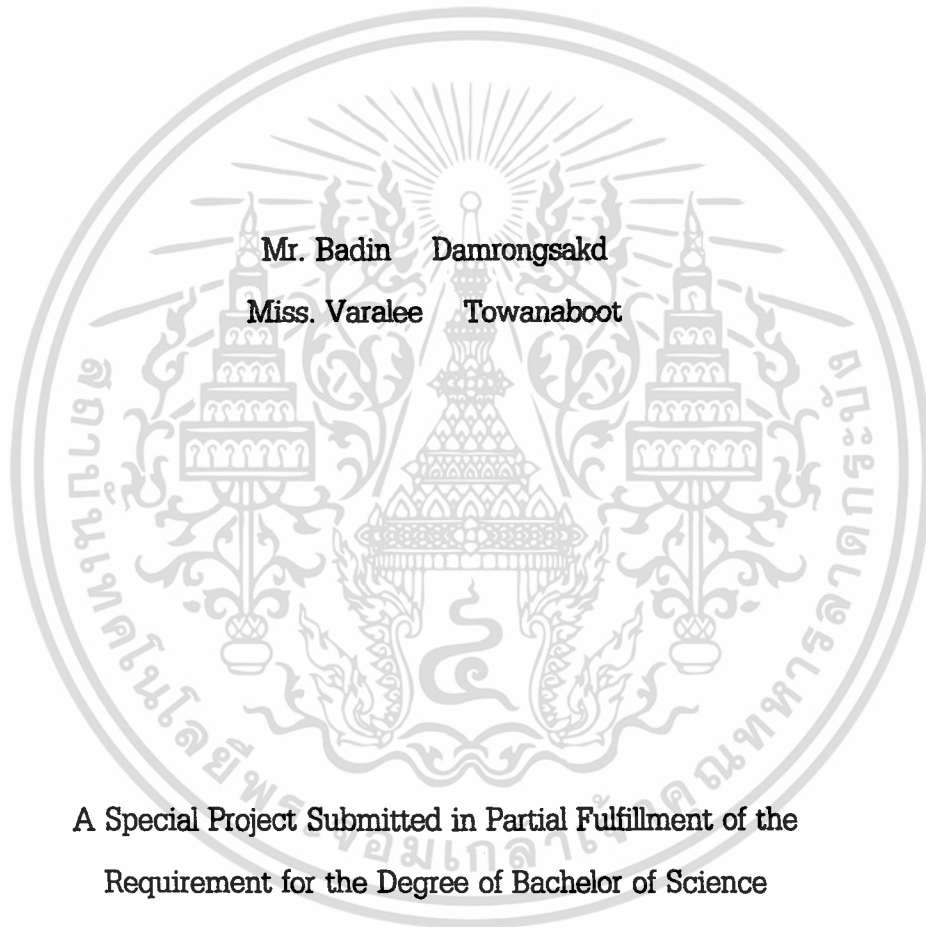
คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๓๙

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RF-linked Miniature Analog Signal Sender



Mr. Badin Damrongsakd

Miss. Varalee Towanaboot

**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

1996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ ตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุขนาดเล็ก

RF-linked Miniature Analog Signal Sender

โดย นายบัณฑิต คำรงค์ศักดิ์

นางสาววารลีย์ โทวณะบุตร

ภาควิชา ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง อนุมัติให้นำโครงการพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิทยาศาสตรบัณฑิต



(รองศาสตราจารย์สุรพล รักvijit)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

คณะกรรมการโครงการพิเศษ



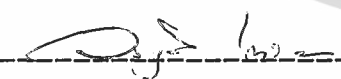
(ผู้ช่วยศาสตราจารย์วิชิต ศิริโชติ)

ประธานกรรมการ




(รองศาสตราจารย์สุรพล รักvijit)

กรรมการ



(ผู้ช่วยศาสตราจารย์ ดร.วารวูฒิ เถาลัดดา)

กรรมการ



(ผู้ช่วยศาสตราจารย์ ดร.ศิริศักดิ์ เตชะทวีกุล)

กรรมการ

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ	ตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุขนาดเล็ก
โดย	นายบัณฑิต คำรงค์ศักดิ์ นางสาววราลี ไทฉะบุตร
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์วิจิต ศิริโชติ
ภาควิชา	ฟิสิกส์ประยุกต์
ปีการศึกษา	2539

บทคัดย่อ

โครงการพิเศษนี้เป็นการพัฒนาระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม (Centralized Remote Monitoring System , CRMS) โดยได้ทำการปรับปรุงและพัฒนา 3 ส่วนด้วยกัน ได้แก่ ส่วนตัววัดปริมาณทางฟิสิกส์ผ่านคลื่นวิทยุ , โมเด็มส่งผ่านคลื่นวิทยุ และซอฟต์แวร์สำหรับควบคุมบนไมโครคอมพิวเตอร์

ตัววัดปริมาณทางฟิสิกส์ผ่านคลื่นวิทยุได้ทำการพัฒนาให้สามารถสื่อสารได้ 2 ทางแบบผลัดกันส่ง (Half Duplex) , เพิ่มเติมส่วนควบคุมระยะไกล และออกแบบวงจรให้มีขนาดเล็กลง โดยวงจรประกอบไปด้วยส่วนแปลงสัญญาณจากอนาลอกเป็นดิจิทัลชนิด Successive Approximation ความละเอียด 10 บิต จำนวน 8 ช่องสัญญาณ วัดสัญญาณได้ตั้งแต่ช่วง 0 - 4.096V , ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งเก็ลชิปโมเด็ม FSK รับ-ส่งข้อมูลแบบอะซิงโครนัส 1200 บิตต่อวินาที , ส่วนขับอุปกรณ์กำลัง และวิทยุสื่อสารย่านความถี่ 144-146 MHz

ส่วนของโมเด็มส่งผ่านคลื่นวิทยุได้ทำการพัฒนาให้สามารถสื่อสารได้ 2 ทางแบบผลัดกันส่งเช่นเดียวกัน อีกทั้งทำการเชื่อมต่อกับไมโครคอมพิวเตอร์ทาง RS-232 โดยใช้ซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้ควบคุมการทำงานบนระบบปฏิบัติการวินโดวส์

Special Project Title RF-lined Miniature Analog Signal Sender
Name Mr. Badin Damrongsakd
 Miss Varalee Towanabout
Special Project Advisor Asstt. Prof. Wichit Sirichote
Department Applied Physics
Academic Year 1996

Abstract

A device used for sending an analog signal by radio wave has been developed. The 8 channels analog voltage ranging from 0 to 4.096V was digitized by a 10 bits resolution successive approximation A/D converter, MAX192. A 20 pins DIP microcontroller, AT89c2051, was used to control the A/D converter. The special project has been developed for Centralized Remote Monitoring System, CRMS. Three parts of it are developed - RF-Linked Analog Sender, RF Modem, and Software on Microcomputer.

RF-Linked Analog Sender is developed to Half-duplex communication, minimize printed circuit board, and include Telecontrolling. This schematic consists of Successive Approximation type of Analog to Digital Conversion (ADC) which have 8 channels of signal and 10-bit resolutions range 0 - 4.096V, MCS-51 microcontroller, Asynchronous 1200 bps FSK Single Chip Modem, Power device, eg. Stepping Motor, driver and 2 metres radio transceiver.

RF Modem, also developed to Half-duplex communication, interfaced with microcomputer via RS-232 serial port which is controlled by software developing for Windows operating system.

กิติกรรมประกาศ

โครงการพิเศษนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากความอนุเคราะห์จากบุคคลหลายฝ่ายดังนี้

คุณพ่อ-คุณแม่ และ ญาติพี่น้องทุกคน
อาตุ๋, อภิรัชย์ ดำรงค์ศักดิ์

ผู้เป็นกำลังใจ และคอยห่วงใยเราเรื่อยมา
อาที่แสนดี และคอยถามไถ่ถึงโครงการ
พิเศษนี้ แม้อลังลับไปแล้ว

ผู้ช่วยศาสตราจารย์วิชิต ศิริโชติ

ผู้ให้การสอนสั่ง , ถ่ายทอดความรู้
และให้คำแนะนำต่างๆเสมอมา

อาจารย์ทุกท่าน

ผู้ถ่ายทอดความรู้แก่เราเรื่อยมา

พี่บ๊ิก พี่สิริส รุ่ง 10

พี่ชายที่คอยให้ คำแนะนำ

พี่แอน Math รุ่ง 11

พี่สาวที่ให้ข้อมูลทางอินเทอร์เน็ต

สำนักงานพัฒนาเทคโนโลยี

สำหรับอุปกรณ์และเครื่องมือต่างๆ

เพื่อนๆทุกคน

ผู้ให้กำลังใจ ด้วยความเต็มใจจริงๆ

ภาควิชาฟิสิกส์ประยุกต์

ที่ให้สิ่งดีๆตั้งแต่ปี 1 ถึง ปี 4

ขอขอบพระคุณทุกท่านเป็นอย่างสูง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	4
2.1 ระบบโทรมาตร	4
2.2 พื้นฐานของตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล	7
2.3 ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล แบบประมาณค่าที่บีท	8
2.3.1 หลักการทำงาน	9
2.3.2 การเชื่อมต่อกับตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล	9
2.3.2.1 การเชื่อมต่อแบบอนุกรม	10
2.3.2.2 การเชื่อมต่อแบบขนาน	11
2.4 ระบบการสื่อสารข้อมูลเบื้องต้น	11
2.4.1 ระบบสื่อสารแบบดิจิตอล	12
2.4.2 ทิศทางของการส่งสัญญาณ	12
2.4.2.1 การส่งสัญญาณแบบทิศทางเดียว	12
2.4.2.2 การส่งสัญญาณแบบสองทิศทางแต่ต่างเวลา	13
2.4.2.3 การส่งสัญญาณแบบสองทิศทางที่เวลาเดียวกัน	13
2.4.3 ลักษณะของการรับ-ส่งข้อมูลแบบดิจิตอล	14
2.4.3.1 การส่งข้อมูลแบบอนุกรม	14
2.4.3.2 การส่งข้อมูลแบบขนาน	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4	การสื่อสารแบบซิงโครนัสและอะซิงโครนัส	16
2.4.4.1	การสื่อสารแบบซิงโครนัส	16
2.4.4.2	การสื่อสารแบบอะซิงโครนัส	16
2.5	การมอดูเลตข้อมูลดิจิทัล	17
2.6	การสื่อสารผ่านระบบแพคเกจเรดิโอ	18
บทที่ 3	หลักการทํางานในส่วนซอฟต์แวร์	20
3.1	ระบบเฝ้าตรวจระวัง	21
3.1.1	การตรวจสอบระบบฮาร์ดแวร์	22
3.1.2	การส่งข้อมูลเพื่อควบคุมการทํางาน	22
3.1.3	การรับข้อมูลจากตัววัดปริมาณทางฟิสิกส์	23
3.1.4	การตรวจสอบผลรวม	24
3.1.5	การตรวจระบบเตือนภัย	24
3.1.6	ระบบควบคุมการทํางานของมอเตอร์	25
3.2	ระบบฐานข้อมูล	26
3.3	การเขียนโปรแกรมติดต่อผ่านพอร์ตอนุกรมด้วยวิซวลเบสิก	27
บทที่ 4	การดำเนินการวิจัย	30
4.1	ตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุขนาดเล็ก	30
4.1.1	ไมโครคอนโทรลเลอร์ AT89c2051	31
4.1.2	โมเด็มชิปเดี่ยว TCM 3105	35
4.1.3	การแปลงสัญญาณอนาลอกไปเป็นดิจิทัลโดยใช้ MAX 192	38
4.1.4	หลักการทํางานทํางานของโปรแกรม	44
4.2	โมเด็มส่งผ่านคลื่นวิทยุ	47
4.2.1	การรับส่งข้อมูลดิจิทัลผ่านพอร์ต RS-232 โดยใช้ DS275	48
4.2.1.1	การทํางานเบื้องต้น	49
4.2.1.2	การประยุกต์ใช้งาน	51
4.2.2	หลักการทํางานทํางานของโปรแกรม	52
4.3	วงจรควบคุมการกคดีย้ออกอากาศ	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	4.4 รูปแบบของเพกเกจข้อมูลที่ทำกรส่ง	54
บทที่ 5	ผลการทดลอง	56
บทที่ 6	สรุปผลการทดลองและแนวพัฒนา	61
ภาคผนวก		



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 ระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม	1
รูปที่ 1.2 โครงสร้างของตัวตรวจวัดปริมาณทางฟิสิกส์ส่งผ่านคลื่นความถี่วิทยุ	2
รูปที่ 1.3 โครงสร้างของโมเด็มส่งผ่านคลื่นความถี่วิทยุ	2
รูปที่ 2.1 ระบบโทรมาตรผ่านคลื่นความถี่วิทยุแบบพื้นฐาน	4
รูปที่ 2.2 ระบบโทรมาตรแบบใหม่	6
รูปที่ 2.3 ตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบประมาณค่าที่ละบิท	9
รูปที่ 2.4 การเชื่อมต่อตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบอนุกรมโดยทั่วไป	10
รูปที่ 2.5 การเชื่อมต่อตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบขนาน โดยทั่วไป	11
รูปที่ 2.6 การส่งสัญญาณแบบทิศทางเดียว	13
รูปที่ 2.7 การส่งสัญญาณแบบสองทิศทางแต่ต่างเวลา	13
รูปที่ 2.8 การส่งสัญญาณแบบสองทิศทางที่เวลาเดียวกัน	13
รูปที่ 2.9 การส่งข้อมูลแบบอนุกรม	14
รูปที่ 2.10 การส่งข้อมูลแบบขนาน	15
รูปที่ 2.11 การส่งข้อมูลแบบอะซิงโครนัส	16
รูปที่ 2.12 การมอดูเลตข้อมูลดิจิทัล	17
รูปที่ 2.13 ระบบแพคเกจรีโอ	18
รูปที่ 2.14 แสดงขาเจ็คสำหรับวิทยุสื่อสารชนิดมือถือ	19
รูปที่ 3.1 แผนภาพแสดงขั้นตอนการทำงานของโปรแกรม	21
รูปที่ 3.2 แสดงการทำงานในระบบเฝ้าตรวจระวัง	22
รูปที่ 3.3 แสดงการทำงานในส่วนการส่งข้อมูลเพื่อควบคุมตัววัดปริมาณทางฟิสิกส์	23
รูปที่ 3.4 แสดงลักษณะของข้อมูลที่รับจากตัววัดปริมาณทางฟิสิกส์	23
รูปที่ 3.5 แสดงการทำงานในส่วนของการตรวจสอบข้อมูล	24
รูปที่ 3.6 แสดงการทำงานในส่วนของการเตือนภัย	25
รูปที่ 3.7 แสดงการทำงานในระบบควบคุมการทำงานของมอเตอร์	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2	แสดงขาของ AT89C2051	33
รูปที่ 4.3	แสดงขาสัญญาณและไดอะแกรมวงจรภายในของ 89C2051	34
รูปที่ 4.4	ไดอะแกรมของโมเด็มชิปเดี่ยว TCM3105	35
รูปที่ 4.5	วงจรโดยทั่วไปของ TCM3105	36
รูปที่ 4.6	รูปร่างการจัดขาของ MAX192	39
รูปที่ 4.7	บล็อกไดอะแกรมภายในตัวไอซี	41
รูปที่ 4.8	(ก) สัญญาณเอาต์พุตแบบซิงเกิลเอนด์ (ข) สัญญาณเอาต์พุตแบบดิฟเฟอเรนเชียล	42
รูปที่ 4.9	ตัวอย่างบล็อกไดอะแกรมการต่อใช้งานร่วมกับ CPU	43
รูปที่ 4.10	วงจรตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุ	46
รูปที่ 4.11	โครงสร้างการทำงานของโมเด็มส่งผ่านคลื่นวิทยุ	47
รูปที่ 4.12	ลักษณะการจัดตำแหน่งขาของไอซี	48
รูปที่ 4.13	การจัดวงจรภายในของ DS275	50
รูปที่ 4.14	วงจรประยุกต์ใช้งานเบื้องต้น	51
รูปที่ 4.15	วงจรควบคุมการกดคีย์ออกอากาศ	53
รูปที่ 4.16	วงจรโมเด็มส่งผ่านคลื่นวิทยุ	55
รูปที่ 5.1	โมเด็มส่งผ่านคลื่นวิทยุ	57
รูปที่ 5.2	ตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุ	57
รูปที่ 5.3	การเชื่อมต่อโมเด็มส่งผ่านคลื่นวิทยุกับคอมพิวเตอร์	58
รูปที่ 5.4	การเชื่อมต่อตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุกับอุปกรณ์	58
รูปที่ 5.5	ผลการทดลองเมื่อใช้การโปรแกรมที่พัฒนาขึ้น	60

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานกับอนุกรม	15
ตารางที่ 4.1 แสดงโหมคในการทำงาน	37



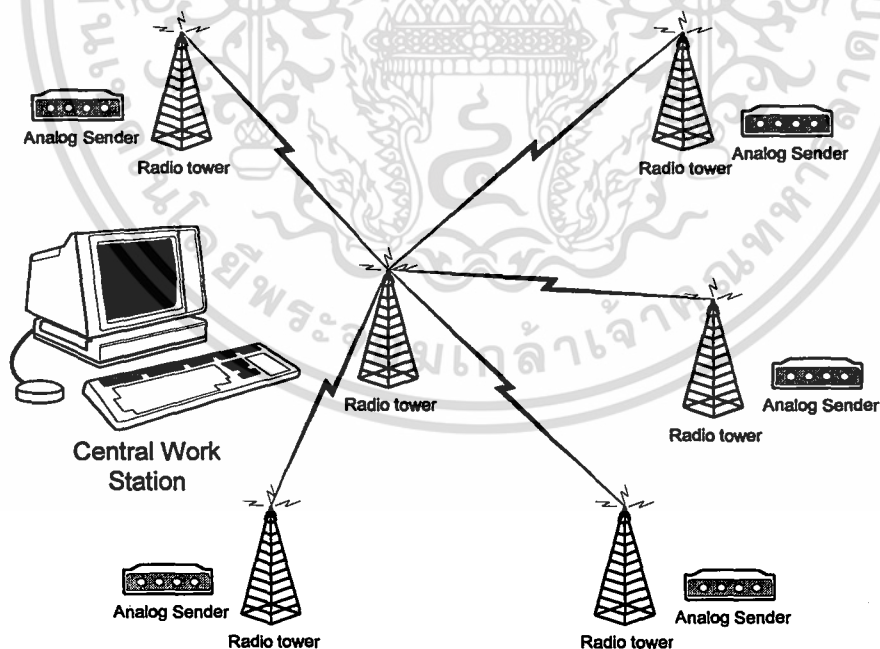
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันเรามีความต้องการตรวจวัดปริมาณทางฟิสิกส์ เพื่อนำมาทำการทดลอง, วิจัย หรือตรวจสอบ ซึ่งปริมาณทางฟิสิกส์ที่เราจะนำมาตรวจสอบนั้นมีอยู่มากมายด้วยกัน เช่น การวัดค่า pH ของน้ำเพื่อตรวจสอบการปล่อยน้ำเสียออกจากโรงงาน, การวัดระดับของน้ำในเขื่อน, การวัดอุณหภูมิภายในระบบ, การส่งสัญญาณภาพสำหรับคุณภาพการจราจรหรือเฝ้าสังเกตปรากฏการณ์ต่างๆ และอื่นๆ

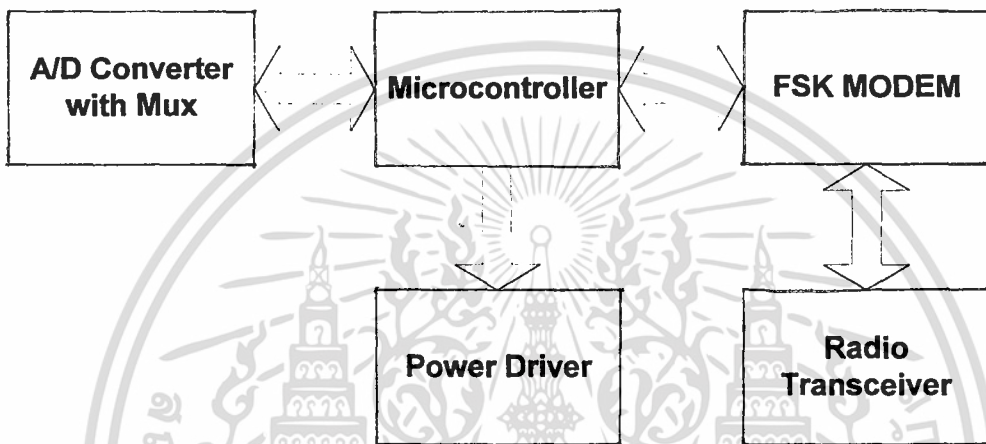
เนื่องจากเราไม่สามารถที่จะคอยเฝ้าตรวจวัดได้ตลอดเวลา หรือไม่สามารถเข้าไปทำการวัดได้โดยตรง ด้วยเหตุนี้เราจึงอาศัยระบบการวัดระยะไกลแบบศูนย์กลางรวม (Centralized Remote Monitoring System : CRMS) เป็นตัวเชื่อมโยงเครือข่ายระหว่างสถานที่ที่ทำการตรวจวัดหลายๆจุด (Remote site) กับสถานที่แสดงผล (Local site) โดยอาศัยการสื่อสารแบบใช้สายและการสื่อสารแบบไม่ใช้สายช่วยในการถ่ายทอดข้อมูลที่ทำการวัด



รูปที่ 1.1 ระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาเราจึงได้ทำการพัฒนาระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม เพื่อให้มีประสิทธิภาพดียิ่งขึ้น โดยทำการพัฒนา ส่วนตัววัดปริมาณทางฟิสิกส์ส่งผ่านคลื่นความถี่วิทยุ , โมเด็มส่งผ่านทางคลื่นความถี่วิทยุ และซอฟต์แวร์สำหรับควบคุมการทำงานบนไมโครคอมพิวเตอร์ในระบบปฏิบัติการวินโดวส์



รูปที่ 1.2 โครงสร้างของตัวตรวจวัดปริมาณทางฟิสิกส์ส่งผ่านคลื่นความถี่วิทยุ



รูปที่ 1.3 โครงสร้างของโมเด็มส่งผ่านคลื่นความถี่วิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม
2. เพื่อศึกษาระบบโทรมาตรแบบไร้สาย
3. สร้างตัวตรวจวัดปริมาณทางฟิสิกส์ส่งผ่านคลื่นความถี่วิทยุขนาดเล็ก
4. สร้างโมเด็มส่งผ่านทางคลื่นความถี่วิทยุ
5. ออกแบบซอฟต์แวร์สำหรับควบคุมการทำงานบนไมโครคอมพิวเตอร์ในระบบปฏิบัติการ วินโดวส์

1.2 ประโยชน์ที่ได้รับ

1. เข้าใจการทำงานของระบบตรวจวัดระยะไกลแบบศูนย์กลางรวม
2. เป็นแนวทางในการพัฒนาระบบตรวจวัดระยะไกลต่อไปในอนาคต
3. สามารถนำไปใช้เพื่อเป็นประโยชน์ต่อส่วนรวมได้

บทที่ 2

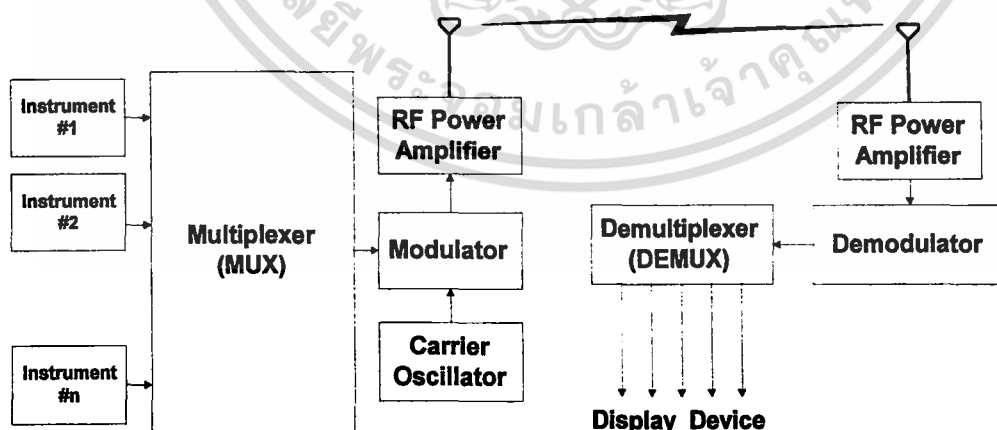
ทฤษฎี

2.1 ระบบโทรมาตร (Telemetry System)

ระบบโทรมาตร หรือระบบตรวจวัดระยะไกล แบ่งออกเป็น 2 ส่วนด้วยกัน ได้แก่ จุดตรวจวัด (Remote Site) และ จุดแสดงผล (Local Site) โดยจุดตรวจวัดเป็นส่วนของการวัดปริมาณทางฟิสิกส์ต่างๆในสถานที่ ที่เข้าไปได้ยากหรือไม่สามารถเข้าไปได้ ตัวอย่างของจุดตรวจวัดได้แก่ หุ่นลอยในทะเล ,หุ่นยนต์ ,ดาวเทียม หรืออุปกรณ์ส่งข้อมูลในป่าดงดิบ ส่วนจุดแสดงผลจะเป็นส่วนที่ใช้ในการแสดงผลข้อมูลที่ได้จากจุดตรวจวัด ซึ่งเข้าถึงข้อมูลได้ง่าย

ส่วนใหญ่แล้วนั้นระบบตรวจวัดระยะไกลมักจะใช้พาหะความถี่สูงมามอดูเลตกับข้อมูลเพื่อทำการส่งข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยระบบที่นิยมใช้กันเป็นระบบโทรมาตรแบบไร้สายผ่านทางคลื่นความถี่วิทยุ โดยอาศัยเสาอากาศเป็นตัวรับ-ส่งสัญญาณ นอกจากนี้แล้วยังใช้ระบบโทรมาตรแบบใช้สายผ่านทางอุปกรณ์ส่ง ซึ่งอาศัยกระแสไฟฟ้าเป็นพาหะในการส่งข้อมูลผ่านทางสายไฟฟ้า เช่น สายโทรศัพท์ หรือ สายไฟฟ้าโดยทั่วไป

พื้นฐานของระบบโทรมาตรผ่านทางคลื่นความถี่วิทยุแสดงดังรูปที่ 2.1



รูปที่ 2.1 ระบบโทรมาตรผ่านคลื่นความถี่วิทยุแบบพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการวัดปริมาณทางฟิสิกส์จากเครื่องมือวัดได้แล้ว ก็ทำการส่งข้อมูลที่ได้ออกไปยัง มัลติเพล็กซ์เซอร์ (Multiplexer , MUX) ซึ่งจะเป็นตัวนำข้อมูลที่ได้ออกมารวมเข้า เป็นสัญญาณๆ เดียว จากนั้นจึงส่งสัญญาณต่อไปให้กับมอดูเลเตอร์ (Modulator) เพื่อที่จะทำการมอดูเลตข้อมูลกับความถี่พาห้ (Carrier Frequency) แล้วนำสัญญาณที่ได้มา ขยายกำลังโดยตัวขยายกำลังย่านความถี่วิทยุ (RF Power Amplifier) ก่อนที่จะส่งออก จากเสาอากาศ (เสาอากาศแบบทิศทางจะมี gain ในการรับ-ส่งสูงกว่าแต่จะรับ-ส่งได้ดี ในทิศทางเดียว หรือเสาอากาศแบบรอบตัวจะมีทิศทางในการรับ-ส่งได้ดีเท่ากันหมดใน ทุกๆทิศทางแต่มี gain ต่ำ) เมื่อเสาอากาศของตัวรับได้รับสัญญาณที่ส่งออกมา ก็จะ ทำการขยายสัญญาณที่รับได้ก่อนจะผ่านดีมอดูเลเตอร์ (Demodulator) เพื่อแยกเอา ความถี่พาห้ออกจากสัญญาณข้อมูลซึ่งต้องเป็นจังหวะเดียวกัน (Synchronized) กับมอดูเลเตอร์ของตัวส่ง ในบางครั้งข้อมูลที่ได้จะถูกแสดงผลโดยตรง บางครั้งก็ทำการนำ ข้อมูลที่ได้ไปประมวลผลสัญญาณก่อนจะนำข้อมูลไปแสดงผลต่อไป

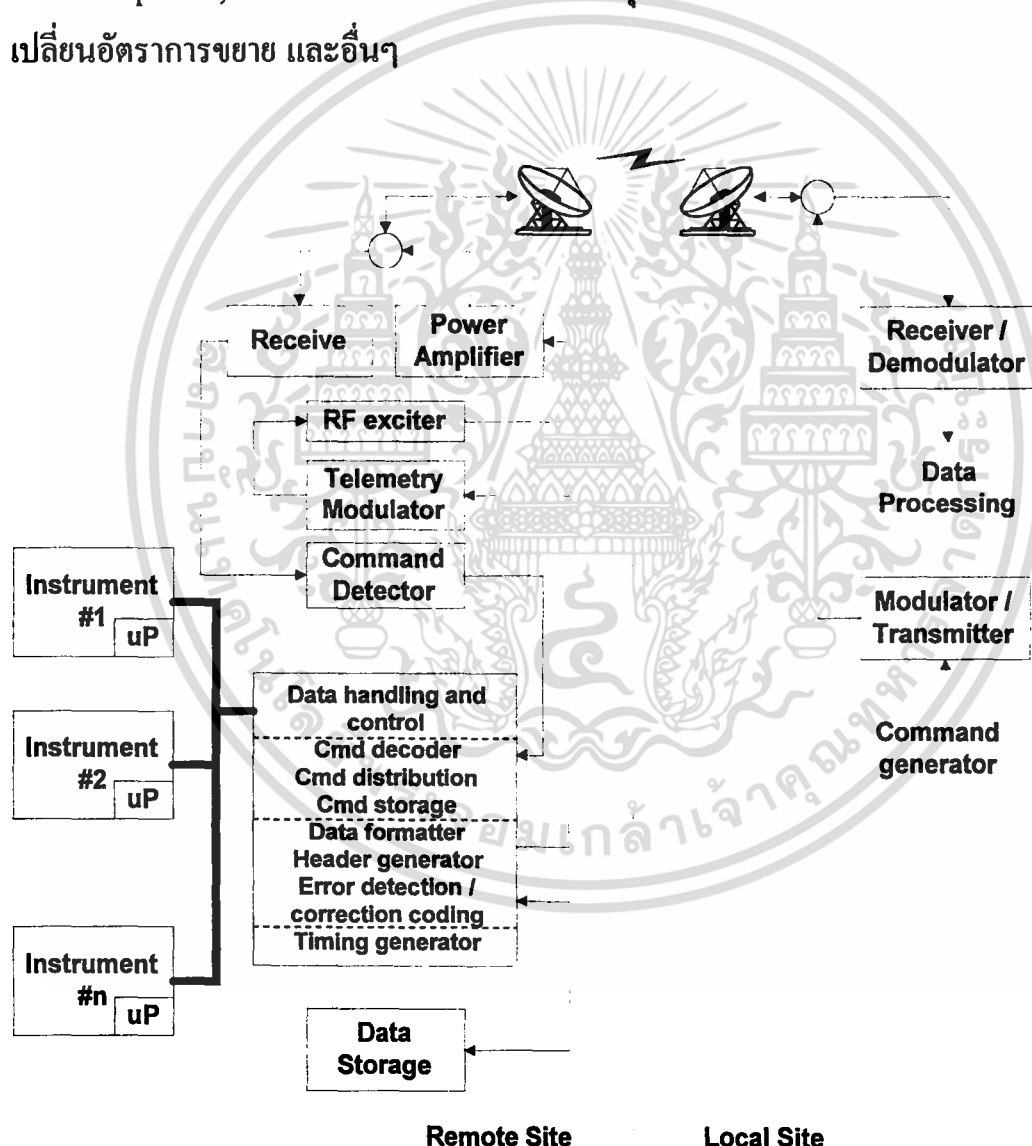
ในระบบโทรมาตรดั้งเดิมนั้น ข้อมูลที่ส่งออกไปยังมัลติเพล็กซ์เซอร์จะอยู่ในฟอร์ม ของอนาลอก ซึ่งในระบบสมัยใหม่นี้ข้อมูลที่ได้จะอยู่ในรูปของดิจิตอลแทน (เนื่องจาก สัญญาณอนาลอกเกิดการผิดพลาดของข้อมูลเนื่องจากสัญญาณรบกวนได้ง่ายกว่า สัญญาณดิจิตอล) โดยที่ส่วนปรับสภาพสัญญาณ (Signal Conditioner) ของเครื่องมือ วัดจะประกอบไปด้วยตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter , ADC) หรือไมก็อาศัยตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลในการดิจิไทซ์ สัญญาณจากเอาท์พุทที่ได้จากเครื่องมือวัด ซึ่งอยู่ในฟอร์มของอนาลอกข้อมูลแบบดิจิ ตอลจะอยู่ในรูปของเลข 0 กับ 1 (หรือ ปิด/เปิด) แล้วอาศัยการมอดูเลตด้วยวิธี Frequency Shift Keying (FSK) ซึ่งเป็นการใช้ความถี่ที่แตกต่างกันระหว่างเลข 0 กับ 1 หรือใช้วิธี Phase Shift Keying (PSK) ซึ่งเป็นการเปลี่ยนเฟสของข้อมูลแทนการ เปลี่ยนความถี่

ชนิดของการมัลติเพล็กซ์เซอร์ก็มีความสำคัญเช่นกัน ซึ่งเราแบ่งออกเป็น 2 แบบ ได้แก่ การมัลติเพล็กซ์แบบแบ่งตามความถี่ (Frequency - Division Multiplexing , FDM) ซึ่งเทคนิคนี้อาศัยการส่งข้อมูลออกไปด้วยความถี่ที่แตกต่างกัน โดยสามารถที่จะแสดงผลข้อมูลได้ต่อเนื่องกันทุกๆสัญญาณ แต่วิธีที่นิยมใช้กันมากที่สุดได้แก่การมัลติเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบแบ่งตามเวลา (Time - Division Multiplexing , TDM) เป็นการส่งข้อมูลที่ใช้ช่วงเวลาที่แตกต่างกันของข้อมูลแต่ละชนิด ทำให้ข้อมูลที่ได้แสดงผลไม่ต่อเนื่องกัน

รูปที่ 2.2 แสดงระบบโทรมาตรที่ได้รับการพัฒนาขึ้นแล้ว จะเห็นว่าในระบบใหม่นี้ข้อมูลไม่จำเป็นต้องส่งจากจุดตรวจวัดมายังจุดแสดงผล (เป็นแบบ Simplex) เท่านั้นแต่จุดแสดงผลสามารถที่จะส่งคำสั่งไปควบคุมเครื่องมือวัดได้อีกด้วย (เป็นแบบ Half Duplex) ตัวอย่างคำสั่งที่ใช้ในการควบคุมก็ได้แก่การเปิดหรือปิดเครื่องมือวัด, เปลี่ยนอัตราการขยาย และอื่นๆ



รูปที่ 2.2 ระบบโทรมาตรแบบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 พื้นฐานของตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล

ในโลกของเรามีปริมาณทางฟิสิกส์ต่างๆที่เราคุ้นเคย เช่น แสง, อุณหภูมิ, ความดัน, เสียง ตลอดจนกระแสไฟฟ้า ล้วนแล้วแต่เป็นปริมาณที่มีการเปลี่ยนแปลงกับเวลาอย่างต่อเนื่อง หรือที่เราเรียกว่า สัญญาณอนาลอก เนื่องจากสัญญาณอนาลอกเป็นฟังก์ชันที่มีการเปลี่ยนแปลงอย่างต่อเนื่องไม่มีที่สิ้นสุด จึงเป็นการยากที่เราจะสามารถควบคุมสัญญาณให้ทำงานตามที่เรต้องการได้ แต่ในทางตรงข้ามเราพบว่าสัญญาณอีกชนิดหนึ่งซึ่งไม่เป็นฟังก์ชันกับเวลาอย่างต่อเนื่อง หรือที่เราเรียกกันว่า สัญญาณดิจิตอล สามารถที่จะควบคุมได้ง่ายกว่า โดยการใช้วงจรอย่างง่ายหรือไมโครคอนโทรลเลอร์

ดังนั้นเมื่อเราต้องการที่จะทำการวัดและควบคุมปริมาณทางฟิสิกส์หรือสัญญาณอนาลอกนั้น เนื่องจากเราไม่สามารถที่จะเชื่อมต่อสัญญาณอนาลอกเข้ากับสัญญาณดิจิตอลได้โดยตรง เราจึงจำเป็นต้องแปลงสัญญาณอนาลอกให้อยู่ในรูปของสัญญาณดิจิตอลเสียก่อน โดยอาศัยวงจรสำหรับแปลงรูปสัญญาณหรือวงจรรวม (Integrated Circuit . IC) โดยมีเทคนิคพื้นฐานที่ใช้ในการแปลงข้อมูลหลายวิธีด้วยกัน ซึ่งแต่ละวิธีก็มีข้อดี - ข้อเสียแตกต่างกันออกไป

ตัวแปลงแบบแฟลช (Flash converter) : เป็นเทคนิคในการแปลงสัญญาณอนาลอกเป็นดิจิตอลที่มีความเร็วในการแปลงสูงที่สุด โดยอาศัยตัวเปรียบเทียบสัญญาณ (Comparator) $2^n - 1$ ตัว (n คือ จำนวนบิต) โดยทั่วไปตัวแปลงสัญญาณชนิดนี้ถูกนำไปใช้ในอุปกรณ์ตรวจสอบอัตโนมัติที่มีราคาแพง

ตัวแปลงแบบสโลปคู่ (Dual-slope converter) : เป็นเทคนิคการแปลงที่มีค่ารีโซลูชัน (resolution) สูง โดยอาศัยการอินทิเกรตสัญญาณอินพุตที่ไม่ทราบค่ากับการอินทิเกรตสัญญาณอ้างอิงที่ทราบค่าแล้ว โดยทั่วไปจะใช้ตัวแปลงสัญญาณชนิดนี้กับอุปกรณ์ประเภทเครื่องมือวัด

ตัวแปลงแบบประมาณค่าทีละบิต (Successive-Approximation converter) : เป็นเทคนิคที่นิยมนำมาใช้มากที่สุด ซึ่งมีความเร็วในการแปลงข้อมูลค่อนข้างสูงและมีจำนวนรีโซลูชันค่อนข้างสูง โดยอาศัยการประมาณค่าสัญญาณอินพุตด้วยตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก (Digital to Analog Converter , DAC) ซึ่งนำเอาค่าที่ได้

มาเปรียบเทียบกับสัญญาณจริง โดยเริ่มเปรียบเทียบจากบิตนัยสำคัญสูงสุด (MSB) จนกระทั่งถึงบิตนัยสำคัญต่ำสุด (LSB)

ตัวแปลงแบบเดลต้า - ซิกม่า (Delta - Sigma converter) : เป็นเทคนิคที่มีรีโซลูชันสูงที่สุด โดยอาศัยวงจรโอเวอร์แซมปลิง และการมอดูเลต ซึ่งโดยส่วนใหญ่ได้ใช้ทำเป็นเครื่องมือวัดและงานด้านออดิโอ

จากทฤษฎีการสุ่มสัญญาณของนายควิสท์ (Nyquist sampling theorem) กล่าวไว้ว่า "การจะสุ่มสัญญาณอินพุตนั้นความถี่ของการสุ่มจะต้องมากกว่าหรือเท่ากับสองเท่าของความถี่สูงสุดของสัญญาณอินพุต ($f_s \geq 2f_{max}$)" ซึ่งในโครงการตัววัดปริมาณทางฟิสิกส์ส่งผ่านคลื่นความถี่วิทยุขนาดเล็ก จึงเลือกใช้ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลชนิดประมาณค่าทีละบิตเนื่องจากมีรีโซลูชันค่อนข้างสูงและมีอัตราเร็วในการสุ่มข้อมูลค่อนข้างสูงเช่นกัน

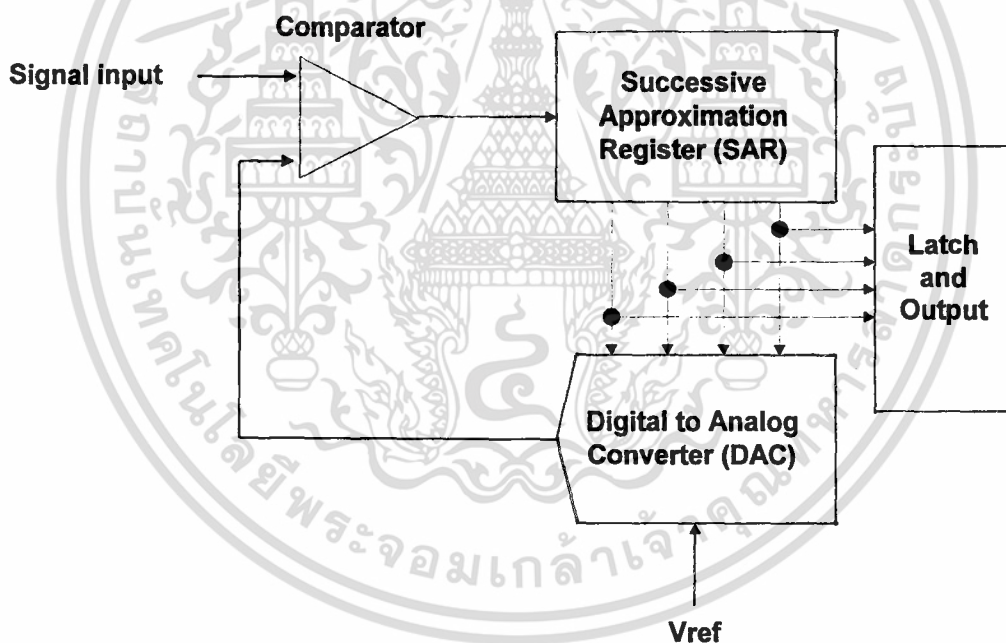
2.3 ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าทีละบิต

ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าทีละบิตบางครั้งถูกเรียกว่าตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบสุ่ม (Sampling ADC) โดยอาศัยหลักการทดสอบแต่ละบิตของรีโซลูชันจากบิตนัยสำคัญสูงสุดจนถึงบิตนัยสำคัญต่ำสุด ตัวแปลงสัญญาณชนิดนี้เป็นที่นิยมใช้กันมาก เนื่องจากมันมีช่วงในการทำงานกว้าง

ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าทีละบิตจะสุ่มสัญญาณอินพุตทีละบิตในการแปลงข้อมูลในแต่ละครั้ง โดยเวลาในการได้รับข้อมูล (Acquisition time) คือช่วงเวลาในการสุ่มสัญญาณอินพุตจนตัวเก็บประจุในวงจรสุ่มและคงค่า (Sample and Hold circuit , S/H) ถูกเก็บประจุจนเต็ม ซึ่งฐานเวลาที่เหมาะสมที่สุดก็คือเวลาที่เกิดจากตัวต้านทานและตัวเก็บประจุภายในตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล

2.3.1 หลักการทำงาน

จะทำการประมาณตัวเลขในหลักนัยสำคัญสูงสุดก่อน โดยให้เป็นลอจิก "1" แล้วนำลอจิกไปผ่านวงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกเพื่อนำสัญญาณอนาลอกที่ได้ไปเปรียบเทียบกับสัญญาณที่ป้อนเข้ามาด้วยวงจรเปรียบเทียบ ถ้าสัญญาณที่ได้มีค่าน้อยกว่าสัญญาณที่ถูกป้อนเข้ามาก็จะส่งลอจิกไปบอก SAR ให้คงสถานะลอจิก "1" ของบิตนี้เอาไว้ แต่ถ้าสัญญาณที่ได้มีค่ามากกว่าสัญญาณที่ป้อนเข้ามาก็จะทำการส่งลอจิกไปบอก SAR เช่นกันให้เปลี่ยนลอจิกจาก "1" เป็น "0" จากนั้นจึงทำการประมาณค่าข้อมูลในบิตต่อไป เมื่อทำการสุ่มแปลงสัญญาณทีละบิตเรียบร้อยแล้วก็จะแลกรหัสข้อมูลหรือองค์ค่าข้อมูลไว้ที่เอาท์พุท



รูปที่ 2.3 ตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบประมาณค่าทีละบิต

2.3.2 การเชื่อมต่อกับตัวแปลงสัญญาณอนาลอกเป็นดิจิทัล

พื้นฐานของการเชื่อมต่อระหว่างตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลกับไมโครคอนโทรลเลอร์แบ่งออกเป็น 2 ชนิดด้วยกันได้แก่

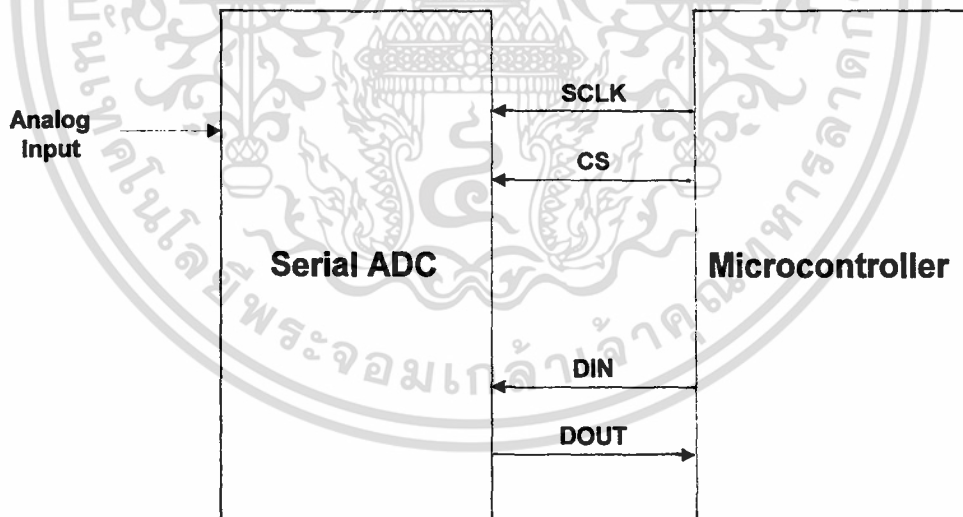
- การเชื่อมต่อแบบอนุกรม
- การเชื่อมต่อแบบขนาน

2.3.2.1 การเชื่อมต่อแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ของหลายๆบริษัทด้วยกันเช่น โมโตโรลา, เท็กซัส อินสตรูเมนต์ และ เนชันแนล เซมิคอนดักเตอร์ จะได้บรรจุคุณสมบัติการเชื่อมต่อแบบ การเลื่อนรีจิสเตอร์แบบอนุกรม (Serial shift register) ไว้ในคำสั่งของซอฟต์แวร์ด้วย ซึ่ง จากคุณสมบัตินี้ทำให้สามารถที่จะเชื่อมต่อกับตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลได้ ง่ายมาก

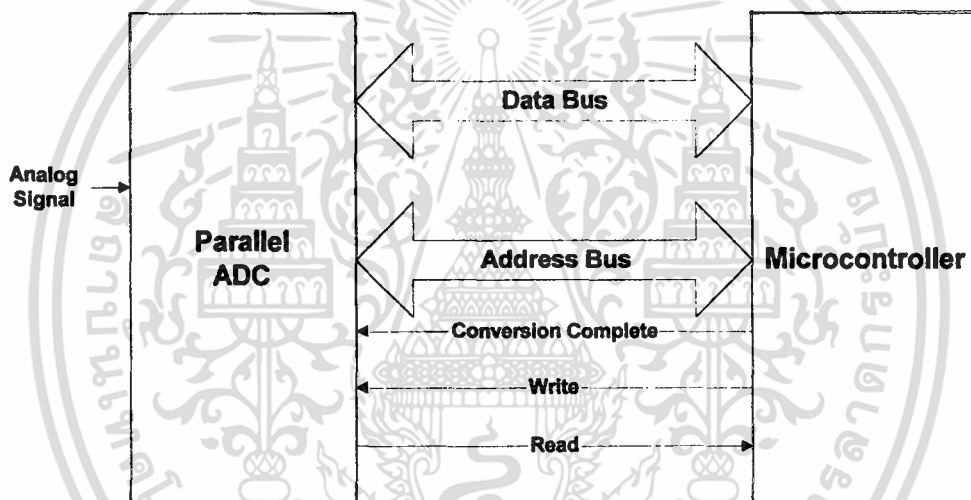
การเชื่อมต่อนั้นจะมีขาที่ใช้ในการควบคุมการเริ่มแปลงสัญญาณ เช่น ขา Chip Select (CS) เมื่อให้ลอจิก "0" กับขา CS ก็จะทำให้ตัวแปลงสัญญาณอนาลอก เป็นดิจิตอลเริ่มทำการแปลง เป็นต้น โดยมีอัตราในการแปลงและช่วงเวลาในการรับ-ส่ง ข้อมูลกับไมโครคอนโทรลเลอร์เป็นไปตามสัญญาณนาฬิกาที่ป้อนให้ทางขา Serial Clock (SCLK) นอกจากนี้เราสามารถที่จะควบคุมการทำงานอื่นๆของตัวแปลง สัญญาณได้ด้วยการส่งคำสั่งผ่านทางขา Data input (DIN) ของตัวแปลงสัญญาณ และ รับข้อมูลที่แปลงสัญญาณเสร็จแล้วจากขา Data output (DOUT)



รูปที่ 2.4 การเชื่อมต่อตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบอนุกรมโดยทั่วไป

2.3.2.2 การเชื่อมต่อแบบขนาน

โดยปกติแล้วมีสองวิธีด้วยกันในการเชื่อมต่อแบบขนานกับไมโครคอนโทรลเลอร์ ได้แก่ วิธีการอั่งแอดเดรส กับ วิธีการกำหนดอินพุตและเอาต์พุต ซึ่งทั้งสองวิธีนี้เหมือนกันเกือบทั้งหมด แต่แตกต่างกันตรง วิธีการอั่งแอดเดรสจะใช้การอ่าน-เขียนข้อมูลเหมือนการอ่าน-เขียนข้อมูลจากหน่วยความจำภายนอก ซึ่งไมโครคอนโทรลเลอร์มีชุดคำสั่งอยู่แล้ว ส่วนวิธีการกำหนดอินพุตและเอาต์พุตเราจำเป็นต้องเขียนคำสั่งในการติดต่อกับตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลเพื่อใช้ควบคุมเอง



รูปที่ 2.5 การเชื่อมต่อตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลแบบขนานโดยทั่วไป

2.4 ระบบการสื่อสารข้อมูลเบื้องต้น

ในชีวิตประจำวันของมนุษย์เรานั้น แทบจะกล่าวได้ว่าต้องเกี่ยวข้องกับการสื่อสารในรูปแบบใดรูปแบบหนึ่งตลอดเวลา ไม่ว่าจะเป็นการสนทนากัน, การอ่านหนังสือ, การรับส่งจดหมาย, การดูหนังฟังเพลง หรือการนั่งฟังสัมมนา ล้วนเป็นการสื่อสารข้อมูลซึ่งมีรูปแบบของการสื่อสารแตกต่างกันไปตามคุณสมบัติเฉพาะของแต่ละชนิดซึ่งล้วนแล้วแต่มีจุดมุ่งหมายที่จะส่งข่าวสารจากจุดหนึ่งไปยังอีกจุดหนึ่งอย่างถูกต้องทั้งสิ้น

เราสามารถแบ่งชนิดของระบบสื่อสารออกเป็น 2 แบบ ตามลักษณะสัญญาณที่ใช้ในระบบ ได้แก่

- แบบสัญญาณอนาลอก ตัวอย่างเช่น เสียงพูด เป็นต้น
- แบบสัญญาณดิจิทัล ตัวอย่างเช่น เลขฐานสอง เป็นต้น

ซึ่งในที่นี้เราจะกล่าวถึงเฉพาะระบบสื่อสารแบบดิจิทัลที่นำมาใช้ในโครงการงานตัววัดปริมาณทางฟิสิกส์ส่งผ่านทางคลื่นความถี่วิทยุขนาดเล็ก

2.4.1 ระบบสื่อสารแบบดิจิทัล

ลักษณะข้อมูลที่ใช้ในระบบนี้จะอยู่ในรูปฟอร์มของตัวเลข "0" กับ "1" เช่น ตัวเลขฐานสอง, ตัวเลขฐานสิบหก เป็นต้น

บางครั้งอาจมีความต้องการส่งสัญญาณอนาลอกผ่านทางระบบดิจิทัลจึงต้องมีการเปลี่ยนสัญญาณอนาลอกให้เป็นดิจิทัลเสียก่อนเรียกว่า การสุ่มตัวอย่าง (Sampling) ซึ่งเป็นวิธีทางคณิตศาสตร์ ค่าที่ได้จากการสุ่มตัวอย่างจัดเป็นเลขฐานสอง (binary code) ที่สามารถจัดตามเทคนิคทางดิจิทัลได้ อย่างเช่น การส่งข้อมูลแบบขนานหรืออนุกรม และแบบซิงโครนัสหรือแบบอะซิงโครนัส เป็นต้น

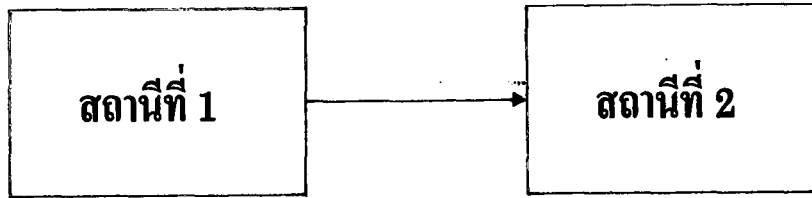
2.4.2 ทิศทางของการส่งสัญญาณ

เราสามารถแบ่งทิศทางของการส่งสัญญาณได้เป็น 3 วิธีด้วยกัน

- การส่งผ่านแบบทิศทางเดียว (Simplex)
- การส่งผ่านแบบสองทิศทางแต่ต่างเวลากัน (Half - duplex)
- การส่งผ่านแบบสองทิศทางที่เวลาเดียวกัน (Full - duplex)

2.4.2.1 การส่งสัญญาณแบบทิศทางเดียว

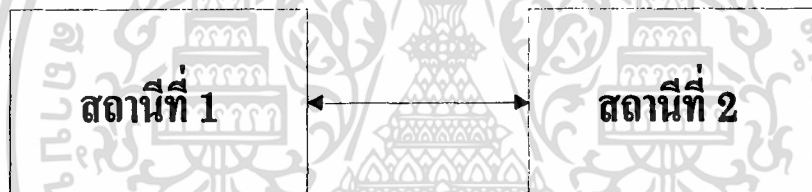
หมายถึง รูปแบบการส่งสัญญาณให้ด้านรับได้ฝ่ายเดียว โดยไม่สามารถโต้ตอบผ่านการติดต่อได้ เช่น การกระจายเสียงของวิทยุหรือสัญญาณโทรทัศน์ เป็นต้น



รูปที่ 2.6 การส่งสัญญาณแบบทิศทางเดียว

2.4.2.2 การส่งสัญญาณแบบสองทิศทางแต่ต่างเวลากัน

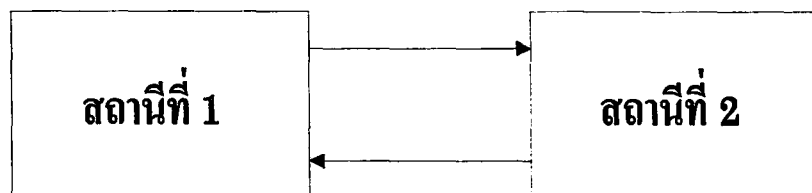
หมายถึง รูปแบบการส่งสัญญาณที่สถานีทั้งสองฝ่ายสามารถรับและส่งสัญญาณระหว่างกันได้ โดยกำหนดว่าต้องมีด้านใดด้านหนึ่งเป็นด้านรับเสมอ ตัวอย่าง เช่น การใช้วิทยุสมัครเล่นในการติดต่อสื่อสาร เป็นต้น



รูปที่ 2.7 การส่งสัญญาณแบบสองทิศทางแต่ต่างเวลากัน

2.4.2.3 การส่งสัญญาณแบบสองทิศทางที่เวลาเดียวกัน

หมายถึง รูปแบบการส่งสัญญาณที่ทั้งด้านส่งและด้านรับ สามารถที่จะส่งสัญญาณในเวลาเดียวกันได้ โดยไม่จำเป็นต้องสลับด้านกันด้วย ตัวอย่างได้แก่ การสนทนาทางโทรศัพท์



รูปที่ 2.8 การส่งสัญญาณแบบสองทิศทางที่เวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ลักษณะของการรับ - ส่งข้อมูลแบบดิจิทัล

โดยทั่วไป หลักใหญ่ของการรับส่งข้อมูลในรูปแบบของสัญญาณดิจิทัลมีลักษณะการส่งข้อมูลอยู่ 2 แบบด้วยกัน คือ

- การส่งแบบอนุกรม
- การส่งแบบขนาน

2.4.3.1 การส่งข้อมูลแบบอนุกรม

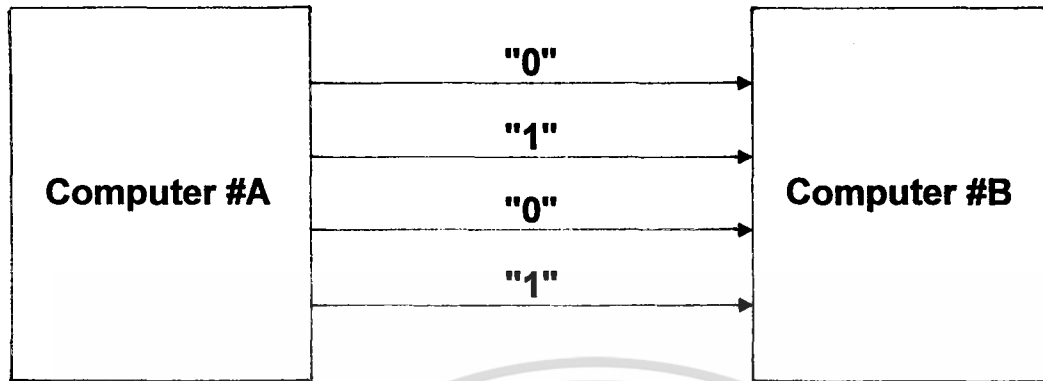
ข้อมูลแต่ละบิตจะถูกส่งเรียงกันออกไปเป็นลำดับ ต่อเนื่องกันทีละบิต เช่น ข้อมูลเป็น 0101 : เลข 1 ทางซ้ายมือสุดซึ่งเป็นบิตนัยสำคัญต่ำสุดก็จะถูกส่งออกไปก่อน ตามด้วยเลข 0 , เลข 1 และ เลข 0 ทางขวามือสุดซึ่งเป็นบิตนัยสำคัญสูงสุดตามลำดับ โดยมีสายส่งอยู่เพียงเส้นเดียวเท่านั้น



รูปที่ 2.9 การส่งข้อมูลแบบอนุกรม

2.4.3.2 การส่งข้อมูลแบบขนาน

ข้อมูลทุกๆบิตจะถูกส่งออกไปพร้อมๆกันในครั้งเดียว เช่น ถ้าข้อมูลเป็น 0101 ทั้งสี่บิตก็จะถูกส่งออกไปพร้อมกันหมด โดยผ่านสายส่งสัญญาณจำนวน 4 เส้น โดยแต่ละบิตจะส่งในสายคนละเส้นกัน



รูปที่ 2.10 การส่งข้อมูลแบบขนาน

ตารางที่ 2.1 ตารางเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานกับอนุกรม

ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานกับการส่งข้อมูลแบบอนุกรม		
	แบบขนาน	แบบอนุกรม
1. ระยะทาง	ปกติจะน้อยกว่า 100 ฟุต	ส่งได้ตั้งแต่ระยะสั้นจนถึงยาวเป็นกิโลเมตร
2. ความเร็ว	อัตราเร็วสูงมากในระยะใกล้	อัตราเร็วช้ากว่าแบบขนานมาก
3. ระดับของสัญญาณ	ใช้ระดับสัญญาณลอจิก "0" กับ "1" ตามลำดับ	มีหลายมาตรฐาน อาจจะใช้ลอจิกแบบ TTL หรือ EIA-232 บางกรณีอาจใช้แบบลูปกระแสได้
4. ความผิดพลาดของสัญญาณ	ถ้าส่งในระยะไกลๆจะผิดพลาดได้ง่าย	การผิดพลาดมีน้อยกว่า
5. ค่าใช้จ่าย	ถ้าส่งในระยะไกลจะสิ้นเปลืองมาก เพราะต้องใช้สายส่งหลายเส้น	สิ้นเปลืองน้อยกว่าหลายเท่า ถึงแม้ว่าจำเป็นต้องใช้อุปกรณ์ในการเปลี่ยนข้อมูลแบบขนานไปเป็นอนุกรม และจากข้อมูลแบบอนุกรมไปเป็นขนานก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การสื่อสารแบบซิงโครนัสและแบบอะซิงโครนัส

2.4.4.1 การสื่อสารแบบซิงโครนัส (Synchronous transmission)

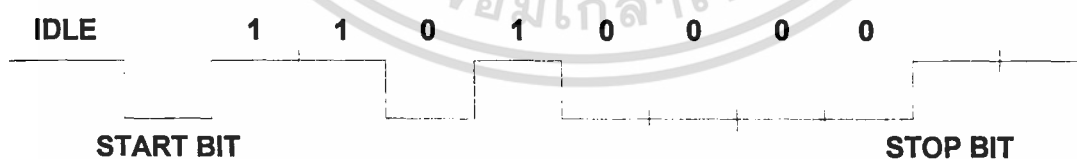
เป็นการรับ-ส่งข้อมูล โดยอาศัยสัญญาณนาฬิกาช่วยในการกำหนดจังหวะการทำงานของด้านรับกับด้านส่งให้มีความสัมพันธ์กัน เมื่อจังหวะเวลาถูกตั้งให้ซิงกับด้านรับแล้ว ข้อมูลจะถูกส่งออกไปบนทางติดต่อในแบบบิตต่อบิตต่อเนื่องกันไปอาศัยช่วงเวลาระหว่างบิตต่อบิตมีค่าเท่ากัน โดยไม่ต้องมีบิตเริ่มส่งหรือบิตจบคอยกำกับ ทำให้ความเร็วในการส่งข้อมูลมีสูง

ข้อเสียของวิธีนี้คือ การที่ต้องมีสัญญาณนาฬิกาขนานไปกับข้อมูล ทำให้ต้องการช่องสื่อสารเพื่อเป็นทางติดต่อเพิ่ม โดยเฉพาะกรณีสื่อสารระยะไกลๆเป็นการยากมากที่จัดหาช่องทางติดต่อแยกต่างหากสำหรับสัญญาณนาฬิกา

นอกจากนี้ทางด้านรับต้องมีวงจรเฟสล็อกเพิ่ม ทำหน้าที่รับข้อมูลจังหวะเวลาจากด้านส่ง และสร้างสัญญาณนาฬิกาขึ้นใหม่ในด้านรับ เพื่อให้เกิดการซิงขึ้น ดังนั้นการส่งแบบซิงโครนัสจึงมีค่าใช้จ่ายสูงกว่า

2.4.4.2 การสื่อสารแบบอะซิงโครนัส(Asynchronous transmission)

เป็นการรับส่งข้อมูลโดยอาศัยบิตเริ่มส่งหรือบิตจบคอยกำกับ โดยไม่จำเป็นต้องมีความสัมพันธ์ระหว่างข้อมูลกับเวลาที่แน่นอน ซึ่งข้อมูลแต่ละตัวสามารถทิ้งช่วงห่างกันเท่าไรก็ได้ขึ้นกับบิตเริ่มส่งหรือบิตจบ โดยวงจรที่ใช้ก็มีความง่ายไม่ต้องยุ่งยากเหมือนกับการสื่อสารแบบซิงโครนัส



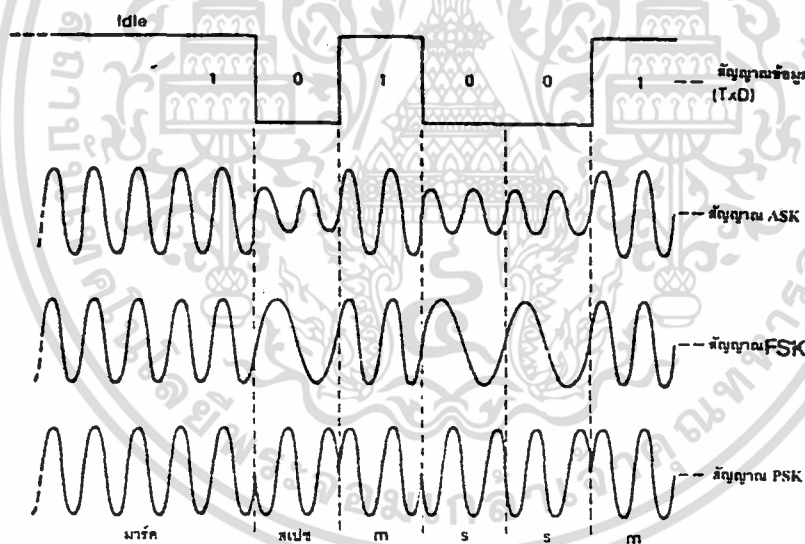
รูปที่ 2.11 การส่งข้อมูลแบบอะซิงโครนัส

ดังนั้นโครงการตัววัดปริมาณทางฟิสิกส์ส่งผ่านทางคลื่นความถี่วิทยุขนาดเล็กจึงเลือกใช้วิธีการส่งแบบอะซิงโครนัส

2.5 การมอดูเลตข้อมูลดิจิทัล

พื้นฐานการมอดูเลตข้อมูลดิจิทัล มีอยู่ด้วยกัน 3 วิธี

- การมอดูเลตดิจิทัลทางขนาด (Amplitude Shift Keying : ASK)
- การมอดูเลตดิจิทัลทางความถี่ (Frequency Shift Keying : FSK)
- การมอดูเลตดิจิทัลทางเฟส (Phase Shift Keying : PSK)



รูปที่ 2.12 การมอดูเลตข้อมูลดิจิทัล

แบบ ASK จัดได้ว่ามีประสิทธิภาพต่ำสุด และใช้ในสายสื่อสารที่ต้องการความเร็วของข้อมูลต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

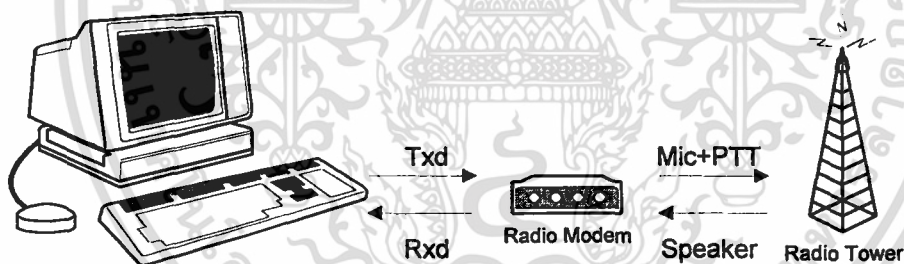
แบบ FSK เป็นแบบที่นิยมใช้ในโมเด็มความเร็วต่ำถึงปานกลาง โดยเป็นการส่งแบบอะซิงโครนัส ที่ใช้ได้ถึง 1200 บิต/วินาที

ส่วนแบบ PSK จะใช้ร่วมกับการส่งแบบซิงโครนัส ในการส่งข้อมูลที่อัตราเร็ว 2400 บิต/วินาที และ 4800 บิต/วินาที ทั้งนี้เพราะเทคนิค PSK ใช้ช่วงความถี่แคบน้อยกว่าแบบ FSK

ในโครงการตัววัดปริมาณทางฟิสิกส์ส่งผ่านทางคลื่นความถี่วิทยุขนาดเล็กเลือกใช้โมเด็มแบบ FSK เนื่องจากสามารถที่จะหาซื้อได้ง่ายและมีราคาถูกกว่าแบบ PSK มาก

2.6 การสื่อสารผ่านระบบแพคเกจเรดิโอ (Packet Radio Communication)

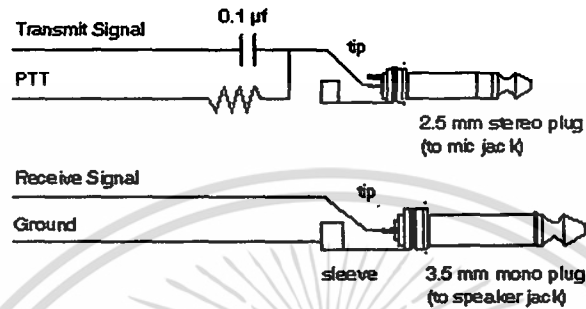
แพคเกจเรดิโอ คือ การรับ-ส่งข้อมูลผ่านทางวิทยุสื่อสาร โดยการนำเอาข้อมูลข่าวสารมารวมเป็นแพคเกจแล้วทำการส่งข้อมูลซึ่งมีการตรวจสอบความผิดพลาดของข้อมูลด้วย ระบบแพคเกจเรดิโอประกอบไปด้วยเครื่องไมโครคอมพิวเตอร์ส่วนบุคคล, เรดิโอโมเด็ม และวิทยุสื่อสาร แสดงดังรูปที่ 2.13



รูปที่ 2.13 ระบบแพคเกจเรดิโอ

เรดิโอโมเด็มจะทำหน้าที่คล้ายกับโมเด็มทั่วไป โดยทำหน้าที่เป็นตัวมอดูเลตสัญญาณข้อมูลที่ถูกส่งออกมาจากคอมพิวเตอร์ ก่อนจะส่งข้อมูลที่ได้รับการมอดูเลตแล้วผ่านทางวิทยุสื่อสารโดยเชื่อมต่อผ่านทางช่องเสียบ mic และ PTT (Push-to-talk) ซึ่งเป็นวงจรที่วิทยุสื่อสารใช้สำหรับกำหนดการส่งออกอากาศ โดยส่วนใหญ่แล้ววิทยุสื่อสารชนิดมือถือมักจะใช้วิธีกำหนดการส่งออกอากาศโดยการตรวจค่าอิมพีแดนซ์ระหว่างขากลางและขาราวนของแจ๊คเสียบไมโครโฟนว่ามีค่าตามที่กำหนดหรือไม่ และเพื่อ

ป้องกันไม่ให้กระแสไฟตรงจากวงจรไปรบกวนจุดไบแอสในวิทยุสื่อสาร จึงมีการต่อตัวเก็บประจุขนาด 0.1 ไมโครฟารัดกั้นสัญญาณไฟตรงเอาไว้



รูปที่ 2.14 แสดงขาแจ๊คสำหรับวิทยุสื่อสารชนิดมือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

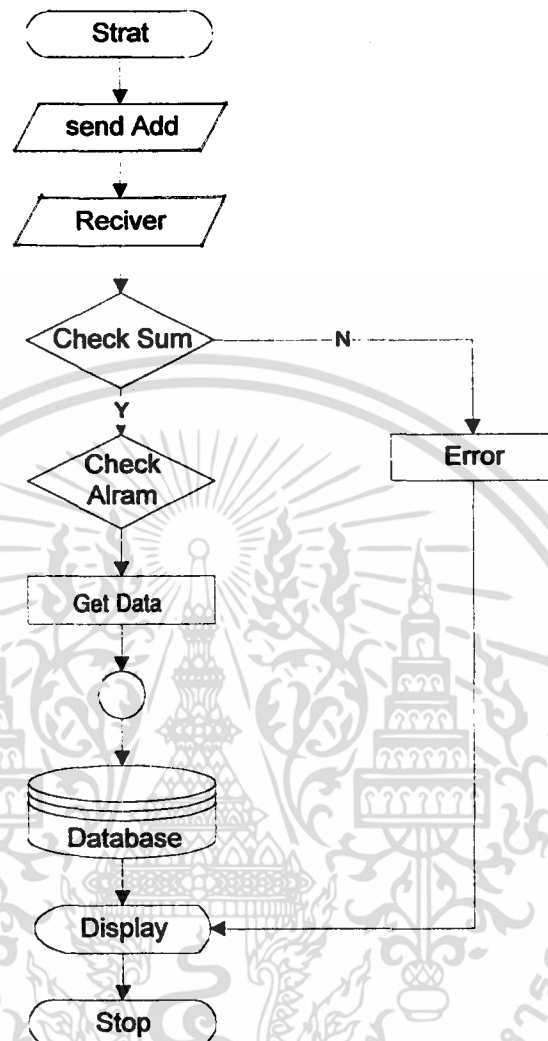
บทที่ 3

หลักการทํางานในส่วนของซอฟต์แวร์

หลักการ โดยรวมของซอฟต์แวร์สามารถแบ่งออกเป็น 2 ส่วนใหญ่

1. ระบบเฝ้าตรวจระวัง (Monitoring System)
2. ระบบฐานข้อมูล (Database System)

ระบบการทํางานของซอฟต์แวร์ ทั้ง 2 ส่วนจะทํางานสอดคล้องกัน โดยมีเมนูหลักเป็นตัวควบคุมการทํางานซึ่งสามารถเลือกทํางานในระบบใดระบบหนึ่งได้ และในขณะที่ระบบเฝ้าตรวจระวังทํางานก็จะมีกรเก็บข้อมูลเข้าในระบบฐานข้อมูล จึงสามารถกล่าวได้ว่าระบบทั้ง 2 ระบบจะทํางานไปพร้อม ๆ กัน และในส่วนของระบบเฝ้าตรวจระวังจะมีระบบควบคุมการทํางานของมอเตอร์รวมอยู่ด้วย โดยส่วนควบคุมมอเตอร์และส่วนเฝ้าตรวจระวังจะทํางานคนละเวลา คือเมื่อทํางานในส่วนเฝ้าตรวจระวัง ส่วนของระบบควบคุมมอเตอร์จะไม่สามารถทํางานและถ้าทํางานในส่วนควบคุมมอเตอร์ ส่วนของระบบเฝ้าตรวจระวังก็ไม่สามารถทํางานได้เช่นกัน เนื่องจากทั้ง 2 ระบบใช้พอร์ต อนุกรมร่วมกัน ซอฟต์แวร์นี้ถูกออกแบบมาเพื่อใช้สำหรับระบบปฏิบัติการวินโดวส์ (Windows) และเป็นซอฟต์แวร์ที่ใช้สำหรับการติดต่อสื่อสารแบบอนุกรม ระหว่าง คอมพิวเตอร์ กับ โมเด็มที่ใช้คลื่นวิทยุสื่อสาร (RF Modem) ด้วยอัตราเร็ว 9600 บิต ต่อ วินาที (Baud Rate =9600bps) และสามารถนำไปใช้กับตัววัดปริมาณทางฟิสิกส์ (Analog Sender) ได้พร้อมกัน 3 ตัว และตัวปริมาณดังกล่าวนี้จะต้องมี แอดเดรส ที่ต่างกัน โดยระบบจะมี แอดเดรส ได้ตั้งแต่ “ @ 0 ” ถึง “ @ 7 ” รวม 8 ตัว และมีระบบรักษาความปลอดภัยในการเปลี่ยน แอดเดรส โดยต้องมีรหัสผ่านถึงจะสามารถเปลี่ยนแอดเดรส ได้

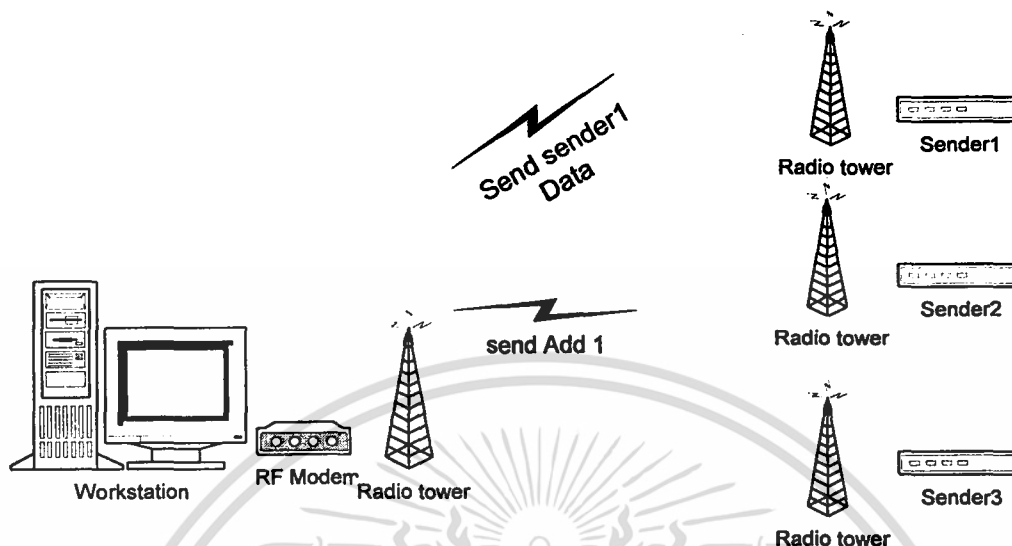


รูปที่ 3.1 แผนภาพแสดงขั้นตอนการทำงานของ โปรแกรม

3.1 ระบบเฝ้าตรวจระวัง (Monitoring System)

หลักการการทำงานของระบบเฝ้าตรวจระวังคือ คอมพิวเตอร์จะทำการส่ง แอคเตรสของตัววัดปริมาณทางฟิสิกส์ที่ต้องการจะทราบข้อมูลออกไป เมื่อตัววัดปริมาณทางฟิสิกส์ทั้งหมดได้รับ แอคเตรส นั้นก็จะทำการตรวจสอบ ถ้า แอคเตรส ที่ส่งมาตรงกับแอคเตรส ของตัววัดปริมาณทางฟิสิกส์ตัวใด ตัววัดปริมาณทางฟิสิกส์ตัวนั้นก็จะส่งข้อมูลในเวลานั้น กลับมาที่คอมพิวเตอร์และ ซอฟแวร์จะนำข้อมูลไปทำการประมวลผลและแสดงผลในส่วนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการทำงานในระบบเฟ้าตรวจระวัง

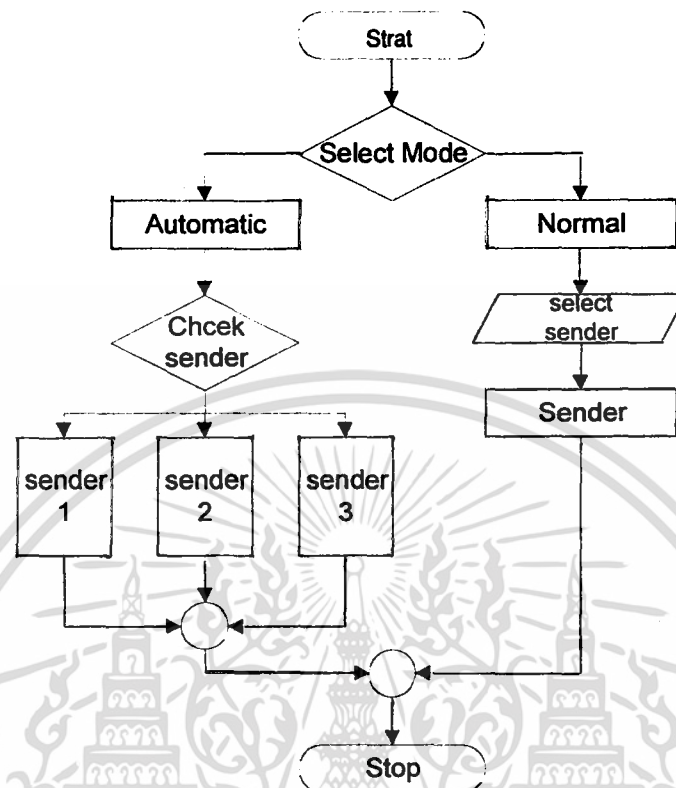
ขั้นตอนในการทำงานสามารถแบ่งออกเป็นส่วน ๆ ตามลำดับการทำงานดังนี้

3.1.1 การตรวจสอบระบบ ฮาร์ดแวร์ โดยส่งค่าที่ว่าง (space) และรอรับค่าที่ส่งจากตัววัดปริมาณทางฟิสิกส์ ถ้าค่าที่รับเข้ามาเป็น “ * ” แสดงว่าติดต่อกลับตัววัดปริมาณได้แล้ว แต่ถ้าค่าที่รับเข้ามาเป็นไม่เท่ากับ “ * ” แสดงว่าฮาร์ดแวร์มีข้อผิดพลาดซึ่งจะเป็นผลให้การทำงานในส่วนอื่นไม่สามารถทำงานได้

3.1.2 การส่งข้อมูลเพื่อควบคุมการทำงาน เป็นการส่งแอดเดรส ของตัววัดปริมาณทางฟิสิกส์ที่ต้องการทราบค่าออกไป ซึ่งการส่งข้อมูลจะแบ่งเป็น 2 แบบ คือ

- การส่งข้อมูลแบบอัตโนมัติ เป็นการส่งข้อมูลตามเวลาที่กำหนดคือตั้งเวลาการส่ง แอดเดรส ของตัววัดปริมาณทางฟิสิกส์ ยกตัวอย่างเช่น ตั้งเวลาให้มีการส่งแอดเดรส ของตัววัดปริมาณ ทางฟิสิกส์ ตัวที่ 1 และ 3 ทุก ๆ 20 วินาที จะทำให้มีข้อมูลของตัววัดปริมาณทางฟิสิกส์ตัวที่ 1 และ 3 เข้ามาทุก ๆ 20 วินาที
- การส่งข้อมูลแบบปกติ เป็นการส่งโดยกำหนดตัววัดปริมาณทางฟิสิกส์ที่จะส่ง และจะทำการส่งเพียงครั้งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงการทำงานในส่วนการส่งข้อมูลเพื่อควบคุมตัววัดปริมาณทางฟิสิกส์

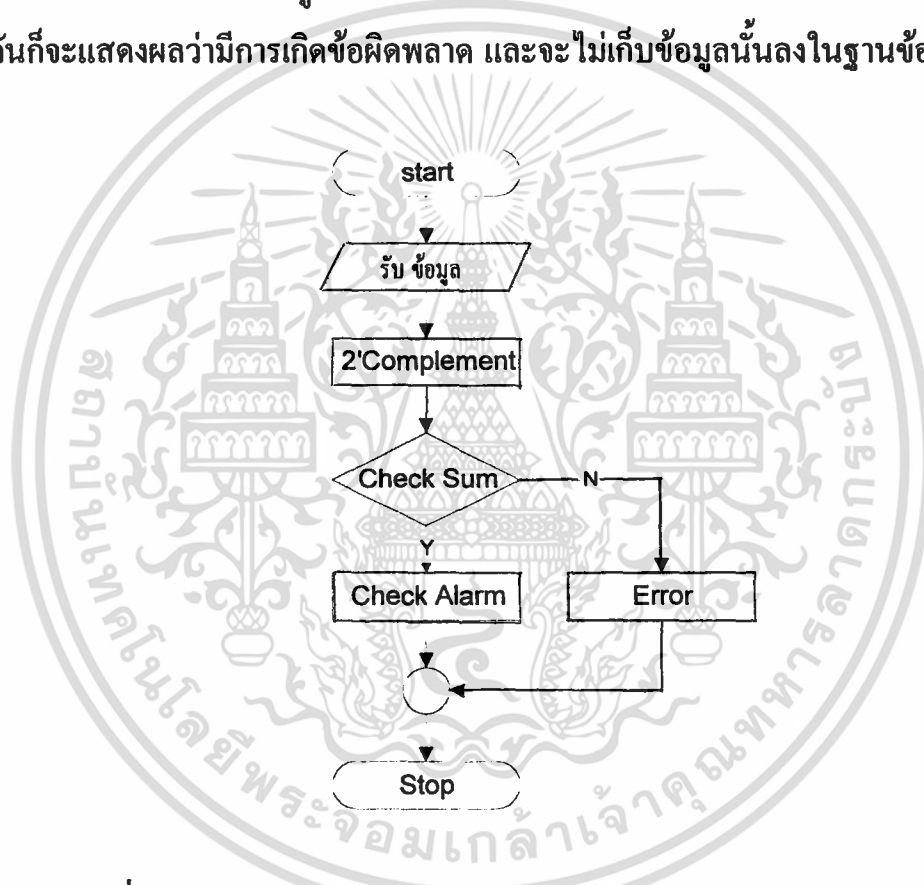
3.1.3 การรับข้อมูลจากตัววัดปริมาณทางฟิสิกส์ หลังจากส่ง แอดเดรสตัววัดปริมาณทางฟิสิกส์ที่ต้องการทราบค่าออกไปแล้ว ซอฟต์แวร์จะมาทำงานในส่วนรับข้อมูล ข้อมูลที่รับเข้ามาจะถูกเก็บไว้ที่ บัฟเฟอร์ของคอมพิวเตอร์ เพื่อมิให้ข้อมูลผิดพลาดจะต้องลบข้อมูลใน บัฟเฟอร์ ให้หมดก่อนรับข้อมูล เมื่อข้อมูลทั้งหมดเข้ามาอยู่ใน บัฟเฟอร์ แล้ว จะทำการถ่ายข้อมูลจาก บัฟเฟอร์ ไปใส่ในตัวแปร (string) ที่กำหนดไว้จากนั้นนำข้อมูลในตัวแปรนั้นมาแบ่งออกเป็น 2 ส่วนคือ ส่วนของ ข้อมูลทั้ง 8 ช่องสัญญาณ และส่วนของตรวจสอบผลรวม (Check sum) แล้วนำข้อมูลในแต่ละส่วนเก็บในตัวแปรข้อมูล และตัวแปรตรวจสอบตามลำดับ

- ข้อมูลทั้ง 8 ช่องสัญญาณ	ตรวจสอบผลรวม
---------------------------	--------------

รูปที่ 3.4 แสดงลักษณะของข้อมูลที่ได้รับจากตัววัดปริมาณทางฟิสิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การตรวจสอบผลรวม ส่วนของการตรวจสอบผลรวม (Check sum) ที่ส่งมาจากตัววัดปริมาณ ทางฟิสิกส์ทำการคำนวณค่าโดยนำข้อมูลทั้ง 8 ช่องสัญญาณบวกกันแล้วทำทุกคอมพลิเมนต์ เพราะฉะนั้น เมื่อรับข้อมูลเข้ามาก็จำเป็นต้องนำข้อมูลในตัวแปรข้อมูลมาบวกกัน แต่จะต้องแปลงค่าข้อมูลในอยู่ในรูปเลขฐาน 16 ก่อน จากนั้นทำทุกคอมพลิเมนต์ แล้วนำค่ามา ตรวจสอบกับข้อมูลในตัวแปรตรวจสอบ(Check sum) ถ้าตรวจสอบแล้วตรงกันก็จะนำข้อมูลไปตรวจสอบในส่วนต่อไป แต่ถ้าตรวจสอบแล้วค่าไม่ตรงกันก็จะแสดงผลว่ามีการเกิดข้อผิดพลาด และจะไม่เก็บข้อมูลนั้นลงในฐานข้อมูล



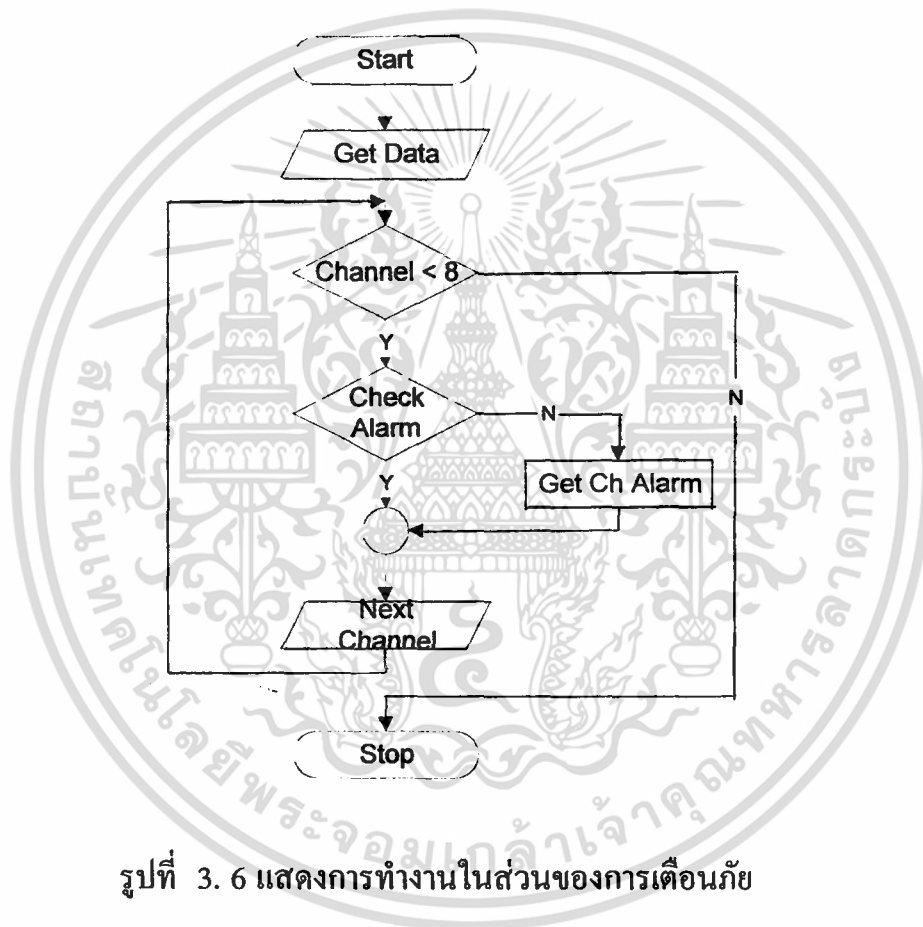
รูปที่ 3. 5 แสดงการทำงานในส่วนของการตรวจสอบข้อมูล

3.1.5 การตรวจระบบเตือนภัย เป็นการตรวจสอบข้อมูลที่รับเข้ามาทั้ง 8 ช่องสัญญาณ โดยจะมีการตั้งค่าสูงสุดและต่ำสุดในแต่ละช่องสัญญาณไว้ แล้วนำข้อมูลในตัวแปรข้อมูลมาแบ่งเป็นข้อมูลในแต่ละช่องสัญญาณแล้วนำข้อมูลในช่องสัญญาณนั้น ๆ ไปเปรียบเทียบกับค่าที่กำหนดไว้ และถ้าค่าของข้อมูลใดมากกว่า หรือ น้อยกว่าค่าที่

กำหนดไว้ก็จะมีการแสดงผลการเตือนภัย โดยจะแสดงผล เวลาและช่องสัญญาณที่เกิด ความผิดพลาด

ยกตัวอย่างเช่น

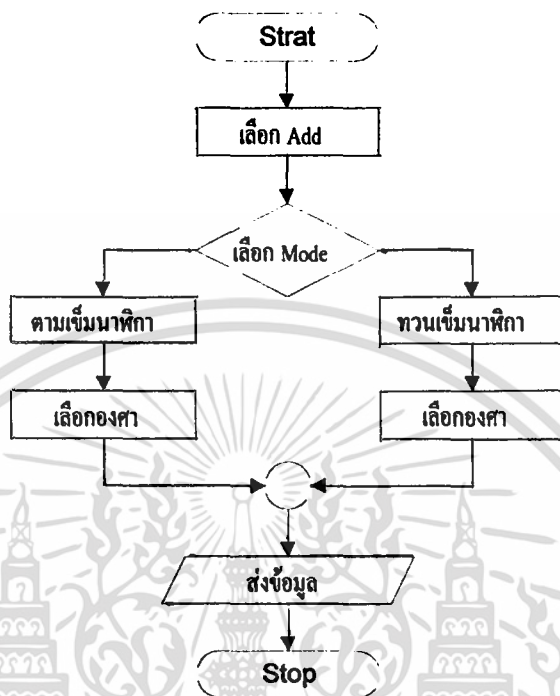
12:02:14 Alarm On Ch 1 3 7 คือมีการเตือนภัยที่เวลา 12:02:14 ช่อง สัญญาณ ที่ 1 , 3 และ 7



รูปที่ 3. 6 แสดงการทำงานในส่วนของการเตือนภัย

3.1.6 ระบบควบคุมการทำงานของมอเตอร์ โดยในตัววัดปริมาณทางฟิสิกส์แต่ละตัวจะมีมอเตอร์อยู่ด้วย เราสามารถที่จะควบคุมการทำงานของมอเตอร์ได้ 2 ลักษณะ คือ หมุนตามเข็มนาฬิกา และ ทวนเข็มนาฬิกา โดยจะหมุนตามค่าที่ตั้งไว้ คือ 45 องศา , 90 องศา , 180 องศา 360 องศา ในการส่งค่าเพื่อควบคุมมอเตอร์นั้นจำเป็นต้องระบุ แอ็คเตส ของตัววัดปริมาณทางฟิสิกส์ที่จะให้มอเตอร์ทำงาน จากนั้นส่งคำสั่งว่าจะใน หมุนตามเข็มนาฬิกา หรือทวนเข็มนาฬิกา และหมุนไปที่องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 7 แสดงการทำงานในระบบควบคุมการทำงานของมอเตอร์

2. ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูลที่สร้างขึ้นมานี้เป็นการจัดการกับข้อมูลโดยอาศัยตัวแปรแบบ ออบเจ็ค เรคอร์ดเซ็ท ซึ่งเป็นที่วิธีที่มีประสิทธิภาพ และมีความยืดหยุ่นต่อการใช้งานใน ลักษณะต่าง ๆ และ ออบเจ็ค เรคอร์ดเซ็ท จะแบ่งได้อีก 3 แบบ ซึ่งแบบที่ได้นำมาใช้คือ ออบเจ็คตาราง ซึ่งจะต้องกำหนดค่าต่างๆ ตั้งแต่ตอนสร้างฐานข้อมูล ฐานข้อมูลที่สร้างขึ้นนี้ประกอบด้วย 3 ตาราง เป็นตารางของตัววัดปริมาณทางฟิสิกส์แต่ละตัว ในแต่ละ ตารางประกอบด้วย 12 ฟิลด์ (Field) และการเก็บข้อมูลแต่ละครั้งจะถือเป็น 1 เรคอร์ด

ระบบฐานข้อมูลจะสามารถแสดงข้อมูลได้ โดยการเลือกตัววัดปริมาณทาง ฟิสิกส์ที่ต้องทราบค่าแล้วใส่ค่า เดือน , ปี ที่ต้องการดูข้อมูล จากนั้นจะมีวันที่ได้ทำการ ส้ารวจในเดือน , ปีนั้นขึ้นมาให้เลือก เมื่อเลือกวันที่จะมีข้อมูลที่ทำกรส้ารวจในวันนั้น แสดงผลพร้อมกราฟในแต่ละช่องสัญญาณ

3. การเขียนโปรแกรมติดต่อสื่อสารผ่านพอร์ตอนุกรม (Communications Control) ด้วย วิชาการเบสิก

ฟังก์ชันในการเขียนโปรแกรมติดต่อสื่อสารกับพอร์ตอนุกรม จะมีอยู่ใน วิชาการเบสิกชุดโปรแกรมเมอร์ (Visual Basic Professional Editor)หรือเวอร์ชัน ที่สูงกว่า โปรแกรมที่ติดต่อกับพอร์ตอนุกรมส่วนใหญ่จะใช้ในการแสดงผลหรือแสดงการทำงานของฮาร์ดแวร์ซึ่งจะทำให้สามารถเข้าใจของระบบการทำงานของฮาร์ดแวร์ได้ง่ายขึ้น โดยมีคำสั่งต่าง ๆที่ใช้ในการควบคุมหลักดังนี้

1.Commport เป็นการตั้งค่าพอร์ตอนุกรมว่าจะตั้งเป็นพอร์ตใดโดยการใส่เลขของพอร์ตอนุกรมที่จะทำการติดต่อ

Mscomm.CommPort = 2 ติดต่อกับพอร์ต 2

2.Settings เป็นการตั้งความเร็ว และ คำต่าง ๆ ที่ใช้ในการส่งข้อมูลผ่านโมเด็ม ของพอร์ทอนุกรมที่ทำการติดต่อ

Mscomm.Settings = “ 9600,N,8,1 ”

3.PortOpen เปิด หรือ ปิด พอร์ตอนุกรมที่ได้ตั้งค่าต่าง ๆ ไว้

Mscomm.PortOpen = True เปิดพอร์ต

Mscomm.PortOpen = False ปิดพอร์ต

4. Input นำค่าข้อมูลที่อยู่ใน บัฟเฟอร์ตัวรับข้อมูล มาเก็บ ใน string ที่กำหนด คือข้อมูลที่ส่งเข้ามาที่คอมพิวเตอร์ที่ถูกเก็บไว้ บัฟเฟอร์ตัวรับข้อมูล

MsgInput = Mscomm.Input ค่าที่ส่งเข้ามาจะถูกเก็บใน String MsgInput

5. Output ส่งข้อมูลออกโดยเขียนข้อมูลที่ บัฟเฟอร์ตัวส่งข้อมูล

Mscomm.Output = “ Hello” ส่งข้อมูล “ Hello”

6. InBufferCount เป็นการนำข้อมูลใน บัฟเฟอร์ตัวรับข้อมูล เพื่อให้ได้จำนวนของข้อมูลที่ต้องการ และสามารถตรวจสอบข้อมูลที่ได้รับเข้ามาได้

Do

Do Event

Loop Until Mscomm.BufferCount >= 20

Msg = Mscomm.Input

โปรแกรมจะวนลูปจะกว่า ข้อมูลในบัฟเฟอร์ตัวรับข้อมูล จะมีค่า มากกว่าเท่ากับ 20 ตัว เป็นการตรวจสอบว่าได้ข้อมูลครบตามที่กำหนดหรือไม่

7.OutBufferCount ใช้ในการเคลียร์บัฟเฟอร์ตัวส่งข้อมูล

Mscomm.OutBufferCount = 0

ใช้ในการเคลียร์บัฟเฟอร์ตัวส่งข้อมูล หลังจากทีส่งข้อมูลใด ๆ ออกไป เพื่อไม่ให้เกิดข้อผิดพลาดในการส่งข้อมูลครั้งต่อไป

โปรแกรมตัวอย่าง ในการติดต่อกับพอร์ตอนุกรม

Sub Form_Load ()

Dim Instring as String

Mscomm.CommPort = 2 ' Com 2

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

Mscomm.PortOpen = True

Mscomm.OutPort = "AT" + Chr\$(13)

Mscomm.OutBufferCount = 0

Do

Do Events ()

Loop Until Mscomm.InBufferCount >= 2

Instring\$ = Mscomm.Input

Mscomm.PortOpen = False

End Sub

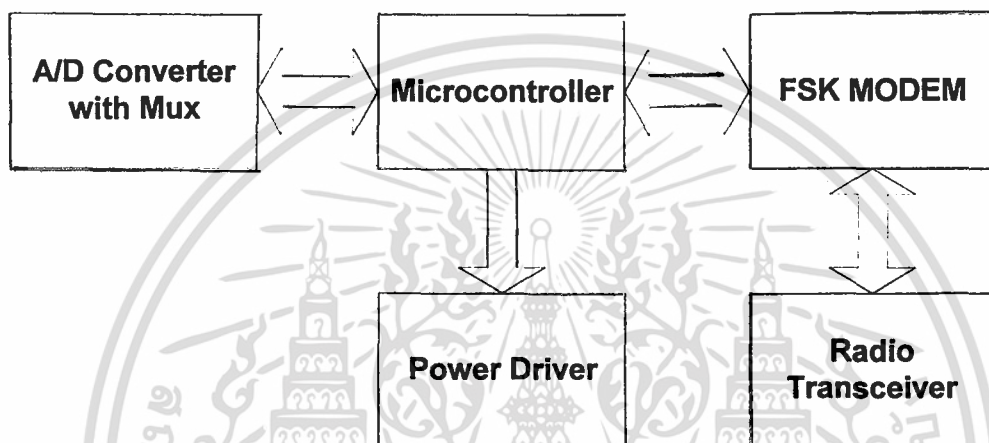
ในการเขียนโปรแกรมในการทำงานจริงควรเขียนในลักษณะของโปรแกรมย่อย
แล้วนำมาเรียกใช้เมื่อต้องการ เช่นมีโปรแกรมย่อยในการรับข้อมูล และส่งข้อมูล แต่
ส่วนการกำหนดคุณสมบัติของฟอร์ตอนุกรมควรเขียนไว้ในส่วนเริ่มต้นโปรแกรม
เพราะถ้ามีการกำหนดซ้ำซ้อนจะทำให้เกิดข้อผิดพลาดได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การดำเนินการวิจัย

4.1 ตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุขนาดเล็ก



รูปที่ 4.1 โครงสร้างการทำงานของตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุ

จากรูปที่ 4.1 แสดงไดอะแกรมการทำงานของตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุ ซึ่งประกอบไปด้วย

1. วิทยุสื่อสาร ทำหน้าที่ในการส่งและรับข้อมูลในรูปแบบของสัญญาณที่ได้รับการมอดูเลตแล้ว
2. โมเด็ม ทำหน้าที่ในการมอดูเลตสัญญาณข้อมูลดิจิทัลกับความถี่พาห์ เพื่อใช้ส่งข้อมูลผ่านไปยังวิทยุสื่อสาร อีกทั้งยังทำการดีมอดูเลตสัญญาณให้อยู่ในรูปแบบของสัญญาณดิจิทัล
3. ไมโครคอนโทรลเลอร์ ทำหน้าที่ในการควบคุมการทำงานของทุกส่วนในระบบ ไม่ว่าจะเป็นการรับ-ส่งข้อมูลผ่านทางโมเด็ม, การสวิตช์เพื่อให้วิทยุทำการส่งสัญญาณออกอากาศหรือรับข้อมูล, การควบคุมการทำงานของตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลและมัลติเพล็กซ์เซอร์, และการควบคุมอุปกรณ์กำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตัวแปลงสัญญาณอนาลอกเป็นดิจิทัล ทำหน้าที่ในการแปลงสัญญาณอนาลอกซึ่งอยู่ในรูปของแรงดันไฟฟ้าที่ได้จากตัวเซนเซอร์ ไปเป็นข้อมูลดิจิทัลเพื่อส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์ประมวลผลอีกทีหนึ่ง โดยมีมัลติเพล็กซ์เซอร์ใช้ในการเลือกช่องที่ใช้ในการวัดเซนเซอร์แต่ละชนิด

5. ตัวขับอุปกรณ์กำลังสูง ทำหน้าที่ในการเชื่อมต่อระหว่างอุปกรณ์กำลังกับไมโครคอนโทรลเลอร์

4.1.1 ไมโครคอนโทรลเลอร์ AT89C2051

AT89C2051 เป็นไมโครคอนโทรลเลอร์ 8 บิต ขนาด 20 ขา สร้างด้วยเทคโนโลยี Atmel's high density nonvolatile memory technology โดยมีหน่วยความจำโปรแกรมภายในขนาด 2 กิโลไบต์ แบบแฟลช หรือที่เรียกกันว่า Programmable and Erasable Read Only Memory (PEROM) , หน่วยความจำข้อมูลภายในขนาด 128 ไบต์, มีอินพุตและเอาต์พุตจำนวน 15 บิต, ตัวนับและตัวตั้งเวลา ขนาด 16 บิต 2 ตัว, แหล่งอินเทอร์รัพต์ 5 แหล่ง, การสื่อสารอนุกรมแบบ Full duplex, อนาลอกคอมพารเตอร์ และวงจรมอนิเตอร์ภายในชิป

เนื่องจากชิป AT89C2051 สร้างด้วยเทคโนโลยี CMOS จึงสามารถเขียนโปรแกรมควบคุมการใช้พลังงานได้ 2 โหมด คือ

- โหมด Idle
- โหมด Power Down

เราสามารถที่จะใช้โหมด Idle ได้โดยการไปเซ็ตบิต Idle ที่รีจิสเตอร์ PCON ซึ่งจะหยุดการจ่ายสัญญาณนาฬิกาให้กับซีพียู แต่ยังคงจ่ายสัญญาณนาฬิกาให้กับแหล่งอินเทอร์รัพต์ทั้ง 5 แหล่งอยู่ ได้แก่ สัญญาณอินเทอร์รัพต์จากภายนอก, ตัวนับและตัวตั้งเวลา และ การสื่อสารอนุกรม ดังนั้นซีพียูจะทำงานเมื่อได้รับสัญญาณอินเทอร์รัพต์จากทั้ง 5 แหล่งที่กล่าวไปแล้ว

ส่วนในโหมด Power Down จะใช้ได้โดยการเซ็ตบิต PD ที่รีจิสเตอร์ PCON ทำให้หยุดการจ่ายสัญญาณนาฬิกาให้กับทุกส่วนของชิป แต่ไฟเลี้ยงที่ลดลงเหลือเพียง

2 โวลท์ยังเพียงพอที่จะรักษาข้อมูลภายในหน่วยความจำข้อมูลเอาไว้ได้ การจะหยุดสถานะ Power Down สามารถทำได้วิธีเดียวคือการรีเซ็ตที่ฮาร์ดแวร์

พอร์ต P1 และ P3 ใช้เป็นได้ทั้งอินพุตและเอาต์พุต ซึ่งสามารถขับกระแสซิงค์ได้ถึง 20 มิลลิแอมป์ จึงสามารถที่จะขับ LED ได้โดยตรง ที่ขาพอร์ตทุกขาจะได้รับการพูลอัพภายในไว้แล้วยกเว้นแต่ P1.0 กับ P1.1 ซึ่งถูกใช้เป็นอนาล็อกคอมพาราเตอร์ โดยมี P3.6 เป็นเอาต์พุตของอนาล็อกคอมพาราเตอร์นั้น (P3.6 ไม่มีการเชื่อมต่อทางขาออกสู่วงจรภายนอก)

4.1.1.1 คุณสมบัติพื้นฐานของ AT89C2051

AT89C2051 เป็นชิปไมโครคอนโทรลเลอร์ขนาดจิ๋ว ผลิตโดย ATMEL ชุดคำสั่งและสถาปัตยกรรมภายใน จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS - 51 ซึ่งผลิตโดยอินเทล มีหน่วยความจำภายในเป็น PEROM หรือเรียกว่า Flash memory ขนาด 2 KB สามารถเขียนและลบใหม่ได้ไม่น้อยกว่า 1000 ครั้ง โปรแกรมที่บันทึกไว้ภายในชิปสามารถเก็บไว้ได้นานถึง 10 ปี จุดเด่นของชิป AT89C2051 มีดังต่อไปนี้

1. สามารถใช้แทนไมโครคอนโทรลเลอร์ ตระกูล MCS - 51
2. หน่วยความจำโปรแกรมเป็น Flash memory ขนาด 2 KB
3. ใช้ไฟเลี้ยงได้ตั้งแต่ 2.7 - 6 V
4. ออสซิลเลเตอร์สามารถป้อนได้สองระดับ
5. สามารถล๊อคโปรแกรมได้สองระดับ
6. หน่วยความจำข้อมูลบนชิปขนาด 128 ไบท์
7. มีอินพุต / เอาต์พุตพอร์ต 15 บิต
8. มีตัวนับและตั้งเวลาขนาด 16 บิต 2 ตัว
9. แหล่งอินเทอร์พท์มี 5 แหล่ง
10. พอร์ตอนุกรมแบบ UART โปรแกรมความเร็วในการส่งข้อมูลได้
11. เอาต์พุตพอร์ตสามารถขับ LED ได้โดยตรง ด้วยกระแส sink 20 mA
12. มีอนาล็อกคอมพาราเตอร์บนชิป
13. มีโหมด Idle และ Power down

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 ตัวถังและขาสัญญาณของชิป AT89C2051

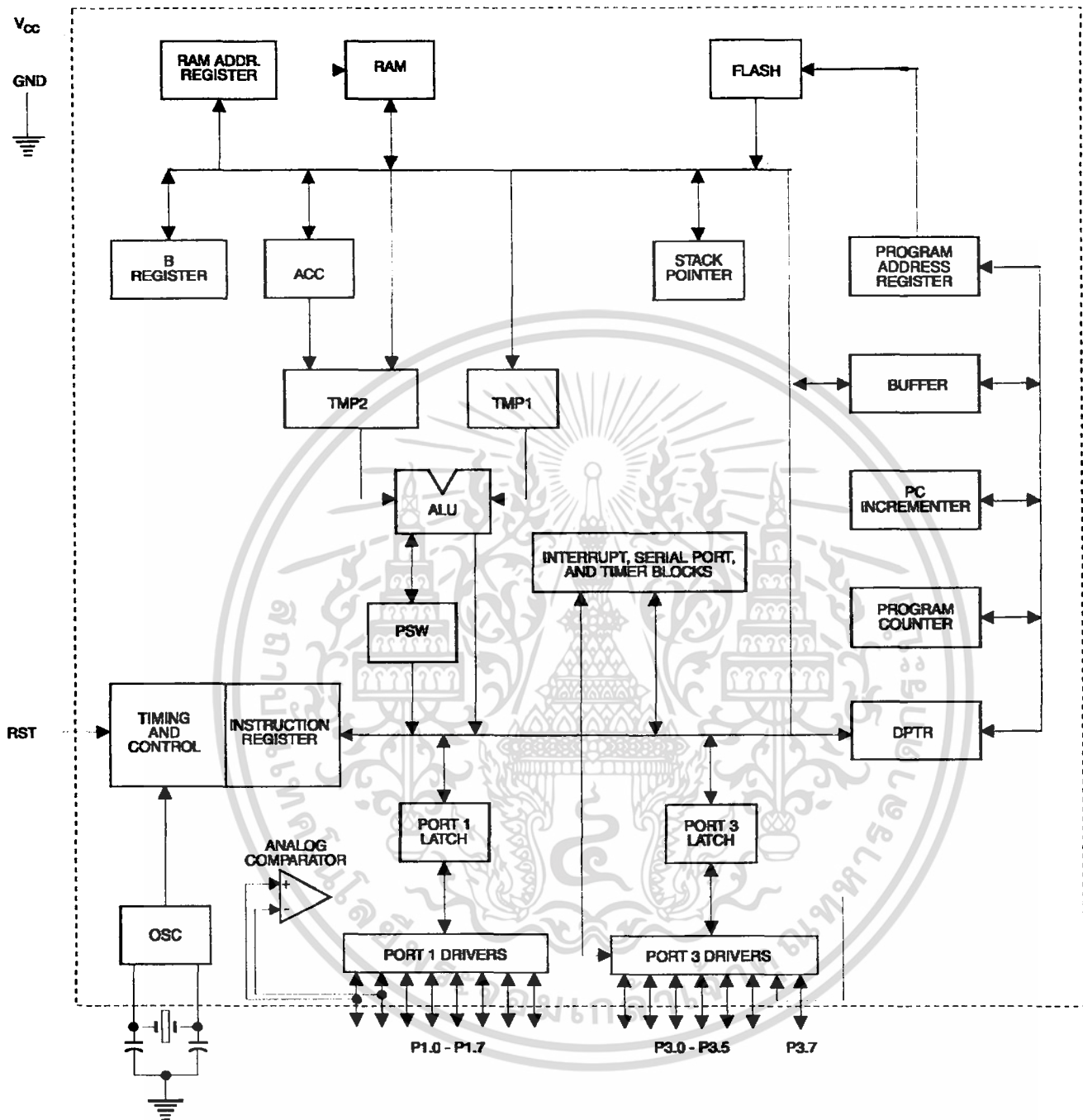
ลักษณะตัวถังเป็นแบบ PDIP (Plastic Dual - Inline - Package) ขนาด 20 ขา แสดงดังรูปที่ 4.2 หากเราจำได้กับระบบ 8051 จะทำความเข้าใจได้ไม่ยาก ขาที่แตกต่างกันสำคัญที่เราสังเกตได้ชัดเจนคือ I/O พอร์ต มีเพียงพอร์ต 1 จำนวน 8 บิต ระบุด้วย P1.0 - P1.7 ข้อสังเกตจะพบว่า P1.0 และ P1.1 มีฟังก์ชันเสริมคือ AIN0 และ AIN1 เป็นอนาล็อกอินพุทของวงจรคอมพิวเตอร์ รูปที่ 4.3 แสดงผังวงจรภายใน

Pin Configuration

PDIP/SOIC			
RST	1	20	VCC
(RXD) P3.0	2	19	P1.7
(TXD) P3.1	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
(INT0) P3.2	6	15	P1.3
(INT1) P3.3	7	14	P1.2
(T0) P3.4	8	13	P1.1 (AIN1)
(T1) P3.5	9	12	P1.0 (AIN0)
GND	10	11	P3.7

รูปที่ 4.2 แสดงขาของ AT89C2051

จะเห็นว่าเราสามารถใช้งานเปรียบเทียบสัปดาห์ก่อนเข้ามาที่ขา P1.0 และ P1.1 ได้โดยเอาที่พุทจะต่อกับ P3.6 ซึ่งสามารถอ่านค่าด้วยโปรแกรม ดังนั้นพอร์ต P3.6 จึงมีได้ต่อกับขาภายนอก พอร์ต 3 จึงมีเพียง P3.1 - P3.5 และ P3.7 สำหรับฟังก์ชันเพื่อเลือกยังคงใช้งานได้เหมือนกับชิปตระกูล 8051 ของอินเทล รายละเอียดสัญญาณทางไฟฟ้าทั้งตรงและไฟสลับของขาสัญญาณแต่ละขา ดูได้ในภาคผนวก



รูปที่ 4.3 แสดงขาสัญญาณและไดอะแกรมวงจรรภายในของ 89C2051

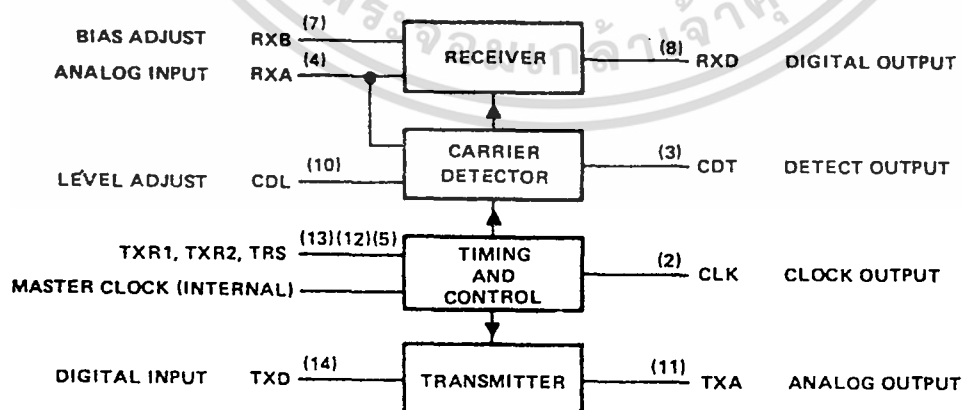
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 โมเด็มชิปเดี่ยว TCM3105 (Single Chip Modem)

TCM3105 เป็นโมเด็มชิปเดี่ยว แบบอะซิงโครนัส FSK โดยสร้างขึ้นจากเทคโนโลยี Silicon Gate CMOS technology ร่วมกันกับสถาปัตยกรรม switched capacitor ซึ่งคอมแพคทีเบิ้ลกับทั้งมาตรฐาน CCITT V.23 และ BELL 202 โดยมีรายละเอียดของแต่ละขาที่จำเป็นดังนี้

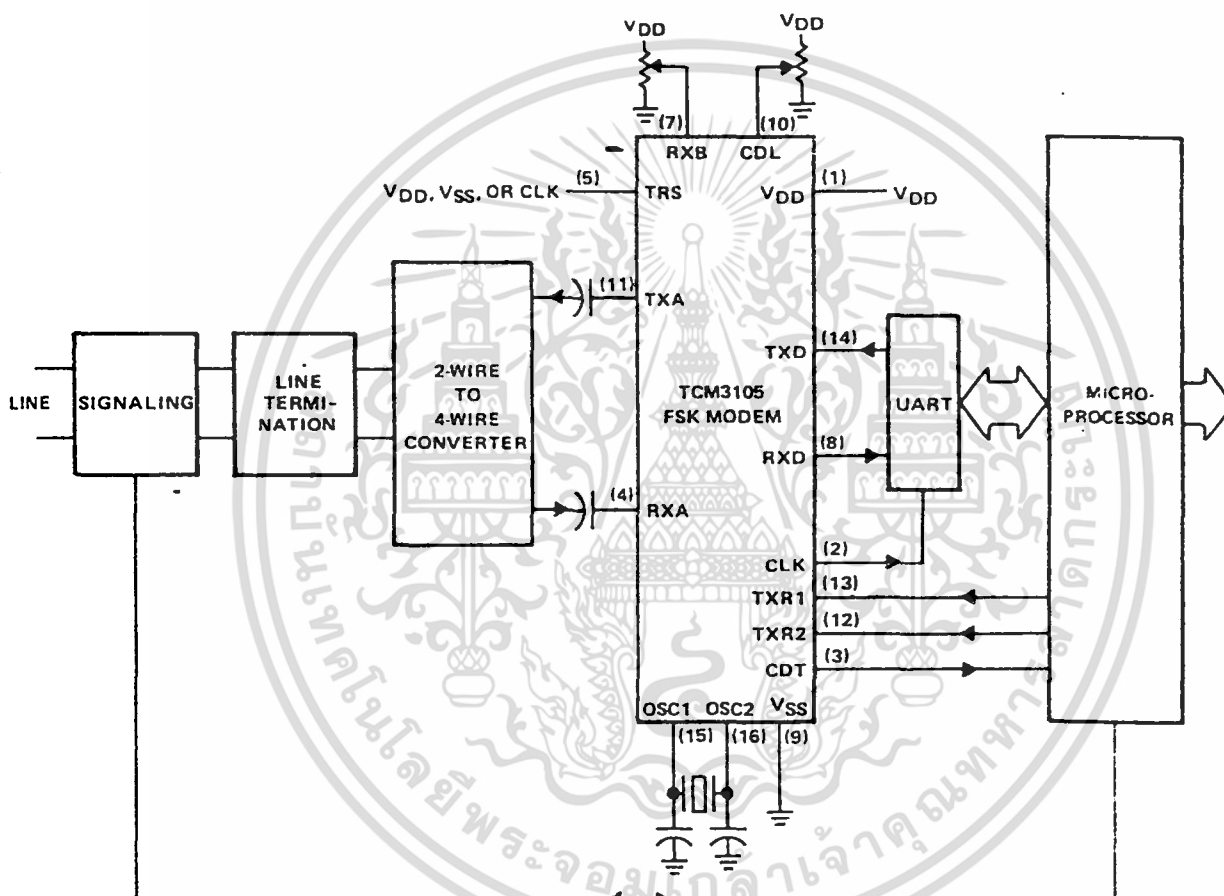
- ขา V_{DD} และ V_{SS} เป็นขาสำหรับป้อนสัญญาณไฟเลี้ยง
- ขา TXR1, TXR2 และ TXRS ใช้เป็นตัวกำหนดอัตราเร็วหรือ baud rate ในการสื่อสารข้อมูล ดูได้จากตารางที่ 4.1
- ขา RxD และ TxD เป็นขาที่ใช้ในการรับส่งข้อมูลแบบดิจิทัล
- ขา RxA และ TxA เป็นขาที่ใช้ในการรับส่งสัญญาณแบบอนาล็อก
- ขา RxB ใช้เป็นขาในการปรับไบแอสจากภายนอกของภาครับเพื่อให้มีการสูญเสียหรือถูกทำลายต่ำสุด
- ขา Osc1 และ Osc2 เป็นขาที่ใช้ในการเชื่อมต่อกับออสซิลเลเตอร์ (โดยทั่วไปแล้วจะใช้คริสตอล 4.4336 MHz) ถ้าใช้ออสซิลเลเตอร์ภายนอกจะต้องต่อเข้ากับขา Osc1 โดยขา Osc2 ปลอยลอยเอาไว้

รูปที่ 4.4 ได้แสดงไคอะแกรมของ TCM3105 และวงจรพื้นฐานของชิปแสดงไว้ดังรูปที่ 4.5 สามารถศึกษารายละเอียดเพิ่มเติมได้จาก Datasheet ในภาคผนวก



รูปที่ 4.4 ไคอะแกรมของโมเด็มชิปเดี่ยว TCM3105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 วงจรโดยทั่วไปของ TCM3105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STANDARD	TRS	TXR1	TXR2	TRANSMITTED BAUD RATE	RECEIVED BAUD RATE	TRANSMIT FREQUENCY ASSIGNMENTS (Hz)	RECEIVE FREQUENCY ASSIGNMENTS (Hz)	CLK FREQUENCY (kHz)
CCITT V.23	L	L	L	1200	1200	M 1300 S 2100	M 1300 S 2100	19.11
	H	L	L	1200	75	M 1300 S 2100	M 390 S 450	19.11
	L	L	H	600	75	M 1300 S 1700	M 390 S 450	9.56
	H	L	H	600	600	M 1300 S 1700	M 1300 S 1700	9.56
	L	H	L	75	1200	M 390 S 450	M 1300 S 2100	19.11
	H	H	L	75	600	M 390 S 450	M 1300 S 1700	9.56
	L	H	H	75	75	M 390 S 450	M 390 S 450	1.19
BELL 202	$\overline{\text{CLK}}$	L	L	1200	1200	M 1200 S 2200	M 1200 S 2200	19.11
	$\overline{\text{CLK}}/8$	L	H	1200	150	M 1200 S 2200	M 387 S 487	19.11
	$\overline{\text{CLK}}/8$	L	H	1200	5	M 1200 S 2200	M 387 S 0	19.11
	CLK	H	L	150	1200	M 387 S 487	M 1200 S 2200	19.11
	CLK	H	H	150	150	M 387 S 487	M 387 S 487	2.39
	CLK [†]	H [†]	L [†]	5	1200	M 387	M 1200	19.11
	H [†]	H [†]	H [†]			S 0	S 2200	
	H	H	H	Transmit Disabled	1200	Transmit Disabled	M 1200 S 2200	19.11

ตารางที่ 4.1 แสดงโหมดในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การแปลงสัญญาณอนาล็อกไปเป็นดิจิทัลโดยใช้ MAX 192

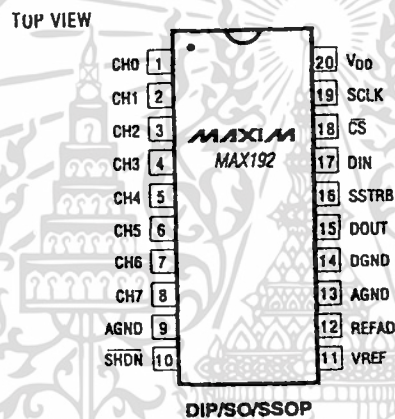
MAX 192 เป็นอุปกรณ์ไอซีแปลงสัญญาณอนาล็อกเป็นดิจิทัล ขนาด 10บิต มีฟังก์ชันการทำงานที่กว้างกว่าเบอร์อื่นมาก ๆ ในทางอินพุตจะมีทั้งหมด 8 ช่องสัญญาณ มาทำการรวมกันแบบมัลติเพล็กซ์ภายในมีช่วงรับสัญญาณแรงดันทางอินพุตที่กว้างและสูง พร้อมกับทางเอาต์พุตสามารถอินเทอร์เฟสได้แบบอนุกรมมีอัตราการแปลงสัญญาณที่ความเร็วสูงมากพร้อมกับคุณลักษณะในการกินกำลังงานต่ำเป็นพิเศษขณะที่ทำงานระดับแรงดันไฟเลี้ยงตัวไอซีจะเป็นแรงดันเดียวขนาด +5 โวลต์ นอกเหนือจากนั้นแล้ว ไอซีเบอร์นี้ สามารถจัดวงจรทางอินพุตให้เป็นการทำงานแบบตรวจจับความแตกต่างทางอินพุตได้เป็น 4 ช่องสัญญาณ ดังกล่าวถึงคุณลักษณะเด่นเบื้องต้นดังต่อไปนี้

- ช่องสัญญาณทางอินพุต 8 ช่องสัญญาณแบบซิงเกิลเอนด์ หรือ 4 ช่องแบบตรวจจับความแตกต่างทางอินพุต
- รับแรงดันไฟเลี้ยงเท่ากับวงจรดิจิทัลอื่น ๆ คือ +5 โวลต์
- กินกำลังงานต่ำที่ 1.5 มิลลิแอมป์ขณะทำงาน และ 2 ไมโครแอมป์ขณะสแตนด์บาย
- มีฟังก์ชันการทำงาน แทร็ค / โฮลด์ (track/hold) ภายในและมีอัตราการแซมปลิงสัญญาณ 133 กิโลเฮิร์ตซ์
- มีแหล่งจ่ายแรงดันอ้างอิงภายในขนาด 4.096 โวลต์
- อินเทอร์เฟสแบบอนุกรม 4 สายและเข้ากันได้ในการอินเทอร์เฟสกับ SPI , QSPI , TMS320
- มีขาใช้งานทั้งหมด 20 ขาบนตัวถังบรรจุ DIP , SO , SSOP
- ตำแหน่งขาทั้งหมดสามารถเข้ากันได้และเหมือนกันกับไอซีขนาด 12 บิต เช่นเบอร์ MAX 186 หรือ MAX 188

ในรูปที่ 4.6 แสดงการจัดตำแหน่งขาของตัวไอซี ซึ่งมีขาใช้งานทั้งหมด 20 ขา ในการอินเทอร์เฟสแบบอนุกรม 4 สายนั้น สามารถต่อเข้าได้โดยตรงกับอุปกรณ์ของ SPI , QSPI โดยไม่ต้องใช้การควบคุมจากระดับลอจิกภายนอก สัญญาณที่ออกจากขาสไทรบอนุกรมเอาต์พุต (SSTRB) สามารถที่จะต่อเข้ากับไอซีดิจิทัลในตระกูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMS320 เพื่อทำการประมวลสัญญาณดิจิทัลได้โดยตรงทางด้านสัญญาณนาฬิกาได้ทั้งภายใน และภายนอกที่ผ่านมาทางการอินเตอร์ทางการอินเตอร์เฟสแบบอนุกรมได้ โดยสัญญาณนาฬิกานั้นจะต้องมีคุณภาพเพื่อเป็นตัวการสำคัญในคุณภาพของการแปลงสัญญาณจากอนาลอกเป็นดิจิทัล และการอินเตอร์เฟสอนุกรมนั้นสามารถทำงานได้ที่สัญญาณนาฬิกาจากภายในตัวไอซีเองที่มีความถี่ 4 เมกะเฮิร์ตซ์พร้อมทั้งมีแหล่งจ่ายแรงดันอ้างอิงภายในขนาด 4.096 โวลต์ ที่มีค่าความแตกต่าง ± 30 พัลส์ต่อนาที่โดยปกติ นอกจากนั้นแล้ววงจรขยายบัฟเฟอร์อ้างอิงภายใน ยังเป็นตัวกำหนดอัตราการขยายเพื่อเป็นการลดความผิดพลาดจากกระบวนการควอนไทซ์



รูปที่ 4.6 รูปร่างการจัดขาของ MAX192

หน้าที่ของขาใช้งาน

ขา 1-8 (Ch0-Ch 7) เป็นขารับสัญญาณอนาลอกแอมพลิงอินพุตมีทั้งหมด 8 อินพุต

ขา 9,13 (A_{GND}) เป็นขากราวด์สัญญาณอนาลอก เมื่อต้องทำหน้าที่แปลงสัญญาณแบบซิงเกิลเอนด์ของสัญญาณที่มาเข้าทางอินพุต (Ch 0 - Ch 7) ขากราวด์อนาลอกนี้ จะต้องต่อกับกราวด์ของแหล่งจ่ายสัญญาณอนาลอกที่มาเข้าที่ทางอินพุตแปลงสัญญาณ

ขา 10 SHDN เป็นขาคควบคุมการหยุดทำงานแบบ 3 ระดับ ถ้าที่ขา SHDN ได้รับลอจิกสถานะ “0” จะทำให้ไอซีตั้งกระแสสำหรับการทำงานต่ำกว่า 10 ไมโครแอมป์ แต่ถ้าต้องการให้ไอซีมีการทำงานอย่างเต็มที่ก็ต้องป้อนลอจิก “1” เข้าที่ขา SHDN นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะทำให้วงจรขยายบัฟเฟอร์อ้างอิงภายในเริ่มทำงาน แต่ถ้าที่ขานี้ไม่ได้รับการกระตุ้นที่ลอจิกใด ๆ เลย ก็จะต้องใช้การกระตุ้นวงจรถ่ายขยายบัฟเฟอร์อ้างอิงจากภายนอก

ขา 11 (V_{REF}) เป็นขาจ่ายแรงดันอ้างอิงสำหรับการแปลงสัญญาณอนาลอกไปเป็นดิจิทัล และที่เอาต์พุตของวงจรถ่ายขยายบัฟเฟอร์อ้างอิงจะต้องต่อตัวเก็บประจุค่า 4.7 ไมโครฟารัดแบบกับกราวด์ เมื่อต้องใช้การกระตุ้นการทำงานจากภายนอก

ขา 12 (REF_{ADJ}) อินพุตของการวงจรถ่ายขยายแรงดันบัฟเฟอร์อ้างอิง หากไม่ต้องการให้วงจรถ่ายขยายบัฟเฟอร์อ้างอิงทำงานต้องต่อขา REF_{ADJ} เข้ากับแรงดัน V_{DD}

ขา 14 (D_{GND}) ขากราวด์ของสัญญาณดิจิทัล

ขา 15 (D_{OUT}) เป็นขาอนุกรมสัญญาณเอาต์พุต สัญญาณข้อมูลคือสัญญาณนาฬิกาเอาต์พุตที่ขอบขาลงของสัญญาณ SCLK จะมีค่าอิมพีแดนซ์สูงเมื่อขา CS มีสถานะเป็น “ 1 ”

ขา 16 ($SSTRB$) เป็นขาอนุกรมสโตรบเอาต์พุต ในโหมดของการใช้สัญญาณนาฬิกาภายในตัวไอซีขา SSTRB จะมีสถานะเป็น “ 0 ” เมื่อไอซีเริ่มทำการแปลงสัญญาณและจะเป็น “ 1 ” เมื่อการแปลงสัญญาณสิ้นสุดลง ในโหมดของการใช้สัญญาณนาฬิกาภายนอก พัลส์ที่ขา SSTRB จะเป็น “ 1 ” ที่หนึ่งคาบเวลาของสัญญาณนาฬิกา ก่อนที่จะถึง MSB ที่ขา SSTRB จะมีค่าอิมพีแดนซ์สูงเมื่อขา CS เป็น “ 1 ”

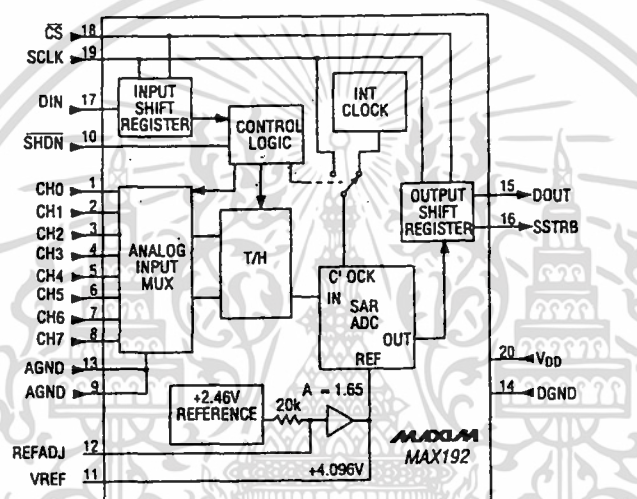
ขา 17 (D_{IN}) เป็นขารับสัญญาณอนุกรมอินพุต ข้อมูลของสัญญาณนาฬิกาที่ขา นี้หรือสัญญาณขอบขาขึ้นของ SCLK

ขา 18 (CS) เป็นขาชิปซีเล็กต์จะแอกทีฟที่ “ 0 ” สัญญาณข้อมูลที่มาเข้าที่ขา D_{IN} นี้จะไม่ใช้สัญญาณนาฬิกาที่ไปเข้าที่ขา D_{IN} และไม่ใช่ “ 0 ” เมื่อขา CS เป็น “ 1 ” ที่ขา D_{OUT} จะมีค่าอิมพีแดนซ์สูง

ขา 19 ($SCLK$) เป็นขารับสัญญาณนาฬิกาอินพุต เป็นสัญญาณนาฬิกาอินพุต และเอาต์พุตของการอินเตอร์เฟสแบบอนุกรม ในโหมดของการใช้สัญญาณนาฬิกาภายใน ขา SCLK จะเป็นการเซตความเร็วในการแปลงสัญญาณ

ขา 20 (V_{DD}) เป็นขารับแรงดันไฟบวก + 5 โวลต์ ± 5 เปอร์เซ็นต์

รายละเอียดในการทำงานในแต่ละส่วนของMAX192 นี้ จะอ้างอิงวงจรบล็อกไดอะแกรมในรูปที่ 4.7 ซึ่งแสดงถึงเทคนิคของการทำงานในแต่ละส่วนและในแต่ละหน้าที่ในการทำงาน โดยที่การทำงานภายในที่สำคัญต่อการแปลงสัญญาณอนาล็อกไปเป็นดิจิทัลที่อัตรา 10 บิตดิจิทัลเอาต์พุต



รูปที่ 4.7 บล็อกไดอะแกรมภายในตัวไอซี

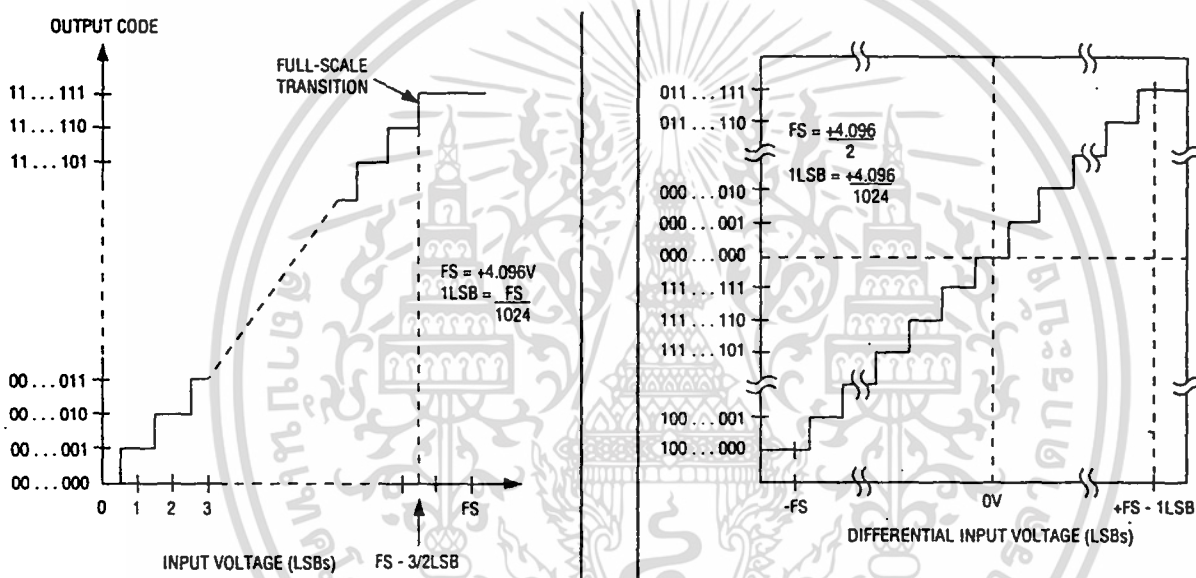
- การเปลี่ยนสัญญาณ

จะเริ่มขึ้นเมื่อมีการควบคุมไบต์ของสัญญาณนาฬิกาเข้ามาเข้าที่ขา DIN ที่ขอบขาขึ้นของสัญญาณที่ขา SCLK และต้องทำให้ขา CS อยู่ในสถานะ “ 0 ” สัญญาณนาฬิกาแต่ละบิตจากขา DIN จะเข้าสู่ส่วนชิปรีจิสเตอร์ภายในตัวไอซีหลังจากที่ขา CS ตกลงลอจิกอันดับแรกซึ่งเป็นบิตข้อมูลลอจิก “ 1 ” จะเป็น MSB ของส่วนไบต์ควบคุม จนกระทั่งบิตข้อมูลแรกหมายถึงการเริ่มต้นจะปรากฏขึ้น ในทุกๆจำนวนลอจิกที่เป็น “ 0 ” สามารถที่จะป้อนเข้าทางขา DIN ได้โดยไม่มีผลเสียแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณดิจิตอลเอาต์พุต

ในการจัดอินพุตแบบขั้วเดียว สัญญาณเอาต์พุตดิจิตอลที่ถูกแปลงออกมาจะมีอยู่ในลักษณะของลำดับหรือแถวของรหัสสัญญาณแบบไบนารี ดังรูปที่ 4.8 (ก) แต่ในการจัดอินพุตแบบสองขั้วดิฟเฟอเรนเชียลนั้น สัญญาณทางเอาต์พุตจะเป็นสัญญาณแบบทวอพลีเมนต์ ดังรูปที่ 4.8 (ข) สัญญาณข้อมูลนาฬิกาเอาต์พุตที่ขอบขาลงของ SCLK จะอยู่ในรูปแบบของ MSB อันดับที่หนึ่ง



รูปที่ 4.8 (ก) สัญญาณเอาต์พุตแบบซิงเกิลเอนด์

(ข) สัญญาณเอาต์พุตแบบดิฟเฟอเรนเชียล

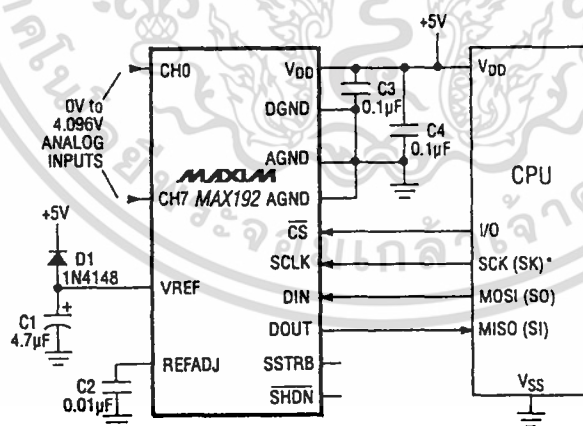
- สัญญาณนาฬิกาจากภายนอก

ในโหมดของการเลือกรับเอาสัญญาณนาฬิกาจากภายนอกนั้นสัญญาณนาฬิกาจะไม่ทำให้เกิดการเลื่อนของข้อมูลทั้งอินพุตและเอาต์พุต เพราะสัญญาณนี้จะเป็นตัวจับลำดับของการแปลงสัญญาณออกไปเป็นดิจิตอลเท่านั้น โดยที่ SSTRB จะมีพัลส์ "1" ในทุก ๆ หนึ่งคาบเวลาของสัญญาณนาฬิกาหลังจากบิตสุดท้ายของไบต์ควบคุมผ่านไป ลำดับคร่าว ๆ ของบิตข้อมูลจะถูกทำและสร้างขึ้นและจ่ายไปที่ขา D_{OUT} ในทุก ๆ คาบเวลาที่ 12 ของขอบขาลงของสัญญาณนาฬิกาที่ขา SCLK ในสลิปบิตแรกของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณข้อมูลที่แท้จริง และสองบิตสุดท้ายจะเป็นบิตข้อมูลย่อย (sub - LSB) ที่ขา SSTRB และขา D_{OUT} จะมีค่าอิมพีแดนซ์ที่สูงมาก เมื่อขา CS เป็น “1” หลังจากที่พัลส์ที่ขา CS ลุกลดต่อไปเป็นขอบขาลง ขา SSTRB จะให้เอาต์พุตออกมาเป็น “0” ในการแปลงสัญญาณแต่ละวงรอบที่เสร็จสมบูรณ์ลงจะใช้เวลาที่สั้นมาก

- การประยุกต์ใช้งาน

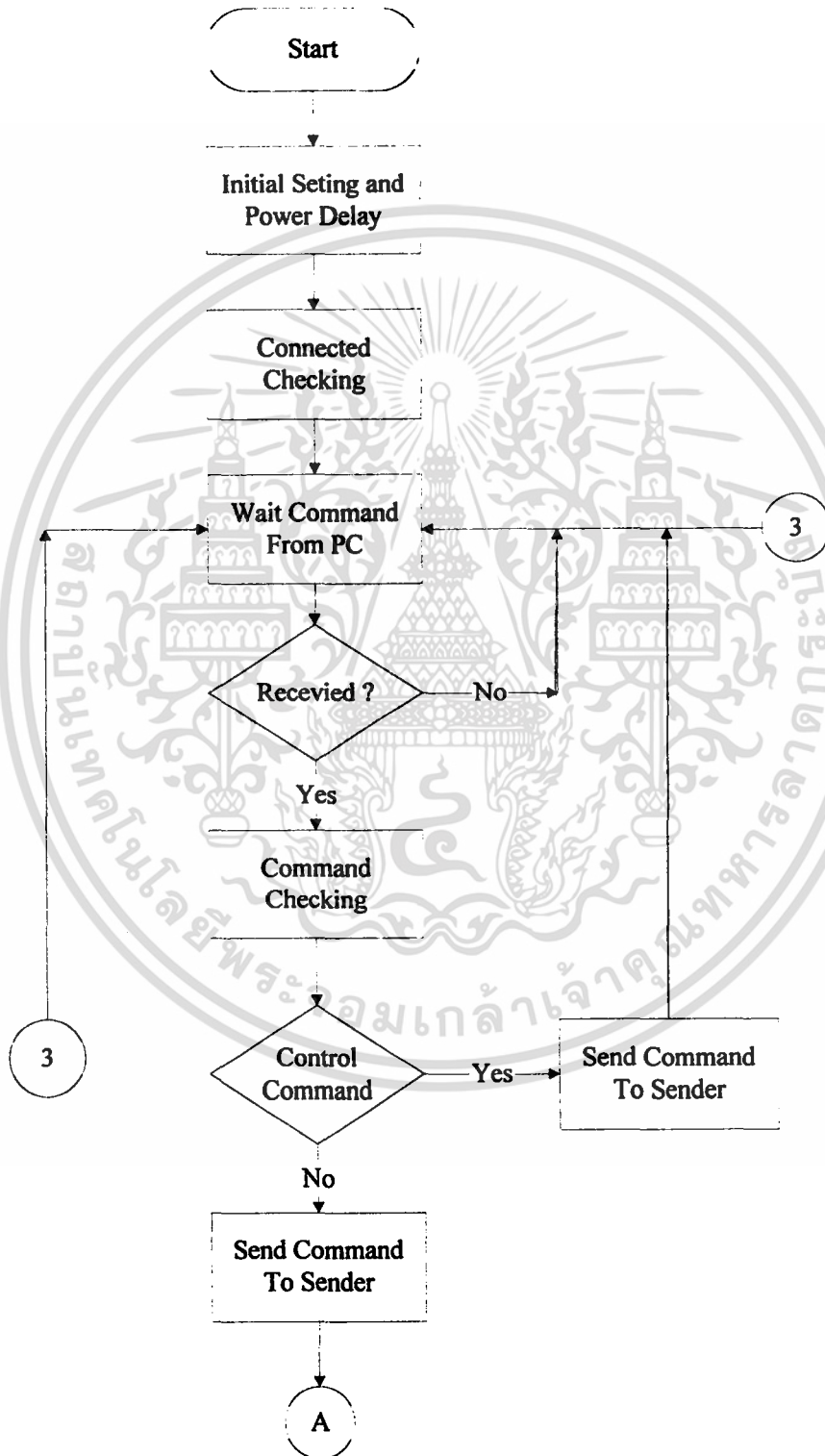
ตามที่กล่าวมาแล้วว่าไอซีตัวนี้สามารถประยุกต์ใช้งานได้หลายคุณสมบัติ เช่น การประยุกต์ใช้งานในรถยนต์เกี่ยวกับระบบตรวจจับต่าง ๆ ใช้ในระบบตรวจจับของหุ่นยนต์ ใช้ในงานด้านตรวจสอบสภาพแบตเตอรี่ เป็นต้น ในรูปที่ 4.9 แสดงลักษณะบล็อกไดอะแกรมหลักการง่าย ๆ ซึ่งขั้นตอนการใช้งานร่วมกับหน่วยประมวลผลทางคอมพิวเตอร์ก็ขึ้นอยู่กับ การเขียนคำสั่งหรือใช้งานร่วมกับซอฟต์แวร์อื่น ๆ



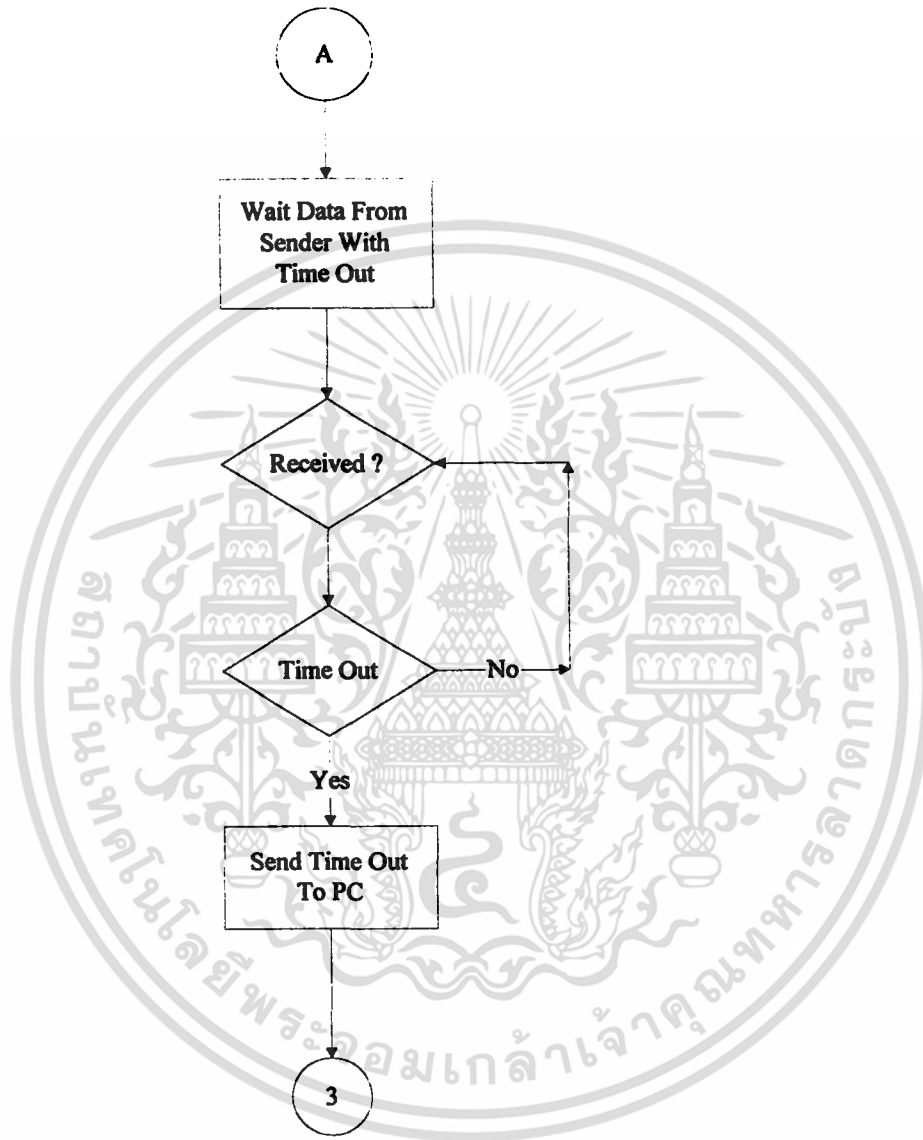
รูปที่ 4.9 ตัวอย่างบล็อกไดอะแกรมการต่อใช้งานร่วมกับ CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

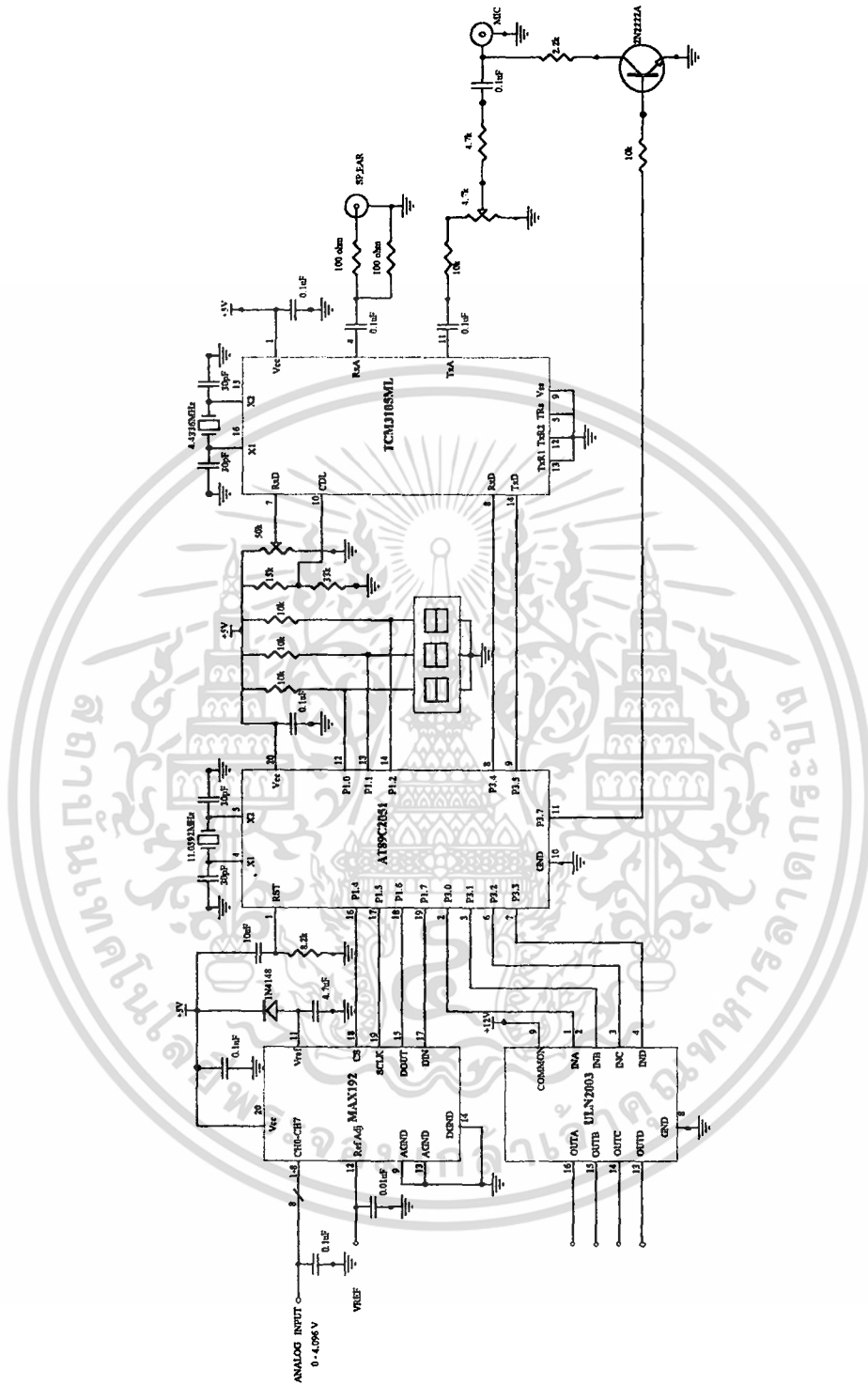
4.1.4 หลักการทำงานของโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



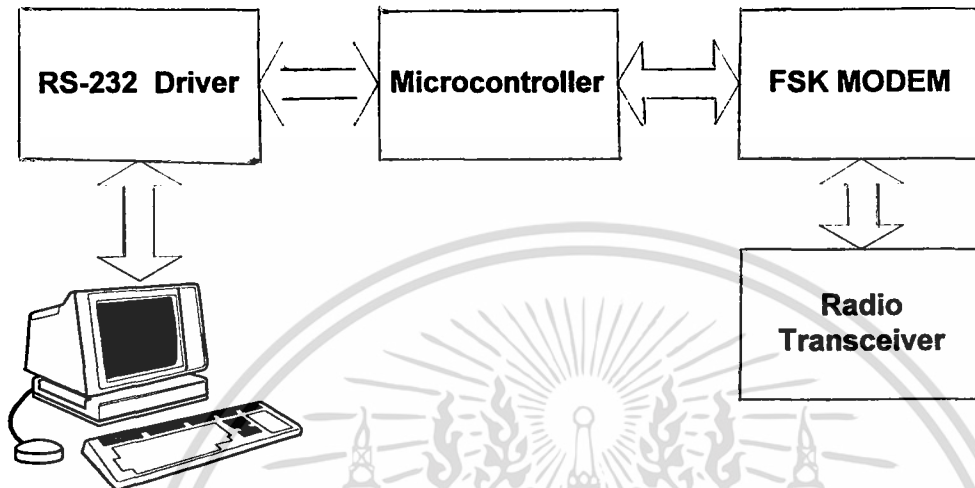
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 วงจรตัวส่งสัญญาณอนาลอกผ่านคลื่นวิทยุขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โมเด็มส่งผ่านคลื่นวิทยุ



รูปที่ 4.11 โครงสร้างการทำงานของโมเด็มส่งผ่านคลื่นวิทยุ

จากรูปที่ 4.11 แสดงไดอะแกรมการทำงานของโมเด็มส่งผ่านคลื่นวิทยุ ซึ่งประกอบไปด้วย

1. วิทยุสื่อสาร ทำหน้าที่ในการส่งและรับข้อมูลในรูปแบบของสัญญาณที่ได้รับการมอดูเลตแล้ว
2. โมเด็ม ทำหน้าที่ในการมอดูเลตสัญญาณข้อมูลดิจิทัลกับความถี่พาห์ เพื่อใช้ส่งข้อมูลผ่านไปยังวิทยุสื่อสาร อีกทั้งยังทำการดีมอดูเลตสัญญาณให้อยู่ในรูปแบบของสัญญาณดิจิทัล
3. ไมโครคอนโทรลเลอร์ ทำหน้าที่ในการควบคุมการทำงานของทุกส่วนในระบบ โดยจะใช้ในการควบคุมการส่งข้อมูลเป็นเฟรมในรูปแบบที่กำหนด และใช้ทำการแปลงข้อมูลจาก NRZ เป็น NRZI และจาก NRZI เป็น NRZ
4. ตัวขับสัญญาณตามมาตรฐาน EIA-232 ทำหน้าที่ในการแปลงลอจิกจากลอจิกแบบ TTL ซึ่งเป็น positive logic ให้เป็นลอจิกแบบ RS-232 ซึ่งเป็น negative logic และแปลงลอจิกจากลอจิกแบบ RS-232 ไปเป็น TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

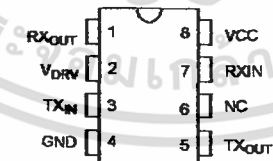
5. ไมโครคอมพิวเตอร์ ทำหน้าที่ในการส่งคำสั่งไปควบคุมยังตัวส่งสัญญาณอนาล็อกและรับข้อมูลที่ตอบกลับมา เพื่อทำการแสดงผล, เก็บเข้าฐานข้อมูล หรือ

4.2.1 การรับส่งข้อมูลดิจิทัลผ่านพอร์ต RS-232 โดยใช้ DS 275

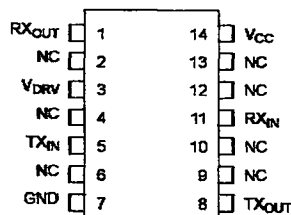
DS 275 นี้เป็นอุปกรณ์รับ - ส่ง ข้อมูลทางดิจิทัลที่มีกำลังในการส่งสัญญาณข้อมูลไปตามสายส่งสัญญาณมากโดยผ่านทางพอร์ตอนุกรม RS-232 ไอซีตัวนี้จัดอยู่ในตระกูลซีมอส ที่มีราคาค่อนข้างต่ำ และสิ้นเปลืองกำลังต่ำมากในการอินเทอร์เฟสข้อมูลผ่านทางพอร์ตอนุกรม RS-232 และนอกจากนั้นระดับของสัญญาณที่รับเข้ามาทางอินพุต RX จะอยู่ในระดับปกติของสัญญาณซีมอส และ TTL สำหรับการส่งข้อมูลนั้นจะเลือกใช้วงจรที่เป็นแบบเฉพาะและจะเป็นการดึงเอากระแสจากสัญญาณขารับของ RS-232 เมื่อสัญญาณนั้นถูกจัดให้อยู่ในแบบของเนกาตีฟ และโดยมากแล้วการสื่อสารข้อมูลผ่านพอร์ตอนุกรมนี้จะนิยมใช้สัญญาณแบบเนกาตีฟ เหตุที่เป็นเช่นนี้ก็เนื่องจากการส่งผ่านแบบนี้จะเป็นการส่งสัญญาณให้เป็นผลสำเร็จได้ดีที่สุดและลดอัตราการสูญเสียกำลังงานสถิตย์ลงไปได้มาก

คุณสมบัติทั่วไปของ DS275 คือเป็นไอซีเพียงตัวเดียวที่ทำหน้าที่ทั้งรับและส่งข้อมูลได้พร้อมกับใช้อุปกรณ์ภายนอกประกอบรวมน้อยมากพร้อมมีตัวถังบรรจุหลายแบบ DIP 8 ขา , SOIC 8 ขา , TSSOP 14ขา ดังแสดงในรูป 4.12

PIN ASSIGNMENT



DS275 8-PIN DIP (300 MIL)
DS275S 8-PIN SOIC (150 MIL)



DS275E 14-PIN TSSOP

รูปที่ 4.12 ลักษณะการจัดตำแหน่งขาของไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของขาต่อใช้งาน

RX_{OUT}	สัญญาณเอาต์พุตของด้านรับ RS-232
V_{DRV}	ขารับแรงดัน +V ของด้านส่ง
TX_{IN}	ขารับสัญญาณอินพุตด้านส่ง RS-232
GND	กราวด์
TX_{OUT}	สัญญาณเอาต์พุตด้านส่ง RS-232
NC	ขาไม่ใช้งาน
RX_{IN}	ขารับสัญญาณอินพุตด้านรับ RS-232
V_{CC}	ขารับแรงดันไฟเลี้ยงบวก (+5 โวลต์)

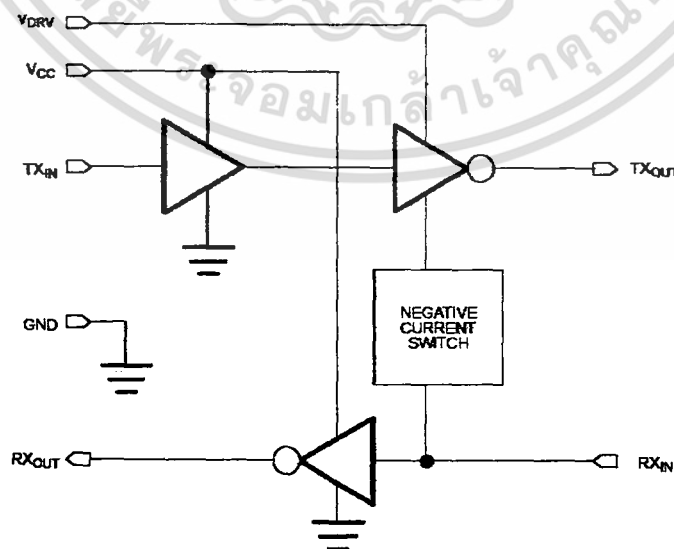
4.2.1.1 การทำงานเบื้องต้น

ไอซีตัวนี้ถูกออกแบบมาสำหรับระบบที่ต้องการหรือมีแบตเตอรี่สำรอง และจะกินกำลังต่ำมากเมื่อถูกใช้งานเชื่อมต่อแบบกึ่งส่วนทางกัน (half-duplex) ทางพอร์ตอนุกรมRS-232 โดยปกตินั้นมีการออกแบบใช้อุปกรณ์ตัวนี้กับพอร์ตอนุกรมพร้อมกับการดึงเอากำลังที่ใช้เลี้ยงวงจรออกมาทางพอร์ตของระบบระหว่างแรงดันบวกและลบของสัญญาณที่ทำการส่งผ่านพอร์ต RS-232 ถ้าการเชื่อมต่อเข้ากับพอร์ตอนุกรมตามลักษณะของวงจรในรูปที่ 4.13 หลังจากเวลาผ่านไปการสื่อสารข้อมูลก็จะสิ้นสุดลง และกำลังงานภายในพอร์ตก็จะมีสถานะการไหลแบบคงที่ จากนั้นกำลังงานจากแบตเตอรี่ก็จะถูกนำมาใช้ DS275 จะทำการขจัดกระแสที่ค้างไหลแบบคงที่นี้ด้วยการดึงกระแสจากไลน์ด้านรับ (ขา RX_{IN}) ของพอร์ตRS-232 เมื่อที่ไลน์ด้านรับนี้มีสถานะเป็นระบบด้านลบ นอกจากนั้นการสื่อสารข้อมูลแบบอซิงโครนัสผ่านพอร์ต RS-232 ตามปกติแล้วจะมีการกำหนดสถานะโดยเมื่อไม่มีการเริ่มส่งข้อมูล DS275 ก็จะไม่มีการกินกำลังงานจากระบบของพอร์ตในขณะนั้นแต่อย่างใด กำลังงานของระบบจะถูกใช้เมื่อเริ่มมีสถานะเป็นบวกในขณะที่ต้องการจะส่งข้อมูลผ่าน RS-232 เอาต์พุต (ที่ขา TX_{OUT}) ก็จะเป็นการเริ่มต้นการส่งข้อมูล และการสื่อสารข้อมูลแบบอซิงโครนัสนี้ตามปกติ จะมีค่าควิต์ไช้เกิดต่ำมาก ๆ จึงทำให้ระบบดึงกำลังงานและมีการสูญเสียกำลังงานต่ำด้วย

การเข้ากันได้กับ RS-232 DS275 มีคุณสมบัติที่เหมาะสมสำหรับการเชื่อมต่อ RS-232 หลายประการ ตั้งแต่การกินกำลังต่ำ จึงเหมาะสำหรับการเชื่อมต่อมีระยะทางหรือความยาวไม่เกิน 10 ฟุต

4.2.1.2 การประยุกต์ใช้งาน

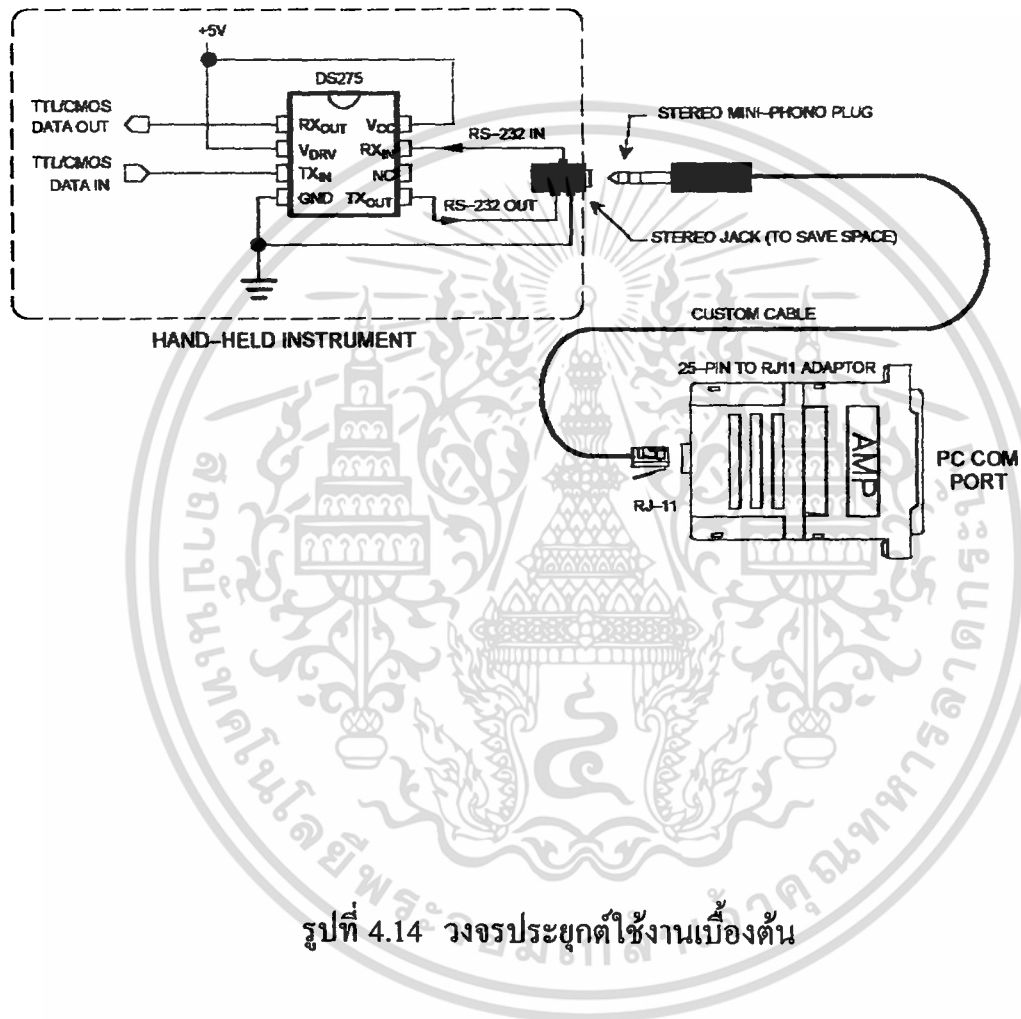
DS275 จะมุ่งเน้นถึงการส่ง-รับข้อมูลผ่านพอร์ตอนุกรม RS-232 เป็นหลัก ใน รูปที่ 4.14 เป็นวงจรเบื้องต้นของการประยุกต์ใช้งาน ในขณะที่การทำงานตามปกติเมื่อสัญญาณข้อมูลถูกส่งจากระบบคอมพิวเตอร์ออกมาผ่าน TX_{OUT} เอาต์พุต DS275 ก็จะสร้างหรือก่อกำเนิดกระแสขับที่ขา V_{DRV} เมื่อที่ขา TX_{OUT} ที่ส่งออกมามีสถานะเป็นบวกกระแสจะไหลจากระบบคอมพิวเตอร์ด้านส่งไปยังพอร์ต RS-232 ของด้านรับโดยจะมีค่าความต้านทานโหลดค่าประมาณ 3 - 7 กิโลโอห์ม เกิดขึ้นภายในระบบสายส่งเมื่อสัญญาณ TX_{OUT} มีสถานะเป็น “มาร์ค” (สัญญาณด้านลบ) กระแสขับขาที่ขา V_{DRV} ก็จะตกลงมาเป็นแรงดันด้านลบและกำหนดได้จากสัญญาณด้านส่งจนถึงปลายสายของเคเบิล จากรูปที่ 4.14 จะสังเกตเห็นที่ปลายคอนเน็กเตอร์ DB-25 จะต่อกับอะแดปเตอร์ RJ11 ที่ตอนท้ายสุดจะเป็นคอนเน็กเตอร์ของโทรศัพท์ซึ่งสะดวกเวลาถอดออกและใส่เข้าที่ปลายต่ออีกด้านหนึ่งจะเป็นแจ็กสเตอริโอสำหรับต่อเข้ากับชุดรับส่ง DS275 ที่ได้สร้างขึ้นดังนั้นสายเชื่อมต่อจึงสามารถทำให้ยาวขึ้นได้เกิน 10 ฟุต



รูปที่ 4.13 การจัดวงจรภายในของ DS275

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

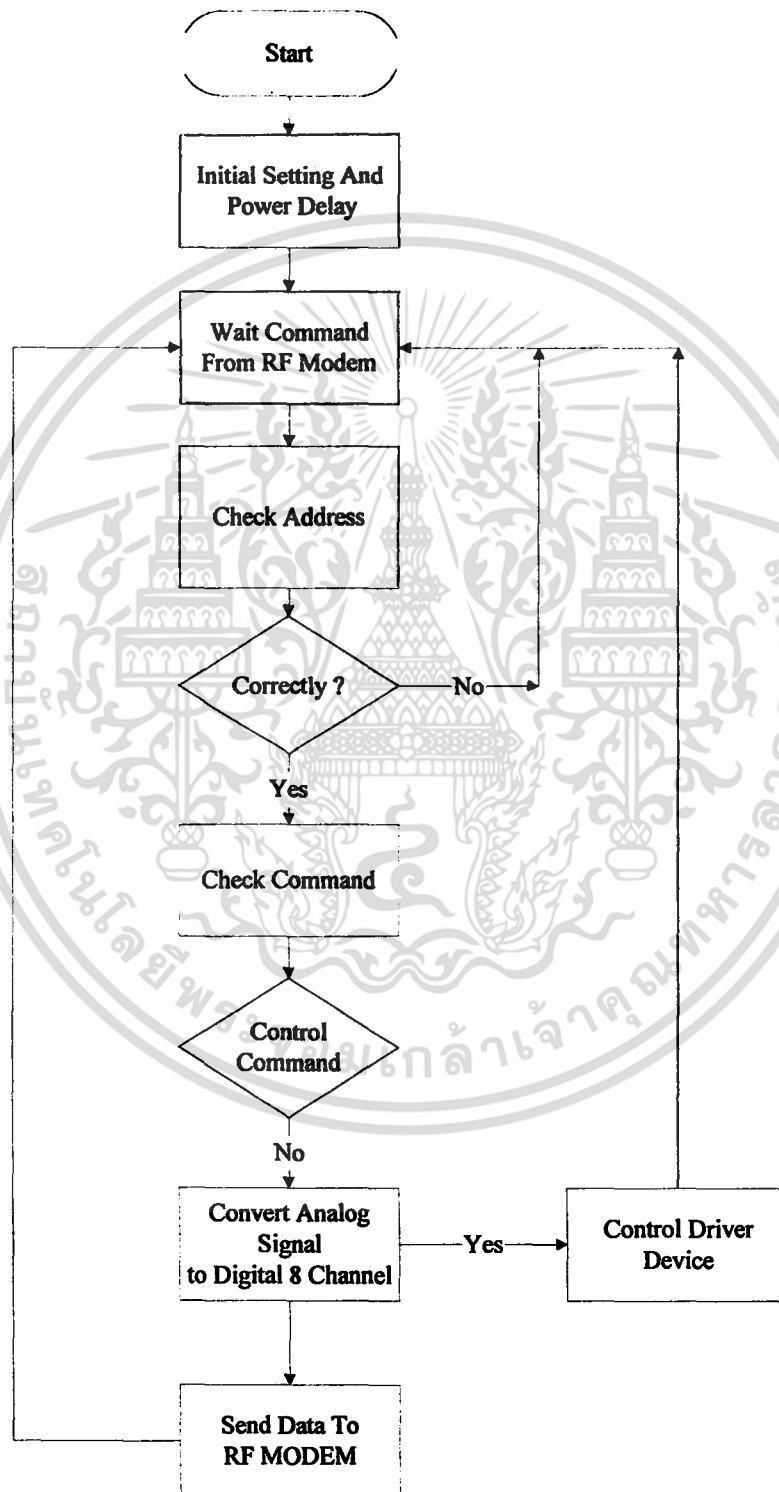
HANDHELD RS-232-C APPLICATION USING A STEREO MINI-JACK Figure 2



รูปที่ 4.14 วงจรประยุกต์ใช้งานเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

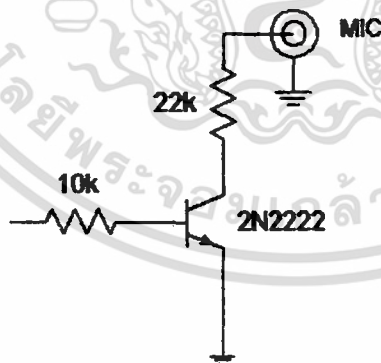
4.2.2 หลักการทำงานของโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรควบคุมการกดคีย์ออกอากาศ (Push to talk, PTT, Control)

จากที่ได้กล่าวไว้แล้วว่าโดยส่วนใหญ่แล้ววิทยุสื่อสารชนิดมือถือมักจะใช้วิธีกำหนดการส่งออกอากาศโดยการตรวจค่าอิมพีแดนซ์ระหว่างขากลาง(ขาสัญญาณ)และขากราว์ของแจ๊คเสียบไมโครโฟนว่ามีค่าตามที่กำหนดหรือไม่ วงจรที่ใช้ควบคุมการกดปล่อยคีย์ออกอากาศที่ได้ออกแบบไว้ ใช้วงจรสวิทช์ทรานซิสเตอร์อย่างง่ายในการควบคุม ดังรูป 4.15 เมื่อให้ลอจิกต่ำที่ขาเบสของทรานซิสเตอร์ ทำให้แรงดันไบแอสที่ขา V_{BE} ต่ำกว่าจุดไบแอส ดังนั้นทรานซิสเตอร์จะไม่ทำงาน เป็นผลให้มีอิมพีแดนซ์สูงมากเมื่อมองจากขาสัญญาณเทียบกับขากราว์ แต่เมื่อให้ลอจิกสูงกับขาเบสของทรานซิสเตอร์ ทำให้แรงดันตกคร่อมขาเบสกับอิมิตเตอร์ถึงจุดไบแอส ทรานซิสเตอร์ก็จะทำงาน ซึ่งเมื่อมองดูจากขาสัญญาณเทียบกับขากราว์จะเห็นว่ามีอิมพีแดนซ์ใกล้เคียงกับความต้านทานที่ขาคอลเลกเตอร์



รูปที่ 4.15 วงจรควบคุมการกดคีย์ออกอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 รูปแบบของแพคเกจข้อมูลที่ทำการส่ง

ในการรับ-ส่งข้อมูลปริมาณอนาล็อกผ่านทางคลื่นวิทยุ จะมึรูปแบบของแพคเกจเป็นดังนี้

SOH SOH xxxx xxxx xxxx xxxx xxxx xxxx xxxx CHECKSUM EOT

โดย

SOH คือ Start of Head เป็นไบต์ที่ใช้ในการกำหนดหัวของข้อมูลในแพคเกจ โดยเท่ากับ 01H ในรหัสแอสกี

EOT คือ End of Text เป็นไบต์ที่ใช้ในการกำหนดการจบของข้อมูลในแพคเกจนั้นๆ ซึ่งมีค่าเท่ากับ 04H ในรหัสแอสกี

CHECKSUM เป็นไบต์ที่ใช้ในการตรวจสอบความผิดพลาดของข้อมูล โดยจะทำการรวมข้อมูลที่ได้ในแต่ละช่องแล้วนำผลที่ได้มาทำ 2's complement

xxxx คือ ข้อมูลที่แปลงได้จะอยู่ในช่วง 0 - 4.096 โวลท์

ตัวอย่างของชุดข้อมูล

SOH SOH 0136 0000 0000 0000 0000 0000 0000 0000 F6 EOT

บทที่ 5

ผลการทดลอง

ในการทดลองขั้นต้นนี้ได้ทำการทดลองโดยการทดสอบการติดต่อสื่อสารระหว่าง RF MODEM กับ RF-linked Analog Signal Sender แบบ Half-duplex โดยได้ทำการทดสอบ 2 วิธีด้วยกัน

1. ใช้โปรแกรม XTALK
2. ใช้โปรแกรมที่ได้พัฒนาขึ้น

วิธีการทดลอง

1. เพื่อให้ RF MODEM เชื่อมต่อเข้ากับไมโครคอมพิวเตอร์ ดังรูปที่ 5.3 คอมพิวเตอร์จะต้องส่งรหัสแอสกีของตัว SPACE ออกไปยัง RF MODEM ซึ่งเมื่อ RF MODEM ได้รับแล้วก็จะส่งค่าแอสกีของ * คืนกลับมายังคอมพิวเตอร์
2. จากนั้นทำการส่งชุดคำสั่งออกไปยัง RF MODEM เพื่อไปควบคุม SENDER อีกทีหนึ่ง โดยมีรูปแบบของคำสั่งแสดงดังข้างล่าง

> @xyy

x คือ ค่าแอสกีตั้งแต่ 0-7

yy คือ คำสั่งในการควบคุม

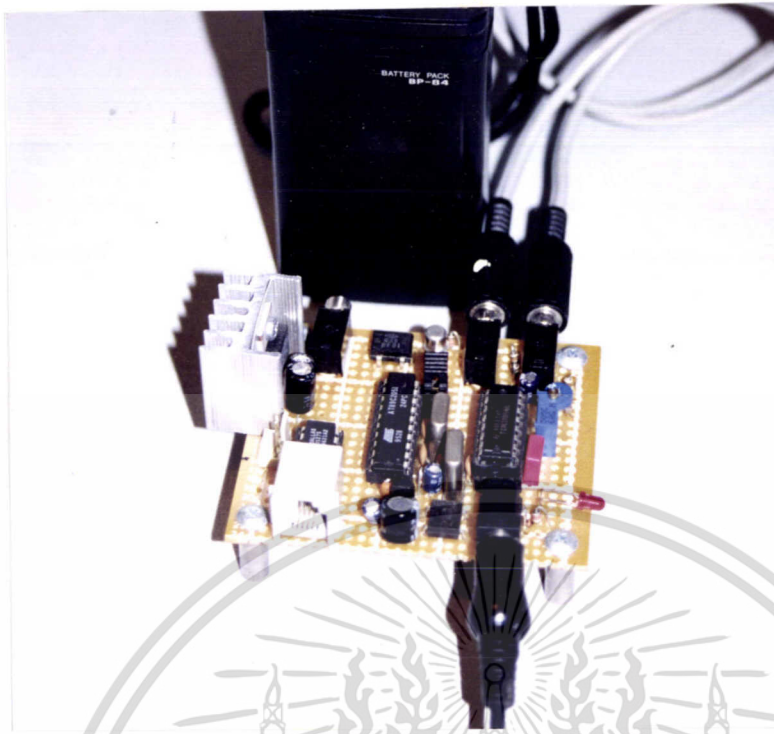
se หมายถึง การเรียกดูข้อมูลทั้ง 8 ช่อง

ry หมายถึง การควบคุมมอเตอร์ให้หมุนไปทางขวา

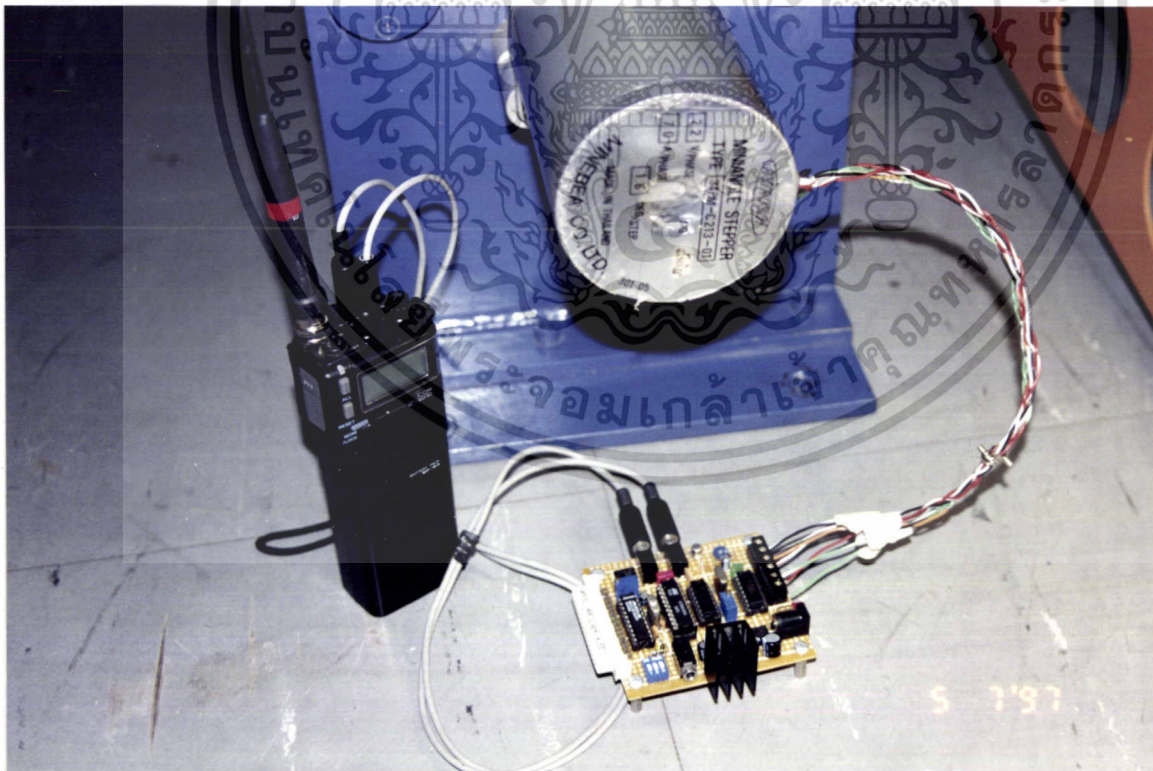
ly หมายถึง การควบคุมมอเตอร์ให้หมุนไปทางซ้าย

หมายเหตุ: ตอนทำการทดลองครั้งแรกนี้ ได้แทนคำสั่งควบคุมมอเตอร์เป็นการกำหนดการเปิดหรือปิดสถานะแทน

3. สังเกตดูผลที่เอาท์พุต

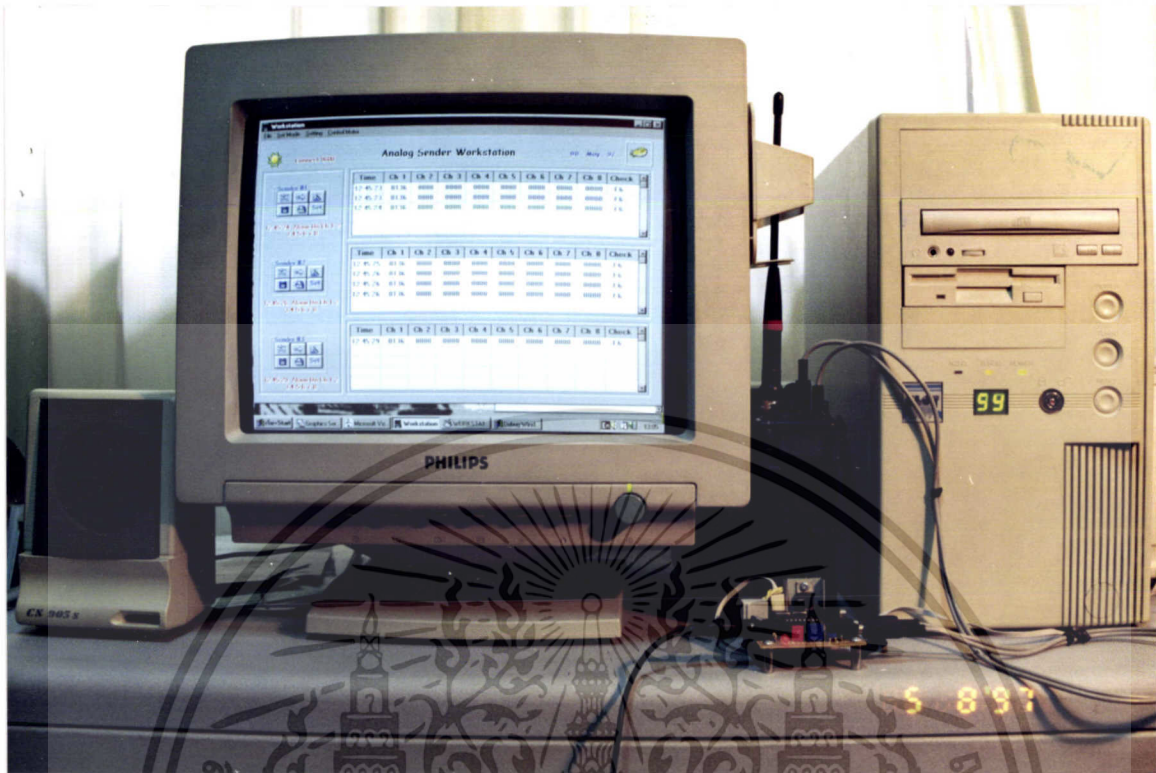


รูปที่ 5.1 โมเต็มส่งผ่านคลื่นวิทยุ

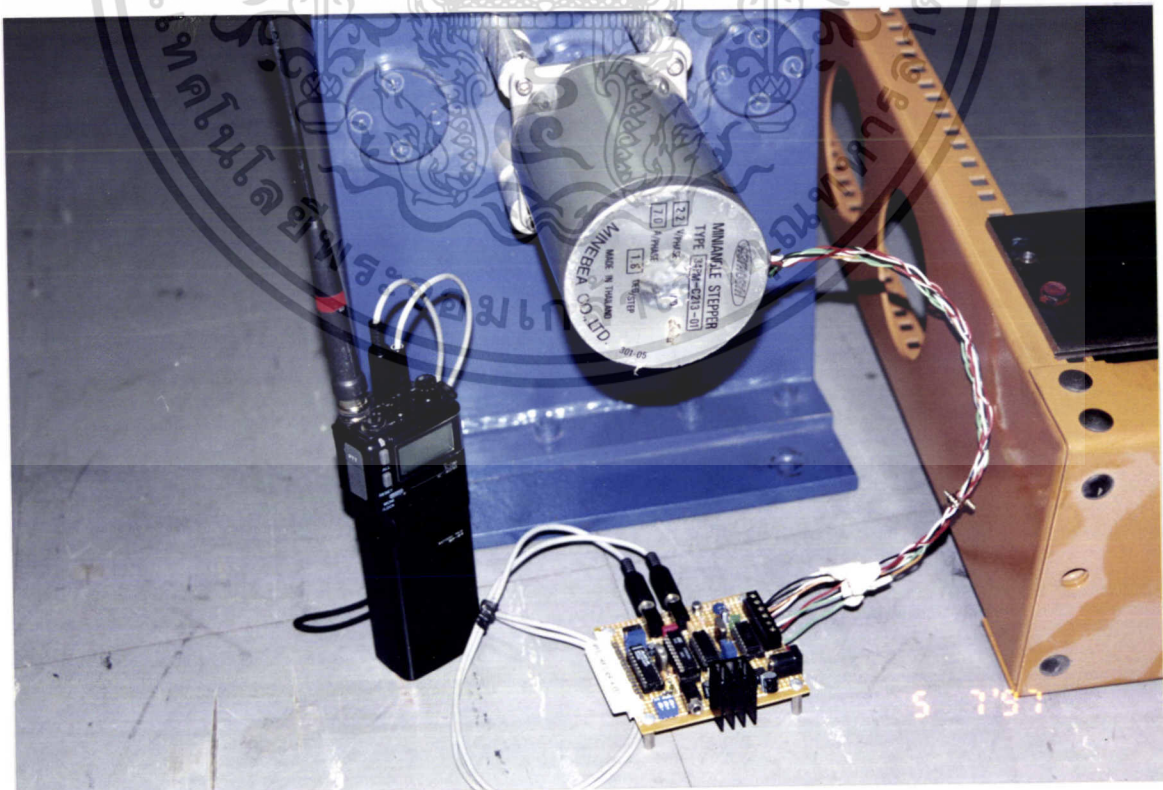


รูปที่ 5.2 ตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 การเชื่อมต่อโมเด็มส่งผ่านคลื่นวิทยุ กับ คอมพิวเตอร์



เอกสารนี้เป็นเอกสารรูปที่ 5.4 การเชื่อมต่อตัวส่งสัญญาณอนาล็อกผ่านคลื่นวิทยุ กับ อุปกรณ์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองเมื่อใช้โปรแกรม XTALK

*

>@1se:02320176011201080084006401000044BF
 >@1se:02200184010401240092006400840032BD
 >@1se:02360168010401200096005600840040B4
 >@1se:02160160010401080080006801040036BE
 >@1se:02040180010401160096006800760032B5
 >@1se:02200168010401160088006800840036AD
 >@1se:02280176010001040080006801000040C5
 >@000*****Timeout*****
 >@2se*****Timeout*****
 >eee*****Timeout*****
 >@1se*****Timeout*****
 >@1se*****Timeout*****
 >@1se:02200176010001040092005600920036BE
 >eee*****Timeout*****
 >@1r2
 >@112
 >

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Workstation File Set Mode Setting Control Moter

Connect 9600

Analog Sender Workstation

10 - Jun - 97

Sender #1

22:34:42 Alarm On Ch 1 2
3 4 5 6 7 8

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check
22:34:41	0136	0000	0000	0000	0000	0000	0000	0000	F6
22:34:42	0136	0000	0000	0000	0000	0000	0000	0000	F6

Sender #2

22:34:44 Alarm On Ch 1 2
3 4 5 6 7 8

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check
22:34:43	0136	0000	0000	0000	0000	0000	0000	0000	F6
22:34:44	0136	0000	0000	0000	0000	0000	0000	0000	F6

Sender #3

22:34:46 Alarm On Ch 1 2
3 4 5 6 7 8

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check
22:34:45	0136	0000	0000	0000	0000	0000	0000	0000	F6
22:34:45	0136	0000	0000	0000	0000	0000	0000	0000	F6
22:34:46	0136	0000	0000	0000	0000	0000	0000	0000	F6
22:34:46	0136	0000	0000	0000	0000	0000	0000	0000	F6

ผลการทดลองเมื่อใช้โปรแกรมที่พัฒนาขึ้น

รูปที่ 5.5 ผลการทดลองเมื่อใช้การโปรแกรมที่พัฒนาขึ้น

บทที่ 6

สรุปผลการทดลองและแนวทางการพัฒนา

6.1 สรุปและการประยุกต์ใช้งาน

โครงการพิเศษตัวส่งสัญญาณอนาล็อกผ่านความถี่วิทยุขนาดเล็กนี้ เป็นโครงการที่พัฒนาขึ้นเพื่อเป็นต้นแบบในการใช้งานด้านระบบโทรมาตรและการควบคุมระยะไกลแบบไร้สาย ซึ่งสามารถทำการสื่อสารได้แบบสองทางชนิด Half Duplex โดยเราสามารถที่จะเลือกควบคุมตัวส่งสัญญาณอนาล็อกได้ทั้งการวัดค่าและการควบคุมระยะไกลแบบรูปเปิด

ซึ่งเราสามารถที่จะนำไปประยุกต์ใช้ในการวัดระยะไกลที่คนเข้าถึงไม่สะดวกหรือระบบวัดแบบไร้สายไม่สามารถวัดได้ เช่น การวัดค่า pH ของน้ำตามแหล่งน้ำต่างๆ หรือจะเป็นระบบวัดบนดาวเทียม ซึ่งประสิทธิภาพของระบบจะขึ้นอยู่กับวิทยุสื่อสาร, กำลังส่ง และระบบเสาอากาศอีกด้วย นอกจากนี้แล้วยังสามารถประยุกต์ใช้ในการควบคุมอุปกรณ์อื่นๆ ได้อีก เช่น การควบคุมสแต็ปมอเตอร์เพื่อใช้ในการหมุนหันเสาอากาศให้ตรงกันระหว่างผู้ส่งกับผู้รับ หรือการควบคุมกล้อง CCD ให้สามารถหันไปเก็บภาพได้ตามมุมที่เราต้องการ และอื่นๆ เป็นต้น

ปัญหาที่พบในโครงการนี้ ได้แก่สัญญาณรบกวนซึ่งทำให้ผลของข้อมูลผิดพลาดไป และความไม่สมดุลกันของภาครับกับภาคส่งของวิทยุสื่อสาร

6.2 แนวทางในการพัฒนาโครงการต่อไป

1. เพิ่มประสิทธิภาพให้กับระบบสื่อสารและการป้องกันสัญญาณรบกวน
2. ระบบควบคุมระยะไกลควรเพิ่มให้เป็นระบบควบคุมแบบปิด เพื่อความแน่นอนในการควบคุมจากระยะไกล
3. เราสามารถเลือกใช้ชนิดของตัวแปลงสัญญาณจากอนาล็อกเป็นดิจิทัลได้ตามแต่สัญญาณที่ต้องการวัด

เอกสารอ้างอิง

1. Ayala , Kenneth J. , The 8051 Microcontroller Architecture , Programming , and Application , pp.11-43 , West Publishing Company , ST. Paul , 1991
2. Norton , Harry N. , Electronic Analysis Instruments , pp 26-30 , Prentice-Hall , New Jersey , 1992
3. Daugherty , Kevin M. , Analog-to-Digital Conversion : A Practical Approach , pp 8-9, 129-144 , McGraw-Hill, Inc. , Singapore , 1995
4. วิสันต์ อาชาเดโชพล , ระบบโทรศัพท์ดิจิทัล , หน้า 9-86 , หจก. สำนักพิมพ์ฟิสิกส์เซ็นเตอร์ , 2536
5. HS3DF , " ใช้วิทยุให้คุ้มค่าด้วยระบบแพคเกจเรดิโอ " , 100 วัตต์ , ฉบับที่ 44 , หน้า 68-70 , 1996
6. วิจิต ศิริโชติและบุญชัย พจนานสมมาน , " Remote Data Acquisiton Module for Physical Quantity " , วารสารของสมาคมฟิสิกส์ไทย , ฉบับที่ 1 (1995) , หน้า 28-31
7. <http://prairie.lakes.com~medcalf/ztx/tnc-bk/>

ประวัติผู้ดำเนินงานวิจัย

นายบดินทร์ ดำรงค์ศักดิ์ เกิดวันอังคารที่ 10 กุมภาพันธ์ พ.ศ. 2519 ที่จังหวัด กรุงเทพฯ สำเร็จการศึกษาระดับมัธยมจากโรงเรียน สาธิต มศว. ประสานมิตร เข้าศึกษา ในระดับปริญญาตรี ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะ วิทยาศาสตร์ ภาควิชาฟิสิกส์ประยุกต์ พ.ศ. 2536 และสำเร็จการศึกษา พ.ศ. 2540 ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาโท ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ ทหารลาดกระบัง คณะวิทยาศาสตร์ ภาควิชาฟิสิกส์ประยุกต์

นางสาววรลณี ไตวณะบุตร เกิดวันอังคารที่ 28 มกราคม พ.ศ. 2519 ที่จังหวัด กรุงเทพฯ สำเร็จการศึกษาระดับมัธยมจากโรงเรียนบดินทรเดชา (สิงห์ สิงหเสนี) เข้า ศึกษาในระดับปริญญาตรี ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะวิทยาศาสตร์ ภาควิชาฟิสิกส์ประยุกต์ พ.ศ. 2536 และสำเร็จการศึกษา พ.ศ. 2540 ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาโท ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ ทหารลาดกระบัง คณะเทคโนโลยีสารสนเทศ แขนงวิชาวิทยาการสารสนเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **คู่มือการใช้ซอฟต์แวร์ เวิร์คสเตชัน (Workstation)**

เวิร์คสเตชัน เป็นซอฟต์แวร์ที่ออกแบบให้ใช้กับระบบการทำงานของตัววัดปริมาณทางฟิสิกส์ส่งผ่านความถี่วิทยุขนาดเล็ก (RF-Link Miniature Analog Sender) โดยใช้ในการติดต่อสื่อสารผ่านพอร์ทอนุกรมระหว่างคอมพิวเตอร์ที่สถานีกลาง กับ ตัววัดปริมาณทางฟิสิกส์ในสถานที่ต่าง ๆ เพื่อทำการเฝ้าตรวจระวัง ควบคุมการทำงานของมอเตอร์ และเก็บข้อมูลที่ส่งผ่านเข้ามาเข้าในระบบฐานข้อมูล

สิ่งที่จำเป็นสำหรับซอฟต์แวร์ เวิร์คสเตชัน

1. คอมพิวเตอร์ 80486 Dx4-100 หรือสูงกว่า
2. ระบบปฏิบัติการวินโดวส์ 3.11 หรือสูงกว่า
3. หน่วยความจำ (RAM) 8 เมกะไบต์
4. ความจุบนฮาร์ดดิสก์ 4 เมกะไบต์
5. โมเด็ม ที่ใช้คลื่นวิทยุสื่อสาร
6. วิทยุสื่อสาร
7. ตัววัดปริมาณทางฟิสิกส์ (Analog Sender)

การติดตั้งซอฟต์แวร์

หลังจากเรามีอุปกรณ์ตั้งที่กล่าวมาแล้ว เราก็จะเริ่มติดตั้งตามขั้นตอนดังนี้

1. เข้าสู่ระบบปฏิบัติการวินโดวส์
2. ใส่แผ่น เซ็ทอัพ ดิสก์ ลงในไดรฟ์ 2.5 นิ้ว (1.44 เมกะไบต์)
3. เลือกคำสั่ง รัน จากเมนู ไฟล์ ที่อยู่ใน โปรแกรม เมนเนเจอร์ ของ วินโดวส์
4. พิมพ์ A: \ Setup แล้วกด เ็นเตอร์

ทำตามคำสั่งที่ปรากฏบนหน้าจอสำหรับติดตั้ง จากนั้นโปรแกรมการติดตั้งจะติดตั้ง
เวริคสเทชัน ลงบนฮาร์ดดิสก์ และเติมกลุ่มวินโดว์ของ เวริคสเทชัน และไอคอนเข้าไป
ในวินโดว์

ส่วนประกอบของซอฟต์แวร์

เมนูหลัก (Main Menu)

เป็นส่วนควบคุมการทำงานของซอฟต์แวร์ เมนูจะไปตัวหลักในการเชื่อมโยงการ
ทำงานของระบบเฝ้าตรวจระวัง (Monitoring) กับ ระบบฐานข้อมูล (Database)



รูปที่ 1 จอภาพเริ่มต้นของ เวริคสเทชัน ซึ่งเป็นส่วนของเมนูหลัก

เมื่อกดปุ่มเมาส์ที่ปุ่ม Monitoring ซอฟต์แวร์จะทำการเปิดโปรแกรมในส่วน of ระบบเฝ้า
ตรวจระวังขึ้นมา และเมื่อกดปุ่มเมาส์ที่ปุ่ม DataBase ซอฟต์แวร์จะทำการเปิดโปรแกรม
ในส่วน of ระบบฐานข้อมูล ส่วนปุ่ม Exit เป็นการจบการทำงาน of ซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Workstation
File Set Mode Setting Control Motor

HardWare Error

Analog Sender Workstation 16 - Mar - 97

Sender #1

		Set

Sender #2

		Set

Sender #3

		Set

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check

Time	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8	Check

รูปที่ 2 จอภาพแสดงการทำงานในส่วนระบบเฝ้าตรวจระวัง (Monitoring)

ระบบเฝ้าตรวจระวัง (Monitoring)

เมื่อเข้ามาในส่วนนี้จะมีการตรวจสอบการทำงานตัววัดปริมาณทางฟิสิกส์ ถ้าข้อความขึ้นว่า HardWare Error ว่าระบบไม่สามารถทำงานได้ แต่ถ้าขึ้นว่า Connect 9600 แสดงว่าระบบพร้อมจะทำงาน



HardWare Error

รูปที่ 3 แสดงผลการตรวจสอบตัววัดปริมาณทางฟิสิกส์

เมนูหลักของระบบเฝ้าตรวจระวังที่ใช้ในการทำงาน

1. File จะมีส่วนของกลับไปเมนูหลัก และ จบการทำงาน
2. Set Mode ตั้งโหมดการทำงานของระบบว่าจะเป็น อัตโนมัติ หรือ ธรรมดา ตั้งเวลาพอร์ตอนุกรม
3. Setting เป็นส่วนที่ใช้ในการตั้งเวลาการทำงานในโหมดอัตโนมัติ การตั้งค่าพอร์ตอนุกรม และเป็นส่วนการตั้งค่าในการเตือนภัยโดยจะให้ใส่ค่า น้อยที่สุด และมากที่สุดที่จะไม่ทำให้เกิดความเสียหายในทุก ๆ ช่องสัญญาณ ของตัววัดปริมาณทางฟิสิกส์ทั้ง 3 ตัว ถ้าข้อมูลที่เข้ามา มีค่าน้อยกว่า หรือ มากกว่าค่าที่ตั้งไว้จะมีการเตือนภัย



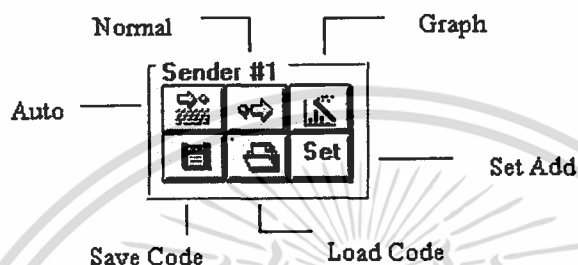
02:56:24 Alarm On Ch 2 7

รูปที่ 4 การแสดงผลการเตือนภัยในของตัววัดปริมาณทางฟิสิกส์ตัวที่ 1 ที่เวลา 02:56:24 มีการเตือนภัยที่ ช่องสัญญาณ ที่ 2 และ 7

4.Control Motor เป็นส่วนการควบคุมการหมุนของมอเตอร์ โดยสร้าง แอดเดรส แล้วเลือกปุ่มการหมุนของมอเตอร์ การทำงานในส่วนนี้ถ้ามีการส่ง แอดเดรส

ของการทำงานของระบบเฝ้าตรวจระวังอยู่ การส่ง แอดเดรส จะหยุดแล้วให้ส่วนการควบคุมการหมุนของมอเตอร์ทำงานก่อน

เมื่อตั้งค่าต่าง ๆ ที่เมนูเสร็จเข้ามาทำงานในระบบเฝ้าตรวจระวังจะมีทูลบ็อกซ์ของตัววัดปริมาณทางฟิสิกส์แต่ละตัว



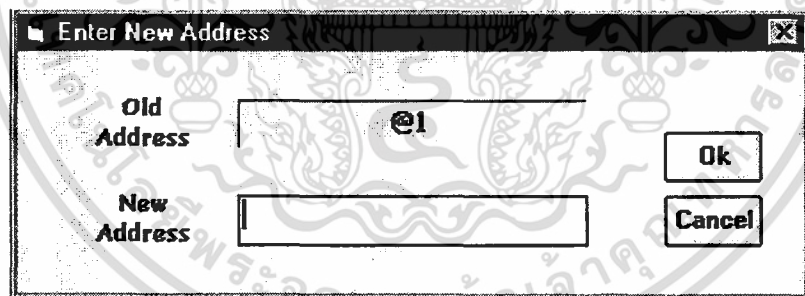
รูปที่ 5 ทูลบ็อกซ์ของตัววัดปริมาณทางฟิสิกส์ตัวที่ 1

ทูลบ็อกซ์ของตัววัดปริมาณทางฟิสิกส์ จะประกอบด้วย 6 ส่วน

1. ปุ่ม Auto เป็นปุ่มที่ใช้ในการส่ง แอดเดรส แบบอัตโนมัติ ลักษณะของปุ่มจะไม่เหมือนปุ่มธรรมดา จะเป็นปุ่มที่กดแล้วจะค้างอยู่แล้วกดอีกครั้งจะกลับสภาพเดิม เมื่อมีการทำงานในโหมดอัตโนมัติ แล้วมีการกดปุ่ม Auto ก็จะมีการส่ง แอดเดรส ของ ตัววัดปริมาณทางฟิสิกส์ตัวนั้น และจะหยุดส่งเมื่อกดปุ่ม Auto อีกครั้งเพื่อให้อยู่ในสภาพเดิม
2. ปุ่ม Normal เป็นปุ่มที่ใช้ในการส่ง แอดเดรส แบบธรรมดา คือเมื่อกดปุ่มก็จะมีการส่ง แอดเดรส ของ ตัววัดปริมาณทางฟิสิกส์ตัวนั้น ตามจำนวนครั้งที่กด
3. ปุ่ม Graph จะใช้ในการดูกราฟของข้อมูลที่ส่งจากตัววัดปริมาณทางฟิสิกส์ โดยจะสามารถดูได้ตั้งแต่ค่าเริ่มต้นจนถึงค่าสุดท้ายก่อนกดปุ่ม เข้ามาดูกราฟ และกราฟจะแสดงข้อมูลระหว่าง ข้อมูล และ เวลา ในส่วนของเมนูในกราฟ จะมีแบบของกราฟให้เลือกที่ เมนู Type ของกราฟโดยมีให้เลือกทั้งหมด 7 แบบ คือ กราฟวงกลมแบบ 2 มิติ (Pie 2D) , กราฟวงกลมแบบ 3 มิติ (Pie

3D) , กราฟแท่งแบบ 2 มิติ (Bar 2D) , กราฟแท่งแบบ 3 มิติ (Bar 3D) ,
กราฟเส้นตรง , กราฟแสดงพื้นที่ (Area) , กราฟที่แสดงเฉพาะจุดข้อมูล
(Scatter)

4. ปุ่ม Save Code เมื่อกดจะทำการบันทึกข้อมูลของตัววัดปริมาณทางฟิสิกส์ตั้ง
แต่เวลา เริ่ม ต้นจนถึงเวลาที่กดปุ่ม การบันทึกจะบันทึกเป็น . Cod แต่การ
บันทึกในส่วนนี้จะไม่เกี่ยวข้องกับระบบฐานข้อมูล
5. ปุ่ม Load Code เมื่อกดจะทำการเปิดข้อมูลที่มีลักษณะ . Cod แล้วข้อมูลจะ
แสดงผลบนตารางของตัววัดปริมาณทางฟิสิกส์นั้น
6. ปุ่ม Set เป็นปุ่มที่ใช้สำหรับตั้งค่า แอดเดรส ที่จะส่งสามารถตั้งได้ตั้งแต่ @1
- @7 รวม 8 ตัว ถ้าใส่ แอดเดรส ต่างจากที่กำหนดก็จะไม่มีข้อมูลเข้ามา และ
เมื่อไม่มีข้อมูลเข้ามาจนถึงเวลาหนึ่งระบบจะมีการแสดงผล Time Out
ในการเข้าไปตั้ง แอดเดรส นั้นจะต้องใส่รหัสผ่านก่อน ซึ่งรหัสนี้คือ
system เมื่อใส่รหัสถูกต้องจึงเข้าไปเปลี่ยน แอดเดรส

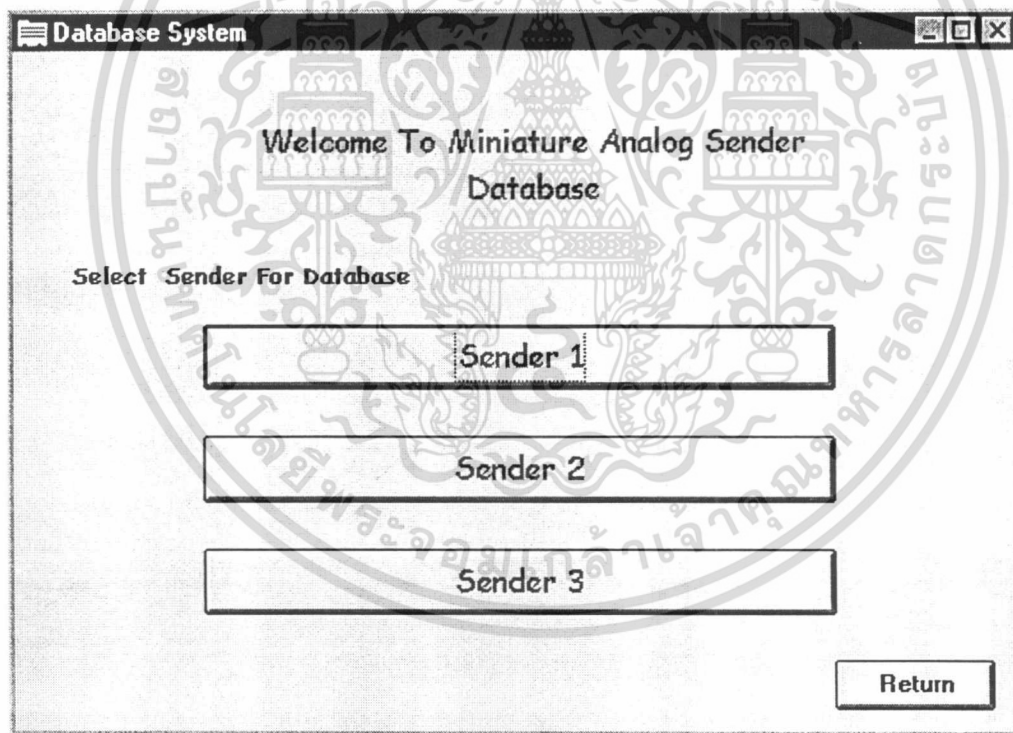


รูปที่ 6 แสดงภาพการเปลี่ยน แอดเดรส ซึ่ง แอดเดรส เก่า คือ @1

ระบบฐานข้อมูล (Database)

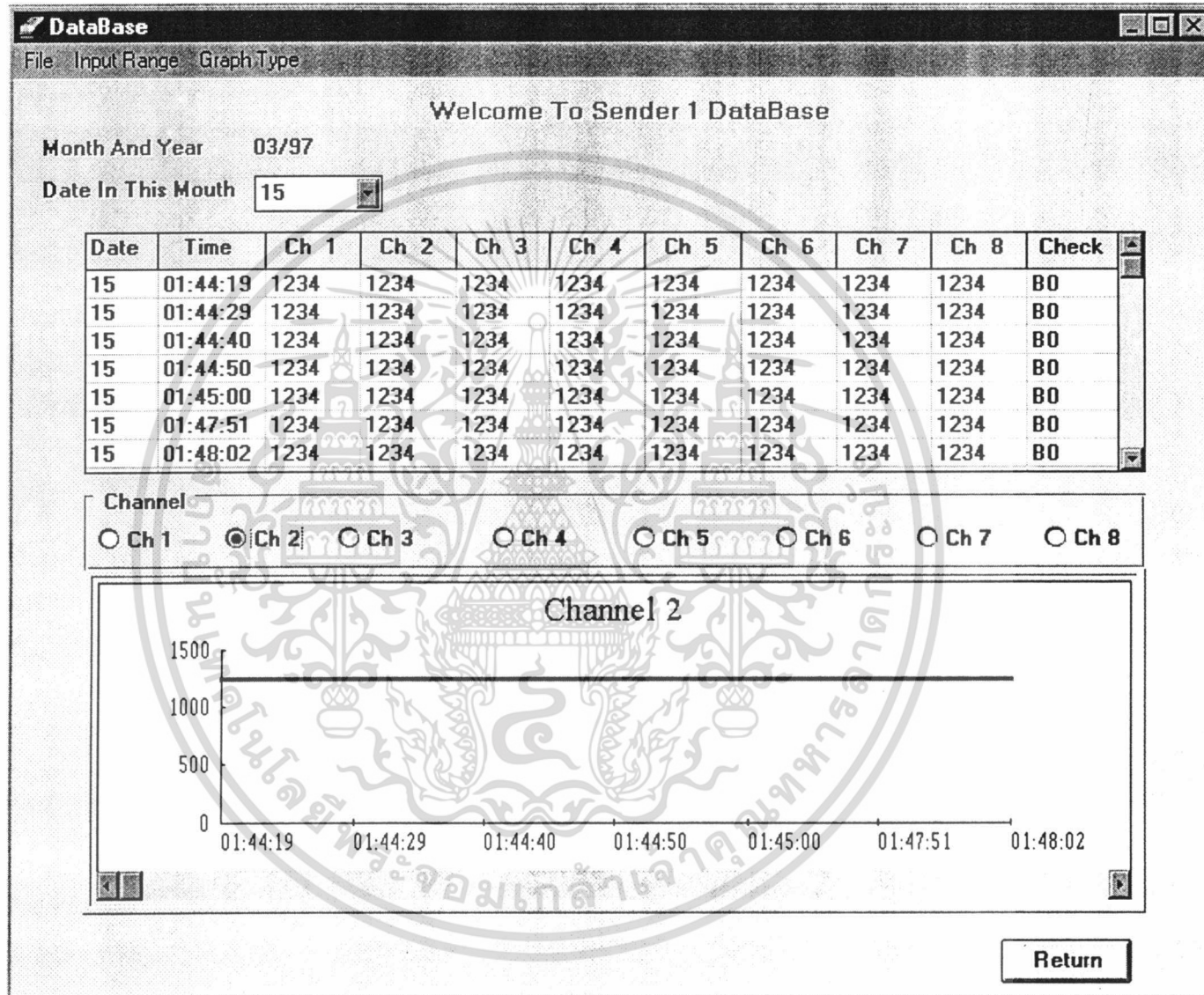
การเก็บข้อมูลในระบบฐานข้อมูลจะทำการเก็บข้อมูลที่ผ่านการตรวจผลรวม (Check Sum) โดยที่ไม่ต้องกดปุ่มใด และการเก็บข้อมูลจะเก็บเข้าในส่วนตารางของตัววัดปริมาณทางฟิสิกส์

เมื่อเข้าสู่ระบบฐานข้อมูลจะต้องเลือกว่าจะดูข้อมูลของตัววัดปริมาณทางฟิสิกส์ตัวใดจะเลือกได้จากเมนูหลักของระบบฐานข้อมูล เมื่อเลือกตัววัดปริมาณทางฟิสิกส์ตัวใดระบบจะเข้าสู่ฐานข้อมูลของตัววัดปริมาณทางฟิสิกส์ตัวนั้น แต่กดปุ่ม Return จะกลับที่เมนูหลักของซอฟต์แวร์



รูปที่ 7 แสดงเมนูหลักในการเลือกดูข้อมูลตัววัดปริมาณทางฟิสิกส์ในระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

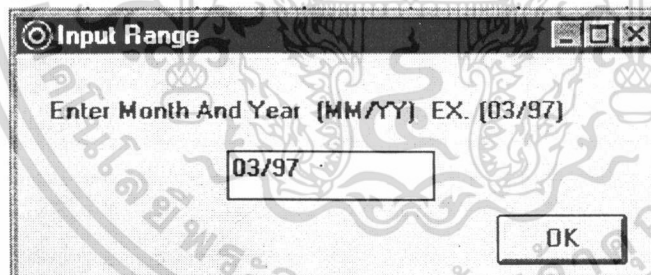


รูปที่ 8 การแสดงข้อมูลในระบบฐานข้อมูล

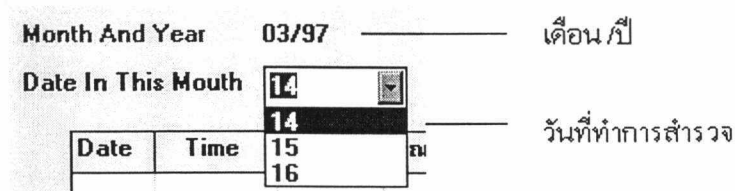
เมื่อเลือกตัววัดปริมาณทางฟิสิกส์ที่จะดูข้อมูลแล้วซอฟต์แวร์จะเปิดหน้าจอสำหรับแสดงข้อมูลในฐานข้อมูล

การแสดงผลข้อมูลในระบบฐานข้อมูล เป็นการแสดงผลข้อมูลของตัววัดปริมาณทางฟิสิกส์ที่เลือกในเมนูหลักของระบบฐานข้อมูล เมนูของการแสดงผลข้อมูลในระบบฐานข้อมูลมีรายละเอียดดังนี้

1. File จะมีส่วนของกลับไปเมนูหลัก และ จบการทำงาน
2. Input Range เป็นการกำหนดช่วงเดือนและปีที่จะดูข้อมูล ในลักษณะ เดือน / ปี เป็นตัวเลข ยกตัวอย่างเช่น 03/97 คือการดูข้อมูลของ เดือนมีนาคม ปี 1997 จากนั้นจะมีวันที่ ที่ทำการสำรวจเก็บข้อมูลขึ้นมาให้เลือก ดังแสดงในรูปที่ 3.3 เมื่อเลือกวันที่แล้วจึงมีข้อมูลของวันนั้นแสดงผล แต่ถ้ากำหนดช่วงเดือนและปีที่จะดูข้อมูลแล้วไม่มีข้อมูลของช่วงเดือนและปีนั้นในฐานข้อมูล จะแสดงข้อความว่าหาข้อมูลในช่วงนั้นไม่พบ และถ้าใส่ข้อมูลไม่ถูกต้องตามที่กำหนดไว้ ก็จะไม่มีการแสดงผลเช่นกัน



รูปที่ 9 แสดงการใส่ เดือน/ปี ในการดูข้อมูลในระบบฐานข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 10 แสดงการแสดงผลเมื่อใส่ เดือน/ปี ที่มีในฐานข้อมูล

3. Graph Type ใช้เลือกชนิดของกราฟที่จะแสดงกราฟของข้อมูลโดยจะมีให้เลือก 7 แบบ คือ กราฟวงกลมแบบ 2 มิติ (Pie 2D) , กราฟวงกลมแบบ 3 มิติ (Pie 3D) , กราฟแท่งแบบ 2 มิติ (Bar 2D) , กราฟแท่งแบบ 3 มิติ (Bar 3D) , กราฟเส้นตรง , กราฟแสดงพื้นที่ (Area) , กราฟที่แสดงเฉพาะจุดข้อมูล (Scatter) เมื่อมีข้อมูลบนตารางก็สามารถดูกราฟของข้อมูลในวันนั้นได้โดยเลือกช่อง สัญญาที่ต้องการ

ในหน้าจอของการแสดงข้อมูลในระบบฐานข้อมูล จะมีปุ่ม Return สำหรับกลับไปเมนูหลักของฐานข้อมูล ข้อมูลที่อยู่ในฐานข้อมูลจะถูกเก็บอยู่ในไฟล์ Sender . Mdb และ Sender . ldb

```

#####
#####;
;      Miniature Analog Sender      ;
;      MAS-01Z                       ;
#####
#####;
;Designed and Developed by Badin
Damrongsakd
;3rd Jan 1997
;Modified by Badin Damrongsakd
;4th Mar 1997
;
;Hardware: AT89c2051 ( 8-Bit Microcontroller
with 2KBytes Flash PEROM )
;      MAX192 ( 8-Ch., Serial 10-Bit Successive
Approximation ADC )
;      TCM3105 ( 1200 BPS FSK Modem )
;Optioned: ULN2003 ( High Current/Voltage
Darlington Drivers )

      CPU "8051.TBL"
      HOF "INT8"

;
;MCS-51 INTERNAL REGISTERS
;
B:      EQU    0F0H    ;B REGISTER
ACC:    EQU    0E0H    ;ACCUMULATOR
PSW:    EQU    0D0H    ;PROGRAM
STATUS WORD
IPC:    EQU    0B8H    ;INTERRUPT
PRIORITY
P3:     EQU    0B0H    ;PORT 3
IEC:    EQU    0A8H    ;INTERRUPT
ENABLE
P2:     EQU    0A0H    ;PORT 2
SBUF:   EQU    99H     ;SEND BUFFER
SCON:   EQU    98H     ;SERIAL CONTROL
P1:     EQU    90H     ;PORT 1
TH1:    EQU    8DH     ;TIMER 1 HIGH
TH0:    EQU    8CH     ;TIMER 0 HIGH
TL1:    EQU    8BH     ;TIMER 1 LOW
TL0:    EQU    8AH     ;TIMER 0 LOW
TMOD:   EQU    89H     ;TIMER MODE
TCON:   EQU    88H     ;TIMER CONTROL
PCON:   EQU    87H     ;POWER CONTROL
REGISTER
DPH:    EQU    83H     ;DATA POINTER
HIGH
DPL:    EQU    82H     ;DATA POINTER
LOW
SP:     EQU    81H     ;STACK POINTER
P0:     EQU    80H     ;PORT 0
;
;MCS-51 INTERNAL BIT ADDRESSES
;
CY:     EQU    0D7H    ;CARRY FLAG
AC:     EQU    0D6H    ;AUXILIARY-
CARRY-FLAG
F0:     EQU    0D5H    ;USER FLAG 0
RS1:    EQU    0D4H    ;REGISTER SELECT
MSB
RS0:    EQU    0D3H    ;REGISTER SELECT
LSB
OV:     EQU    0D2H    ;OVERFLOW FLAG
P:      EQU    0D0H    ;PARITY FLAG
PS:     EQU    0BCH    ;PRIORITY SERIAL
PORT
PT1:    EQU    0BBH    ;PRIORITY TIMER 1
PX1:    EQU    0BAH    ;PRIORITY
EXTERNAL 1
PT0:    EQU    0B9H    ;PRIORITY TIMER 0
PX0:    EQU    0B8H    ;PRIORITY
EXTERNAL 0
EA:     EQU    0AFH    ;ENABLE ALL
INTERRUPT
ES:     EQU    0ACH    ;ENABLE SERIAL
INTERRUPT
ET1:    EQU    0ABH    ;ENABLE TIMER 1
INTERRUPT
EX1:    EQU    0AAH    ;ENABLE EXTERNAL
1 INTERR
ET0:    EQU    0A9H    ;ENABLE TIMER 0
INTERRUPT
EX0:    EQU    0A8H    ;ENABLE EXTERNAL
0 INTERR
SM0:    EQU    09FH    ;SERIAL MODE 0
SM1:    EQU    09EH    ;SERIAL MODE 1
SM2:    EQU    09DH    ;SERIAL MODE 2
REN:    EQU    09CH    ;SERIAL RECEPTION
ENABLE
TB8:    EQU    09BH    ;TRANSMITT BIT 8
RB8:    EQU    09AH    ;RECEIVE BIT 8
TI:     EQU    099H    ;TRANSMIT
INTERRUPT FLAG
RI:     EQU    098H    ;RECEIVE INTERRUPT
FLAG
TF1:    EQU    08FH    ;TIMER 1 OVERFLOW
FLAG
TR1:    EQU    08EH    ;TIMER 1 RUN
CONTROL BIT
TF0:    EQU    08DH    ;TIMER 0 OVERFLOW
FLAG
TR0:    EQU    08CH    ;TIMER 0 RUN
CONTROL BIT
IE1:    EQU    08BH    ;EXT INTERR. 1 EDGE
FLAG
IT1:    EQU    08AH    ;EXT INTERR. 1 TYPE
FLAG
IE0:    EQU    089H    ;EXT INTERR. 0 EDGE
FLAG
IT0:    EQU    088H    ;EXT INTERR. 0 TYPE
FLAG
P0.0:   EQU    080H    ;
P0.1:   EQU    081H    ;
P0.2:   EQU    082H    ;
P0.3:   EQU    083H    ;
P0.4:   EQU    084H    ;
P0.5:   EQU    085H    ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P0.6: EQU 086H ;
P0.7: EQU 087H ;
XOFF: EQU 13H
XON: EQU 11H
BREAK: EQU 00H

P1.0: EQU 090H ; Set Address of
Sender
P1.1: EQU 091H ; Set Address of
Sender
P1.2: EQU 092H ; Set Address of
Sender
P1.3: EQU 093H ;
P1.4: EQU 094H ; Chip Selected for
MAX192
P1.5: EQU 095H ; Serial Clock for
MAX192
P1.6: EQU 096H ; Data Out for
MAX192
P1.7: EQU 097H ; Data In for MAX192

p2.0: equ 0a0h
p2.1: equ 0a1h
p2.2: equ 0a2h
p2.3: equ 0a3h
p2.4: equ 0a4h
p2.5: equ 0a5h
p2.6: equ 0a6h
p2.7: equ 0a7h

p3.0: equ 0b0h ; Drive Stepping Motor
Phase 1
p3.1: equ 0b1h ; Drive Stepping Motor
Phase 2
p3.2: equ 0b2h ; Drive Stepping Motor
Phase 3
p3.3: equ 0b3h ; Drive Stepping Motor
Phase 4
p3.4: equ 0b4h ; Rxd_NRZI
p3.5: equ 0b5h ; Tx_NRZI
p3.6: equ 0b6h ;
p3.7: equ 0b7h ; PTT

p3.0: equ 0b0h
p3.1: equ 0b1h
p3.2: equ 0b2h
p3.3: equ 0b3h
p3.4: equ 0b4h
p3.5: equ 0b5h
p3.6: equ 0b6h
p3.7: equ 0b7h

SOH: EQU 01H
EOT: EQU 04H
NAK: EQU 15H
ACK: EQU 06H
FLAG: EQU 7EH
BS: EQU 08H
FS: EQU 0CH
CR: EQU 0DH
LF: EQU 0AH
EOS: EQU 10H
SYN: EQU 16H
SPACE: EQU 20H

XOFF: EQU 13H
XON: EQU 11H
BREAK: EQU 00H

ORG 0000H
LJMP MAIN_TIMER

ORG 0100H
;HEX TO ASCII CONVERTER
;=====
;ENTRY: A
;EXIT: A
HEX2ASCII: CLR C
MOV R6,A ; Save hex value in R6
SUBB A,#0AH ; if A < 0AH then C
= 1
JC HEX_09 ; else C = 0
HEX_AF: ADD A,#41H ; 41h is "A"
(ascii)
RET
HEX_09: MOV A,R6 ; Return first
value
ADD A,#30H ; 30h is "0" (ascii)
RET
;ASCII TO HEX CONVERTER
;=====
;ENTRY: A ( 30-39 for 0-9, 41-46 for A-F, 61-66 for
a-f )
;EXIT: A
ASCII2HEX: CLR C
MOV R6,A ; Save ascii value in
R6
SUBB A,#61H ; If A < 61h ("a")
then C = 1
JC ASCII_AF ; else C = 0
ADD A,#0AH
RET
ASCII_AF: CLR C
MOV A,R6 ; Return first value
SUBB A,#41H ; If A < 41h ("A")
then C = 1
JC ASCII_09 ; else C = 0
ADD A,#0AH
RET
ASCII_09: CLR C
MOV A,R6
SUBB A,#30H ; If A is a number in
ascii
RET ; Result A = number in
hex
;16 BIT BINARY TO BCD CONVERSION
;=====
;ENTRY: LO_BIN,HI_BIN
;EXIT: LO_BCD,HI_BCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LO_BIN: EQU 2AH
HI_BIN: EQU 2BH
LO_BCD: EQU 2CH
HI_BCD: EQU 2DH
BIN2BCD: MOV LO_BCD,#00H ;
First, Clearing value in memory
MOV HI_BCD,#00H ; of
LO_BCD and HI_BCD
MOV A,LO_BIN
CJNE A,#00H,CONV1 ; Check
LO_BIN = 0 ?
SJMP CHK_HI_BYTE ; If
LO_BIN = 0 then Check HI_BIN
CONV1: MOV R6,LO_BIN ; Put
LO_BIN to R6 for turns of loop
CON: CLR C ; Clear carry flag
MOV A,LO_BCD
ADD A,#01H ; LO_BCD =
LO_BCD + 1
DAA
MOV LO_BCD,A
MOV A,HI_BCD
ADDC A,#00H ; HI_BCD =
HI_BCD + C
DAA
MOV HI_BCD,A
DJNZ R6,CON
CHK_HI_BYTE: MOV A,HI_BIN
CJNE A,#00H,CONV2 ; Check
HI_BIN = 0 ?
RET ; If HI_BIN = 0 then
end
CONV2: DEC HI_BIN ; else
HI_BIN = HI_BIN - 1
MOV R6,#00H ; Put zero to R6
for 256 turns of loop
SJMP CON

*****
***;
; SERVICE ROUTINES FOR TCM3105
;
; 1200BPS FSK MODEM ;
*****
***;
; Designed by Wichit Sirichote
; Modified by Badin Damrongsakd
;
; P3.4 => RX
; P3.5 => TX
; P3.7 => PTT
PTT: EQU P3.7 ; PUSH TO
TALK SIGNAL TO RADIO
RX: EQU P3.4 ; RXD FROM
TCM3105
TX: EQU P3.5 ; TXD TO
TCM3105
TX_NRZI: EQU 14H ; SERIAL
BUFFER NRZI DATA BIT
RXD_NRZI: EQU 21H ; 24H BIT
1 CORRESPONDING TO 0A0H P2.1

BCC: EQU 33H ; BYTE CHECK
SUM 2'S COMPLEMENT OF SUM BYTE
RXD_NRZ: EQU 25H

;SEND_NRZI SEND A TO TxD P3.4 BY
SOFTWARE GENERATED ASYNCHRONOUS
NRZI
;=====
;ENTRY: A
;EXIT: TX_NRZI

SEND_NRZI: SETB RS0 ; select bank 3
SETB RS1
PUSH ACC
CPL TX_NRZI ; INVERT LAST
STATE IN SBUF BECAUSE
; WE'RE GOING TO SEND
START BIT '0'
MOV C,TX_NRZI
MOV TX,C ; SEND TO P1.1 TxD
LCALL DELAY_1200BPS
; THEN D0,D1...D7
MOV R7,#08D
NRZI1: RRC A
JC NO_CHANGE ; NO CHANGE IF
BIT TO BE SENT IS '1'
CPL TX_NRZI ; ELSE 0 BIT TO
BE SENT THEN COMPLEMENT
MOV C,TX_NRZI
MOV TX,C ; SEND TO TxD
SJMP DELAY1200
NO_CHANGE:
MOV C,TX_NRZI
MOV TX,C ; NO CHANGE FOR
'1'
DELAY1200: LCALL DELAY_1200BPS
DJNZ R7,NRZI1
MOV C,TX_NRZI ; SEND STOP
BIT '1'i.e., NO CHANGE
MOV TX,C
LCALL DELAY_1200BPS
LCALL DELAY_1200BPS
CLR RS0 ; Resume bank 0
CLR RS1
POP ACC
RET
;DELAY_1200BPS DELAY 833 MICROSECOND
(1/1200 BIT/S)
DELAY_1200BPS: MOV R5,#02H
DELAY1: MOV R4,#0BDH
DELAY2: DJNZ R4,DELAY2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DJNZ R5,DELAY1
        RET

;DELAY_1200_1.5 DELAY 833 + 833/2
MICROSECOND
DELAY_1200_1.5: MOV R5,#03H
DELAY3:  MOV R4,#0BDH
DELAY4:  DJNZ R4,DELAY4
        DJNZ R5,DELAY3
        RET

;DELAY_1200_0.5 DELAY 833/2
MICROSECOND

DELAY_1200_0.5: MOV R5,#01H
DELAY5:  MOV R4,#0BDH
DELAY6:  DJNZ R4,DELAY6
        DJNZ R5,DELAY5
        RET

;DELAY_PTT DELAY FOR PRESSING PTT
BEFORE SEND MESSAGE (200 ms)

DELAY_PTT: MOV R6,#02D
DELAY9:  MOV R5,#240D
DELAY7:  MOV R4,#0BDH
DELAY8:  DJNZ R4,DELAY8
        DJNZ R5,DELAY7
        DJNZ R6,DELAY9
        RET

;RECEIVES_NRZI
;=====
;ENTRY: -
;EXIT: A

RECEIVES_NRZI:          ; CHECK
START BIT BEFORE FOR
;
SYNCHRONIZATION EACH FRAME
SAMPLING1: SETB RX          ; WRITE
'I' BEFORE THEN READ BACK
        MOV C,RX
        MOV 0E1H,C
        XRL A,24H          ; BIT 1 IN 24H
WE SAVE LAST STATE
        ; OF RXD_NRZI
        JB 0E1H,FOUND_CHANGE
        SJMP SAMPLING1      ; WAIT
UNTIL FOUND SIGNAL CHANGE

FOUND_CHANGE: SETB RX
        MOV C,RX
        MOV RXD_NRZI,C      ; UPDATE
RXD TO NEW VALUE
        LCALL DELAY_1200_1.5 ; GET
DO FIRST

        MOV R7,#08D
GET_NRZI: SETB RX
        MOV C,RX
        MOV 0E1H,C
        XRL A,24H
        JB 0E1H,FOUND_ZERO
        SETB C
        MOV A,RXD_NRZ
        RRC A
        MOV RXD_NRZ,A
        SJMP GET_NEXT_BIT

FOUND_ZERO: SETB RX
        MOV C,RX
        MOV RXD_NRZI,C      ; UPDATE
BEFORE
        CLR C          ; PUT '0' TO RXD_NRZ
        MOV A,RXD_NRZ
        RRC A
        MOV RXD_NRZ,A

GET_NEXT_BIT: LCALL DELAY_1200BPS
        DJNZ R7,GET_NRZI
        RET          ; IGNOR STOP BIT

        LCALL DELAY_1200BPS ; STOP BIT
NO CHANGE LAST STATE
        ; FROM SENDER
        MOV A,RXD_NRZ
        RET

;SEND CR AND LF IN NRZI FORM
;=====
TXD_CRLF_NRZI: MOV A,#CR
        LCALL SEND_NRZI
        MOV A,#LF
        LCALL SEND_NRZI
        RET

;SEND SPACE IN NRZI FORM
;=====
TXD_SPACE_NRZI: MOV A,#SPACE
        LCALL SEND_NRZI
        RET

;SEND STRING IN NRZI FORM
;=====
;ENTRY: DPTR
;EXIT: -
SEND_STRING_NRZI: CLR A
        MOVC A,@A+DPTR
        CJNE A,#EOS,SEND_STRING2
        RET
SEND_STRING2: LCALL SEND_NRZI
        INC DPTR
        SJMP SEND_STRING_NRZI

;SEND BYTE IN NRZI FORM ( with optioned 2's
complements )
;=====
;ENTRY: A
;EXIT: -

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEND_BYTE_NRZI: MOV R4,A ;
Save value to R4
SWAP A ; For send high
nibble first
ANL A,#0FH ; Clear old low
nibble
LCALL HEX2ASCII

push acc ; Sum data in ascii
form
add a,bcc
mov bcc,a
pop acc

LCALL SEND_NRZI
MOV A,R4 ; Return value in
R4 to A
ANL A,#0FH
LCALL HEX2ASCII

push acc
add a,bcc
mov bcc,a
pop acc

LCALL SEND_NRZI
RET

;Send analog data in NRZI digital format to
Central Work Station
=====
;ENTRY: -
;EXIT: -

SEND_FRAME: MOV A,#SOH
LCALL SEND_NRZI
MOV A,#SOH
LCALL SEND_NRZI
LCALL SEND_ALL

mov a,bcc
cpl a
inc a

LCALL SEND_BYTE_NRZI
MOV A,#EOT
LCALL SEND_NRZI
RET

;*****
; SERVICE ROUTINES FOR MAX192 ;
; 8 ch. SERIAL 10-BIT S/A ADC ;
;*****
; P1.7 => DIN (DATA INPUT)
; P1.6 => DOUT (DATA OUTPUT)
; P1.5 => SCLK (SERIAL CLOCK)
; P1.4 => CS (CHIP SELECT ACTIVE "
LOW")
DOUT: EQU P1.6 ; DATA OUTPUT
DIN: EQU P1.7 ; DATA INPUT
SCLK: EQU P1.5 ; Serial Clock
CS: EQU P1.4 ; Chip Select
TB1: EQU 30H ; TRANSMIT BYTE
RB2: EQU 31H ; RECIEVE HIGH BYTE
RB3: EQU 32H ; RECIEVE LOW BYTE

;TRANSMIT INITIAL DATA FOR CONTROL
MAX192
=====
;
; CONTROL-BYTE FORMAT
;
; B7 B6 B5 B4 B3 B2 B1 B0
; STRT SEL2 SEL1 SEL0 U/BI S/DF PDN1
PDN0
;
; NOTE: Sending control-byte at rising edge on SCLK
; Send "start" bit first (MAX192 operate when CS
= low and first logic
; "1" arriving to DIN)
;
; ENTRY: A=CONTROL-BYTE
; EXIT: -

SEND_TB1: CLR C
MOV R7,#08H
RB1: RLC A ; Send MSB first
MOV DIN,C
SETB SCLK
CLR SCLK
DJNZ R7,RB1
RET

;RECEIVE OUTPUT DATA HIGH BYTE (B9-B2)
=====
;
; NOTE: Data is clocked out at the falling edge of
SCLK in MSB-first
; ENTRY: -
; EXIT: A

RECEIVE_RB2: CLR A
CLR C
MOV R7,#08H
REC1: SETB SCLK
CLR SCLK
MOV C,DOUT
RLC A ; Receive MSB first
DJNZ R7,REC1
RET

;RECEIVE OUTPUT DATA LOW BYTE (B1-B0,S1-
S0,4 ZEROS)
=====
;
; NOTE: Data is clocked out at the falling edge of
SCLK in MSB-first

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; S1 and S0 are Sub-LSB
;
;ENTRY: -
;EXIT: A

```

```

RECEIVE_RB3: CLR A
              CLR C
              MOV R7,#08H
REC2:        SETB P1.5
              CLR P1.5
              MOV C,DOUT
              RLC A
              DJNZ R7,REC2
              RET

```

```

;MAX192 10 BIT A/D CONVERSION
;=====
;
;
; NOTE: Step of conversion
; 1. Define value in control-byte
; 2. CS = low
; 3. Send control-byte (TB1)
; 4. Receive RB2
; 5. Receive Rb3
; 6. CS = high
;
;ENTRY: TB1
;EXIT: RB2,RB3 (31H,32H)

```

```

CONV_192:  PUSH ACC
           MOV A,TB1
           CLR CS
           LCALL SEND_TB1
           LCALL RECEIVE_RB2
           MOV RB2,A
           LCALL RECEIVE_RB3
           MOV RB3,A
           SETB CS
           POP ACC
           RET

```

```

;ENTRY: TB1
;EXIT: RB2,RB3 (31H,32H)

```

```

CONV_192:  PUSH ACC
           MOV A,TB1
           CLR CS
           LCALL SEND_TB1
           LCALL RECEIVE_RB2
           MOV RB2,A
           LCALL RECEIVE_RB3
           MOV RB3,A
           SETB CS
           POP ACC
           RET

```

```

;SEND CONVERSION DATA FROM A/D TO
MODEM
;=====
;
;ENTRY: RB2,RB3
;EXIT: -

```

```

;ENTRY: RB2,RB3
;EXIT: -

```

```

SEND_192:  MOV R7,#06H ; This
           loop is turn bit0 - bit7 to
CHANGE:    CLR C ; RB3 and
           bit8 - bit9 to RB2
           MOV A,RB2
           RRC A
           MOV RB2,A
           MOV A,RB3
           RRC A
           MOV RB3,A
           DJNZ R7,CHANGE

```

```

MOV HI_BIN,RB2 ; Put RB2 to
HI_BIN , RB3 to LO_BIN for
MOV LO_BIN,RB3 ; using subroutine
BIN2BCD and MULTI4
LCALL MULTI4 ; Used for
maximum result = int. Vref
LCALL BIN2BCD
MOV A,HI_BCD
LCALL SEND_BYTE_NRZI
MOV A,LO_BCD
LCALL SEND_BYTE_NRZI
RET

```

```

;SCAN CHANNELS OF MAX192 AND KEEP
;=====
;
;ENTRY: TB1
;EXIT: 40H - 4FH
(CH0,CH2,CH4,CH6,CH1,CH3,CH5,CH7)
; CH0 = 40H (KEEP HIGH BYTE) AND 41H
(KEEP LOW BYTE)

```

```

SCAN_192:  CLR A
           MOV R0,#40H
           MOV R7,#08H
SCAN:      MOV B,A ; Keep in B
           SWAP A ; Bit 4-6 are channel
           selecting bit
           ORL A,TB1 ; Change channel of
           adc in bit4 - bit6
           MOV TB1,A ; on control-byte
           MOV A,B ; Return to A
           INC A ; Change channel by
           increment A
           LCALL CONV_192
           MOV @R0,RB2
           INC R0
           MOV @R0,RB3
           INC R0
           DJNZ R7,SCAN
           RET

```

```

;DISPLAY ALL CHANNELS OF MAX192
;=====
;
;ENTRY: 40H-4FH
;EXIT: -

```

```

;ENTRY: 40H-4FH
;EXIT: -

```

```

SEND_ALL:  MOV TB1,#8FH
           LCALL CONV_192
           LCALL SEND_192
           MOV TB1,#0CFH
           LCALL CONV_192
           LCALL SEND_192
           MOV TB1,#9FH
           LCALL CONV_192
           LCALL SEND_192
           MOV TB1,#0DFH
           LCALL CONV_192
           LCALL SEND_192
           MOV TB1,#0AFH
           LCALL CONV_192
           LCALL SEND_192

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV TB1,#0EFH
LCALL CONV_192
LCALL SEND_192
MOV TB1,#0BFH
LCALL CONV_192
LCALL SEND_192
MOV TB1,#0FFH
LCALL CONV_192
LCALL SEND_192
RET

;16 BIT MULTIPLY BY 4 FOR CONVERT
1.024V TO 4.096V (REAL VALUE)
;=====
;ENTRY: LO_BIN,HI_BIN
;EXIT: LO_BIN,HI_BIN

CARRY: EQU 20H

MULTI4: push b
MOV A,LO_BIN
MOV B,#04H
MUL AB
MOV LO_BIN,A
MOV CARRY,B ; Carry of
multiply on low byte must be
MOV A,HI_BIN ; keep for plus
on high byte
MOV B,#04H
MUL AB
ADD A,CARRY
MOV HI_BIN,A
pop b
RET

;=====
TEST13: MOV R0,#50H ; ADDRESS
TO PUT COMMAND TO MEMORY

TEST14: LCALL RECEIVES_NRZI
CJNE A,#SOH,TEST14 ; WAIT UNTIL
FOUND START OF HEAD
LCALL RECEIVES_NRZI ; IF THE
SECOND BYTE WAS NOT SOH
; THEN LOOP
CJNE A,#SOH,TEST14 ; UNTIL
FOUND TWO CONSECUTIVE BYTE
; OF SOH
; FOUND SOH SOH THEN
GET BYTE UNTIL EOT
TEST15: LCALL RECEIVES_NRZI
; DO UNTIL FOUND EOT
CJNE A,#EOT,TEST16 ;
SJMP CHK_ADDR ; GO OUT LOOP

TEST16: MOV @R0,A
INC R0
SJMP TEST15
;Remark Ascii Format
;=====
;50h is @
;51h is address (0-7)
;52h is main command (R,L,S for SE)
;53h is step (1-4,E for SE)
;54h is check sum byte

;CHKSUM_CMD: MOV R2,#04H
; MOV R1,#50H
; CLR A
;CHECK_CMD: ADD A,@R1
; INC R1
; DJNZ R2,CHECK_CMD
; CJNE A,54H,ERROR
; SJMP CHK_ADDR
;ERROR: LJMP TEST13

;PI.0 - P1.2 is assumed to hardware address
;PORT 1.0 1.1 1.2
; 0 0 0 -> @0
; 1 0 0 -> @1
; 0 1 0 -> @2
; 1 1 0 -> @3
; 0 0 1 -> @4
; 1 0 1 -> @5
; 0 1 1 -> @6
; 1 1 1 -> @7

CHK_ADDR: MOV A,51H
ANL A,#00000111B
MOV 51H,A
MOV A,P1 ; READ HARDWARE
ADDRESS FROM PORT 1
ANL A,#00000111B ; ALSO READ
P1.0-P1.2
CJNE A,51H,TEST13 ; CHECK
ADDRESS WITH 51H
; IF ADDRESS TRUE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ; CONVERT SIGNAL  
FROM 8 CHANNELS AND  
                ; SEND PACKET TO  
RF MODEM
```

```
CHK_COMMAND: MOV  A,52H  
              CJNE A,#"r",COMMAND1  
              SETB P3.0  
              LJMP TEST13  
COMMAND1:    CJNE  
A,#"l",START_SEND  
              CLR  P3.0  
              LJMP TEST13
```

```
START_SEND:  MOV  B,#03H
```

```
RETURN_SEND: MOV  BCC,#00H  
            LCALL DELAY_PTT  
            SETB PTT  
            LCALL DELAY_PTT  
            LCALL SEND_FRAME  
            LCALL DELAY_PTT  
            CLR  PTT  
            LCALL DELAY_PTT  
            DJNZ B,RETURN_SEND  
            LJMP TEST13  
            END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#####
; RF Modem ;
; RFM-01W ;
#####
;Designed and Developed by Badin
Damrongsakd
;29th Jan 1997
;Modified by Badin Damrongsakd
;13rd Mar 1997
;Hardware: AT89c2051 ( 8-Bit Microcontroller
with 2KBytes Flash PEROM )
;Update: DS275 ( Line-Powered RS-232
Transceiver Chip )
; TCM3105 ( 1200 BPS FSK Modem )

CPU "8051.TBL"
HOF "INT8"
;MCS-51 INTERNAL REGISTERS
;
B: EQU 0F0H ;B REGISTER
ACC: EQU 0E0H ;ACCUMULATOR
PSW: EQU 0D0H ;PROGRAM
STATUS WORD
IPC: EQU 0B8H ;INTERRUPT
PRIORITY
P3: EQU 0B0H ;PORT 3
IEC: EQU 0A8H ;INTERRUPT
ENABLE
P2: EQU 0A0H ;PORT 2
SBUF: EQU 99H ;SEND BUFFER
SCON: EQU 98H ;SERIAL CONTROL
P1: EQU 90H ;PORT 1
TH1: EQU 8DH ;TIMER 1 HIGH
TH0: EQU 8CH ;TIMER 0 HIGH
TL1: EQU 8BH ;TIMER 1 LOW
TL0: EQU 8AH ;TIMER 0 LOW
TMOD: EQU 89H ;TIMER MODE
TCON: EQU 88H ;TIMER CONTROL
PCON: EQU 87H ;POWER CONTROL
REGISTER
DPH: EQU 83H ;DATA POINTER
HIGH
DPL: EQU 82H ;DATA POINTER
LOW
SP: EQU 81H ;STACK POINTER
P0: EQU 80H ;PORT 0
;
;MCS-51 INTERNAL BIT ADDRESSES
;
CY: EQU 0D7H ;CARRY FLAG
AC: EQU 0D6H ;AUXILIARY-
CARRY FLAG
F0: EQU 0D5H ;USER FLAG 0
RS1: EQU 0D4H ;REGISTER
SELECT MSB
RS0: EQU 0D3H ;REGISTER
SELECT LSB
OV: EQU 0D2H ;OVERFLOW
FLAG
P: EQU 0D0H ;PARITY FLAG

PS: EQU 0BCH ;PRIORITY SERIAL
PORT
PT1: EQU 0BBH ;PRIORITY TIMER 1
PX1: EQU 0BAH ;PRIORITY
EXTERNAL 1
PT0: EQU 0B9H ;PRIORITY TIMER 0
PX0: EQU 0B8H ;PRIORITY
EXTERNAL 0
EA: EQU 0AFH ;ENABLE ALL
INTERRUPT
ES: EQU 0ACH ;ENABLE SERIAL
INTERRUPT
ET1: EQU 0ABH ;ENABLE TIMER 1
INTERRUPT
EX1: EQU 0AAH ;ENABLE EXTERNAL
1 INTERR
ET0: EQU 0A9H ;ENABLE TIMER 0
INTERRUPT
EX0: EQU 0A8H ;ENABLE EXTERNAL
0 INTERR
SM0: EQU 09FH ;SERIAL MODE 0
SM1: EQU 09EH ;SERIAL MODE 1
SM2: EQU 09DH ;SERIAL MODE 2
REN: EQU 09CH ;SERIAL RECEPTION
ENABLE
TB8: EQU 09BH ;TRANSMITT BIT 8
RB8: EQU 09AH ;RECEIVE BIT 8
TI: EQU 099H ;TRANSMIT
INTERRUPT FLAG
RI: EQU 098H ;RECEIVE INTERRUPT
FLAG
TF1: EQU 08FH ;TIMER 1 OVERFLOW
FLAG
TR1: EQU 08EH ;TIMER 1 RUN
CONTROL BIT
TF0: EQU 08DH ;TIMER 0 OVERFLOW
FLAG
TR0: EQU 08CH ;TIMER 0 RUN
CONTROL BIT
IE1: EQU 08BH ;EXT INTERR. 1 EDGE
FLAG
IT1: EQU 08AH ;EXT INTERR. 1 TYPE
FLAG
IE0: EQU 089H ;EXT INTERR. 0 EDGE
FLAG
IT0: EQU 088H ;EXT INTERR. 0 TYPE
FLAG

P0.0: EQU 080H ;
P0.1: EQU 081H ;
P0.2: EQU 082H ;
P0.3: EQU 083H ;
P0.4: EQU 084H ;
P0.5: EQU 085H ;
P0.6: EQU 086H ;
P0.7: EQU 087H ;

P1.0: EQU 090H ;Rxd_NRZI
P1.1: EQU 091H ;Tx_NRZI
P1.2: EQU 092H ;
P1.3: EQU 093H ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P1.4: EQU 094H ; ;SEND A TO RS-232
P1.5: EQU 095H ; ;
P1.6: EQU 096H ; OPTIONS ;ENTRY: A
P1.7: EQU 097H ; OPTIONS ;EXIT: -

p2.0: equ 0a0h SEND: JNB TI,SEND
p2.1: equ 0a1h CLR TI
p2.2: equ 0a2h MOV SBUF,A
p2.3: equ 0a3h RET
p2.4: equ 0a4h
p2.5: equ 0a5h
p2.6: equ 0a6h ;RECEIVE FROM RS-232
p2.7: equ 0a7h ;
;ENTRY: -
;EXIT: A

p3.0: equ 0b0h RECEIVE: JNB RI,RECEIVE
p3.1: equ 0b1h CLR RI
p3.2: equ 0b2h MOV A,SBUF
p3.3: equ 0b3h RET
p3.4: equ 0b4h
p3.5: equ 0b5h
p3.6: equ 0b6h ;RECEIVE FROM RS-232 WITH ECHO
p3.7: equ 0b7h ;
;ENTRY: -
;EXIT: -

SOH: EQU 01H GET_ECHO: LCALL RECEIVE
EOT: EQU 04H LCALL SEND
NAK: EQU 15H RET
ACK: EQU 06H ;SEND "CR" AND "LF" TO RS-232
;
;ENTRY: -
;EXIT: -

BS: EQU 08H TXD_CRLF: MOV A,#CR
FS: EQU 0CH LCALL SEND
CR: EQU 0DH MOV A,#LF
LF: EQU 0AH LCALL SEND
EOS: EQU 10H RET
SYN: EQU 16H
SPACE: EQU 20H
XOFF: EQU 13H
XON EQU 11H
BREAK: EQU 00H

ORG 0000H
LJMP MAIN_TIMER

ORG 0100H ;SEND STRING FROM TABLE TO RS-232
;
;ENTRY: DPTR ( Point to define string )
;EXIT: -

;INITIAL RS-232 COMMUNICATIONS
;
;ENTRY: -
;EXIT: -
SEND_STRING: CLR A
MOV A,@A+DPTR
CJNE A,#EOS,SEND_STRING1
RET
SEND_STRING1: LCALL SEND
INC DPTR
SJMP SEND_STRING

INIT_RS232: MOV TMOD,#00100001B
MOV SCON,#01010010B ;SEND SPACE TO RS-232
MOV TH1,#0FDH ;
SETB TR1 ;ENTRY: -
RET ;EXIT: -

; BAUD RATE: 9600 BIT/SEC
; DATA LENGTH: 8 BIT
; STOP BIT: 1 BIT
; PARITY: NO
; X-TAL: 11.0592 MHz

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TXD_SPACE:  MOV  A,#SPACE          JC  ASCII_09    ; else C = 0
             LCALL SEND              ADD  A,#0AH
             RET                      RET
;RECEIVE BYTE OF ASCII FROM RS-232 (
;recieve 2 ascii numbers )
;=====
;ENTRY: -
;EXIT: A
;=====

GET_BYTE:   LCALL GET_ECHO    ; or
LCALL RECEIVE
           LCALL ASCII2HEX
           SWAP  A
           MOV  R5,A        ; R5 RESTORE

HIGH_NIBBLE:
RECEIVE    LCALL GET_ECHO    ; or LCALL
           LCALL ASCII2HEX
           ADD  A,R5
           RET

;HEX TO ASCII CONVERTER
;=====
;ENTRY: A
;EXIT: A

HEX2ASCII: CLR  C
           MOV  R6,A        ; Save hex value
           in R6
           SUBB A,#0AH      ; if A < 0AH
           then C = 1
           JC  HEX_09      ; else C = 0
HEX_AF:    ADD  A,#41H      ; 41h is "
A" (ascii)
           RET
HEX_09:    MOV  A,R6        ; Return first
value
           ADD  A,#30H      ; 30h is "0"
(ascii)
           RET

;ASCII TO HEX CONVERTER
;=====
;ENTRY: A ( 30-39 for 0-9, 41-46 for A-F, 61-
66 for a-f)
;EXIT: A

ASCII2HEX: CLR  C
           MOV  R6,A        ; Save ascii value
           in R6
           SUBB A,#61H      ; If A < 61h ("
a") then C = 1
           JC  ASCII_AF     ; else C = 0
           ADD  A,#0AH
           RET
ASCII_AF:  CLR  C
           MOV  A,R6        ; Return first
value
           SUBB A,#41H      ; If A < 41h ("
A") then C = 1

ASCII_09:  CLR  C
           MOV  A,R6
           SUBB A,#30H      ; If A < 30h ("
0") then C = 1
           JC  ASCII_09     ; else C = 0
           ADD  A,#0AH
           RET

SEND_BYTE: MOV  R4,A        ; Save value
to R4
           SWAP  A          ; For send high nibble
           first
           ANL  A,#0FH      ; Clear old low nibble
           LCALL HEX2ASCII
           LCALL SEND
           MOV  A,R4        ; Return value in R4 to
A
           ANL  A,#0FH
           LCALL HEX2ASCII
           LCALL SEND
           RET

;SEND COMMAND PROMPT TO RS-232
;=====
;ENTRY: -
;EXIT: -
SEND_PROMPT: MOV A,#">"
            LCALL SEND
            RET

;16 BIT BINARY TO BCD CONVERSION
;=====
;ENTRY: LO_BIN,HI_BIN
;EXIT: LO_BCD,HI_BCD

LO_BIN:    EQU  2AH
HI_BIN:    EQU  2BH
LO_BCD:    EQU  2CH
HI_BCD:    EQU  2DH

BIN2BCD:   MOV  LO_BCD,#00H ; First,
Clearing value in memory
           MOV  HI_BCD,#00H ; of LO_BCD
and HI_BCD
           MOV  A,LO_BIN
           CJNE A,#00H,CONV1 ; Check
LO_BIN = 0 ?
           SJMP CHK_HI_BYTE ; If LO_BIN = 0
then Check HI_BIN
CONV1:     MOV  R6,LO_BIN ; Put
LO_BIN to R6 for turns of loop
CON:       CLR  C          ; Clear carry flag
           MOV  A,LO_BCD
           ADD  A,#01H      ; LO_BCD =
LO_BCD + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DAA
MOV LO_BCD,A
MOV A,HI_BCD
ADDC A,#00H ; HI_BCD =
HI_BCD + C
DAA
MOV HI_BCD,A
DJNZ R6,CON
CHK_HI_BYTE: MOV A,HI_BIN
CJNE A,#00H,CONV2 ; Check
HI_BIN = 0 ?
RET ; If HI_BIN = 0 then
end
CONV2: DEC HI_BIN ; else
HI_BIN = HI_BIN - 1
MOV R6,#00H ; Put zero to R6
for 256 turns of loop
SJMP CON
;FIND_CHKSUM BYTE ROUTINE ( ADD
DATA AND 2'COMPLEMENT )
;=====
;ENTRY: R0,R7
;EXIT: BCC
; R0 is a address pointer
; R7 is a number of sum data

FIND_CHKSUM: CLR A
CHKSUM: ADD A,@R0
INC R0
DJNZ R7,CHKSUM
CPL A ; 1's complement
INC A ; add 1 for made it to
2's complement
MOV BCC,A
RET

;*****
;***;
; SERVICE ROUTINES FOR TCM3105
;
; 1200BPS FSK MODEM ;
;*****
;***;
; Designed by Wichit Sirichote
; Modified by Badin Damrongsakd
;
; P1.0 => RX
; P1.1 => TX
; P3.7 => PTT

PTT: EQU P3.7 ; PUSH TO
TALK SIGNAL TO RADIO
RX: EQU P1.0 ; RXD FROM
TCM3105
TX: EQU P1.1 ; TXD TO
TCM3105
TX_NRZI: EQU 14H ; SERIAL
BUFFER NRZI DATA BIT
RXD_NRZI: EQU 21H ; 24H BIT
1 CORRESPONDING TO 0A0H P2.1

BCC: EQU 53H ; BYTE CHECK
SUM 2'S COMPLEMENT OF SUM BYTE
RXD_NRZI: EQU 25H

;SEND_NRZI SEND A TO TxD P1.1 BY
SOFTWARE GENERATED ASYNCHRONOUS
NRZI
;=====
;ENTRY: A
;EXIT: TX_NRZI

SEND_NRZI: SETB RS0 ; select bank 3
SETB RS1
PUSH ACC
CPL TX_NRZI ; INVERT LAST
STATE IN SBUF BECAUSE
; WE'RE GOING TO SEND
START BIT '0'
MOV C,TX_NRZI
MOV TX,C ; SEND TO P1.1 TxD

LCALL DELAY_1200BPS
; THEN D0,D1...D7
MOV R7,#08D
NRZI1: RRC A
JC NO_CHANGE ; NO CHANGE IF
BIT TO BE SENT IS '1'
CPL TX_NRZI ; ELSE 0 BIT TO
BE SENT THEN COMPLEMENT
MOV C,TX_NRZI
MOV TX,C ; SEND TO TxD
SJMP DELAY1200

NO_CHANGE:
MOV C,TX_NRZI
MOV TX,C ; NO CHANGE FOR
'1'

DELAY1200: LCALL DELAY_1200BPS
DJNZ R7,NRZI1
MOV C,TX_NRZI ; SEND STOP
BIT '1'i.e., NO CHANGE
MOV TX,C
LCALL DELAY_1200BPS
LCALL DELAY_1200BPS
CLR RS0 ; Resume bank 0
CLR RS1
POP ACC

RET

;=====
;ENTRY: -
;EXIT: -

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY_1200BPS: MOV R5,#02H
DELAY1:  MOV R4,#0BDH
DELAY2:  DJNZ R4,DELAY2
          DJNZ R5,DELAY1
          RET

```

```

;DELAY_1200_1.5 DELAY 833 + 833/2
MICROSECOND

```

```

;ENTRY: -
;EXIT: -
DELAY_1200_1.5: MOV R5,#03H
DELAY3:  MOV R4,#0BDH
DELAY4:  DJNZ R4,DELAY4
          DJNZ R5,DELAY3
          RET

```

```

;DELAY_1200_0.5 DELAY 833/2
MICROSECOND

```

```

;ENTRY: -
;EXIT: -
DELAY_1200_0.5: MOV R5,#01H
DELAYS:  MOV R4,#0BDH
DELAY6:  DJNZ R4,DELAY6
          DJNZ R5,DELAYS
          RET

```

```

;DELAY_PTT DELAY FOR PRESSING PTT
BEFORE SEND MESSAGE

```

```

;ENTRY: -
;EXIT: -
DELAY_PTT:  MOV R6,#03D
DELAY9:  MOV R5,#240D
DELAY7:  MOV R4,#0BDH
DELAY8:  DJNZ R4,DELAY8
          DJNZ R5,DELAY7
          DJNZ R6,DELAY9
          RET

```

```

;RECEIVES_NRZI ( OPTION TIME OUT )

```

```

;ENTRY: -
;EXIT: A
RECEIVES_NRZI: CLR F0
                MOV R2,#00H
                MOV R3,#00H
                MOV R6,#00H
                ; CHECK START BIT
BEFORE FOR
                ;
SYNCHRONIZATION EACH FRAME
SAMPLING1: SETB RX ; WRITE
'1' BEFORE THEN READ BACK

```

```

MOV C,RX
MOV 0E1H,C ; 0E1H IS BIT 1 IN
24H
XRL A,24H ; BIT 1 IN 24H WE
SAVE LAST STATE
                ; OF RXD_NRZI
JB 0E1H,FOUND_CHANGE

```

```

; OPTION HERE

```

```

;-----
INC R2
CJNE R2,#00H,TIMED
INC R3
CJNE R3,#00,TIMED
INC R6
CJNE R6,#0AH,TIMED
SETB F0 ; IF F0=1 INDICATED

```

```

TIME OUT

```

```

RET
; END OPTION

```

```

TIMED: SJMP SAMPLING1 ; WAIT
UNTIL FOUND SIGNAL CHANGE
FOUND_CHANGE: SETB RX
                MOV C,RX
                MOV RXD_NRZI,C ; UPDATE RXD
TO NEW VALUE
LCALL DELAY_1200_1.5 ; GET D0
FIRST

```

```

MOV R7,#08D
GET_NRZI: SETB RX
MOV C,RX
MOV 0E1H,C
XRL A,24H
JB 0E1H,FOUND_ZERO
SETB C
MOV A,RXD_NRZ
RRC A
MOV RXD_NRZ,A
SJMP GET_NEXT_BIT

```

```

FOUND_ZERO: SETB RX
MOV C,RX
MOV RXD_NRZI,C ; UPDATE

```

```

BEFORE
CLR C ; PUT '0' TO RXD_NRZ
MOV A,RXD_NRZ
RRC A
MOV RXD_NRZ,A

```

```

GET_NEXT_BIT: LCALL DELAY_1200BPS
DJNZ R7,GET_NRZI
RET ; IGNOR STOP BIT
LCALL DELAY_1200BPS ; STOP BIT
NO CHANGE LAST STATE
                ; FROM SENDER
MOV A,RXD_NRZ
RET
;SEND CR AND LF IN NRZI FORM

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;=====
;ENTRY: -
;EXIT: -

TXD_CRLF_NRZI: MOV A,#CR
                LCALL SEND_NRZI
                MOV A,#LF
                LCALL SEND_NRZI
                RET

;SEND SPACE IN NRZI FORM
;=====
;ENTRY: -
;EXIT: -

TXD_SPACE_NRZI: MOV A,#SPACE
                LCALL SEND_NRZI
                RET

;SEND STRING IN NRZI FORM
;=====
;ENTRY: DPTR
;EXIT: -

SEND_STRING_NRZI: CLR A
                  MOV A,@A+DPTR
                  CJNE A,#EOS,SEND_STRING2
                  RET
SEND_STRING2: LCALL SEND_NRZI
              INC DPTR
              SJMP SEND_STRING_NRZI
;SEND BYTE IN NRZI FORM
;=====
;ENTRY: A
;EXIT: -

SEND_BYTE_NRZI: MOV R4,A ;
                Save value to R4
                SWAP A ; For send high
                nibble first
                ANL A,#0FH ; Clear old low
                nibble
                LCALL HEX2ASCII
                LCALL SEND_NRZI
                MOV A,R4 ; Return value in
                R4 to A
                ANL A,#0FH
                LCALL HEX2ASCII
                LCALL SEND_NRZI
                RET

;SEND COMMAND TO SENDER
;=====
;ENTRY: -
;EXIT: -

SEND_COMMAND: MOV R0,#30H
               MOV R7,#04H
COMMAND: MOV A,@R0
          LCALL SEND_NRZI
          INC R0

DINZ R7,COMMAND
RET

#####
;# MAIN PROGRAM #
#####

MAIN_TIMER:
                MOV R0,#00H
POWER_DELAY: DJNZ R0,POWER_DELAY

                MOV SP,#60H ; FIRST, STACK
                MUST BE INITIALIZED
                SETB EA
                CLR F0
                LCALL INIT_RS232

                SETB TX_NRZI ; Initialize
                TX_NRZI once
                SETB RXD_NRZI
                CLR PTT

CHK_CONNECT: LCALL RECEIVE ;
              CONNECTED WITH SOFTWARE CHECKING
              CJNE A,#SPACE,CHK_CONNECT
              MOV A,#"" ; RETURN "" TO
              SOFTWARE FOR HARDWARE
              LCALL SEND ; CONNECTED
              AND RECEIVED SPACE KEY

MAIN:
                MOV R0,#30H ; SAVED FIRST
                COMMAND TO ADDRESS 30H
                MOV R7,#04H
RECV_CMD: LCALL RECEIVE
           LCALL SEND ; Remarkd when
           use windows's software
           MOV @R0,A
           INC R0
           DJNZ R7,RECV_CMD
           MOV R0,#30H
           MOV R7,#04H
           LCALL FIND_CHKSUM

TEST0: SETB PTT
        LCALL DELAY_PTT
        MOV A,#SOH
        LCALL SEND_NRZI
        MOV A,#SOH
        LCALL SEND_NRZI
        LCALL SEND_COMMAND
        MOV A,BCC
        LCALL SEND_NRZI
        MOV A,#EOT
        LCALL SEND_NRZI
        LCALL DELAY_PTT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR PTT
LCALL DELAY_PTT

TEST14:  LCALL RECEIVES_NRZI
        JNB F0,NON_TIME_OUT
TIME_OUT: MOV DPTR,#TIMEOUT
        LCALL SEND_STRING
        CLR F0
        AJMP MAIN

NON_TIME_OUT: JB P1.7,TEST16 ;IF
P1.7 = 1 THEN SKIP SOH CHECKING

        CJNE A,#SOH,TEST14 ;WAIT
UNTIL FOUND START OF HEAD
        LCALL RECEIVES_NRZI ;IF THE
SECOND BYTE WAS NOT SOH
        ; THEN LOOP
        CJNE A,#SOH,TEST14 ;UNTIL
FOUND TWO CONSECUTIVE BYTE
        ; OF SOH
        ; FOUND SOH SOH
THEN GET BYTE UNTIL EOT
        MOV A,#"." ;"." IS HEADER
TO SOFTWARE
        LCALL SEND

TEST15:  LCALL RECEIVES_NRZI
        ; DO UNTIL FOUND
EOT
        CJNE A,#EOT,TEST16 ;
LJMP MAIN ; LOOP MAIN

TEST16:  LCALL SEND
        JB P1.7,TEST14 ;IF P1.7 = 1
THEN GET RAW SERIAL DATA
        SJMP TEST15

;STRING CONSTANTS
;=====

TIMEOUT:  DFB "*****"
Timeout*****",EOS

END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' PROGRAM DISPLAY GRAPH FOR  
WORKSTATION  
' PROGRAMER VARALEE TOWANABOOT
```

```
Option Explicit  
Dim X, Y, n, Q, a As Integer  
Dim link As String  
Dim style
```

```
Sub Cmdokg_Click ()  
Form1.Enabled = True  
Unload form3  
Form1.Show  
End Sub
```

```
Sub Form_Load ()  
Dim a  
a = Form1.Datalist(s).ListCount - 7  
Form1.Enabled = False  
hscroll1.Min = 0  
hscroll1.Max = a  
'MsgBox a  
End Sub
```

```
Sub HScroll1_Change ()  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub marea_Click ()  
Graph1.GraphType = 8  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mbar2d_Click ()  
Graph1.GraphType = 3  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mbar3d_Click ()  
Graph1.GraphType = 4  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mline_Click ()  
Graph1.GraphType = 6  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mpie2d_Click ()  
Graph1.GraphType = 1  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mpie3d_Click ()  
Graph1.GraphType = 2  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub mscatter_Click ()  
Graph1.GraphType = 9  
If Opch1.Value = True Then Plot (12)  
If Opch2.Value = True Then Plot (16)  
If Opch3.Value = True Then Plot (20)  
If Opch4.Value = True Then Plot (24)  
If Opch5.Value = True Then Plot (28)  
If Opch6.Value = True Then Plot (32)  
If Opch7.Value = True Then Plot (36)  
If Opch8.Value = True Then Plot (40)  
End Sub
```

```
Sub opCh1_Click (Value As Integer)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 1 "
Graph1.LeftTitle = " Celcuis "
'q = 12
Plot (12)
End Sub

Sub opCh2_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 2 "
Graph1.LeftTitle = " Celcuis "
Plot (16)
End Sub

Sub opCh3_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 3 "
Graph1.LeftTitle = " Celcuis "
Plot (20)
End Sub

Sub opch4_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 4 "
Graph1.LeftTitle = " Celcuis "
Plot (24)
End Sub

Sub opch5_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 5 "
Graph1.LeftTitle = " Celcuis "
Plot (28)
End Sub

Sub opch6_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 6 "
Graph1.LeftTitle = " Celcuis "
Plot (32)
End Sub

Sub opch7_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 7 "
Graph1.LeftTitle = " Celcuis "
Plot (36)
End Sub

Sub opch8_Click (Value As Integer)
Graph1.NumSets = 1
Graph1.NumPoints = 7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 8 "
Graph1.LeftTitle = " Celcuis "
Plot (40)
End Sub

Sub Plot (Q As Integer)
Dim n, i As Integer
Dim Timex As String
n = 0
For i = 1 To 7
Graph1.ThisPoint = i
Timex = Left$(Form1.Datalist(s).List(hscroll1.Value
+ n), 8)
Graph1.LabelText = Timex
Graph1.ThisPoint = i
Graph1.GraphData = Val(Mid$(Form1.Datalist(s).
List(hscroll1.Value + n), Q, 4))
n = n + 1
Next i
Graph1.DrawMode = 2
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' PROGRAM DATABASE FOR
WORKSTATION
' PROGRAMER VARALEE TOWANABOOT
' DATABASE TABLE OBJECT
```

```
Dim Q As Integer
Sub ClearGrid ()
Grid1.SelStartCol = 0
Grid1.SelStartRow = 1
Grid1.SelEndCol = Grid1.Cols - 1
Grid1.SelEndRow = Grid1.Rows - 1
Grid1.FillStyle = 1
Grid1.Text = " "
Grid1.FillStyle = 0
Grid1.SelEndCol = 0
Grid1.SelEndRow = 1
End Sub
```

```
Sub Cmdreport_Click ()
form11.Show
Form8.Enabled = False
End Sub
```

```
Sub CmdReturn_Click ()
Unload Form8
Form9.Show
End Sub
```

```
Sub Combo1_Click ()
Dim N As Integer
Cn = Form8.Combo1.Text
Graph1.Visible = False
Hscroll1.Visible = False
opch1.Value = False
opch2.Value = False
opch3.Value = False
opch4.Value = False
opch5.Value = False
opch6.Value = False
opch7.Value = False
opch8.Value = False
'ClearGrid
LoadToGrid
Framech.Enabled = True
'Set Graph
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If
```

```
Hscroll1.Min = 0
Hscroll1.Max = Cd(Val(Cn)) - N
End Sub
```

```
Sub Form_Load ()
Grid1.Row = 0
Grid1.RowHeight(0) = 300
Grid1.Col = 0
Grid1.ColWidth(0) = 600
Grid1.Text = "Date"
Grid1.Col = 1
```

```
Grid1.ColWidth(1) = 900
Grid1.Text = " Time "
```

```
For I = 2 To 9
Grid1.ColWidth(I) = 800
Grid1.Col = I
Grid1.Text = " Ch " & I - 1
Next
Grid1.Col = 10
Grid1.ColWidth(10) = 750
Grid1.Text = " Check "
```

```
'Check Sender
If S = 0 Then pSender.Caption = " Welcome To
Sender 1 DataBase"
If S = 1 Then pSender.Caption = " Welcome To
Sender 2 DataBase"
If S = 2 Then pSender.Caption = " Welcome To
Sender 3 DataBase"
Cmdreport.Enabled = False
End Sub
```

```
Sub HScroll1_Change ()
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub
```

```
Sub m2dbar_Click ()
Graph1.GraphType = 3
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub
```

```
Sub m2dPie_Click ()
Graph1.GraphType = 1
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub
```

```
Sub m3dbar_Click ()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Graph1.GraphType = 4
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub

```

```

Sub M3dpie_Click ()
Graph1.GraphType = 2
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub

```

```

Sub marea_Click ()
Graph1.GraphType = 8
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub

```

```

Sub MExit_Click ()
End
End Sub

```

```

Sub Minput_Click ()
Form10.Show
End Sub

```

```

Sub mline_Click ()
Graph1.GraphType = 6
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)
If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub

```

```

Sub mscatter_Click ()
Graph1.GraphType = 9
If opch1.Value = True Then Plot (2)
If opch2.Value = True Then Plot (3)
If opch3.Value = True Then Plot (4)
If opch4.Value = True Then Plot (5)

```

```

If opch5.Value = True Then Plot (6)
If opch6.Value = True Then Plot (7)
If opch7.Value = True Then Plot (8)
If opch8.Value = True Then Plot (9)
End Sub

```

```

Sub opCh1_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If

```

```

Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N * 7 Cd(Val(Cn))
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 1 "
'Graph1.LeftTitle = " Celcuis "
Plot (2)
End Sub

```

```

Sub opCh2_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If

```

```

Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 2 "
'Graph1.LeftTitle = " Celcuis "
Plot (3)
End Sub

```

```

Sub opCh3_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If

```

```

Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 3 "
'Graph1.LeftTitle = " Celcuis "
Plot (4)
End Sub

```

```

Sub opch4_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else N = 7
End If
Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 4 "
'Graph1.LeftTitle = " Celcuis "
Plot (5)
End Sub

```

```

Sub opch5_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If

```

```

Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 5 "
'Graph1.LeftTitle = " Celcuis "
Plot (6)
End Sub

```

```

Sub opch6_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If
Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 6 "
'Graph1.LeftTitle = " Celcuis "
Plot (7)
End Sub

```

```

Sub opch7_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If
Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 7 "
'Graph1.LeftTitle = " Celcuis "
Plot (8)
End Sub

```

```

Sub opch8_Click (Value As Integer)
Dim N As Integer
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If
Graph1.Visible = True
Hscroll1.Visible = True
Graph1.NumSets = 1
Graph1.NumPoints = N*7
Graph1.ThisSet = 1
Graph1.GraphTitle = " Channel 8 "
'Graph1.LeftTitle = " Celcuis "
Plot (9)
End Sub

```

```

Sub Plot (Q As Integer)
Dim I, N As Integer
Dim Timex As String
If Cd(Val(Cn)) < 7 Then
N = Cd(Val(Cn))
Else N = 7
End If
For I = 1 To N '7 Cd(Val(Cn))
' MsgBox Cd(Val(Cn))
Grid1.Row = I + Hscroll1.Value
Graph1.ThisPoint = I
Grid1.Col = 1
Timex = Grid1.Text
Graph1.LabelText = Timex
Graph1.ThisPoint = I
Grid1.Col = Q
Graph1.GraphData = Grid1.Text
Next I
Graph1.DrawMode = 2
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' MAIN PROGRAM FOR
WORKSTATION
' PROGRAMER :) VARALEE
TOWANABOOT
```

```
Option Explicit
Dim gettime As Variant
Dim Code() As String
Dim X, N(), L1, L2, L3 As Integer 'for reciver &
clear
Dim S1, S2, S3 'auto
```

```
Sub Checkadd1 ()
Dim X As String, filename, Z, Y As Integer
X = Dir$(Address1_FILE)
If X = "" Then
address1 = "@1"
Exit Sub
End If
filename = FreeFile
Open Address1_FILE For Binary As filename
address1 = ""
Z = 1
While Not EOF(filename)
Get #filename, , Y
address1 = address1 + Chr$(Y + Z)
Z = Z + 1
Wend
address1 = Left$(address1, Len(address1) - 1)
Close #filename
End Sub
```

```
Sub Checkadd2 ()
Dim X As String, filename, Z, Y As Integer
X = Dir$(Address2_FILE)
If X = "" Then
address2 = "@2"
Exit Sub
End If
filename = FreeFile
Open Address2_FILE For Binary As filename
address2 = ""
Z = 1
While Not EOF(filename)
Get #filename, , Y
address2 = address2 + Chr$(Y + Z)
Z = Z + 1
Wend
address2 = Left$(address2, Len(address2) - 1)
Close #filename
End Sub
```

```
Sub Checkadd3 ()
Dim X As String, filename, Z, Y As Integer
X = Dir$(Address3_FILE)
If X = "" Then
address3 = "@3"
Exit Sub
End If
filename = FreeFile
Open Address3_FILE For Binary As filename
```

```
address3 = ""
Z = 1
While Not EOF(filename)
Get #filename, , Y
address3 = address3 + Chr$(Y + Z)
Z = Z + 1
Wend
address3 = Left$(address3, Len(address3) - 1)
Close #filename
End Sub
```

```
Sub ClearGrid ()
N(S) = 1
Grid1(S).SelStartCol = 0
Grid1(S).SelStartRow = 1
Grid1(S).SelEndCol = Grid1(S).Cols - 1
Grid1(S).SelEndRow = Grid1(S).Rows - 1
Grid1(S).FillStyle = 1
Grid1(S).Text = " "
Grid1(S).FillStyle = 0
Grid1(S).SelEndCol = 0
Grid1(S).SelEndRow = 1
End Sub
```

```
Sub Cmdgraph1_Click ()
S = 0
Form3.Show
End Sub
```

```
Sub Cmdgraph2_Click ()
S = 1
Form3.Show
End Sub
```

```
Sub Cmdgraph3_Click ()
S = 2
Form3.Show
End Sub
```

```
Sub Cmdload1_Click ()
S = 0
'On Error GoTo errhand
cmdialog1.Filter = "All Files (*.*)*.*.|Code files (*.
cod)|*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.DialogTitle = " Load Data "
cmdialog1.Action = 1
Filename = cmdialog1.FileName
OpenCode (Filename)
Grid1(0).Rows = Datalist(0).ListCount + 1
L1 = 1
loaddata
'errhand: Exit Sub
End Sub
```

```
Sub Cmdload2_Click ()
S = 1
On Error GoTo errhand1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmdialog1.Filter = "All Files (*.*)*. *|Code files (*.cod)*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.DialogTitle = " Load Data "
cmdialog1.Action = 1
Filename = cmdialog1.Filename
OpenCode (Filename)
Grid1(1).Rows = Datalist(1).ListCount
L2 = 1
loaddata
errhand1: Exit Sub
End Sub

```

```

Sub Cmdload3_Click ()
S = 2
On Error GoTo errhand2
cmdialog1.Filter = "All Files (*.*)*. *|Code files (*.cod)*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.DialogTitle = " Load Data "
cmdialog1.Action = 1
Filename = cmdialog1.Filename
OpenCode (Filename)
Grid1(2).Rows = Datalist(2).ListCount
L3 = 1
loaddata
errhand2: Exit Sub
End Sub

```

```

Sub Cmdnor1_Click ()
S = 0
N(0) = N(0) + 1
If L1 = 1 Then
ClearGrid
Datalist(0).Clear
Comm.Output = address1 & "se"
Reciver
Else
Comm.Output = address1 & "se"
Reciver
End If
End Sub

```

```

Sub Cmdnor2_Click ()
S = 1
N(1) = N(1) + 1
If L2 = 1 Then
ClearGrid
Datalist(1).Clear
Comm.Output = address2 & "se"
Reciver
Else
Comm.Output = address2 & "se"
Reciver
End If
End Sub

```

```

Sub Cmdnor3_Click ()
S = 2
N(2) = N(2) + 1
If L3 = 1 Then

```

```

ClearGrid
Datalist(2).Clear
Comm.Output = address3 & "se"
Reciver
Else
Comm.Output = address3 & "se"
Reciver
End If
End Sub

```

```

Sub Cmdsave1_Click ()
S = 0
On Error GoTo errhandler
cmdialog1.DialogTitle = "Save Data"
cmdialog1.Filter = "All Files (*.*)*. *|Code files (*.cod)*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 2 'save as file
Filename = cmdialog1.Filename
Savecode (Filename)
errhandler:
Exit Sub
End Sub

```

```

Sub Cmdsave2_Click ()
S = 1
On Error GoTo errhandler1
cmdialog1.DialogTitle = "Save Data"
cmdialog1.Filter = "All Files (*.*)*. *|Code files (*.cod)*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 2 'save as file
Filename = cmdialog1.Filename
Savecode (Filename)
errhandler1:
Exit Sub
End Sub

```

```

Sub Cmdsave3_Click ()
S = 2
On Error GoTo errhandler2
cmdialog1.DialogTitle = "Save Data"
cmdialog1.Filter = "All Files (*.*)*. *|Code files (*.cod)*.cod"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 2 'save as file
Filename = cmdialog1.Filename
Savecode (Filename)
errhandler2:
Exit Sub
End Sub

```

```

Sub Cmdseta1_Click ()
S = 0
form5.Show
End Sub

```

```

Sub Cmdseta2_Click ()
S = 1
form5.Show
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Sub Cmdseta3_Click ()
S = 2
form5.Show
End Sub

Sub Form_Load ()
Dim i As Integer
ReDim N(3)
paneldate.Caption = Format(Date, "dd - mmm -
yy")
Timage.Enabled = False
image3.Picture = image1.Picture
Grid1(0).Row = 0
Grid1(0).RowHeight(0) = 300
Grid1(0).Col = 0
For i = 0 To 9
Grid1(0).ColWidth(0) = 900
Next
Grid1(0).Text = " Time "
For i = 1 To 8
Grid1(0).ColWidth(i) = 800
Grid1(0).Col = i
Grid1(0).Text = " Ch " & i
Next
Grid1(0).Col = 9
Grid1(0).ColWidth(9) = 900
Grid1(0).Text = " Check "

Grid1(1).Row = 0
Grid1(1).RowHeight(0) = 300
Grid1(1).Col = 0
For i = 0 To 9
Grid1(1).ColWidth(0) = 900
Next
Grid1(1).Text = " Time "
For i = 1 To 8
Grid1(1).ColWidth(i) = 800
Grid1(1).Col = i
Grid1(1).Text = " Ch " & i
Next
Grid1(1).Col = 9
Grid1(1).ColWidth(9) = 900
Grid1(1).Text = " Check "

Grid1(2).Row = 0
Grid1(2).RowHeight(0) = 300
Grid1(2).Col = 0
For i = 0 To 9
Grid1(2).ColWidth(0) = 900
Next
Grid1(2).Text = " Time "
For i = 1 To 8
Grid1(2).ColWidth(i) = 800
Grid1(2).Col = i
Grid1(2).Text = " Ch " & i
Next
Grid1(2).Col = 9
Grid1(2).ColWidth(9) = 900
Grid1(2).Text = " Check "

'If form2.opcom1.Value = True Then
Comm.CommPort = 1
'If form2.opcom2.Value = True Then
Comm.CommPort = 2
Comm.CommPort = 2
Comm.Settings = "9600,n,8,1"
Comm.InputLen = 0
Comm.PortOpen = True
If Msetman.Checked = True Then
Grstart1.Enabled = False
Grstart2.Enabled = False
Grstart3.Enabled = False
Else
Grstart1.Enabled = True
Grstart1.Enabled = True
Grstart1.Enabled = True
End If
Checkadd1
Checkadd2
Checkadd3
'Check Alarm
For i = 0 To 7
Min1(i) = 500
Min2(i) = 500
Min3(i) = 500
Max1(i) = 1500
Max2(i) = 1500
Max3(i) = 1500
Next i
FristCon
End Sub

Sub FristCon ()
Dim Star, T As String
Dim i As Integer
'For I = 0 To 5
Comm.OutBufferCount = 0
Comm.InBufferCount = 0
Comm.Output = " "
Do
Loop Until Comm.InBufferCount <= 2
Star = Comm.Input
T = "*" & Chr(13) & ">"
i = Len(T)
'MsgBox Str(i)
If i = 3 Then
Pcon.Caption = "Connect 9600"
Else
Pcon.Caption = "HardWare Error"
End If
'Next I
End Sub

Sub GrStart1_Click (Value As Integer)
S = 0
S1 = Grstart1.Value
If L1 = 1 Then
ClearGrid
Datalist(0).Clear
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Grstart2_Click (Value As Integer)
```

```
S = 1
```

```
S2 = Grstart2.Value
```

```
If L2 = 1 Then
```

```
ClearGrid
```

```
Datalist(1).Clear
```

```
End If
```

```
End Sub
```

```
Sub Grstart3_Click (Value As Integer)
```

```
S = 2
```

```
S3 = Grstart1.Value
```

```
If L3 = 1 Then
```

```
ClearGrid
```

```
Datalist(2).Clear
```

```
End If
```

```
End Sub
```

```
Sub loaddata ()
```

```
Dim Code1, Code() As String
```

```
Dim A, i As Integer
```

```
ReDim Code(9)
```

```
For A = 0 To Datalist(S).ListCount - 1
```

```
Code1 = Datalist(S).List(A)
```

```
Code(0) = Left$(Code1, 10)
```

```
Code(1) = Mid(Code1, 12, 4)
```

```
Code(2) = Mid(Code1, 16, 4)
```

```
Code(3) = Mid(Code1, 20, 4)
```

```
Code(4) = Mid(Code1, 24, 4)
```

```
Code(5) = Mid(Code1, 28, 4)
```

```
Code(6) = Mid(Code1, 32, 4)
```

```
Code(7) = Mid(Code1, 36, 4)
```

```
Code(8) = Right(Code1, 4)
```

```
Grid1(S).Row = 1 + A
```

```
Grid1(S).Col = 0
```

```
Grid1(S).Text = Code(0)
```

```
For i = 1 To 8
```

```
Grid1(S).Col = i
```

```
Grid1(S).Text = " " & Code(i)
```

```
Next i
```

```
Next A
```

```
End Sub
```

```
Sub Mcontrol_Click ()
```

```
form12.Show
```

```
End Sub
```

```
Sub MExit_Click ()
```

```
End
```

```
End Sub
```

```
Sub Mfexit_Click ()
```

```
form4.Show
```

```
Unload form1
```

```
End Sub
```

```
Sub Msalarm_Click ()
```

```
Form7.Show
```

```
End Sub
```

```
Sub Msetauto_Click ()
```

```
Msetauto.Checked = True
```

```
Msetman.Checked = False
```

```
Tauto.Enabled = Msetauto.Checked
```

```
Timage.Enabled = Msetauto.Checked
```

```
Cmdnor1.Enabled = False
```

```
Cmdnor2.Enabled = False
```

```
Cmdnor3.Enabled = False
```

```
Grstart1.Enabled = True
```

```
Grstart2.Enabled = True
```

```
Grstart3.Enabled = True
```

```
End Sub
```

```
Sub Msetman_Click ()
```

```
Dim Flag
```

```
Msetman.Checked = True
```

```
Flag = Msetman.Checked
```

```
Msetauto.Checked = Not Flag
```

```
Tauto.Enabled = Msetauto.Checked
```

```
Timage.Enabled = Msetauto.Checked
```

```
Cmdnor1.Enabled = Flag
```

```
Cmdnor2.Enabled = Flag
```

```
Cmdnor3.Enabled = Flag
```

```
Grstart1.Enabled = False
```

```
Grstart2.Enabled = False
```

```
Grstart3.Enabled = False
```

```
Grstart1.Value = False
```

```
Grstart2.Value = False
```

```
Grstart3.Value = False
```

```
End Sub
```

```
Sub Mstime_Click ()
```

```
form2.Show
```

```
End Sub
```

```
Sub Opencode (Filename As String)
```

```
Dim filecode As String
```

```
Dim i As Integer
```

```
Datalist(S).Clear
```

```
For i = 0 To Datalist(S).ListCount - 1
```

```
Next i
```

```
Open Filename For Input As #2
```

```
Do While Not EOF(2)
```

```
Line Input #2, filecode
```

```
Datalist(S).AddItem filecode
```

```
Loop
```

```
Close #2
```

```
End Sub
```

```
Sub Reciver ()
```

```
Dim Msg, msg1, Msg2, Code, PCh, PAlarm, P, Tic,
```

```
TicA, CheckSum, CheckS, Check As String
```

```
Dim i, Z, Sum, No, He
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
Grid1(S).Text = " " & Ch1(i)
Next
```

```
End Sub
```

```
Sub Savecode (Filename As String)
```

```
Dim i
```

```
i = 0
```

```
If Dir(Filename) <> " " Then
```

```
Open Filename For Output As #1
```

```
Do Until i = Datalist(S).ListCount
```

```
Print #1, Datalist(S).List(i)
```

```
i = i + 1
```

```
Loop
```

```
Close #1
```

```
End If
```

```
End Sub
```

```
Sub Tauto_Timer ()
```

```
If S1 = True Then
```

```
S = 0
```

```
N(0) = N(0) + 1
```

```
Comm.Output = address1 & "se"
```

```
Receiver
```

```
End If
```

```
If S2 = True Then
```

```
S = 1
```

```
N(1) = N(1) + 1
```

```
Comm.Output = address2 & "se"
```

```
Receiver
```

```
End If
```

```
If S3 = True Then
```

```
S = 2
```

```
N(2) = N(2) + 1
```

```
Comm.Output = address3 & "se"
```

```
Receiver
```

```
End If
```

```
End Sub
```

```
Sub Timage_Timer ()
```

```
Static T As Integer
```

```
If T Then
```

```
image3.Picture = image1.Picture
```

```
Else
```

```
image3.Picture = image2.Picture
```

```
End If
```

```
T = Not T
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DECLARE DATABASE

Global Sender As Database

Global Sender1 As Table

Global Sender2 As Table

Global Sender3 As Table

'Global Sel As Integer

Global S, K As Integer

Global tmon, Cn As String

Global Max1(7), Min1(7)

Global Max2(7), Min2(7)

Global Max3(7), Min3(7)

Global Ch1(15), Cd(31)

Global filename As String

Global address1 As String

Global address2 As String

Global address3 As String

Global Const Address1_FILE = "ADD1.@@@"

Global Const Address2_FILE = "ADD2.@@@"

Global Const Address3_FILE = "ADD3.@@@"

Sub loaddata ()

Dim Dts, Dt As String

Dim Dy, I As Integer

If S = 0 Then

For I = 1 To 31

Cd(I) = 0

Next I

Dts = ""

Sender1.MoveFirst

Do While (Not Sender1.EOF)

If Sender1!Month = tmon Then

Dt = Sender1!Date

Dy = Val(Dt)

Cd(Dy) = Cd(Dy) + 1

If Dts <> Dt Then

Form8.Combo1.AddItem Sender1!Date

Dts = Dt

End If

Form8.Combo1.Text = Form8.Combo1.List(0)

End If

Sender1.MoveNext

Loop

End If

If S = 1 Then

For I = 1 To 31

Cd(I) = 0

Next I

Dts = ""

Sender2.MoveFirst

Do While (Not Sender2.EOF)

If Sender2!Month = tmon Then

Dt = Sender2!Date

Dy = Val(Dt)

Cd(Dy) = Cd(Dy) + 1

If Dts <> Dt Then

Form8.Combo1.AddItem Sender2!Date

Dts = Dt

End If

Form8.Combo1.Text = Form8.Combo1.List(0)

End If

Sender2.MoveNext

Loop

End If

If S = 2 Then

For I = 1 To 31

Cd(I) = 0

Next I

Dts = ""

Sender3.MoveFirst

Do While (Not Sender3.EOF)

If Sender3!Month = tmon Then

Dt = Sender3!Date

Dy = Val(Dt)

Cd(Dy) = Cd(Dy) + 1

If Dts <> Dt Then

Form8.Combo1.AddItem Sender3!Date

Dts = Dt

End If

Form8.Combo1.Text = Form8.Combo1.List(0)

End If

Sender3.MoveNext

Loop

End If

End Sub

Sub LoadToGrid ()

Dim N As Integer

N = 1

If S = 0 Then

Sender1.MoveFirst

Do While (Not Sender1.EOF)

If (Sender1!Month = tmon And Sender1!Date = Cn) Then

Form8.Grid1.Rows = Cd(Val(Cn)) + 1

Form8.Grid1.Row = N

Form8.Grid1.Col = 0

Form8.Grid1.Text = Sender1!Date

Form8.Grid1.Col = 1

Form8.Grid1.Text = Sender1!Tim

Form8.Grid1.Col = 2

Form8.Grid1.Text = Sender1!Channel1

Form8.Grid1.Col = 3

Form8.Grid1.Text = Sender1!Channel2

Form8.Grid1.Col = 4

Form8.Grid1.Text = Sender1!Channel3

Form8.Grid1.Col = 5

Form8.Grid1.Text = Sender1!Channel4

Form8.Grid1.Col = 6

Form8.Grid1.Text = Sender1!Channel5

Form8.Grid1.Col = 7

Form8.Grid1.Text = Sender1!Channel6

Form8.Grid1.Col = 8

Form8.Grid1.Text = Sender1!Channel7

Form8.Grid1.Col = 9

Form8.Grid1.Text = Sender1!Channel8

Form8.Grid1.Col = 10

Form8.Grid1.Text = Sender1!Check

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

N = N + 1
End If
Sender1.MoveNext
Loop
End If

If S = 1 Then
Sender2.MoveFirst
Do While (Not Sender2.EOF)
If (Sender2!Month = tmon And Sender2!Date =
Cn) Then
Form8.Grid1.Rows = Cd(Val(Cn)) + 1
Form8.Grid1.Row = N
Form8.Grid1.Col = 0
Form8.Grid1.Text = Sender2!Date
Form8.Grid1.Col = 1
Form8.Grid1.Text = Sender2!Tim
Form8.Grid1.Col = 2
Form8.Grid1.Text = Sender2!Channel1
Form8.Grid1.Col = 3
Form8.Grid1.Text = Sender2!Channel2
Form8.Grid1.Col = 4
Form8.Grid1.Text = Sender2!Channel3
Form8.Grid1.Col = 5
Form8.Grid1.Text = Sender2!Channel4
Form8.Grid1.Col = 6
Form8.Grid1.Text = Sender2!Channel5
Form8.Grid1.Col = 7
Form8.Grid1.Text = Sender2!Channel6
Form8.Grid1.Col = 8
Form8.Grid1.Text = Sender2!Channel7
Form8.Grid1.Col = 9
Form8.Grid1.Text = Sender2!Channel8
Form8.Grid1.Col = 10
Form8.Grid1.Text = Sender2!Check
N = N + 1
End If
Sender2.MoveNext
Loop
End If

```

```

If S = 2 Then
Sender3.MoveFirst
Do While (Not Sender3.EOF)
If (Sender3!Month = tmon And Sender3!Date =
Cn) Then
Form8.Grid1.Rows = Cd(Val(Cn)) + 1
Form8.Grid1.Row = N
Form8.Grid1.Col = 0
Form8.Grid1.Text = Sender3!Date
Form8.Grid1.Col = 1
Form8.Grid1.Text = Sender3!Tim
Form8.Grid1.Col = 2
Form8.Grid1.Text = Sender3!Channel1
Form8.Grid1.Col = 3
Form8.Grid1.Text = Sender3!Channel2
Form8.Grid1.Col = 4
Form8.Grid1.Text = Sender3!Channel3
Form8.Grid1.Col = 5
Form8.Grid1.Text = Sender3!Channel4
Form8.Grid1.Col = 6

```

```

Form8.Grid1.Text = Sender3!Channel5
Form8.Grid1.Col = 7
Form8.Grid1.Text = Sender3!Channel6
Form8.Grid1.Col = 8
Form8.Grid1.Text = Sender3!Channel7
Form8.Grid1.Col = 9
Form8.Grid1.Text = Sender3!Channel8
Form8.Grid1.Col = 10
Form8.Grid1.Text = Sender3!Check
N = N + 1
End If
Sender3.MoveNext
Loop
End If
End Sub

```

```

Sub SaveData ()
Dim Mont, Dat, Tic, Check As String
'Add new Data
Mont = Format(Now, "MM/YY")
Dat = Format(Now, "dd")
Tic = Format(Now, "HH:MM:SS ")
If S = 0 Then
BeginTrans
Sender1.AddNew
Sender1!Month = Mont
Sender1!Date = Dat
Sender1!Tim = Tic
Sender1!Channel1 = Ch1(1)
Sender1!Channel2 = Ch1(2)
Sender1!Channel3 = Ch1(3)
Sender1!Channel4 = Ch1(4)
Sender1!Channel5 = Ch1(5)
Sender1!Channel6 = Ch1(6)
Sender1!Channel7 = Ch1(7)
Sender1!Channel8 = Ch1(8)
Sender1!Check = Ch1(9)
Sender1.Update
CommitTrans
End If

```

```

If S = 1 Then
BeginTrans
Sender2.AddNew
Sender2!Month = Mont
Sender2!Date = Dat
Sender2!Tim = Tic
Sender2!Channel1 = Ch1(1)
Sender2!Channel2 = Ch1(2)
Sender2!Channel3 = Ch1(3)
Sender2!Channel4 = Ch1(4)
Sender2!Channel5 = Ch1(5)
Sender2!Channel6 = Ch1(6)
Sender2!Channel7 = Ch1(7)
Sender2!Channel8 = Ch1(8)
Sender2!Check = Ch1(9)
Sender2.Update
CommitTrans
End If

```

```

If S = 2 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BeginTrans
Sender3.AddNew
Sender3!Month = Mont
Sender3!Date = Dat
Sender3!Tim = Tic
Sender3!Channel1 = Ch1(1)
Sender3!Channel2 = Ch1(2)
Sender3!Channel3 = Ch1(3)
Sender3!Channel4 = Ch1(4)
Sender3!Channel5 = Ch1(5)
Sender3!Channel6 = Ch1(6)
Sender3!Channel7 = Ch1(7)
Sender3!Channel8 = Ch1(8)
Sender3!Check = Ch1(9)
Sender3.Update
CommitTrans
End If
End Sub

Function Thedata ()
Dim Sender1TB As New TableDef
ReDim Data1Field(12) As New Field
Dim Sender2TB As New TableDef
ReDim Data2Field(12) As New Field
Dim Sender3TB As New TableDef
ReDim Data3Field(12) As New Field

If Len(Dir$("Sender.MDB")) = 0 Then
' Create and open database
Set Sender = CreateDatabase("
Sender.MDB", DB_LANG_GENERAL)

' Create fields of Sender1 Table
Sender1TB.Name = "Sender1"
Data1Field(1).Name = "Month"
Data1Field(1).Type = DB_TEXT
Data1Field(1).Size = 7
Sender1TB.Fields.Append Data1Field(1)
Data1Field(2).Name = "Date"
Data1Field(2).Type = DB_TEXT
Data1Field(2).Size = 5
Sender1TB.Fields.Append Data1Field(2)
Data1Field(3).Name = "Tim"
Data1Field(3).Type = DB_TEXT
Data1Field(3).Size = 15
Sender1TB.Fields.Append Data1Field(3)
Data1Field(4).Name = "Channel1"
Data1Field(4).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(4)
Data1Field(5).Name = "Channel2"
Data1Field(5).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(5)
Data1Field(6).Name = "Channel3"
Data1Field(6).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(6)
Data1Field(7).Name = "Channel4"
Data1Field(7).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(7)
Data1Field(8).Name = "Channel5"
Data1Field(8).Type = DB_LONG

Sender1TB.Fields.Append Data1Field(8)
Data1Field(9).Name = "Channel6"
Data1Field(9).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(9)
Data1Field(10).Name = "Channel7"
Data1Field(10).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(10)
Data1Field(11).Name = "Channel8"
Data1Field(11).Type = DB_LONG
Sender1TB.Fields.Append Data1Field(11)
Data1Field(12).Name = "Check"
Data1Field(12).Type = DB_TEXT
Data1Field(12).Size = 5
Sender1TB.Fields.Append Data1Field(12)

' Add Sender1 TB To Database
Sender.TableDefs.Append Sender1TB

' Create fields of Sender2 Table
Sender2TB.Name = "Sender2"
Data2Field(1).Name = "Month"
Data2Field(1).Type = DB_TEXT
Data2Field(1).Size = 7
Sender2TB.Fields.Append Data2Field(1)
Data2Field(2).Name = "Date"
Data2Field(2).Type = DB_TEXT
Data2Field(2).Size = 5
Sender2TB.Fields.Append Data2Field(2)
Data2Field(3).Name = "Tim"
Data2Field(3).Type = DB_TEXT
Data2Field(3).Size = 15
Sender2TB.Fields.Append Data2Field(3)
Data2Field(4).Name = "Channel1"
Data2Field(4).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(4)
Data2Field(5).Name = "Channel2"
Data2Field(5).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(5)
Data2Field(6).Name = "Channel3"
Data2Field(6).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(6)
Data2Field(7).Name = "Channel4"
Data2Field(7).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(7)
Data2Field(8).Name = "Channel5"
Data2Field(8).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(8)
Data2Field(9).Name = "Channel6"
Data2Field(9).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(9)
Data2Field(10).Name = "Channel7"
Data2Field(10).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(10)
Data2Field(11).Name = "Channel8"
Data2Field(11).Type = DB_LONG
Sender2TB.Fields.Append Data2Field(11)
Data2Field(12).Name = "Check"
Data2Field(12).Type = DB_TEXT
Data2Field(12).Size = 5
Sender2TB.Fields.Append Data2Field(12)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Set Sender3 = Sender.OpenTable("Sender3")
OpenTheDatabase = True
```

```
'Add Serder2TB To Database
Sender.TableDefs.Append Sender2TB
```

End Function

```
' Create fields of Sender3 Table
Sender3TB.Name = "Sender3"
Data3Field(1).Name = "Month"
Data3Field(1).Type = DB_TEXT
Data3Field(1).Size = 7
Sender3TB.Fields.Append Data3Field(1)
Data3Field(2).Name = "Date"
Data3Field(2).Type = DB_TEXT
Data3Field(2).Size = 5
Sender3TB.Fields.Append Data3Field(2)
Data3Field(3).Name = "Tim"
Data3Field(3).Type = DB_TEXT
Data3Field(3).Size = 15
Sender3TB.Fields.Append Data3Field(3)
Data3Field(4).Name = "Channel1"
Data3Field(4).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(4)
Data3Field(5).Name = "Channel2"
Data3Field(5).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(5)
Data3Field(6).Name = "Channel3"
Data3Field(6).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(6)
Data3Field(7).Name = "Channel4"
Data3Field(7).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(7)
Data3Field(8).Name = "Channel5"
Data3Field(8).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(8)
Data3Field(9).Name = "Channel6"
Data3Field(9).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(9)
Data3Field(10).Name = "Channel7"
Data3Field(10).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(10)
Data3Field(11).Name = "Channel8"
Data3Field(11).Type = DB_LONG
Sender3TB.Fields.Append Data3Field(11)
Data3Field(12).Name = "Check"
Data3Field(12).Type = DB_TEXT
Data3Field(12).Size = 5
Sender3TB.Fields.Append Data3Field(12)
```

```
'Add Serder3TB To Database
Sender.TableDefs.Append Sender3TB
' Must close and re-open to enable multi-
user access
' to a newly created database.
Sender.Close
End If
```

```
Set Sender = OpenDatabase("Sender.MDB")
Set Sender1 = Sender.OpenTable("Sender1")
Set Sender2 = Sender.OpenTable("Sender2")
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 2 Kbytes of Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2 Kbytes 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 MCS-51™ instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2 Kbytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 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 Configuration

PDIP/SOIC

RST	1	20	VCC
(RXD) P3.0	2	19	P1.7
(TXD) P3.1	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
(INT0) P3.2	6	15	P1.3
(INT1) P3.3	7	14	P1.2
(T0) P3.4	8	13	P1.1 (AIN1)
(T1) P3.5	9	12	P1.0 (AIN0)
GND	10	11	P3.7

8-Bit Microcontroller with 2 Kbytes Flash

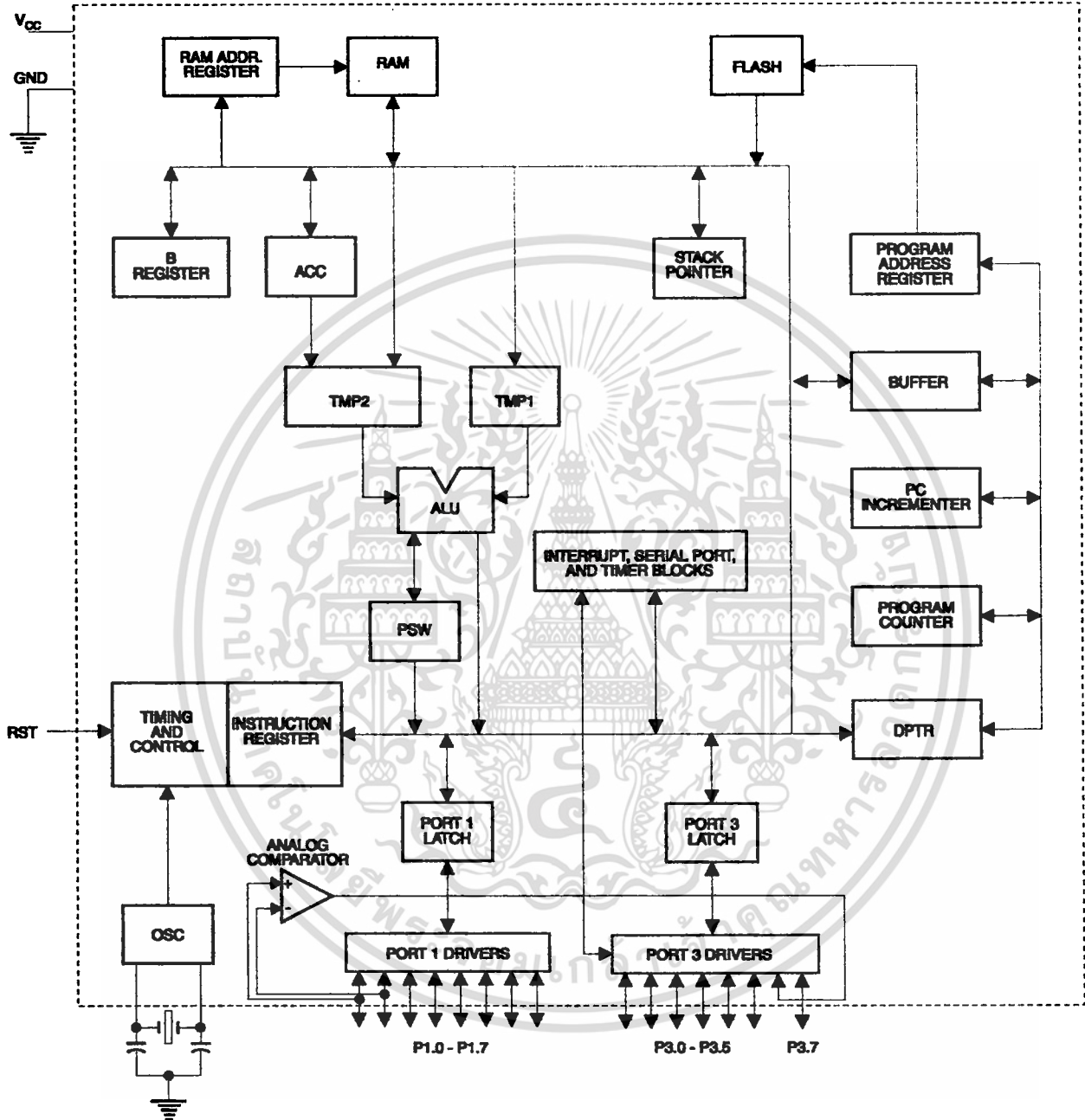
0368C



3-17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and program verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. 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 AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

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

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

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

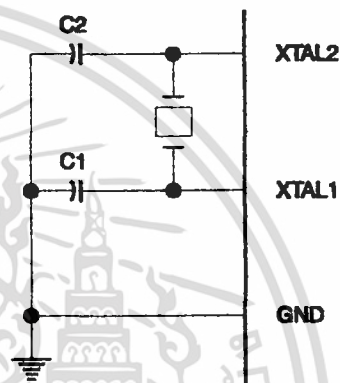
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

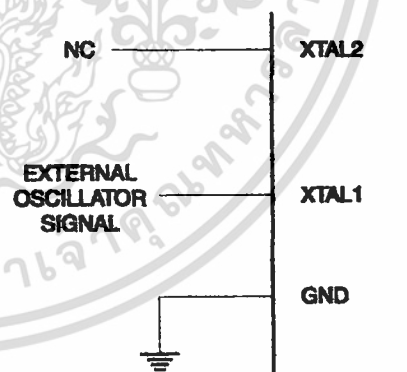
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. 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.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration





Special Function Registers

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

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.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								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		SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C2051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2 Kbytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ
With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.





Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes⁽¹⁾

Program Lock Bits	Lock Bits		Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation

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.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle 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 is terminated by reset, the instruction following the one that invokes idle should not be one that writes 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 Vcc is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

Programming The Flash

The AT89C2051 is shipped with the 2 Kbytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
Apply power between Vcc and GND pins
Set RST and XTAL1 to GND
With all other pins floating, wait for greater than 10 milliseconds
 2. Set pin RST to 'H'
Set pin P3.2 to 'H'
 3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
- To Program and Verify the Array:
4. Apply data for Code byte at location 000H to P1.0 to P1.7.
 5. Raise RST to 12V to enable programming.
 6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
 7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
 8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
 9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2 Kbytes array or until the end of the object file is reached.
 10. Power-off sequence:
set XTAL1 to 'L'
set RST to 'L'
Float all other I/O pins
Turn Vcc power off

Data Polling: The AT89C2051 features 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 P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2 Kbytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.





- (000H) = 1EH indicates manufactured by Atmel
- (001H) = 21H indicates 89C2051

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	P3.2/ PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data ^(1,3)	12V		L	H	H	H
Read Code Data ⁽¹⁾	H	H	L	L	H	H
Write Lock	Bit - 1		H	H	H	H
	Bit - 2		H	H	L	L
Chip Erase	12V	 ⁽²⁾	H	L	L	L
Read Signature Byte	H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.

2. Chip Erase requires a 10 ms PROG pulse.

3. P3.1 is pulled Low during programming to indicate RDY/BSY.



Figure 3. Programming the Flash Memory

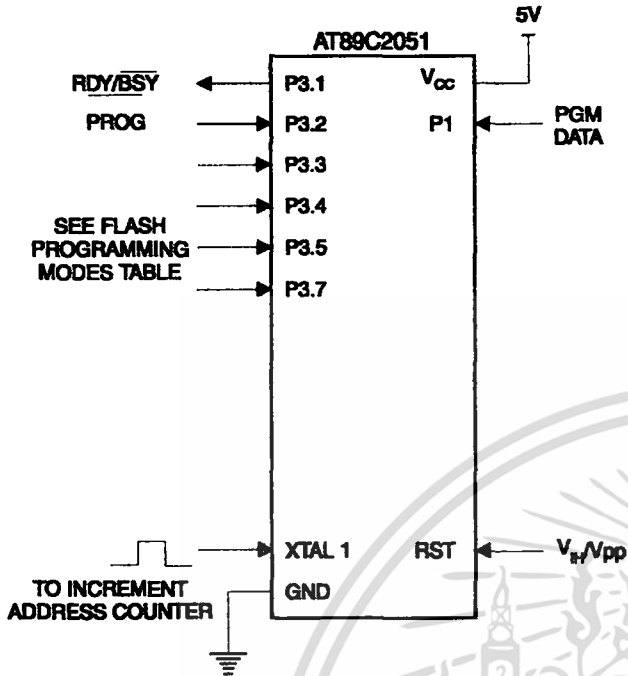
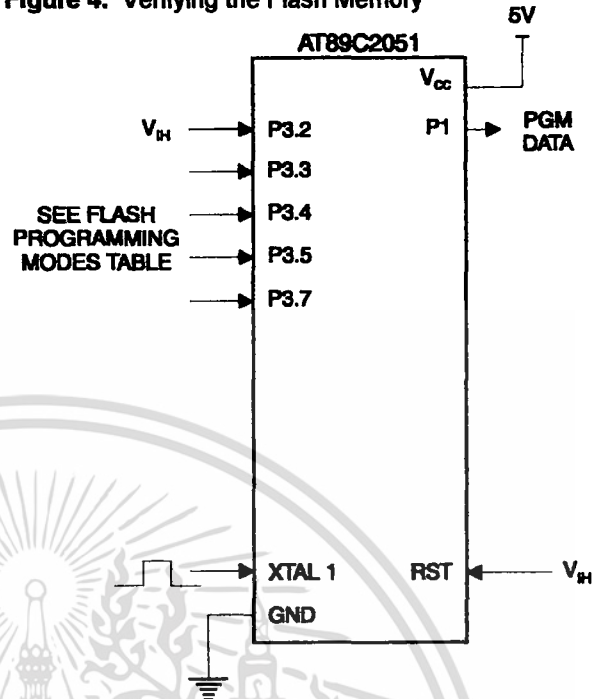


Figure 4. Verifying the Flash Memory



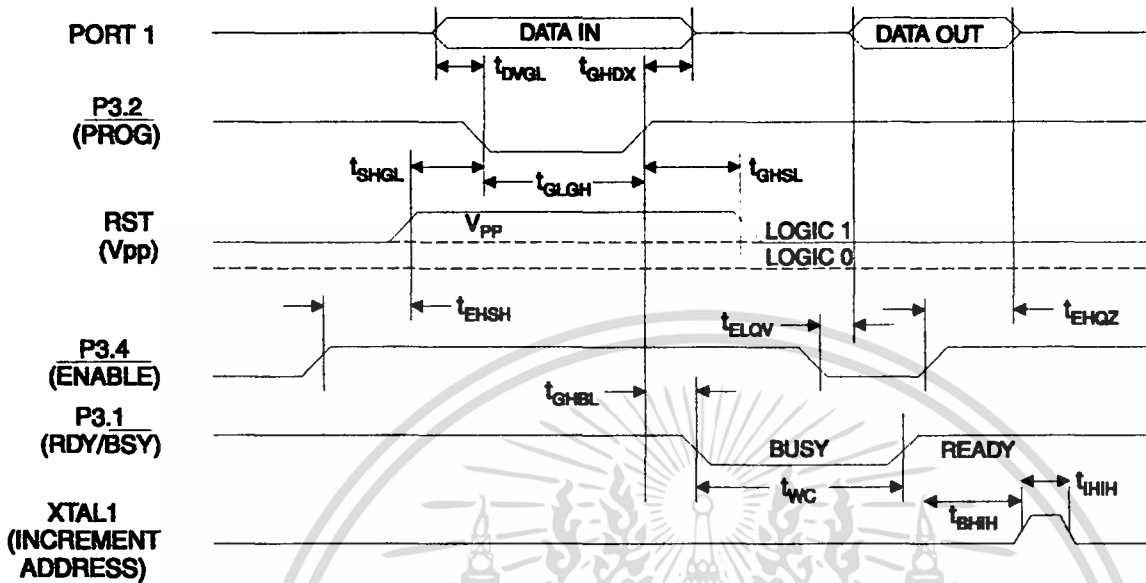
Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C}$ to 27°C , $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	μA
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	1.0		μs
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	1.0		μs
t_{EHS}	P3.4 ($\overline{\text{ENABLE}}$) High to V_{PP}	1.0		μs
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		1.0	μs
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	1.0	μs
t_{GMHL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHH}	$\overline{\text{RDY/BSY}}$ to Increment Clock Delay	1.0		μs
t_{HIL}	Increment Clock High	200		ns

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

Flash Programming and Verification Waveforms



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current.....	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.



D.C. Characteristics

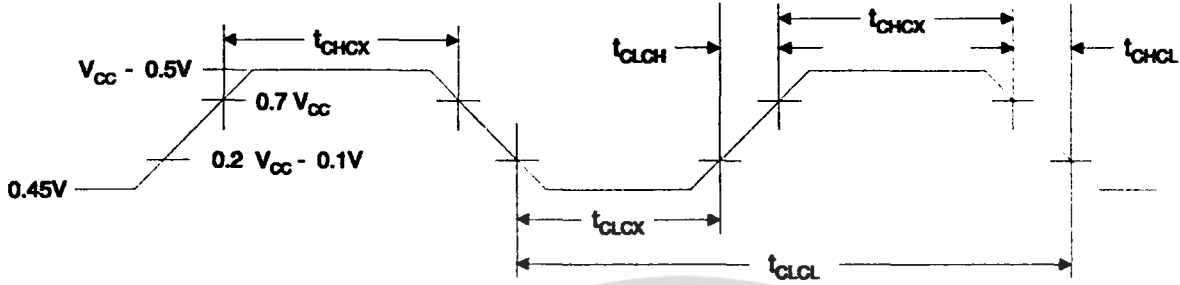
$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 2.7\text{ V}$ to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	$0.2 V_{CC}-0.1$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC}+0.9$	$V_{CC}+0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC}+0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1, 3)	$I_{OL} = 20\text{ mA}$, $V_{CC} = 5\text{ V}$ $I_{OL} = 10\text{ mA}$, $V_{CC} = 2.7\text{ V}$		0.5	V
V_{OH}	Output High Voltage (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	$V_{IN} = 0.45\text{ V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	$V_{IN} = 2\text{ V}$		-750	μA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		± 10	μA
V_{OS}	Comparator Input Offset Voltage	$V_{CC} = 5\text{ V}$		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6\text{ V}/3\text{ V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{ V}/3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		100	μA
		$V_{CC} = 3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		20	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 20 mA
Maximum total I_{OL} for all output pins: 80 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2 V.

External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	V _{CC} = 2.7 V to 6.0 V		V _{CC} = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
1/t _{CLCL}	Oscillator Frequency	0	12	0	24	MHz
t _{CLCL}	Clock Period	83.3		41.6		ns
t _{CHCX}	High Time	30		15		ns
t _{CLCX}	Low Time	30		15		ns
t _{CLCH}	Rise Time		20		20	ns
t _{CHCL}	Fall Time		20		20	ns



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

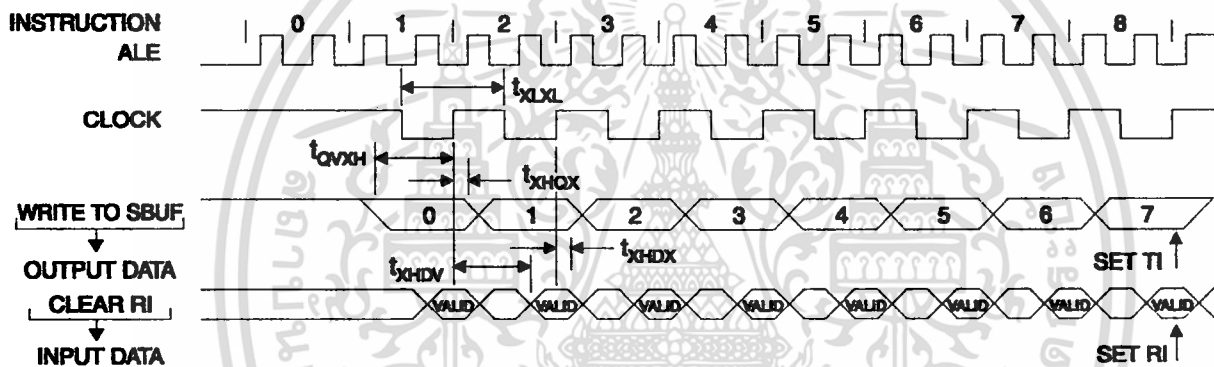


Serial Port Timing: Shift Register Mode Test Conditions

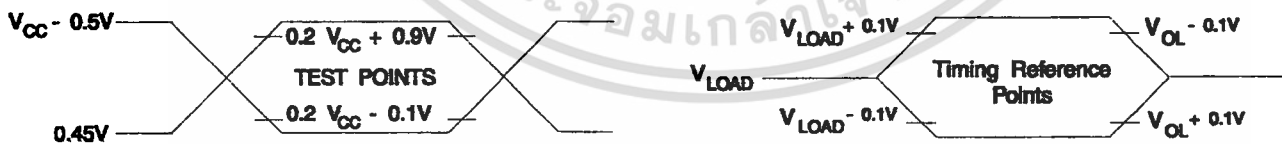
($V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-33$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms



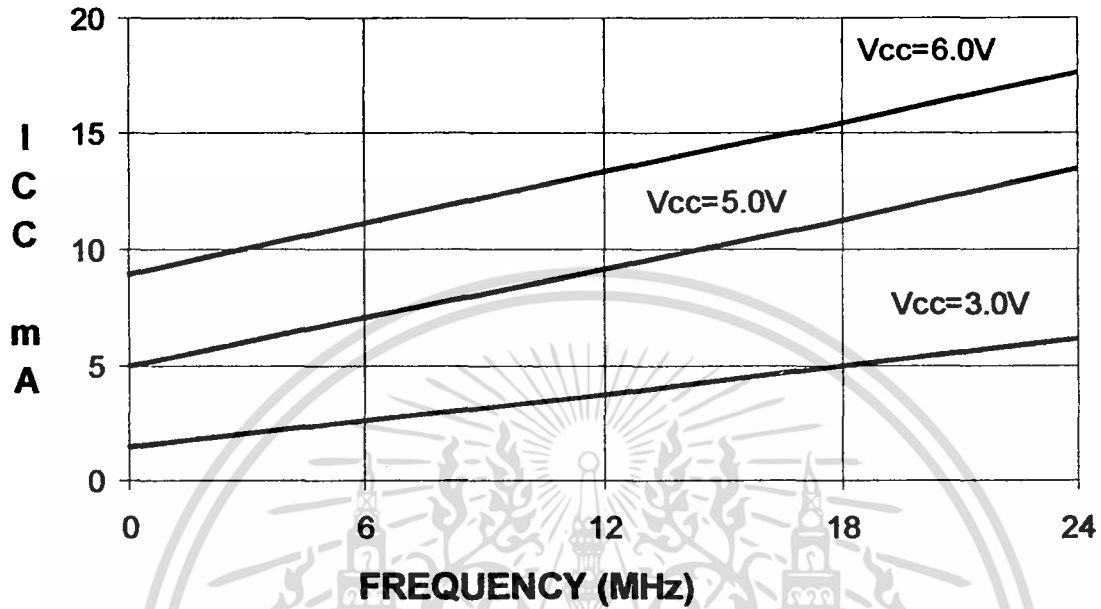
AC Testing Input/Output Waveforms ⁽¹⁾ Float Waveforms ⁽¹⁾



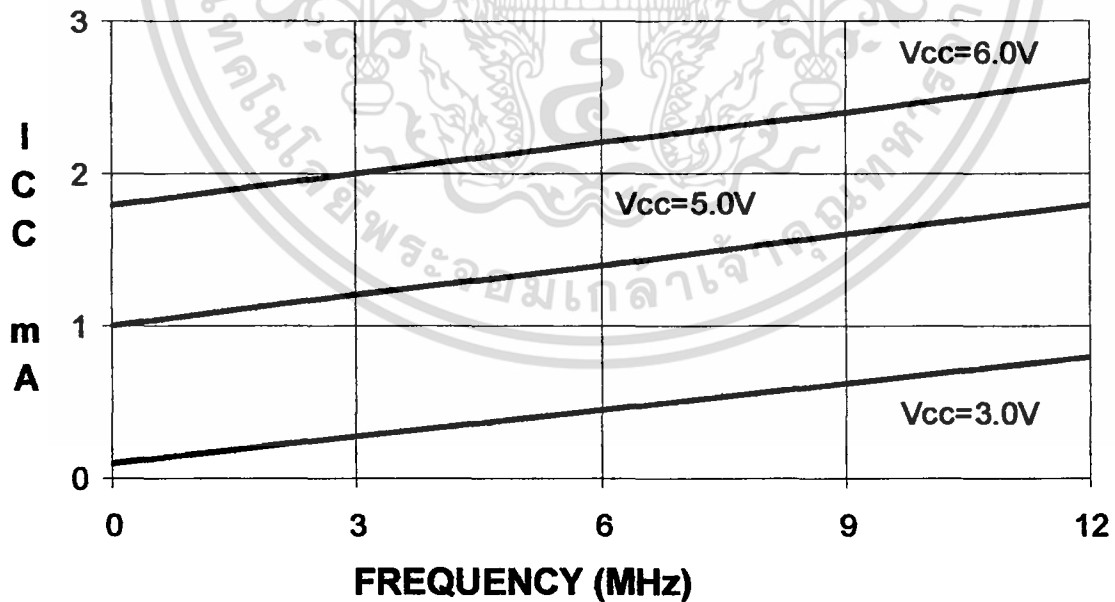
Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5\text{ V}$ for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

AT89C2051
TYPICAL ICC - ACTIVE (85°C)

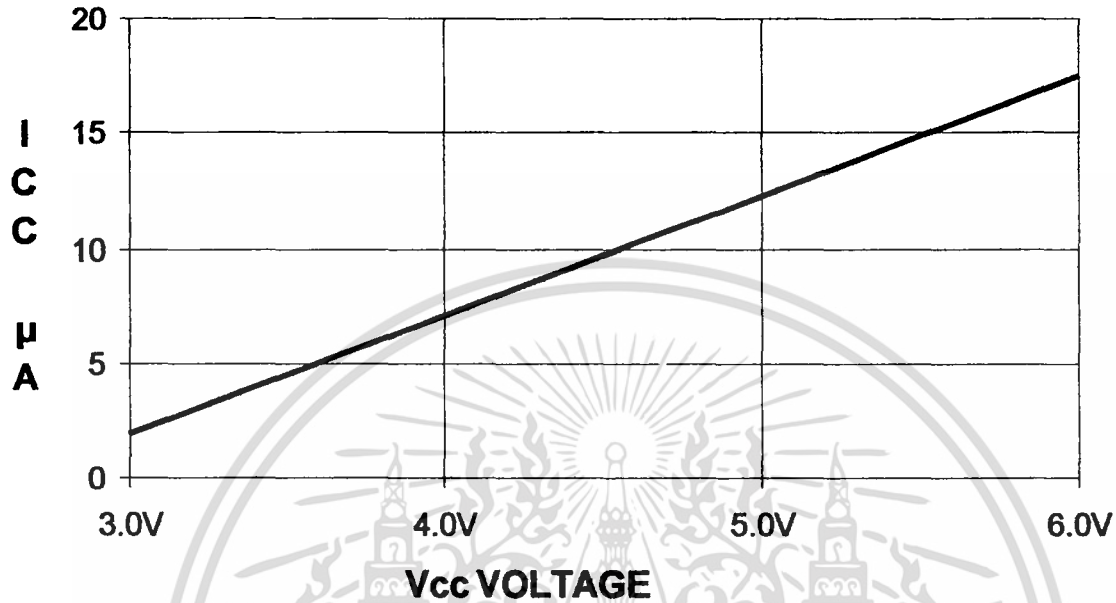


AT89C2051
TYPICAL ICC - IDLE (85°C)



AT89C2051

TYPICAL ICC vs. VOLTAGE- POWER DOWN (85°C)



- Note:
1. XTAL1 tied to GND for I_{CC} (power down).
 2. P.1.0 and P1.1 = V_{CC} or GND.
 3. Lock bits programmed.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6.0 V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0 V to 6.0 V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)



Package Type	
20P3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIM

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

General Description

The MAX192 is a low-cost, 10-bit data-acquisition system that combines an 8-channel multiplexer, high-bandwidth track/hold, and serial interface with high conversion speed and ultra-low power consumption. The device operates with a single +5V supply. The analog inputs are software configurable for single-ended and differential (unipolar/bipolar) operation.

The 4-wire serial interface connects directly to SPI™, QSPI™, and Microwire™ devices, without using external logic. A serial strobe output allows direct connection to TMS320 family digital signal processors. The MAX192 uses either the internal clock or an external serial-interface clock to perform successive approximation A/D conversions. The serial interface can operate beyond 4MHz when the internal clock is used. The MAX192 has an internal 4.096V reference with a drift of ± 30 ppm typical. A reference-buffer amplifier simplifies gain trim and two sub-LSBs reduce quantization errors.

The MAX192 provides a hardwired SHDN pin and two software-selectable power-down modes. Accessing the serial interface automatically powers up the device, and the quick turn-on time allows the MAX192 to be shut down between conversions. By powering down between conversions, supply current can be cut to under 10 μ A at reduced sampling rates.

The MAX192 is available in 20-pin DIP and SO packages, and in a shrink-small-outline package (SSOP) that occupies 30% less area than an 8-pin DIP. The data format provides hardware and software compatibility with the MAX186/MAX188. For anti-aliasing filters, consult the data sheets for the MAX291–MAX297.

Applications

Automotive
Pen-Entry Systems
Consumer Electronics
Portable Data Logging
Robotics
Battery-Powered Instruments, Battery Management
Medical Instruments

See last page for Typical Operating Circuit.

™ SPI and QSPI are trademarks of Motorola Corp.
Microwire is a trademark of National Semiconductor Corp.

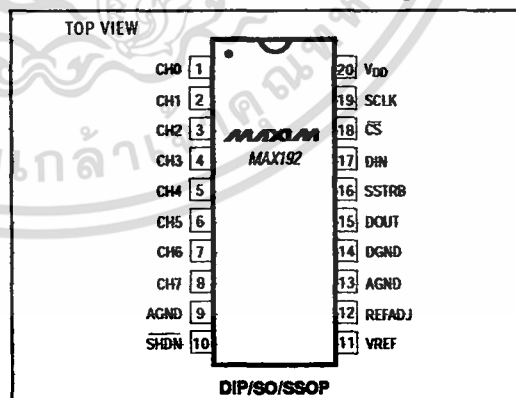
Features

- ◆ 8-Channel Single-Ended or 4-Channel Differential Inputs
- ◆ Single +5V Operation
- ◆ Low Power: 1.5mA (operating)
2 μ A (power-down)
- ◆ Internal Track/Hold, 133kHz Sampling Rate
- ◆ Internal 4.096V Reference
- ◆ 4-Wire Serial Interface is Compatible with SPI, QSPI, Microwire, and TMS320
- ◆ 20-Pin DIP, SO, SSOP Packages
- ◆ Pin-Compatible 12-Bit Upgrade (MAX186/MAX188)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE	NL(LSBs)
MAX192ACPP	0 °C to +70 °C	20 Plastic DIP	$\pm 1/2$
MAX192BCPP	0 °C to +70 °C	20 Plastic DIP	± 1
MAX192ACWP	0 °C to +70 °C	20 Wide SO	$\pm 1/2$
MAX192BCWP	0 °C to +70 °C	20 Wide SO	± 1
MAX192ACAP	0 °C to +70 °C	20 SSOP	$\pm 1/2$
MAX192BCAP	0 °C to +70 °C	20 SSOP	± 1
MAX192AEPP	-40 °C to +85 °C	20 Plastic DIP	$\pm 1/2$
MAX192BEPP	-40 °C to +85 °C	20 Plastic DIP	± 1
MAX192AEWP	-40 °C to +85 °C	20 Wide SO	$\pm 1/2$
MAX192BEWP	-40 °C to +85 °C	20 Wide SO	± 1
MAX192AEAP	-40 °C to +85 °C	20 SSOP	$\pm 1/2$
MAX192BEAP	-40 °C to +85 °C	20 SSOP	± 1
MAX192AMJP	-55 °C to +125 °C	20 CERDIP	$\pm 1/2$
MAX192BMJP	-55 °C to +125 °C	20 CERDIP	± 1

Pin Configuration



MAXIM

Maxim Integrated Products 1

Call toll free 1-800-998-8800 for free samples or literature.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

ABSOLUTE MAXIMUM RATINGS

V _{DD} to AGND	-0.3V to +6V	SSOP (derate 8.00mW/°C above +70 °C)	640mW
AGND to DGND	0.3V to +0.3V	CERDIP (derate 11.11mW/°C above +70 °C)	880mW
CH0-CH7 to AGND, DGND	-0.3V to (V _{DD} + 0.3V)	Operating Temperature Ranges	
VREF to AGND	-0.3V to (V _{DD} + 0.3V)	MAX192_C_P	0 °C to +70 °C
REFADJ to AGND	-0.3V to (V _{DD} + 0.3V)	MAX192_E_P	-40 °C to +85 °C
Digital Inputs to DGND	-0.3V to (V _{DD} + 0.3V)	MAX192_MJP	-55 °C to +125 °C
Digital Outputs to DGND	-0.3V to (V _{DD} + 0.3V)	Storage Temperature Range	
Continuous Power Dissipation (T _A = +70 °C)			-60 °C to +150 °C
Plastic DIP (derate 11.11mW/°C above +70 °C)	889mW	Lead Temperature (soldering, 10sec)	
SO (derate 10.00mW/°C above +70 °C)	800mW	+300 °C	

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{DD} = 5V ±5%, f_{CLK} = 2.0MHz, external clock (50% duty cycle), 15 clocks/conversion cycle (133ksps), 4.7µF capacitor at VREF pin, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25 °C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DC ACCURACY (Note 1)						
Resolution			10			Bits
Relative Accuracy (Note 2)		MAX192A		±1/2		LSB
		MAX192B		±1		LSB
Differential Nonlinearity	DNL	No missing codes over temperature			±1	LSB
Offset Error					±2	LSB
Gain Error		External reference, 4.096V			±2	LSB
Gain Temperature Coefficient		External reference, 4.096V		±0.8		ppm/°C
Channel-to-Channel Offset Matching				±0.1		LSB
DYNAMIC SPECIFICATIONS (10kHz sine-wave input, 4.096V_{p-p}, 133ksps, 2.0MHz external clock)						
Signal-to-Noise + Distortion Ratio	SINAD			66		dB
Total Harmonic Distortion (up to the 5th harmonic)	THD			-70		dB
Spurious-Free Dynamic Range	SFDR			70		dB
Channel-to-Channel Crosstalk		65kHz, V _{IN} = 4.096V _{p-p} (Note 3)		-75		dB
Small-Signal Bandwidth		-3dB rolloff		4.5		MHz
Full-Power Bandwidth				800		kHz
CONVERSION RATE						
Conversion Time (Note 4)	t _{CONV}	Internal clock	5.5		10	µs
		External clock, 2MHz, 12 clocks/conversion	6			
Track/Hold Acquisition Time	t _{AZ}				1.5	µs
Aperture Delay				10		ns
Aperture Jitter				<50		ps
Internal Clock Frequency				1.7		MHz

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

ELECTRICAL CHARACTERISTICS (continued)

($V_{DD} = 5V \pm 5\%$, $f_{CLK} = 2.0\text{MHz}$, external clock (50% duty cycle), 15 clocks/conversion cycle (133ksp/s), 4.7 μF capacitor at VREF pin, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted. Typical values are at $T_A = +25^\circ\text{C}$.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	
External Clock Frequency		External compensation, 4.7 μF	0.1		2.0	MHz	
		Internal compensation (Note 5)	0.1		0.4		
		Used for data transfer only		10			
ANALOG INPUT							
Analog Input Voltage		Common-mode range (any input) (Note 6)	0		V_{DD}	V	
		Single-ended range (unipolar only)	0		V_{REF}		
		Differential range	Unipolar	0			V_{REF}
			Bipolar	$-\frac{V_{REF}}{2}$			$\frac{V_{REF}}{2}$
Multiplexer Leakage Current		On/off leakage current, $V_{IN} = 0V, 5V$		± 0.01	± 1	μA	
Input Capacitance		(Note 5)		16		pF	
INTERNAL REFERENCE (reference buffer enabled)							
VREF Output Voltage		$T_A = +25^\circ\text{C}$	4.076	4.096	4.116	V	
VREF Short-Circuit Current					30	mA	
VREF Tempco				± 30		ppm/ $^\circ\text{C}$	
Load Regulation (Note 7)		0mA to 0.5mA output load		2.5		mV	
Capacitive Bypass at VREF		Internal compensation		0		μF	
		External compensation		4.7			
Capacitive Bypass at REFADJ		Internal compensation		0.01		μF	
		External compensation		0.01			
REFADJ Adjustment Range				± 1.5		%	
EXTERNAL REFERENCE AT VREF (buffer disabled, $V_{REF} = 4.096V$)							
Input Voltage Range			2.5		$V_{DD} + 50\text{mV}$	V	
Input Current				200	350	μA	
Input Resistance			12	20		k Ω	
Shutdown VREF Input Current				1.5	10	μA	
Buffer Disable Threshold REFADJ			$V_{DD} - 50\text{mV}$			V	
EXTERNAL REFERENCE AT REFADJ							
Capacitive Bypass at VREF		Internal compensation mode		0		μF	
		External compensation mode		4.7			
Reference-Buffer Gain				1.678		V/V	
REFADJ Input Current					± 50	μA	

Low-Power, 8-Channel, Serial 10-Bit ADC

ELECTRICAL CHARACTERISTICS (continued)

($V_{DD} = 5V \pm 5\%$, $f_{CLK} = 2.0MHz$, external clock (50% duty cycle), 15 clocks/conversion cycle (133ksps), 4.7 μF capacitor at VREF pin, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted. Typical values are at $T_A = +25^\circ C$.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DIGITAL INPUTS (DIN, SCLK, CS, SHDN)						
DIN, SCLK, CS Input High Voltage	V_{INH}		2.4			V
DIN, SCLK, CS Input Low Voltage	V_{INL}				0.8	V
DIN, SCLK, CS Input Hysteresis	V_{HYST}			0.15		V
DIN, SCLK, CS Input Leakage	I_{IN}	$V_{IN} = 0V$ or V_{DD}			± 1	μA
DIN, SCLK, CS Input Capacitance	C_{IN}	(Note 5)			15	pF
SHDN Input High Voltage	V_{INH}		$V_{DD} - 0.5$			V
SHDN Input Low Voltage	V_{INL}				0.5	V
SHDN Input Current, High	I_{INH}	SHDN = V_{DD}			4.0	μA
SHDN Input Current, Low	I_{INL}	SHDN = 0V	-4.0			μA
SHDN Input Mid Voltage	V_{IM}		1.5		$V_{DD} - 1.5$	V
SHDN Voltage, Floating	V_{FLT}	SHDN = open		2.75		V
SHDN Max Allowed Leakage, Mid Input		SHDN = open	-100		100	nA
DIGITAL OUTPUTS (DOUT, SSTRB)						
Output Voltage Low	V_{OL}	$I_{SINK} = 5mA$			0.4	V
		$I_{SINK} = 16mA$			0.3	
Output Voltage High	V_{OH}	$I_{SOURCE} = 1mA$	4			V
Three-State Leakage Current	I_L	CS = 5V			± 10	μA
Three-State Leakage Capacitance	C_{OUT}	CS = 5V (Note 5)			15	pF
POWER REQUIREMENTS						
Positive Supply Voltage	V_{DD}			$5 \pm 5\%$		V
Positive Supply Current	I_{DD}	Operating mode		1.5	2.5	mA
		Fast power-down		30	70	
		Full power-down		2	10	μA
Positive Supply Rejection (Note 8)	PSR	$V_{DD} = 5V \pm 5\%$; external reference, 4.096V; full-scale input		± 0.06	± 0.5	mV

Note 1: Tested at $V_{DD} = 5.0V$; single-ended, unipolar.

Note 2: Relative accuracy is the deviation of the analog value at any code from its theoretical value after the full-scale range has been calibrated.

Note 3: Grounded on-channel; sine wave applied to all off channels.

Note 4: Conversion time defined as the number of clock cycles times the clock period; clock has 50% duty cycle.

Note 5: Guaranteed by design. Not subject to production testing.

Note 6: The common-mode range for the analog inputs is from AGND to V_{DD} .

Note 7: External load should not change during conversion for specified accuracy.

Note 8: Measured at $V_{SUPPLY} +5\%$ and $V_{SUPPLY} -5\%$ only.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

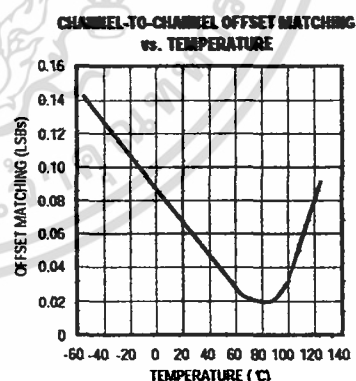
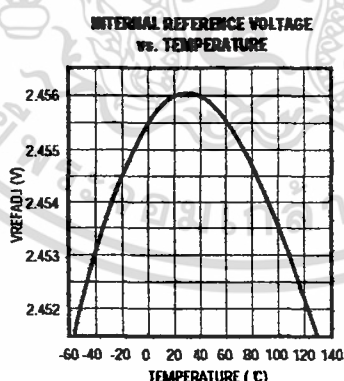
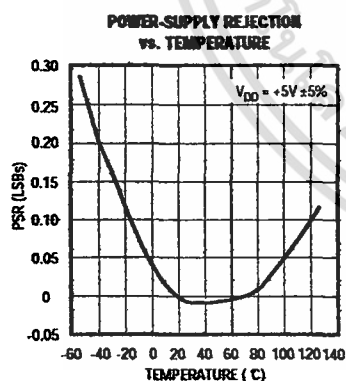
TIMING CHARACTERISTICS

($V_{DD} = 5V \pm 5\%$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Acquisition Time	tAZ		1.5			μ s
DIN to SCLK Setup	tDS		100			ns
DIN to SCLK Hold	tDH				0	ns
SCLK Fall to Output Data Valid	tDO	$C_{LOAD} = 100pF$	20		150	ns
\overline{CS} Fall to Output Enable	tDV	$C_{LOAD} = 100pF$			100	ns
\overline{CS} Rise to Output Disable	tTR	$C_{LOAD} = 100pF$			100	ns
\overline{CS} to SCLK Rise Setup	tCSS		100			ns
\overline{CS} to SCLK Rise Hold	tCSH		0			ns
SCLK Pulse Width High	tCH		200			ns
SCLK Pulse Width Low	tCL		200			ns
SCLK Fall to SSTRB	tSSTRB	$C_{LOAD} = 100pF$			200	ns
\overline{CS} Fall to SSTRB Output Enable (Note 5)	tSDV	External clock mode only, $C_{LOAD} = 100pF$			200	ns
\overline{CS} Rise to SSTRB Output Disable (Note 5)	tSTR	External clock mode only, $C_{LOAD} = 100pF$			200	ns
SSTRB Rise to SCLK Rise (Note 5)	tSCK	Internal clock mode only	0			ns

Note 6: Guaranteed by design. Not subject to production testing.

Typical Operating Characteristics



MAXIM

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADCs

Pin Description

PIN	NAME	FUNCTION
1-8	CH0-CH7	Sampling Analog Inputs
9, 13	AGND	Analog Ground. Also IN- Input for single-enabled conversions. Connect both AGND pins to analog ground.
10	SHDN	Three-Level Shutdown Input. Pulling SHDN low shuts the MAX192 down to 10 μ A (max) supply current, otherwise the MAX192 is fully operational. Pulling SHDN high puts the reference-buffer amplifier in internal compensation mode. Letting SHDN float puts the reference-buffer amplifier in external compensation mode.
11	VREF	Reference Voltage for analog-to-digital conversion. Also, Output of the Reference Buffer Amplifier. Add a 4.7 μ F capacitor to ground when using external compensation mode. Also functions as an input when used with a precision external reference.
12	REFADJ	Reference-Buffer Amplifier Input. To disable the reference-buffer amplifier, tie REFADJ to V _{DD} .
14	DGND	Digital Ground
15	DOUT	Serial Data Output. Data is clocked out at the falling edge of SCLK. High impedance when \overline{CS} is high.
16	SSTRB	Serial Strobe Output. In internal clock mode, SSTRB goes low when the MAX192 begins the A/D conversion and goes high when the conversion is done. In external clock mode, SSTRB pulses high for one clock period before the MSB decision. SSTRB is high impedance when \overline{CS} is high (external mode).
17	DIN	Serial Data Input. Data is clocked in at the rising edge of SCLK.
18	\overline{CS}	Active-Low Chip Select. Data will not be clocked into DIN unless \overline{CS} is low. When \overline{CS} is high, DOUT is high impedance.
19	SCLK	Serial Clock Input. Clocks data in and out of serial interface. In external clock mode, SCLK also sets the conversion speed. (Duty cycle must be 45% to 55%.)
20	V _{DD}	Positive Supply Voltage, +5V \pm 5%

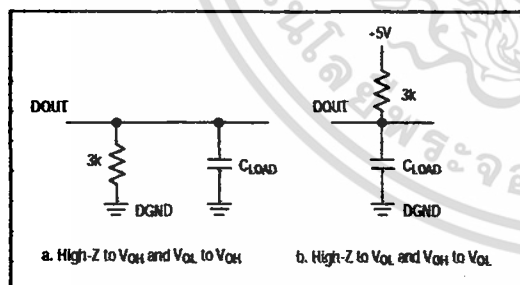


Figure 1. Load Circuits for Enable Time

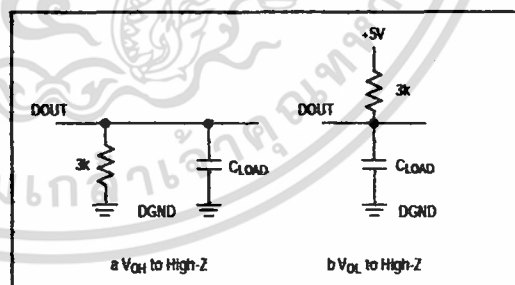


Figure 2. Load Circuits for Disabled Time

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

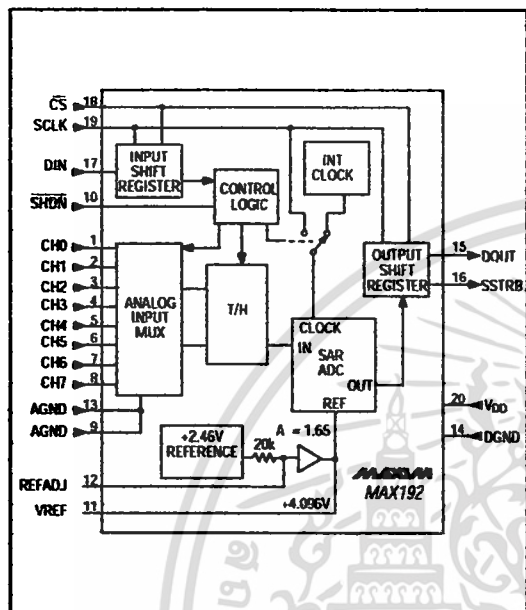


Figure 3. Block Diagram

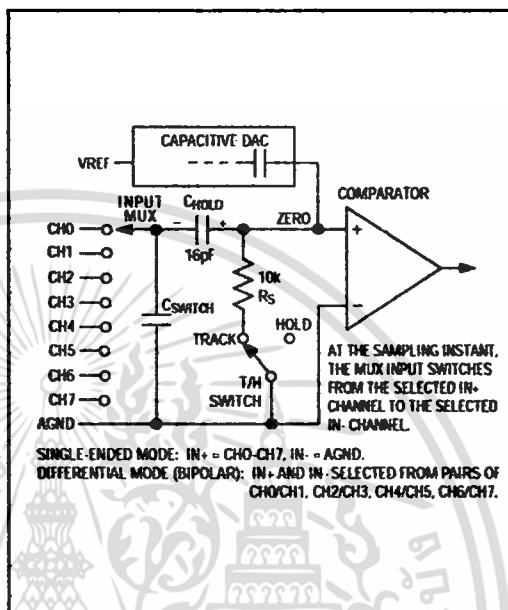


Figure 4. Equivalent Input Circuit

Detailed Description

The MAX192 uses a successive-approximation conversion technique and input track/hold (T/H) circuitry to convert an analog signal to a 10-bit digital output. A flexible serial interface provides easy interface to microprocessors. No external hold capacitors are required. Figure 3 shows the block diagram for the MAX192.

Pseudo-Differential Input

The sampling architecture of the ADC's analog comparator is illustrated in the Equivalent Input Circuit (Figure 4). In single-ended mode, IN+ is internally switched to CH0-CH7 and IN- is switched to AGND. In differential mode, IN+ and IN- are selected from pairs of CH0/CH1, CH2/CH3, CH4/CH5 and CH6/CH7. Refer to Tables 1 and 2 to configure the channels.

In differential mode, IN- and IN+ are internally switched to either one of the analog inputs. This configuration is pseudo-differential to the effect that only the signal at IN+ is sampled. The return side (IN-) must remain stable within $\pm 0.5\text{LSB}$ ($\pm 0.1\text{LSB}$ for best results) with

respect to AGND during a conversion. Accomplish this by connecting a $0.1\mu\text{F}$ capacitor from AIN- (the selected analog input, respectively) to AGND.

During the acquisition interval, the channel selected as the positive input (IN+) charges capacitor C_{HOLD} . The acquisition interval spans three SCLK cycles and ends on the falling SCLK edge after the last bit of the input control word has been entered. At the end of the acquisition interval, the T/H switch opens, retaining charge on C_{HOLD} as a sample of the signal at IN+.

The conversion interval begins with the input multiplexer switching C_{HOLD} from the positive input (IN+) to the negative input (IN-). In single-ended mode, IN- is simply AGND. This unbalances node ZERO at the input of the comparator. The capacitive DAC adjusts during the remainder of the conversion cycle to restore its node ZERO to 0V within the limits of its resolution. This action is equivalent to transferring a charge of $16\text{pF} \times (V_{\text{IN}+} - V_{\text{IN}-})$ from C_{HOLD} to the binary-weighted capacitive DAC, which in turn forms a digital representation of the analog input signal.

Low-Power, 8-Channel, Serial 10-Bit ADC

Track/Hold

The T/H enters its tracking mode on the falling clock edge after the fifth bit of the 8-bit control word has been shifted in. The T/H enters its hold mode on the falling clock edge after the eighth bit of the control word has been shifted in. If the converter is set up for single-ended inputs, IN⁺ is connected to AGND, and the converter samples the "+" input. If the converter is set up for differential inputs, IN⁻ connects to the "-" input, and the difference of |IN⁺ - IN⁻| is sampled. At the end of the conversion, the positive input connects back to IN⁺, and C_{HOLD} charges to the input signal.

The time required for the T/H to acquire an input signal is a function of how quickly its input capacitance is charged. If the input signal's source impedance is high, the acquisition time lengthens and more time must be allowed between conversions. Acquisition time is calculated by:

$$t_{AZ} = 9 (R_S + R_{IN}) 16pF$$

where $R_{IN} = 5k\Omega$, R_S = the source impedance of the input signal, and t_{AZ} is never less than 1.5 μ s. Note that source impedances below 5k Ω do not significantly affect the AC performance of the ADC.

Input Bandwidth

The ADC's input tracking circuitry has a 4.5MHz small-signal bandwidth, so it is possible to digitize high-speed transient events and measure periodic signals with bandwidths exceeding the ADC's sampling rate by using undersampling techniques. To avoid high-frequency signals being aliased into the frequency band of interest, anti-alias filtering is recommended. See the data sheets for the MAX291-MAX297 filters.

Analog Input Range and Input Protection

Internal protection diodes, which clamp the analog input to V_{DD} and AGND, allow the channel input pins to swing from $AGND - 0.3V$ to $V_{DD} + 0.3V$ without damage. However, for accurate conversions near full scale, the inputs must not exceed V_{DD} by more than 50mV, or be lower than AGND by 50mV.

If an off-channel analog input exceeds the supplies by more than 50mV, current will flow through the protection diodes on that input. If this current exceeds 2mA, the accuracy of the on-channel's conversion will be degraded.

The MAX192 can be configured for differential (unipolar or bipolar) or single-ended (unipolar only) inputs, as selected by bits 2 and 3 of the control byte (Table 3).

In the single-ended mode, set the UN/BIP bit to unipolar. In this mode, analog inputs are internally referenced to AGND, with a full-scale input range from 0V to V_{REF} .

In differential mode, both unipolar and bipolar settings can be used. Choosing unipolar mode sets the differential input range at 0V to V_{REF} . The output code is invalid (code zero) when a negative differential input voltage is applied. Bipolar mode sets the differential input range to $\pm V_{REF} / 2$. Note that in this differential mode, the common-mode input range includes both supply rails. Refer to Tables 4a and 4b for input voltage ranges.

Quick Look

To evaluate the analog performance of the MAX192 quickly, use Figure 5's circuit. The MAX192 requires a control byte to be written to DIN before each

Table 1. Channel Selection in Single-Ended Mode (SGL/DIFF = 1)

SEL2	SEL1	SEL0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	AGND
0	0	0	+								-
1	0	0		+							-
0	0	1			+						-
1	0	1				+					-
0	1	0					+				-
1	1	0						+			-
0	1	1							+		-
1	1	1								+	-

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

Table 2. Channel Selection in Differential Mode (SGL/DIFF = 0)

SEL2	SEL1	SEL0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
0	0	0	+	-						
0	0	1			+	-				
0	1	0					+	-		
0	1	1							+	-
1	0	0	-	+						
1	0	1			-	+				
1	1	0					-	+		
1	1	1							-	+

Table 3. Control-Byte Format

Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
START	SEL2	SEL1	SEL0	UNI/BIP	SGL/DIF	PD1	PD0
Bit	Name	Description					
7(MSB)	START	The first logic "1" bit after \overline{CS} goes low defines the beginning of the control byte.					
6 5 4	SEL2 SEL1 SEL0	These three bits select which of the eight channels are used for the conversion. See Tables 1 and 2.					
3	UNI/BIP	1 = unipolar, 0 = bipolar. Selects unipolar or bipolar conversion mode. In unipolar mode, an analog input signal from 0V to VREF can be converted; in differential bipolar mode, the differential signal can range from $-VREF/2$ to $+VREF/2$. Select differential operation if bipolar mode is used.					
2	SGL/DIF	1 = single ended, 0 = differential. Selects single-ended or differential conversions. In single-ended mode, input signal voltages are referred to AGND. In differential mode, the voltage difference between two channels is measured. Select unipolar operation if single-ended mode is used. See Tables 1 and 2.					
1 0(LSB)	PD1 PD0	Selects clock and power-down modes.					
		PD1	PD0	Mode			
		0	0	Full power-down ($I_Q = 2\mu A$)			
		0	1	Fast power-down ($I_Q = 30\mu A$)			
		1	0	Internal clock mode			
		1	1	External clock mode			

Low-Power, 8-Channel, Serial 10-Bit ADC

Table 4a. Unipolar Full Scale and Zero Scale

REFERENCE		ZERO SCALE	FULL SCALE
Internal Reference		0V	+4.096V
External Reference	at REFADJ	0V	VREFADJ (1.678)
	at VREF	0V	VREF

conversion. Tying DIN to +5V feeds in control bytes of \$FF (HEX), which trigger single-ended conversions on CH7 in external clock mode without powering down between conversions. In external clock mode, the SSTRB output pulses high for one clock period before the most significant bit of the conversion result comes out of DOUT. Varying the analog input to CH7 should alter the sequence of bits from DOUT. A total of 15 clock cycles is required per conversion. All transitions of the SSTRB and DOUT outputs occur on the falling edge of SCLK.

How to Start a Conversion

A conversion is started on the MAX192 by clocking a control byte into DIN. Each rising edge on SCLK, with CS low, clocks a bit from DIN into the MAX192's internal shift register. After CS falls, the first arriving logic "1" bit defines the MSB of the control byte. Until this first "start" bit arrives, any number of logic "0" bits can be clocked into DIN with no effect. Table 3 shows the control-byte format.

The MAX192 is compatible with Microwire, SPI, and QSPI devices. For SPI, select the correct clock polarity and sampling edge in the SPI control registers: set CPOL = 0 and CPHA = 0. Microwire and SPI both transmit a byte and receive a byte at the same time. Using the *Typical Operating Circuit*, the simplest software interface requires only three 8-bit transfers to perform a conversion (one 8-bit transfer to configure the ADC, and two more 8-bit transfers to clock out the 12-bit conversion result).

Example: Simple Software Interface

Make sure the CPU's serial interface runs in master mode so the CPU generates the serial clock. Choose a clock frequency from 100kHz to 2MHz.

- 1) Set up the control byte for external clock mode, call it TB1. TB1 should be of the format:

Table 4b. Differential Bipolar Full Scale, Zero Scale, and Negative Full Scale

REFERENCE		NEGATIVE FULL SCALE	ZERO SCALE	FULL SCALE
Internal Reference		-4.096V / 2	0V	+4.096V / 2
External Reference	at REFADJ	-1/2 VREFADJ (1.678)	0V	+1/2 VREFADJ (1.678)
	at VREF	-1/2 VREF	0V	+1/2 VREF

1XXXXX11 binary, where the Xs denote the particular channel and conversion-mode selected.

- 2) Use a general-purpose I/O line on the CPU to pull CS on the MAX192 low.
- 3) Transmit TB1 and simultaneously receive a byte and call it RB1. Ignore RB1.
- 4) Transmit a byte of all zeros (\$00 HEX) and simultaneously receive byte RB2.
- 5) Transmit a byte of all zeros (\$00 HEX) and simultaneously receive byte RB3.
- 6) Pull CS on the MAX192 high.

Figure 6 shows the timing for this sequence. Bytes RB2 and RB3 will contain the result of the conversion padded with one leading zero, two sub-LSB bits, and three trailing zeros. The total conversion time is a function of the serial clock frequency and the amount of dead time between 8-bit transfers. Make sure that the total conversion time does not exceed 120µs, to avoid excessive T/H droop.

Digital Output

In unipolar input mode, the output is straight binary (Figure 15). For bipolar inputs in differential mode, the output is two's-complement (Figure 16). Data is clocked out at the falling edge of SCLK in MSB-first format.

Internal and External Clock Modes

The MAX192 may use either an external serial clock or the internal clock to perform the successive-approximation conversion. In both clock modes, the external clock shifts data in and out of the MAX192. The T/H acquires the input signal as the last three bits of the control byte are clocked into DIN. Bits PD1 and PD0 of the control byte program the clock mode. Figures 7 through 10 show the timing characteristics common to both modes.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

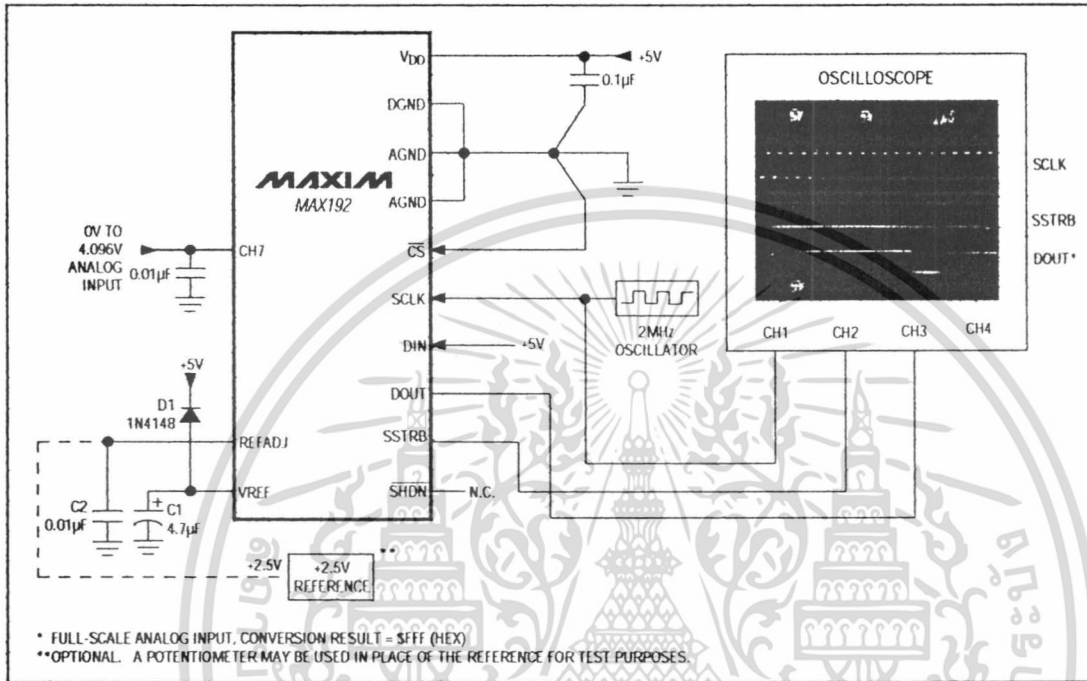


Figure 5. Quick-Look Circuit

External Clock

In external clock mode, the external clock not only shifts data in and out, it also drives the analog-to-digital conversion steps. SSTRB pulses high for one clock period after the last bit of the control byte. Successive-approximation bit decisions are made and appear at DOUT on each of the next 12 SCLK falling edges (see Figure 6). The first 10 bits are the true data bits, and the last two are sub-LSB bits.

SSTRB and DOUT go into a high-impedance state when CS goes high; after the next CS falling edge, SSTRB will output a logic low. Figure 8 shows the SSTRB timing in external clock mode.

The conversion must complete in some minimum time, or else droop on the sample-and-hold capacitors may degrade conversion results. Use internal clock mode if the clock period exceeds 10µs, or if serial-clock interruptions could cause the conversion interval to exceed 120µs.

Internal Clock

In internal clock mode, the MAX192 generates its own conversion clock internally. This frees the microprocessor from the burden of running the SAR conversion clock, and allows the conversion results to be read back at the processor's convenience, at any clock rate from zero to typically 10MHz. SSTRB goes low at the start of the conversion and then goes high when the conversion is complete. SSTRB will be low for a maximum of 10µs, during which time SCLK should remain low for best noise performance. An internal register stores data when the conversion is in progress. SCLK clocks the data out at this register at any time after the conversion is complete. After SSTRB goes high, the next falling clock edge will produce the MSB of the conversion at DOUT, followed by the remaining bits in MSB-first format (Figure 9). CS does not need to be held low once a conversion is started.

Pulling CS high prevents data from being clocked into

Low-Power, 8-Channel, Serial 10-Bit ADC

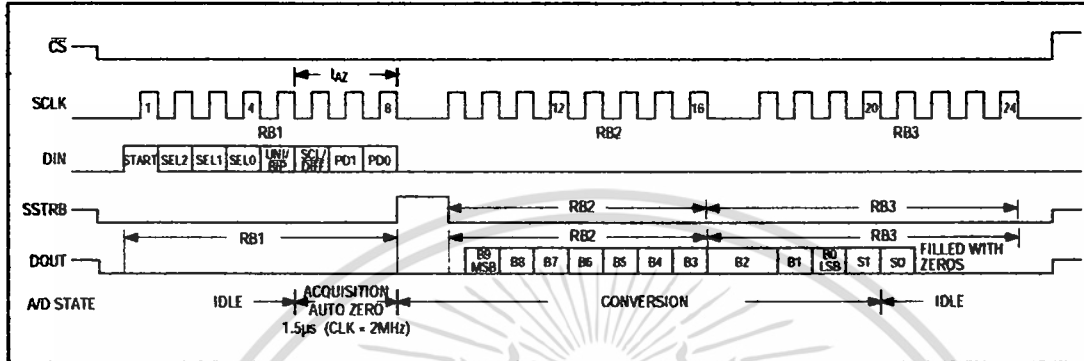


Figure 6. 24-Bit External Clock Mode Conversion Timing (SPI, QSPI and Microwire Compatible)

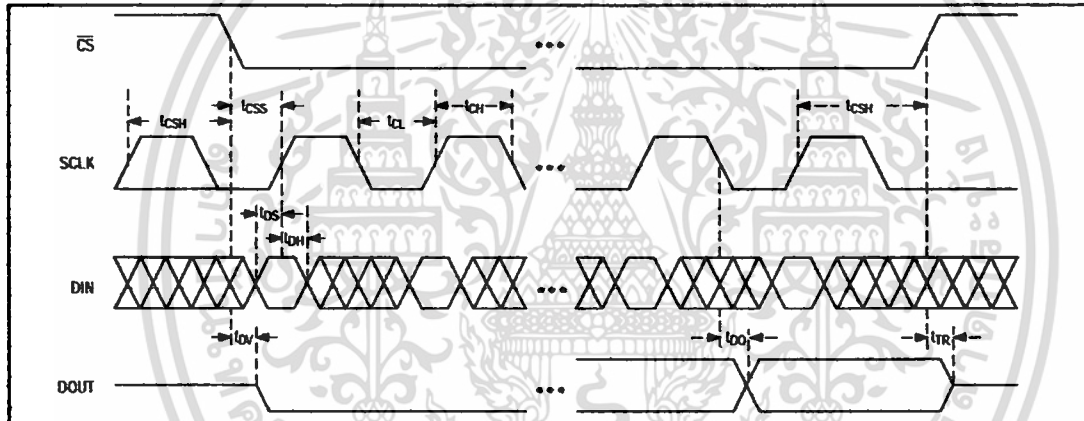


Figure 7. Detailed Serial-Interface Timing

the MAX192 and three-states DOUT, but it does not adversely affect an internal clock-mode conversion already in progress. When internal clock mode is selected, SSTRB does not go into a high-impedance state when CS goes high.

Figure 10 shows the SSTRB timing in internal clock mode. In internal clock mode, data can be shifted in and out of the MAX192 at clock rates exceeding 4.0MHz, provided that the minimum acquisition time, t_{AZ} , is kept above 1.5µs.

Data Framing

The falling edge of \overline{CS} does not start a conversion on the MAX192. The first logic high clocked into DIN is interpreted as a start bit and defines the first bit of the control byte. A conversion starts on the falling edge of SCLK,

after the eighth bit of the control byte (the PD0 bit) is clocked into DIN. The start bit is defined as:

The first high bit clocked into DIN with \overline{CS} low anytime the converter is idle, e.g. after V_{DD} is applied.

OR

The first high bit clocked into DIN after bit 3 of a conversion in progress is clocked onto the DOUT pin.

If a falling edge on \overline{CS} forces a start bit before bit 3 (B3) becomes available, then the current conversion will be terminated and a new one started. Thus, the fastest the MAX192 can run is 15 clocks per conversion. Figure 11a shows the serial-interface timing necessary to perform a conversion every 15 SCLK cycles in external clock mode.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

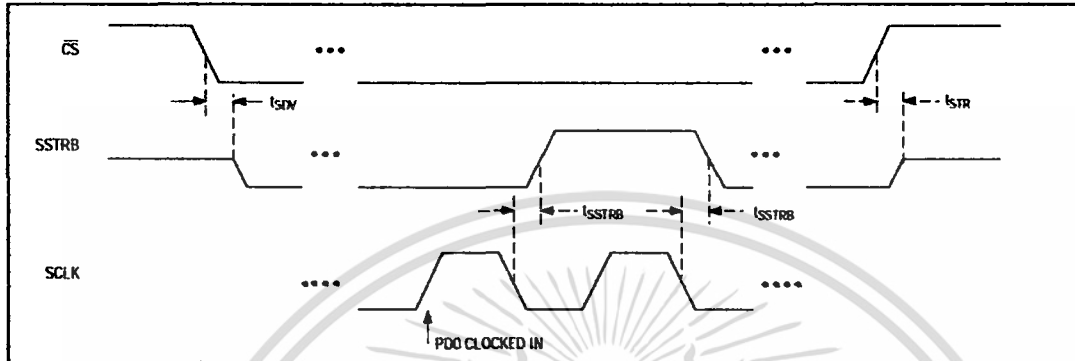


Figure 8. External Clock Mode SSTRB Detailed Timing

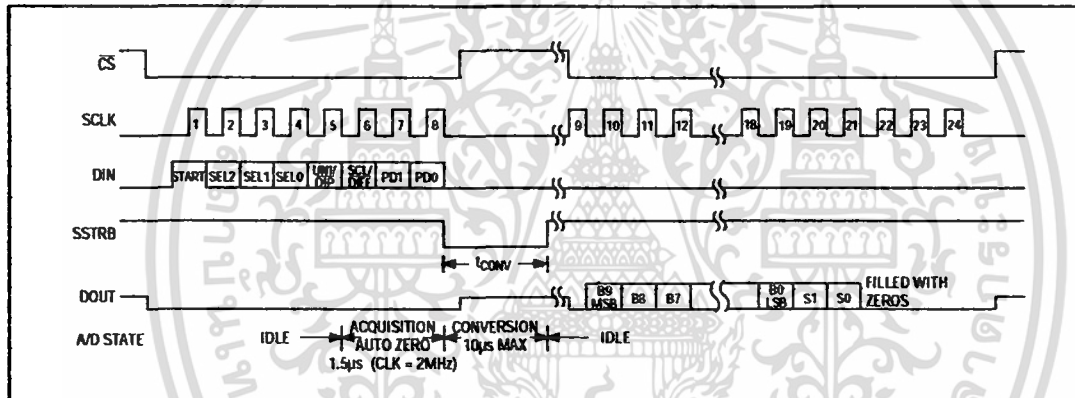


Figure 9. Internal Clock Mode Timing

Most microcontrollers require that conversions occur in multiples of 8 SCLK clocks; 16 clocks per conversion will typically be the fastest that a microcontroller can drive the MAX192. Figure 11b shows the serial-interface timing necessary to perform a conversion every 16 SCLK cycles in external clock mode.

Applications Information

Power-On Reset

When power is first applied and if $\overline{\text{SHDN}}$ is not pulled low, internal power-on reset circuitry will activate the MAX192 in internal clock mode, ready to convert with $\text{SSTRB} = \text{high}$. After the power supplies have been stabilized, the internal reset time is 100 μs and no conversions should be performed during this phase. SSTRB is high on power-up and, if $\overline{\text{CS}}$ is low, the first logical 1 on DIN will be interpreted as a start bit. Until a conversion

takes place, DOUT will shift out zeros.

Reference-Buffer Compensation

In addition to its shutdown function, the $\overline{\text{SHDN}}$ pin also selects internal or external compensation. The compensation affects both power-up time and maximum conversion speed. Compensated or not, the minimum clock rate is 100kHz due to droop on the sample-and-hold.

To select external compensation, float $\overline{\text{SHDN}}$. See the *Typical Operating Circuit*, which uses a 4.7 μF capacitor at VREF . A value of 4.7 μF or greater ensures stability and allows operation of the converter at the full clock speed of 2MHz. External compensation increases power-up time (see the *Choosing Power-Down Mode* section, and Table 5).

Internal compensation requires no external capacitor at VREF , and is selected by pulling $\overline{\text{SHDN}}$ high. Internal compensation allows for shortest power-up times, but is

MAXIM

13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

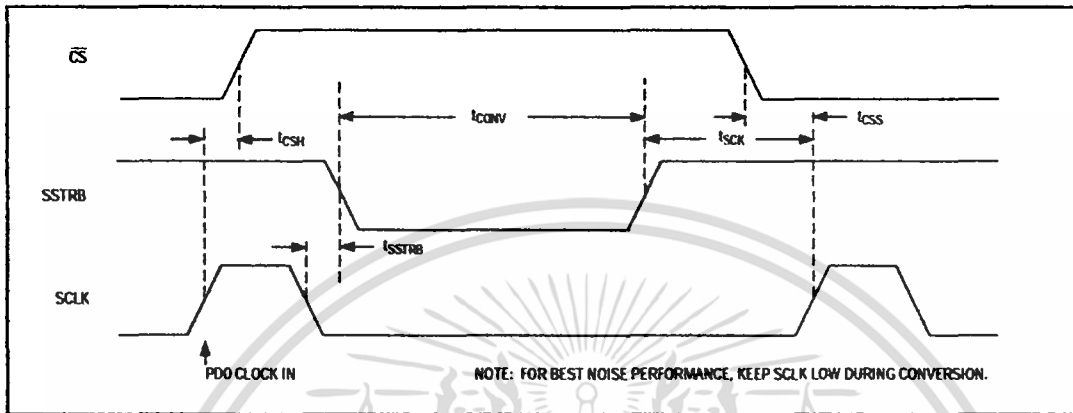


Figure 10. Internal Clock Mode SSTRB Detailed Timing

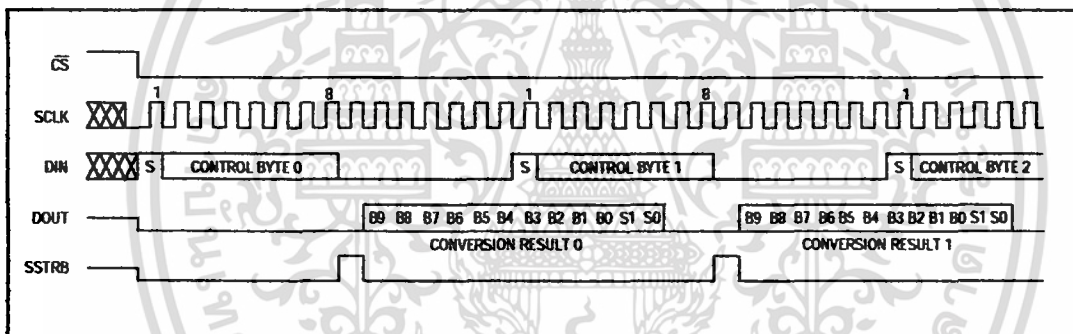


Figure 11a. External Clock Mode, 15 Clocks/Conversion Timing

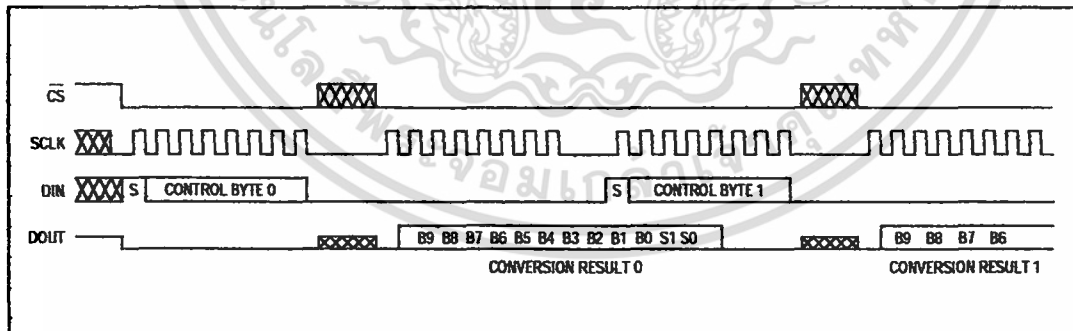


Figure 11b. External Clock Mode, 16 Clocks/Conversion Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

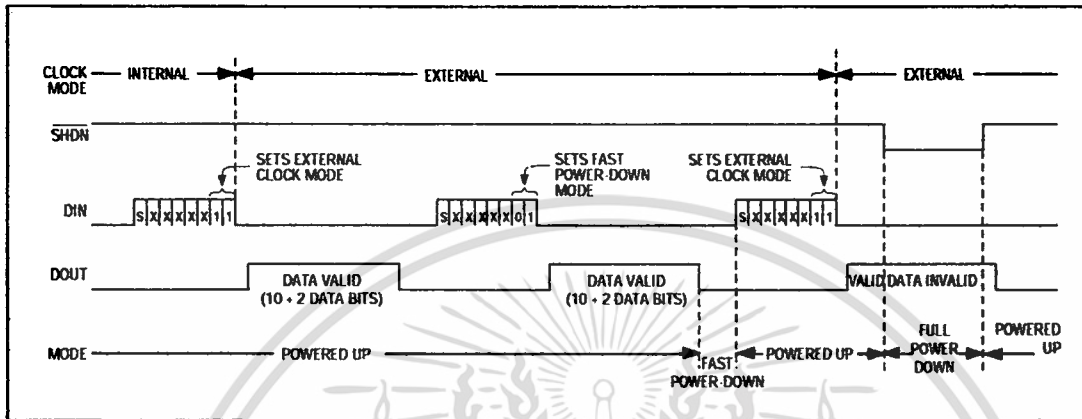


Figure 12a. Timing Diagram Power-Down Modes, External Clock

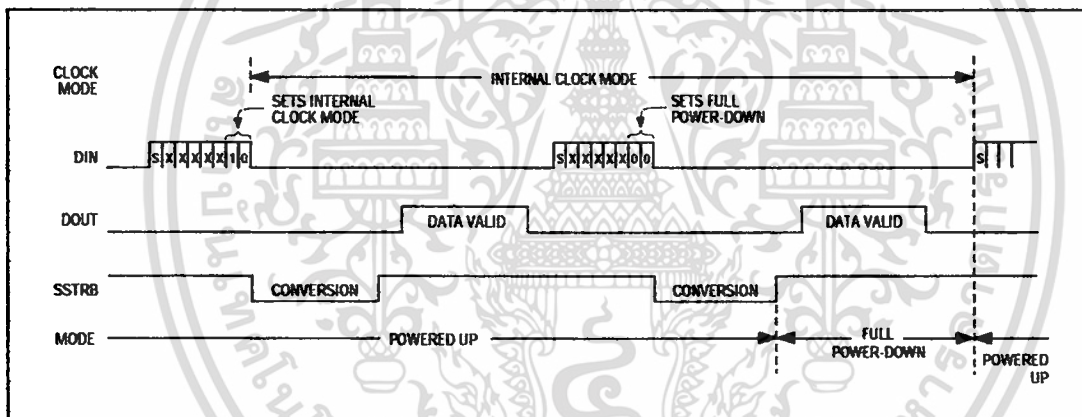


Figure 12b. Timing Diagram Power-Down Modes, Internal Clock

only available using an external clock and reduces the maximum clock rate to 400kHz.

Power-Down

Choosing Power-Down Mode

You can save power by placing the converter in a low-current shutdown state between conversions. Select full power-down or fast power-down mode via bits 7 and 8 of the DIN control byte with SHDN high (see Tables 3 and 6). Pull SHDN low at any time to shut down the converter completely. SHDN overrides bits 7 and 8 of DIN word (see Table 7).

Full power-down mode turns off all chip functions that draw quiescent current, typically reducing I_{DD} to 2 μ A.

Fast power-down mode turns off all circuitry except the bandgap reference. With the fast power-down mode, the supply current is 30 μ A. Power-up time can be shortened to 5 μ s in internal compensation mode.

In both software shutdown modes, the serial interface remains operational, however, the ADC will not convert. Table 5 illustrates how the choice of reference-buffer compensation and power-down mode affects both power-up delay and maximum sample rate.

Low-Power, 8-Channel, Serial 10-Bit ADC

Table 6. Worst-Case Power-Up Delay Times

Reference Buffer	Reference-Buffer Compensation Mode	VREF Capacitor (μF)	Power-Down Mode	Power-Up Delay (sec)	Maximum Sampling Rate (ksps)
Enabled	Internal		Fast	5 μ	26
Enabled	Internal		Full	300 μ	26
Enabled	External	4.7	Fast	See Figure 14c	133
Enabled	External	4.7	Full	See Figure 14c	133
Disabled			Fast	2 μ	133
Disabled			Full	2 μ	133

Table 6. Software Shutdown and Clock Mode

PD1	PD0	Device Mode
1	1	External Clock Mode
1	0	Internal Clock Mode
0	1	Fast Power-Down Mode
0	0	Full Power-Down Mode

In external compensation mode, the power-up time is 20ms with a 4.7 μF compensation capacitor when the capacitor is fully discharged. In fast power-down, you can eliminate start-up time by using low-leakage capacitors that will not discharge more than 1/2LSB while shut down. In shutdown, the capacitor has to supply the current into the reference (1.5 μA typ) and the transient currents at power-up.

Figures 12a and 12b illustrate the various power-down sequences in both external and internal clock modes.

Software Power-Down

Software power-down is activated using bits PD1 and PD0 of the control byte. As shown in Table 6, PD1 and PD0 also specify the clock mode. When software shutdown is asserted, the ADC will continue to operate in the last specified clock mode until the conversion is complete. Then the ADC powers down into a low quiescent-current state. In internal clock mode, the interface remains active and conversion results may be clocked out while the MAX192 has already entered a software power-down.

The first logical 1 on DIN will be interpreted as a start bit, and powers up the MAX192. Following the start bit, the data input word or control byte also determines

Table 7. Hard-Wired Shutdown and Compensation Mode

SHDN State	Device Mode	Reference-Buffer Compensation
1	Enabled	Internal Compensation
Floating	Enabled	External Compensation
0	Full Power-Down	N/A

clock and power-down modes. For example, if the DIN word contains PD1 = 1, then the chip will remain powered up. If PD1 = 0, a power-down will resume after one conversion.

Hardware Power-Down

The SHDN pin places the converter into the full power-down mode. Unlike with the software shutdown modes, conversion is not completed. It stops coincidentally with SHDN being brought low. There is no power-up delay if an external reference is used and is not shut down. The SHDN pin also selects internal or external reference compensation (see Table 7).

Power-Down Sequencing

The MAX192 auto power-down modes can save considerable power when operating at less than maximum sample rates. The following discussion illustrates the various power-down sequences.

Lowest Power at up to 500 Conversions/Channel/Second

The following examples illustrate two different power-down sequences. Other combinations of clock rates, compensation modes, and power-down modes may give lowest power consumption in other applications.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

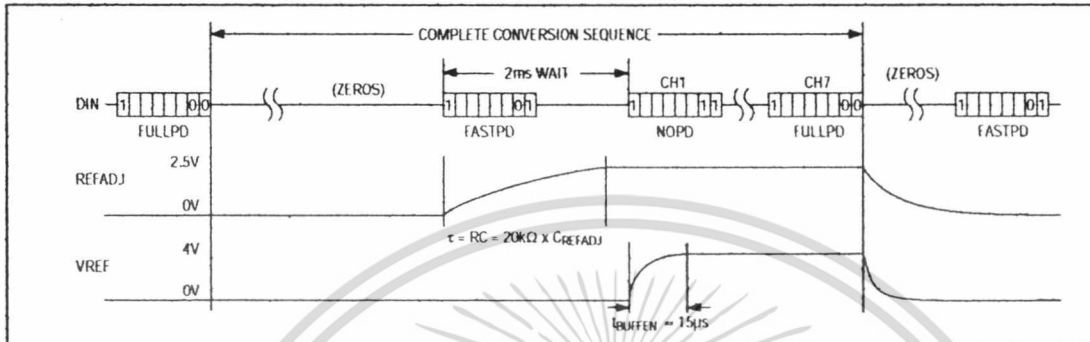


Figure 13. FULLPD/FASTPD Power-Up Sequence

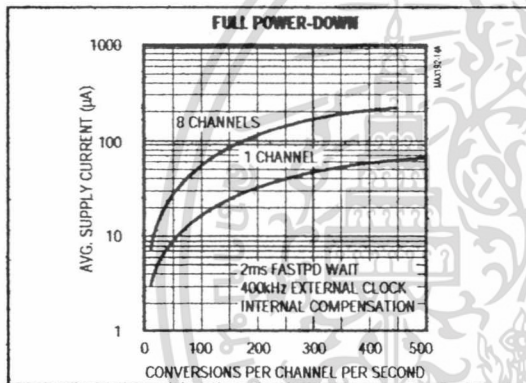


Figure 14a. Supply Current vs. Sample Rate/Second, FULLPD, 400kHz Clock

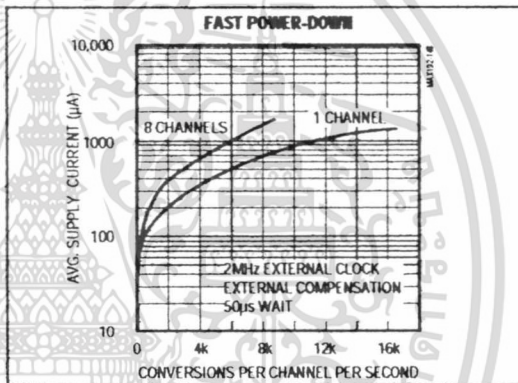


Figure 14b. Supply Current vs. Sample Rate/Second, FASTPD, 2MHz Clock

Figure 14a depicts the MAX192 power consumption for one or eight channel conversions utilizing full power-down mode and internal reference compensation. A 0.01 μ F bypass capacitor at REFADJ forms an RC filter with the internal 20k Ω reference resistor with a 0.2ms time constant. To achieve full 10-bit accuracy, 10 time constants or 2ms are required after power-up. Waiting 2ms in FASTPD mode instead of full power-up will reduce the power consumption by a factor of 10 or more. This is achieved by using the sequence shown in Figure 13.

Lowest Power at Higher Throughputs

Figure 14b shows the power consumption with external-reference compensation in fast power-down, with one and eight channels converted. The external 4.7 μ F compensation requires a 50 μ s wait after power-up, accomplished by 75 idle clocks after a

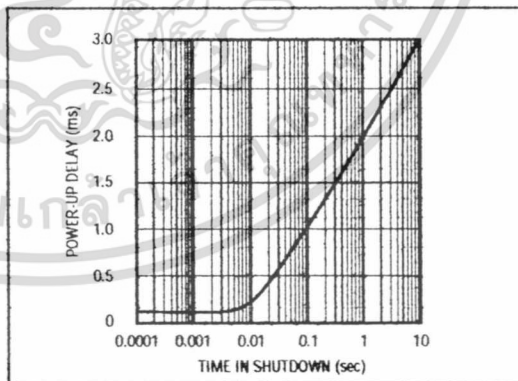


Figure 14c. Typical Power-Up Delay vs. Time in Shutdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

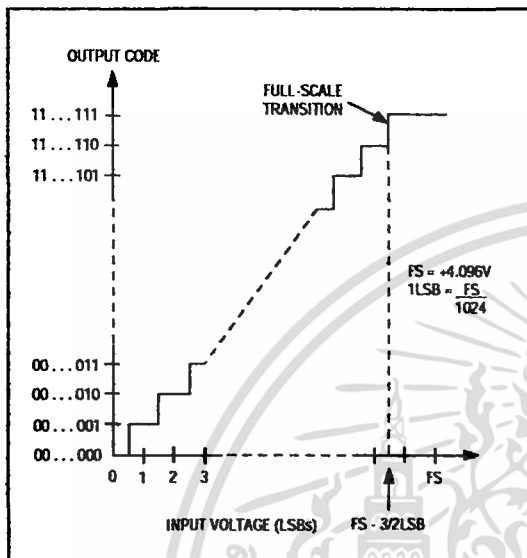


Figure 15. Unipolar Transfer Function, 4.096V = Full Scale

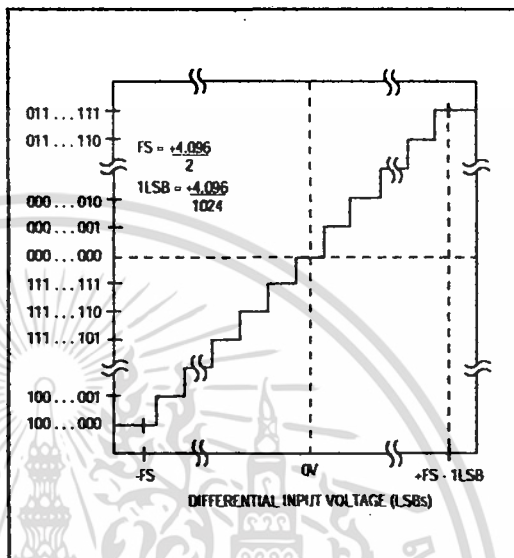


Figure 16. Differential Bipolar Transfer Function, $\pm 4.096V/2 =$ Full Scale

dummy conversion. This circuit combines fast multi-channel conversion with lowest power consumption possible. Full power-down mode may provide increased power savings in applications where the MAX192 is inactive for long periods of time, but where intermittent bursts of high-speed conversions are required.

External and Internal References

The MAX192 can be used with an internal or external reference. Diode D1 shown in the *Typical Operating Circuit* ensures correct start-up. Any standard signal diode can be used. An external reference can either be connected directly at the VREF terminal or at the REFADJ pin.

The MAX192's internally trimmed 2.46V reference is buffered with a gain of 1.678 to scale an external 2.5V reference at REFADJ to 4.096V at VREF.

Internal Reference

The full-scale range of the MAX192 with internal reference is 4.096V with unipolar inputs, and $\pm 2.048V$ with differential bipolar inputs. The internal reference voltage is adjustable to $\pm 1.5\%$ with the Reference-Adjust Circuit of Figure 17.

External Reference

An external reference can be placed at either the input (REFADJ) or the output (VREF) of the internal buffer amplifier. The REFADJ input impedance is typically 20k Ω . At VREF, the input impedance is a minimum of 12k Ω for DC currents. During conversion, an external reference at VREF must be able to deliver up to 350 μA DC load current and have an output impedance of 10 Ω or less. If the reference has higher output impedance or is noisy, bypass it close to the VREF pin with a 4.7 μF capacitor.

Using the buffered REFADJ input avoids external buffering of the reference. To use the direct VREF input, disable the internal buffer by tying REFADJ to V_{DD} .

Transfer Function and Gain Adjust

Figure 15 depicts the nominal, unipolar input/output (I/O) transfer function, and Figure 16 shows the differential bipolar input/output transfer function. Code transitions occur halfway between successive integer LSB values. Output coding is binary with $1LSB = 4.00mV$ ($4.096V / 1024$) for unipolar operation and $1LSB = 4.00mV$ [$(4.096V / 2 - 4.096V / 2) / 1024$] for bipolar operation.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

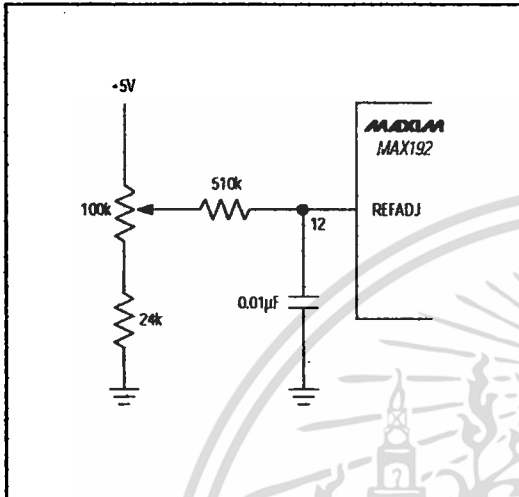


Figure 17. Reference Adjust Circuit

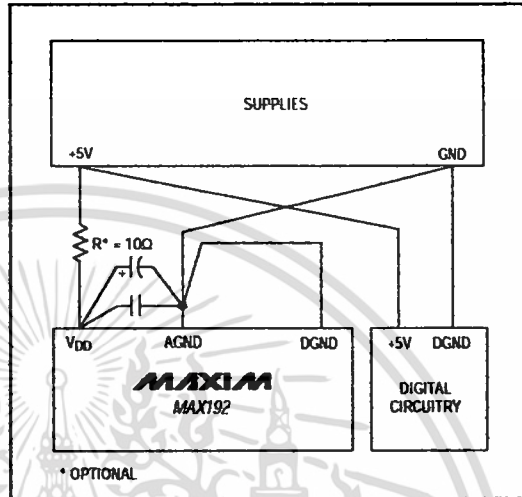


Figure 18. Power Supply Grounding Connection

Figure 17, the Reference-Adjust Circuit, shows how to adjust the ADC gain in applications that use the internal reference. The circuit provides $\pm 1.5\%$ (± 15 LSBs) of gain adjustment range.

Layout, Grounding, Bypassing

For best performance, use printed circuit boards. Wire-wrap boards are not recommended. Board layout should ensure that digital and analog signal lines are separated from each other. Do not run analog and digital (especially clock) lines parallel to one another, or digital lines underneath the ADC package.

Figure 18 shows the recommended system ground connections. A single-point analog ground ("star" ground point) should be established at AGND, separate from the logic ground. All other analog grounds and DGND should be connected to this ground. No other digital system ground should be connected to this single-point analog ground. The ground return to the power supply for this ground should be low impedance and as short as possible for noise-free operation.

High-frequency noise in the V_{DD} power supply may affect the high-speed comparator in the ADC. Bypass these supplies to the single-point analog ground with

0.1 μ F and 4.7 μ F bypass capacitors close to the MAX192. Minimize capacitor lead lengths for best supply-noise rejection. If the +5V power supply is very noisy, a 10 Ω resistor can be connected as a lowpass filter, as shown in Figure 18.

High-Speed Digital Interfacing

The MAX192 can interface with QSPI at high throughput rates using the circuit in Figure 19. This QSPI circuit can be programmed to do a conversion on each of the eight channels. The result is stored in memory without taxing the CPU since QSPI incorporates its own micro-sequencer.

Figure 20 details the code that sets up QSPI for autonomous operation. In external clock mode, the MAX192 performs a single-ended, unipolar conversion on each of the eight analog input channels. Figure 21 shows the timing associated with the assembly code of Figure 20. The first byte clocked into the MAX192 is the control byte, which triggers the first conversion on CH0. The last two bytes clocked into the MAX192 are all zero, and clock out the results of the CH7 conversion.

Low-Power, 8-Channel, Serial 10-Bit ADC

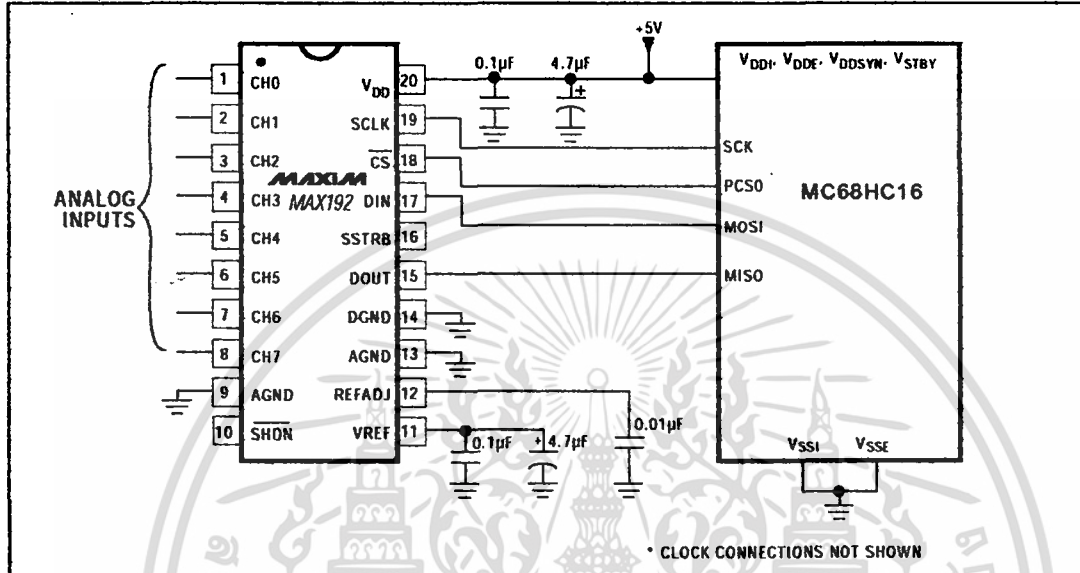


Figure 19. MAX192 QSPI Connection

TMS320 to MAX192 Interface

Figure 22 shows an application circuit to interface the MAX192 to the TMS320 in external clock mode. The timing diagram for this interface circuit is shown in Figure 23.

Use the following steps to initiate a conversion in the MAX192 and to read the results:

- 1) The TMS320 should be configured with CLKX (transmit clock) as an active-high output clock and CLKR (TMS320 receive clock) as an active-high input clock. CLKX and CLKR of the TMS320 are tied together with the SCLK input of the MAX192.
- 2) The MAX192 CS is driven low by the XF_{I/O} port of the TMS320 to enable data to be clocked into DIN of the MAX192.
- 3) An 8-bit word (1XXXXX11) should be written to the

MAX192 to initiate a conversion and place the device into external clock mode. Refer to Table 3 to select the proper XXXXX bit values for your specific application.

- 4) The SSTRB output of the MAX192 is monitored via the FSR input of the TMS320. A falling edge on the SSTRB output indicates that the conversion is in progress and data is ready to be received from the MAX192.
- 5) The TMS320 reads in one data bit on each of the next 16 rising edges of SCLK. These data bits represent the 10-bit conversion result and two sub-LSBs, followed by four trailing bits, which should be ignored.
- 6) Pull CS high to disable the MAX192 until the next conversion is initiated.

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

```

* Description :
*   This is a shell program for using a stand-alone 68HC16 without any external memory. The internal 1K RAM
*   is put into bank $0F to maintain 68HC11 code compatibility. This program was written with software
*   provided in the Motorola 68HC16 Evaluation Kit.
*
*   Roger J.A. Chen, Applications Engineer
*   MAXIM Integrated Products
*   November 20, 1992
*
.....
INCLUDE 'EQUATES.ASM' ;Equates for common reg addr
INCLUDE 'ORG00000.ASM' ;initialize reset vector
INCLUDE 'ORG00008.ASM' ;initialize interrupt vectors
ORG $0200 ;start program after interrupt vectors
INCLUDE 'INITSYS.ASM' ;set EK=F,XK=0,YK=0,ZK=0
;set sys clock at 16.78 MHz, COP off
INCLUDE 'INITRAM.ASM' ;turn on internal SRAM at $10000
;set stack (SK=1, SP=03FE)

MAIN:
JSR INITQSPI
MAINLOOP:
JSR READ192
WAIT:
LDAA SPSR
ANDA #80
BEQ WAIT ;wait for QSPI to finish
BRA MAINLOOP
ENDPROGRAM:

INITQSPI:
;This routine sets up the QSPI microsequencer to operate on its own.
;The sequencer will read all eight channels of a MAX192 each time
;it is triggered. The A/D converter results will be left in the
;receive data RAM. Each 16 bit receive data RAM location will
;have a leading zero, 10 + 2 bits of conversion result and three zeros.
;
;Receive RAM Bits 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
;A/D Result 0 MSB LSB 0 0 0
***** Initialize the QSPI Registers *****
PSHA
PSHB
LDAA #%01111000
STAA QPDR ;idle state for PCS0-3 = high
LDAA #%01111011
STAA QPAR ;assign port D to be QSPI
LDAA #%01111110
STAA QDDR ;only MISO is an input
LDD #8008
STD SPCR0 ;master mode, 16 bits/transfer,
;CPOL=CPHA=0, 1MHz Ser Clock
LDD #30000
STD SPCR1 ;set delay between PCS0 and SCK,
;set delay between transfers

```

Figure 20. MAX192 Assembly-Code Listing

MAXIM

21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

```

LDD #0800
STD SPCR2 ;set ENDQF to $8 for 9 transfers
***** Initialize QSPI Command RAM *****

LDAA #80 ;CONT=1,BITSE=0,DT=0,DSCK=0,PCSO=ACTIVE
STAA $FD40 ;store first byte in COMMAND RAM
LDAA #C0 ;CONT=1,BITSE=1,DT=0,DSCK=0,PCSO=ACTIVE
STAA $FD41
STAA $FD42
STAA $FD43
STAA $FD44
STAA $FD45
STAA $FD46
STAA $FD47
LDAA #40 ;CONT=0,BITSE=1,DT=0,DSCK=0,PCSO=ACTIVE
STAA $FD48
***** Initialize QSPI Transmit RAM *****

LDD #3008F
STD $FD20
LDD #300CF
STD $FD22
LDD #3009F
STD $FD24
LDD #300DF
STD $FD26
LDD #300AF
STD $FD28
LDD #300EF
STD $FD2A
LDD #300BF
STD $FD2C
LDD #300FF
STD $FD2E
LDD #30000
STD $FD30

PULB
PULA
RTS

READ192:
;This routine triggers the QSPI microsequencer to autonomously
;trigger conversions on all 8 channels of the MAX192. Each
;conversion result is stored in the receive data RAM.
PSHA
LDAA #80
ORAA SPCR1
STAA SPCR1 ;just set SPE
PULA
RTS

***** Interrupts/Exceptions *****

BDM: BGND ;exception vectors point here
;and put the user in background debug mode

```

Figure 20. MAX192 Assembly-Code Listing (continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

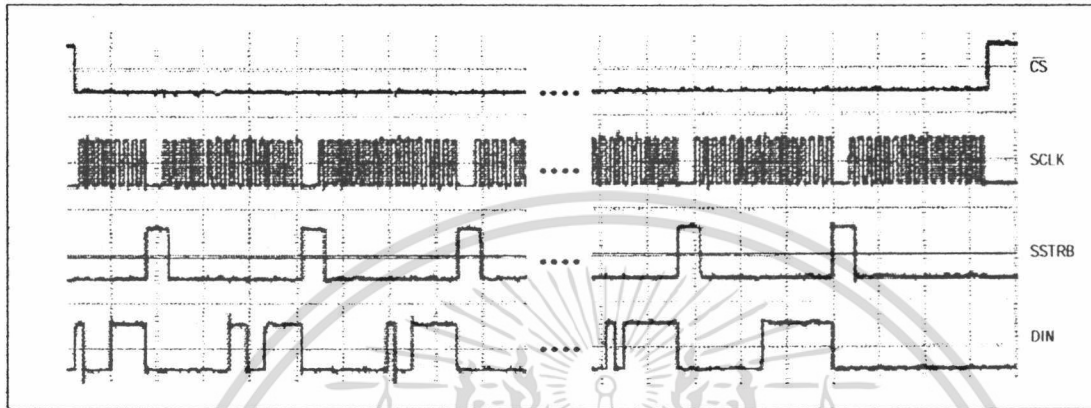


Figure 21. QSPI Assembly-Code Timing

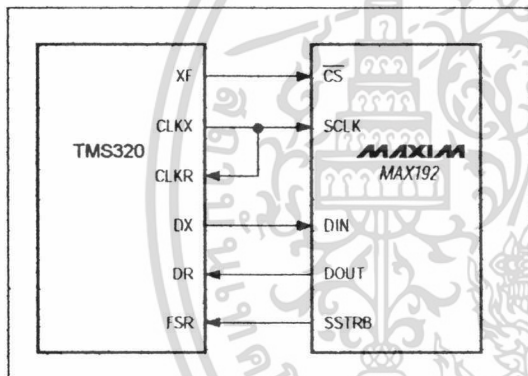


Figure 22. MAX192 to TMS320 Serial Interface

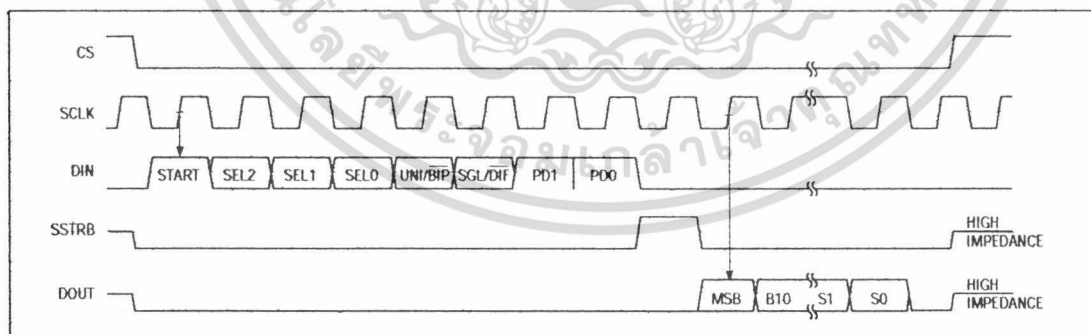


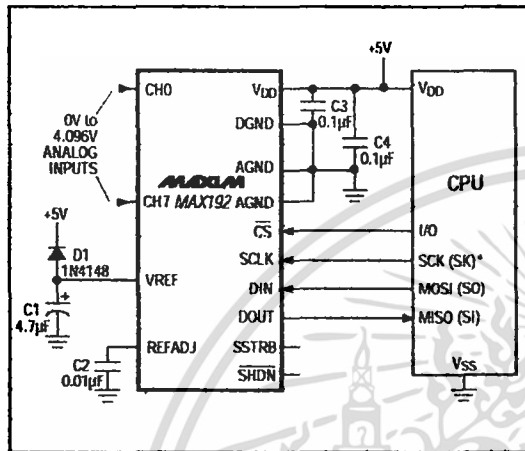
Figure 23. TMS320 Serial-Interface Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Power, 8-Channel, Serial 10-Bit ADC

MAX192

Typical Operating Circuit



TRANSISTOR COUNT: 2278

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

2A Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 (408) 737-7600

© 1994 Maxim Integrated Products

Printed USA

MAXIM is a registered trademark of Maxim Integrated Products.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS
SEMICONDUCTOR

DS275 Line-Powered RS-232 Transceiver Chip

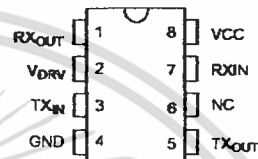
FEATURES

- Low-power serial transmitter/receiver for battery-backed systems
- Transmitter steals power from receive signal line to save power
- Ultra-low static current, even when connected to RS-232-E port
- Variable transmitter level from +5 to +12 volts
- Compatible with RS-232-E signals
- Available in 8-pin, 150-mil wide SOIC package (DS275S) and 14-pin TSSOP package
- Low-power CMOS

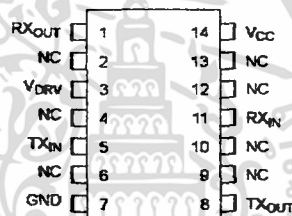
ORDERING INFORMATION

DS275 8-pin DIP
DS275S 8-pin SOIC
DS275E 14-pin TSSOP

PIN ASSIGNMENT



DS275 8-PIN DIP (300 MIL)
DS275S 8-PIN SOIC (150 MIL)



DS275E 14-PIN TSSOP

PIN DESCRIPTION

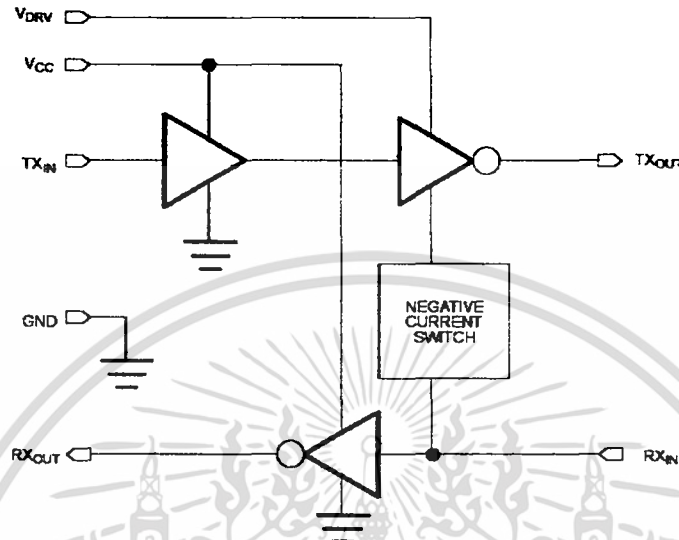
RX _{OUT}	– RS-232 Receiver Output
V _{DRV}	– Transmit Driver +V
TX _{IN}	– RS-232 Driver Input
GND	– System Ground (0V)
TX _{OUT}	– RS-232 Driver Output
NC	– No Connection
RX _{IN}	– RS-232 Receiver Input
V _{CC}	– System Logic Supply (+5V)

DESCRIPTION

The DS275 Line-Powered RS-232 Transceiver Chip is a CMOS device that provides a low-cost, very low-power interface to RS-232 serial ports. The receiver input translates RS-232 signal levels to common CMOS/TTL levels. The transmitter employs a unique circuit which steals current from the receive RS-232 signal when that signal is in a negative state (marking). Since most serial communication ports remain in a negative state statically, using the receive signal for negative

power greatly reduces the DS275's static power consumption. This feature is especially important for battery-powered systems such as laptop computers, remote sensors, and portable medical instruments. During an actual communication session, the DS275's transmitter will use system power (5–12 volts) for positive transitions while still employing the receive signal for negative transitions.

DS275 BLOCK DIAGRAM Figure 1



OPERATION

Designed for the unique requirements of battery-backed systems, the DS275 provides a low-power half-duplex interface to an RS-232 serial port. Typically, a designer must use an RS-232 device which uses system power during both negative and positive transitions of the transmit signal to the RS-232 port. If the connector to the RS-232 port is left connected for an appreciable time after the communication session has ended, power will statically flow into that port, draining the battery capacity. The DS275 eliminates this static current drain by stealing current from the receive line (RX_{IN}) of the RS-232 port when that line is at a negative level (marking). Since most asynchronous communication over an RS-232 connection typically remains in a marking state when data is not being sent, the DS275 will not consume system power in this condition. System power would only be used when positive-going transitions are needed on the transmit RS-232 output (TX_{OUT}) when data is sent. However, since synchronous communication sessions typically exhibit a very low duty-cycle, overall system power consumption remains low.

RECEIVER SECTION

The RX_{IN} pin is the receive input for an RS-232 signal whose levels can range from ± 3 to ± 15 volts. A negative data signal is called a mark while a positive data signal is

called a space. These signals are inverted and then level-shifted to normal +5 volt CMOS/TTL logic levels. The logic output associated with RX_{IN} is RX_{OUT} which swings from $+V_{CC}$ to ground. Therefore, a mark on RX_{IN} produces a logic 1 at RX_{OUT} ; a space produces a logic 0.

The input threshold of RX_{IN} is typically around 1.8 volts with 500 millivolts of hysteresis to improve noise rejection. Therefore, an input positive-going signal must exceed 1.8 volts to cause RX_{OUT} to switch states. A negative-going signal must now be lower than 1.3 volts (typically) to cause RX_{OUT} to switch again. An open on RX_{IN} is interpreted as a mark, producing a logic 1 at RX_{OUT} .

TRANSMITTER SECTION

TX_{IN} is the CMOS/TTL-compatible input for digital data from the user system. A logic 1 at TX_{IN} produces a mark (negative data signal) at TX_{OUT} while a logic 0 produces a space (positive data signal). As mentioned earlier, the transmitter section employs a unique driver design that uses the RX_{IN} line for swinging to negative levels. The RX_{IN} line must be in a marking or idle state to take advantage of this design; if RX_{IN} is in a spacing state, TX_{OUT} will only swing to ground. When TX_{OUT} needs to transition to a positive level, it uses the V_{DRV} power pin

for this level. V_{DRV} can be a voltage supply between 5 to 12 volts, and in many situations it can be tied directly to the +5 volt V_{CC} supply. *It is important to note that V_{DRV} must be greater than or equal to V_{CC} at all times.*

The voltage range on V_{DRV} permits the use of a 9-volt battery in order to provide a higher voltage level when TX_{OUT} is in a space state. When V_{CC} is shut off to the DS275 and V_{DRV} is still powered (as might happen in a battery-backed condition), only a small leakage current (about 50–100 nA) will be drawn. If TX_{OUT} is loaded during such a condition, V_{DRV} will draw current only if RX_{IN} is not in a negative state. During normal operation ($V_{CC}=5$ volts), V_{DRV} will draw less than 2 μ A when TX_{OUT} is marking. Of course, when TX_{OUT} is spacing, V_{DRV} will draw substantially more current—about 3 mA depending upon its voltage and the impedance that TX_{OUT} sees.

The TX_{OUT} output is slow-rate limited to less than 30 volts/ μ s in accordance with RS-232 specifications. In the event TX_{OUT} should be inadvertently shorted to ground, internal current-limiting circuitry prevents damage, even if continuously shorted.

RS-232 COMPATIBILITY

The intent of the DS275 is not so much to meet all the requirements of the RS-232 specification as to offer a low-power solution that will work with most RS-232 ports with a connector length of less than 10 feet. As a prime example, the DS275 will not meet the RS-232 requirement that the signal levels be at least ± 5 volts minimum when terminated by a 3K Ω load and $V_{DRV}=+5$ volts. Typically 4 volts will be present at TX_{OUT} when spacing under this condition. However, since most RS-232 receivers will correctly interpret any voltage over 2 volts as a space, there will be no problem transmitting data.

APPLICATIONS INFORMATION

The DS275 is designed as a low-cost, RS-232-E interface expressly tailored for the unique requirements of battery-operated handheld products. As shown in the electrical specifications, the DS275 draws exceptionally low operating and static current. During normal operation when data from the handheld system is sent from the TX_{OUT} output, the DS275 only draws significant V_{DRV} current when TX_{OUT} transitions positively (spacing). This current flows primarily into the RS-232 receiver's

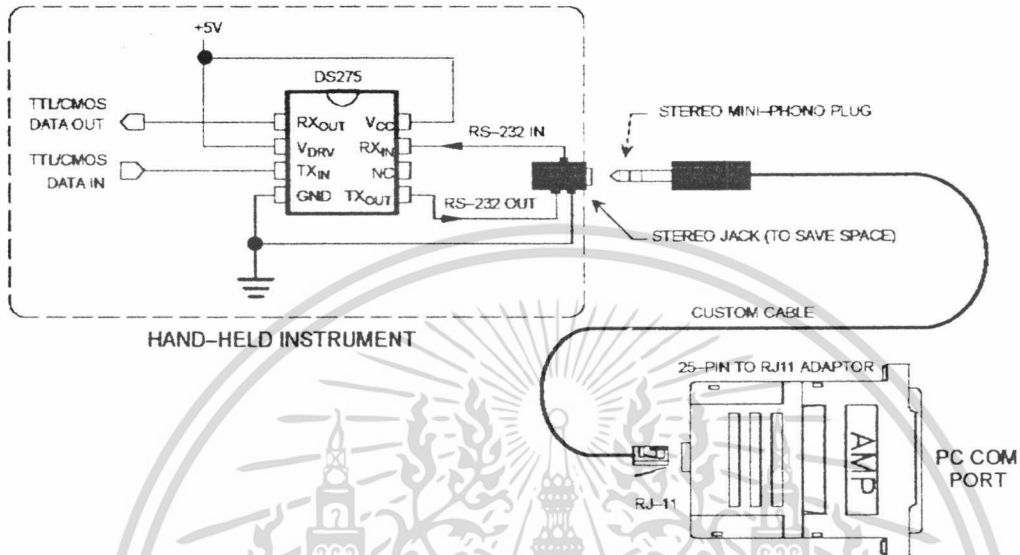
3–7K Ω load at the other end of the attaching cable. When TX_{OUT} is marking (a negative data signal), the V_{DRV} current falls dramatically since the negative voltage is provided by the transmit signal from the other end of the cable. This represents a large reduction in overall operating current, since typical RS-232 interface chips use charge-pump circuits to establish both positive and negative levels at the transmit driver output.

To obtain the lowest power consumption from the DS275, observe the following guidelines. First, to minimize V_{DRV} current when connected to an RS-232 port, always maintain TX_{IN} at a logic 1 when data is not being transmitted (idle state). This will force TX_{OUT} into the marking state, minimizing V_{DRV} current. Second, V_{DRV} current will drop to less than 100 nA when V_{CC} is grounded. Therefore, if V_{DRV} is tied directly to the system battery, the logic +5 volts can be turned off to achieve the lowest possible power state.

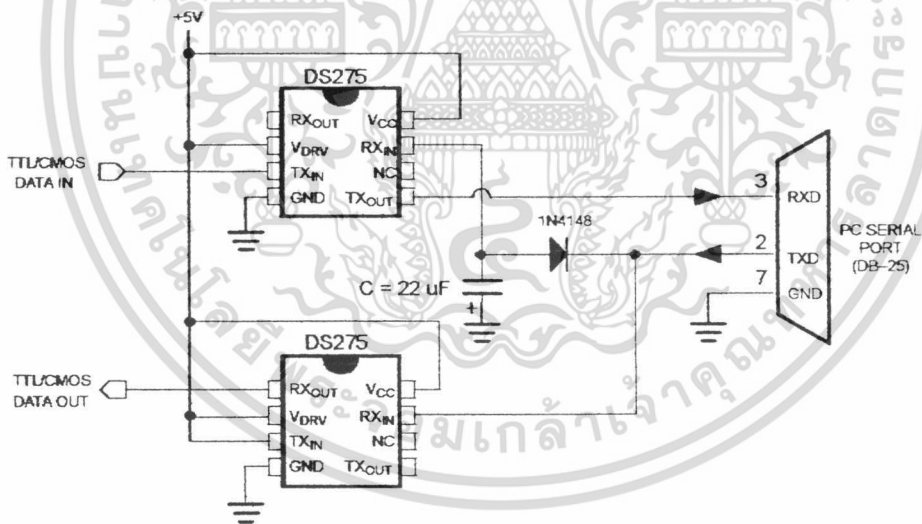
FULL-DUPLEX OPERATION

The DS275 is intended primarily for half-duplex operation; that is, RX_{IN} should remain idle in the marking state when transmitting data out TX_{OUT} and visa versa. However, the part can be operated full-duplex with most RS-232-E serial ports since signals swinging between 0 and +5V will usually be correctly interpreted by an RS-232-E receiver device. The 5-volt swing occurs when TX_{OUT} attempts to swing negative while RX_{IN} is at a positive voltage, which turns on an internal weak pull-down to ground for the TX_{OUT} driver's negative reference. So, transmit mark signals at TX_{OUT} may have voltage jumps from some negative value (corresponding to RX_{IN} marking) to approximately ground. One possible problem that may occur in this case is if the receiver at the other end requires a negative voltage for recognizing a mark. In this situation, the full-duplex circuit shown in Figure 3 can be used as an alternative. The 22 μ F capacitor forms a negative-charge reservoir; consequently, when the TXD line is spacing (positive), TX_{OUT} still has a negative source available for a time period determined by the capacitor and the load resistance at the other end (3–7K Ω). This circuit was tested from 150–19,200 bps with error-free operation using a SN75154 Quad Line Receiver as the receiver for the TX_{OUT} signal. Note that the SN75154 can have a marking input threshold below ground; hence there is the need for TX_{OUT} to swing both positive and negative in full-duplex operation with this device.

HANDHELD RS-232-C APPLICATION USING A STEREO MINI-JACK Figure 2



FULL-DUPLEX CIRCUIT USING NEGATIVE-CHARGE STORAGE Figure 3



NOTE:

The capacitor stores negative charge whenever the TXD signal from the PC serial port is in a marking data state (a negative voltage that is typically -10 volts). The top DS275's TX_{OUT} uses this negative charge reservoir when it is in a marking state. The capacitor will discharge to 0 volts when the TXD line is spacing (and TX_{OUT} is still marking) at a time constant determined by its value and the value of the load resistance reflected back to TX_{OUT}. However, when TXD is marking, the capacitor will quickly charge back to -10 volts. Note that TXD remains in a marking state when idle, which improves the performance of this circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

V_{CC}	-0.3 to +7.0 volts
V_{DRV}	-0.3 to +13.0 volts
RX_{IN}	± 15 volts
TX_{IN}	-0.3 to $V_{CC} + 0.3$ volts
TX_{OUT}	± 15 volts
RX_{OUT}	-0.3 to $V_{CC} + 0.3$ volts
Storage Temperature	-55°C to +125°C
Operating Temperature	0°C to 70°C

- * This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Logic Supply	V_{CC}	4.5	5.0	5.5	V	1
Transmit Driver Supply	V_{DRV}	4.5	5-12	13.0	V	1
Logic 1 Input	V_{IH}	2.0		$V_{CC}+0.3$	V	2
Logic 0 Input	V_{IL}	-0.3		+0.8	V	
RS-232 Input Range (RX_{IN})	V_{RS}	-15		+15	V	
Dynamic Supply Current $TX_{IN} = V_{CC}$	I_{DRV1}		400	800	μA	3
	I_{OC1}		40	100	μA	
$TX_{IN} = GND$	I_{DRV1}		3.8	5.0	mA	
	I_{OC1}		40	100	μA	
Static Supply Current $TX_{IN} = V_{CC}$	I_{DRV2}		1.5	10.0	μA	4
	I_{OC2}		10.0	15.0	μA	
$TX_{IN} = GND$	I_{DRV2}		3.8	5.0	mA	
	I_{OC2}		10.0	20.0	μA	
Driver Leakage Current ($V_{CC} = 0V$)	I_{DRV3}		0.05	1.0	μA	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC ELECTRICAL CHARACTERISTICS

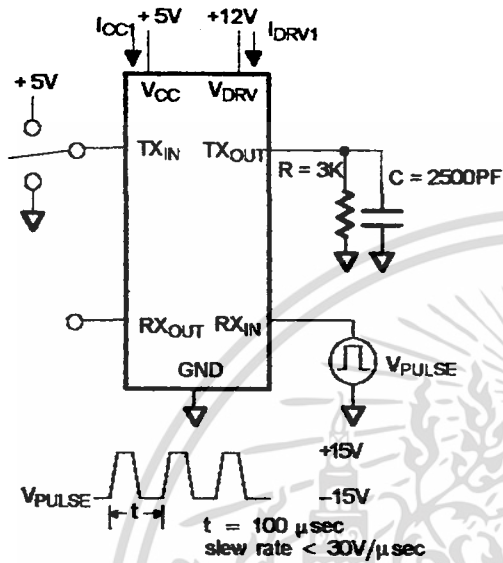
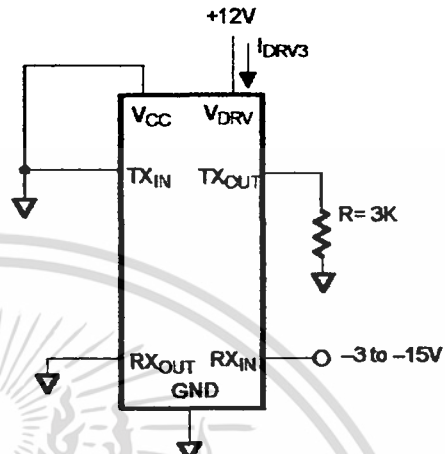
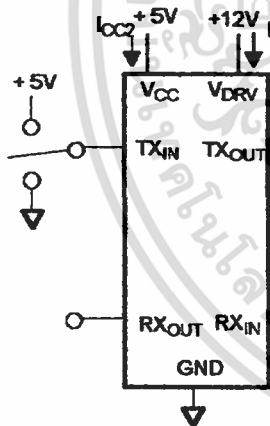
(0°C to 70°C; $V_{CC} = V_{DRV} = 5V \pm 10\%$)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
TX _{OUT} Level High	V _{OTXH}	3.5	4.0	5.0	V	6
TX _{OUT} Level Low	V _{OTXL}	-8.5	-9.0		V	7
TX _{OUT} Short Circuit Current	I _{SC}		+20	+85	mA	
TX _{OUT} Output Slew Rate	t _{SR}			30	V/μs	
Propagation Delay	t _{PD}		5		μs	8
RX _{IN} Input Threshold Low	V _{TL}	0.8	1.2	1.6	V	
RX _{IN} Input Threshold High	V _{TH}	1.6	2.0	2.4	V	
RX _{IN} Threshold Hysteresis	V _{HYS}	0.5	0.8		V	9
RX _{OUT} Output Current @ 2.4 V	I _{OH}	-1.0			mA	
RX _{OUT} Output Current @ 0.4 V	I _{OL}			32	mA	

NOTES:

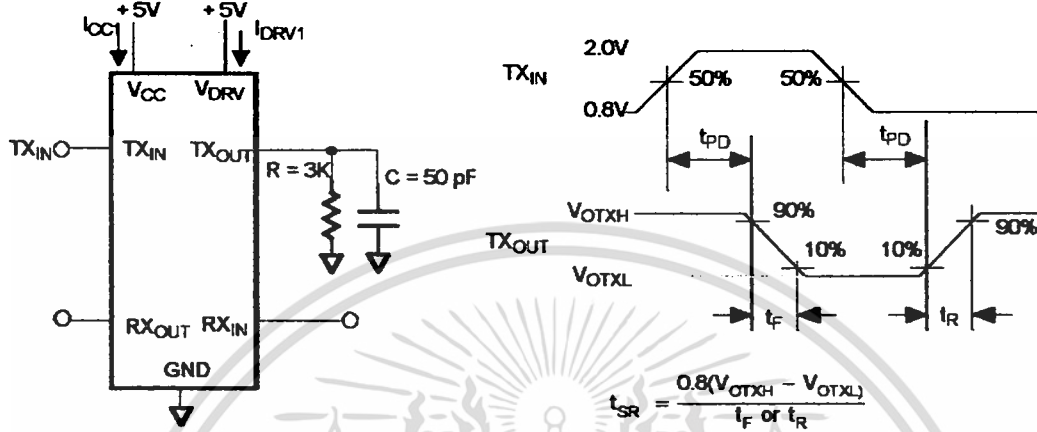
1. V_{DRV} must be greater than or equal to V_{CC} .
2. $V_{CC} = V_{DRV} = 5V \pm 10\%$.
3. See test circuit in Figure 4.
4. See test circuit in Figure 5.
5. See test circuit in Figure 6.
6. TX_{IN} = V_{IL} and TX_{OUT} loaded by 3KΩ to ground.
7. TX_{IN} = V_{IH}, RX_{IN} = -10 volts and TX_{OUT} loaded by 3KΩ to ground.
8. TX_{IN} to TX_{OUT} – see Figure 7.
9. $V_{HYS} = V_{TH} - V_{TL}$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

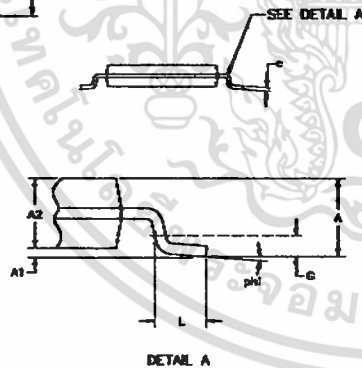
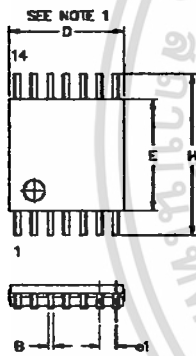
DYNAMIC OPERATING CURRENT TEST CIRCUIT Figure 4**DRIVER LEAKAGE TEST CIRCUIT** Figure 6**STATIC OPERATING CURRENT TEST CIRCUIT** Figure 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROPAGATION DELAY TEST CIRCUIT Figure 7



DS275E 14-PIN TSSOP



DIM	MIN	MAX
A MM	-	1.10
A1 MM	0.05	-
A2 MM	0.75	1.05
B MM	0.18	0.30
C MM	0.09	0.18
D MM	4.90	5.10
E MM	4.40 NOM	
e1 MM	0.65 BSC	
G MM	0.25 REF	
H MM	6.25	6.55
L MM	0.50	0.70
phi	0°	8°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS2003/DS9667/DS2004 High Current/Voltage Darlington Drivers

General Description

The DS2003/DS9667/DS2004 are comprised of seven high voltage, high current NPN Darlington pairs. All units feature common emitter, open collector outputs. To maximize their effectiveness, these units contain suppression diodes for inductive loads and appropriate emitter base resistors for leakage.

The DS2003/DS9667 has a series base resistor to each Darlington pair, thus allowing operation directly with TTL or CMOS operating at supply voltages of 5.0V.

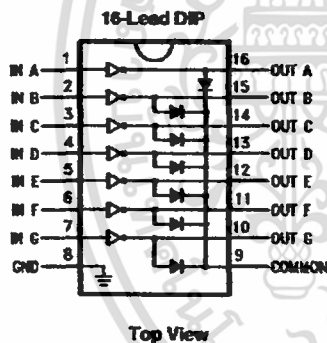
The DS2004 has an appropriate input resistor to allow direct operation from CMOS or PMOS outputs operating from supply voltages of 6.0V to 15V.

The DS2003/DS9667/DS2004 offer solutions to a great many interlace needs, including solenoids, relays, lamps, small motors, and LEDs. Applications requiring sink currents beyond the capability of a single output may be accommodated by paralleling the outputs.

Features

- Seven high gain Darlington pairs
- High output voltage ($V_{CE} = 50V$)
- High output current ($I_C = 350\text{ mA}$)
- TTL, PMOS, CMOS compatible
- Suppression diodes for inductive loads
- Extended temperature range

Connection Diagram



Order Numbers

	J Package Number J16A	N Package Number N16E	M Package Number M16A
DS2003 DS9667	DS2003MJ DS2003TJ DS2003CJ DS9667MJ DS9667TJ DS9667CJ	DS2003TN DS2003CN DS9667TN DS9667CN	DS2003TM DS2003CM
DS2004	DS2004MJ DS2004TJ DS2004CJ	DS2004TN DS2004CN	DS2004TM DS2004CM

Absolute Maximum Ratings (Note 1)

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

Storage Temperature Range

Ceramic DIP	-65°C to +175°C
Molded DIP	-65°C to +150°C

Operating Temperature Range

DS2003M/DS9667M	-55°C to +125°C
DS2004M	-55°C to +125°C
DS2003T/DS9667T	-40°C to +105°C
DS2004T	-40°C to +105°C
DS2003C/DS9667C	0°C to +85°C
DS2004C	0°C to +85°C

Lead Temperature

Ceramic DIP (Soldering, 60 seconds)	300°C
Molded DIP (Soldering, 10 seconds)	265°C

Maximum Power Dissipation* at 25°C

Cavity Package	2016 mW
Molded Package	1836 mW
S.O. Package	926 mW

*Derate cavity package 16.13 mW/°C above 25°C, derate molded DIP package 14.7 mW/°C above 25°C. Derate S.O. package 7.4 mW/°C

Input Voltage	30V
Output Voltage	55V
Emitter-Base Voltage	6.0V
Continuous Collector Current	500 mA
Continuous Base Current	25 mA

Electrical Characteristics $T_A = 25^\circ\text{C}$, unless otherwise specified (Note 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{CEX}	Output Leakage Current	$T_A = 25^\circ\text{C}, V_{CE} = 50\text{V}$ (Figure 1a)			20	μA
		$T_A = 85^\circ\text{C}, V_{CE} = 50\text{V}$ (Figure 1a) for Commercial Grade			100	
		$T_A = 25^\circ\text{C}, V_{CE} = 50\text{V}, V_I = 1.0\text{V}$ (Figure 1b) DS2004			500	
$V_{CE(Sat)}$	Collector-Emitter Saturation Voltage	$I_C = 350\text{ mA}, I_B = 500\ \mu\text{A}$ (Figure 2) (Note 3)		1.25	1.6	V
		$I_C = 200\text{ mA}, I_B = 350\ \mu\text{A}$ (Figure 2)		1.1	1.3	
		$I_C = 100\text{ mA}, I_B = 250\ \mu\text{A}$ (Figure 2)		0.9	1.1	
$I_{I(ON)}$	Input Current	$V_I = 3.85\text{V}$ (Figure 3) DS2003/DS9667		0.93	1.35	mA
		$V_I = 5.0\text{V}$ (Figure 3) DS2004		0.35	0.5	
		$V_I = 12\text{V}$ (Figure 3)		1.0	1.45	
$I_{I(OFF)}$	Input Current (Note 4)	$T_A = 85^\circ\text{C}$ for Commercial $I_C = 500\ \mu\text{A}$ (Figure 4)	50	100		μA
$V_{I(ON)}$	Input Voltage (Note 5)	$V_{CE} = 2.0\text{V}, I_C = 200\text{ mA}$ (Figure 5) DS2003/DS9667			2.4	V
		$V_{CE} = 2.0\text{V}, I_C = 250\text{ mA}$ (Figure 5)			2.7	
		$V_{CE} = 2.0\text{V}, I_C = 300\text{ mA}$ (Figure 5)			3.0	
		$V_{CE} = 2.0\text{V}, I_C = 125\text{ mA}$ (Figure 5) DS2004			5.0	
		$V_{CE} = 2.0\text{V}, I_C = 200\text{ mA}$ (Figure 5)			6.0	
		$V_{CE} = 2.0\text{V}, I_C = 275\text{ mA}$ (Figure 5)			7.0	
C_i	Input Capacitance			15	30	pF
t_{PLH}	Turn-On Delay	$0.5 V_I$ to $0.5 V_O$			1.0	μs
t_{PHL}	Turn-Off Delay	$0.5 V_I$ to $0.5 V_O$			1.0	μs
I_R	Clamp Diode Leakage Current	$V_R = 50\text{V}$ (Figure 6)	$T_A = 25^\circ\text{C}$		50	μA
			$T_A = 85^\circ\text{C}$		100	μA
V_F	Clamp Diode Forward Voltage	$I_F = 350\text{ mA}$ (Figure 7)		1.7	2.0	V

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the devices should be operated at these limits. The tables of "Electrical Characteristics" provide conditions for actual device operation.

Note 2: All limits apply to the complete Darlington series except as specified for a single device type.

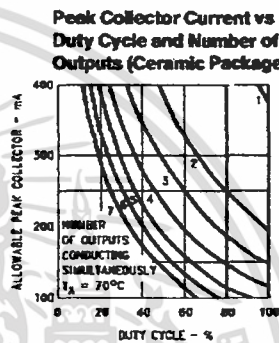
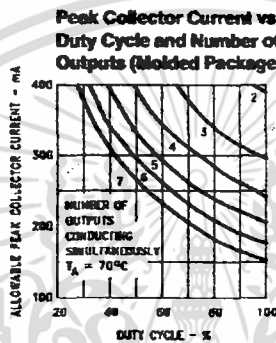
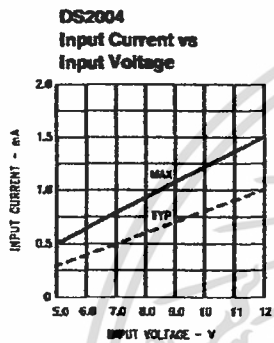
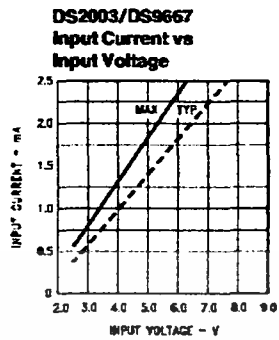
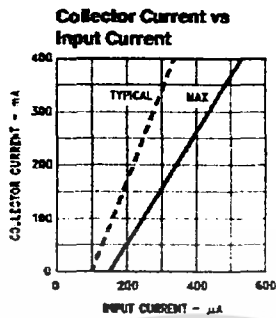
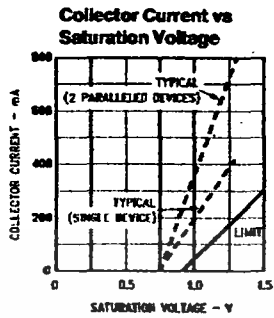
Note 3: Under normal operating conditions these units will sustain 350 mA per output with $V_{CE(Sat)} = 1.6\text{V}$ at 70°C with a pulse width of 20 ms and a duty cycle of 30%.

Note 4: The $I_{I(OFF)}$ current limit guarantees against partial turn-on of the output.

Note 5: The $V_{I(ON)}$ voltage limit guarantees a minimum output sink current per the specified test conditions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

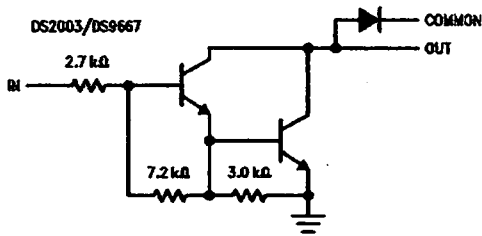
Typical Performance Characteristics



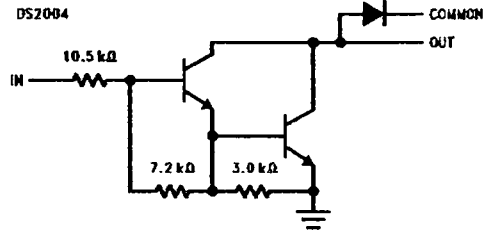
TL/F/9647-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Equivalent Circuits



TL/F/9647-3



TL/F/9647-6

Test Circuits

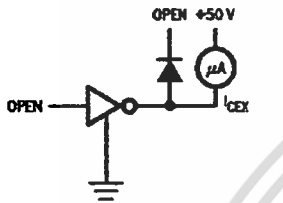


FIGURE 1a

TL/F/9647-7

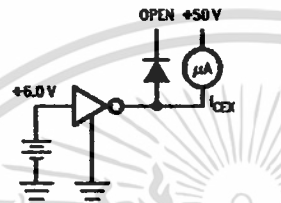


FIGURE 1b

TL/F/9647-8

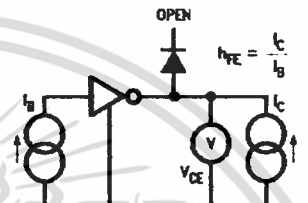


FIGURE 2

TL/F/9647-9

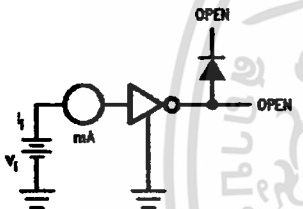


FIGURE 3

TL/F/9647-10

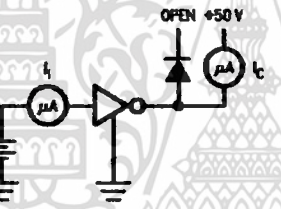


FIGURE 4

TL/F/9647-11

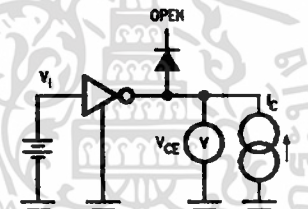


FIGURE 5

TL/F/9647-12

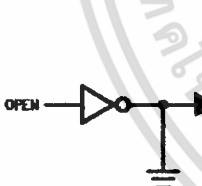


FIGURE 6

TL/F/9647-13

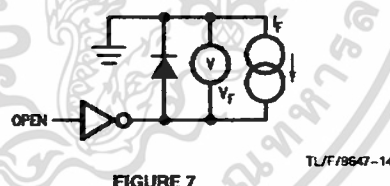


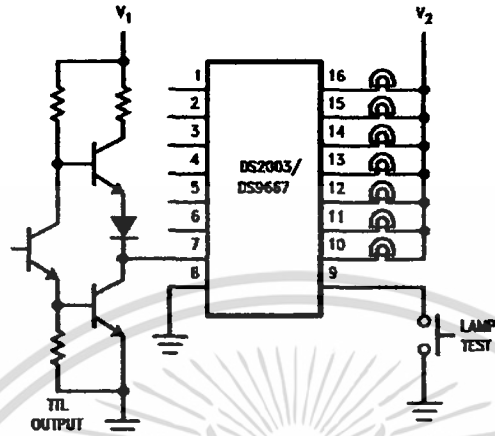
FIGURE 7

TL/F/9647-14

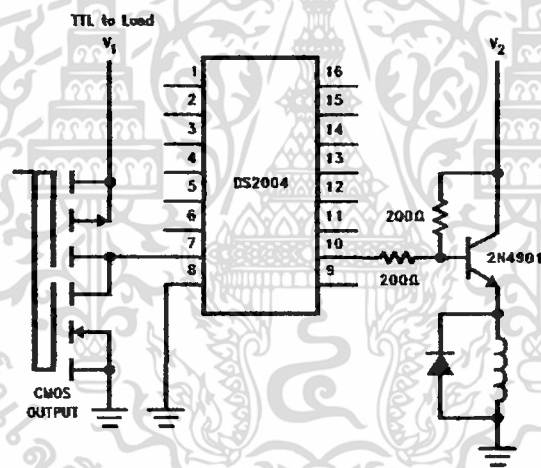
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications

Buffer for Higher Current Loads



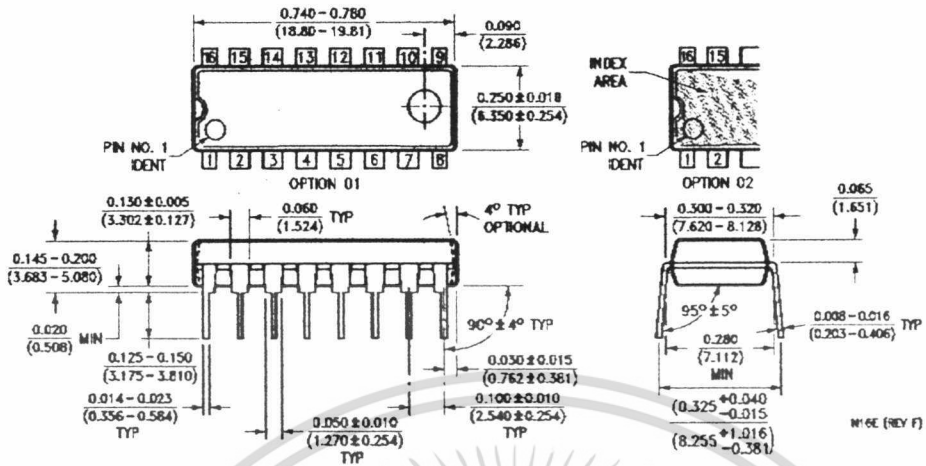
TL/F/9647-16



TL/F/9647-17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Dual-In-Line Package (M)
 Order Number DS2003CN, DS9667CN, DS2003TN, DS9667TN, DS2004CN or DS2004TN
 NS Package Number N16E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
 1111 West Bardin Road
 Arlington, TX 76017
 Tel: 1(800) 272-9959
 Fax: 1(800) 737-7018

<http://www.national.com>

National Semiconductor Europe
 Fax: +49 (0) 180-530 85 86
 Email: europe.support@nsc.com
 Deutsch Tel: +49 (0) 180-530 85 85
 English Tel: +49 (0) 180-532 78 32
 Français Tel: +49 (0) 180-532 83 56
 Italiano Tel: +49 (0) 180-534 16 86

National Semiconductor Hong Kong Ltd.
 13th Floor, Straight Block,
 Ocean Centre, 5 Canton Rd.
 Tsimshatsui, Kowloon
 Hong Kong
 Tel: (852) 2737-1600
 Fax: (852) 2738-9960

National Semiconductor Japan Ltd.
 Tel: 81-043-299-2308
 Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are needed and National reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้