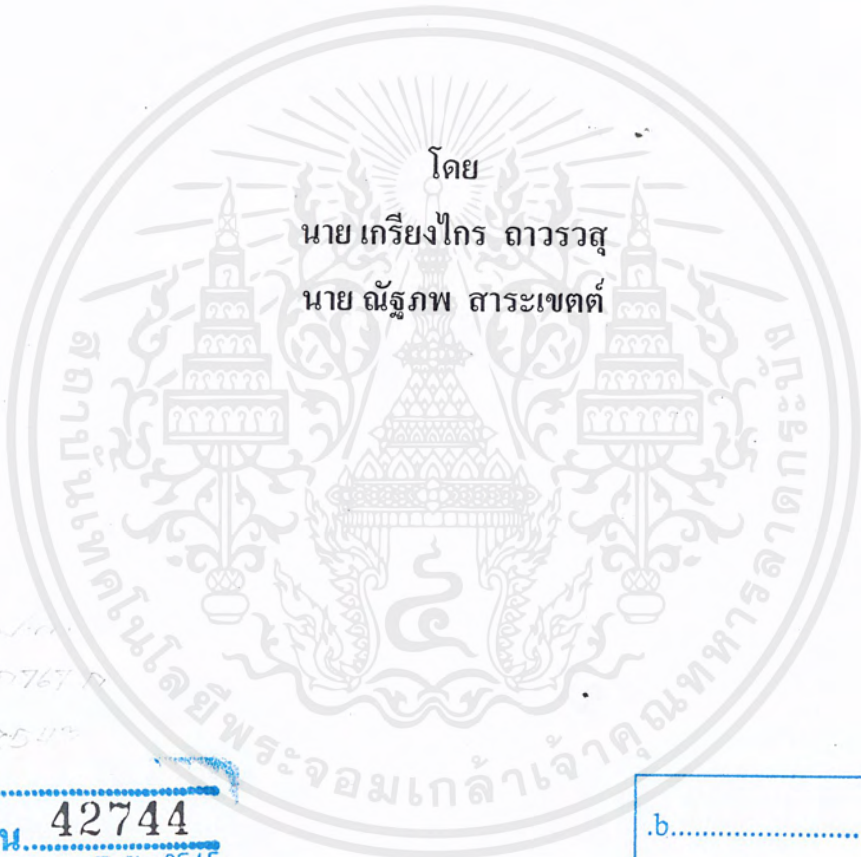


การวัดอุณหภูมิผ่านเครือข่ายอินเทอร์เน็ต

TEMPERATURE MONITORING VIA INTERNET NETWORK



โดย  
นาย เกรียงไกร ถาวรวิสุ  
นาย ธีรภัฏ สาระเขตต์

4

๗/๗  
๗๗๖๗  
๒๕๔๓

เลขหมู่.....  
เลขทะเบียน 42744  
วัน, เดือน, ปี 7 ส.ย. 2545

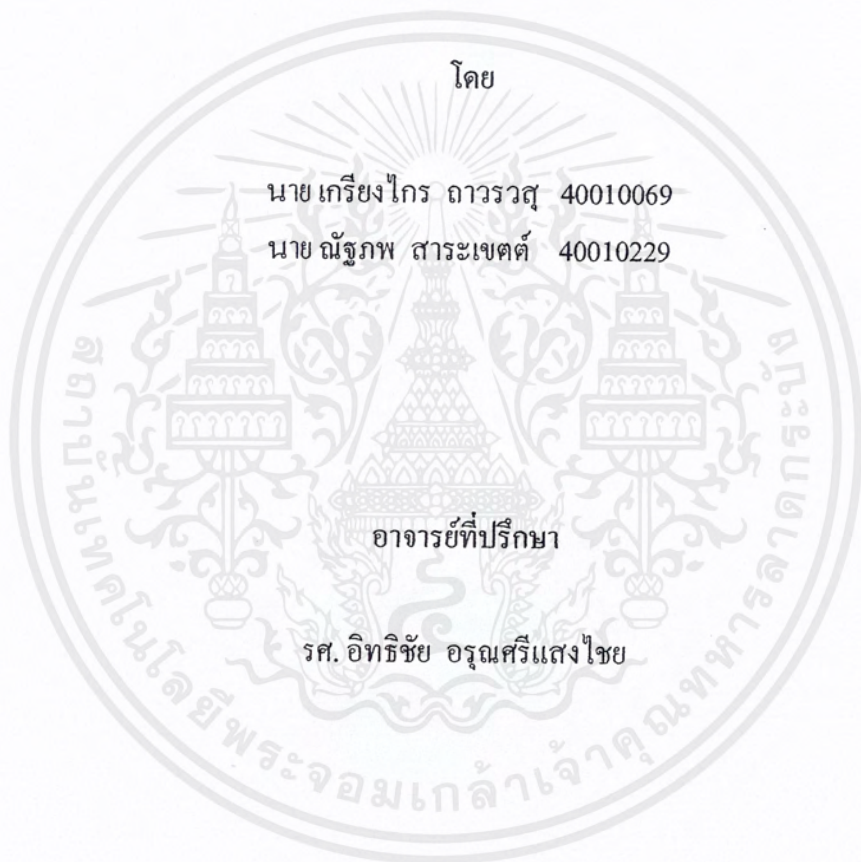
b.....  
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๒๕๔๓

การวัดอุณหภูมิผ่านเครือข่ายอินเทอร์เน็ต  
Temperature Monitoring via Internet Network



ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา อิเล็กทรอนิกส์


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวัดอุณหภูมิผ่านเครือข่ายอินเทอร์เน็ต

ผู้จัดทำ

1. นายเกรียงไกร ถาวรวิสุ 40010069

2. นายณัฐภาพ สาระเขตต์ 40010229



(รศ. อิทธิชัย อรุณศรีแสงไชย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดอุณหภูมิผ่านเครือข่ายอินเทอร์เน็ต

Temperature Monitoring via Internet Network

1. นายเกรียงไกร ถาวรวิสุ 40010069
2. นายณัฐภพ สาระเขตต์ 40010229

โครงการนี้ได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การวัดคุณหมุมิผ่านเครือข่ายอินเทอร์เน็ต

เกรียงไกร ถาวรวิสุ

ญัฐภพ สาระเขตต์

รศ. อธิธิชัย อรุณศรีแสงไชย

ปีการศึกษา 2543

### บทคัดย่อ

การวัดคุณหมุมิผ่านเครือข่ายอินเทอร์เน็ตเป็นการประยุกต์การวัดสัญญาณต่างๆเข้ากับการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต เพื่อทำให้ง่ายต่อการตรวจสอบผลของการวัดเพราะสามารถทำการตรวจสอบได้ทุกที่ที่สามารถเข้าถึงเครือข่ายอินเทอร์เน็ตได้ โดยโครงงานนี้ประกอบด้วยส่วนการวัดค่าคุณหมุมิโดยสามารถใช้งานได้ในช่วงคุณหมุมิตั้งแต่ 0 องศาเซลเซียส ถึง 100 องศาเซลเซียส, ส่วนของวงจรเชื่อมต่อที่สามารถเลือกพอร์ต,ค่าความเร็วในการติดต่อ,รูปแบบของบิตข้อมูลที่ใช้ในการสื่อสารกับคอมพิวเตอร์ได้ และส่วนของโปรแกรมที่พัฒนามาเพื่อให้ผู้ใช้งานสามารถสังเกตผลของการวัดผ่านเครือข่ายอินเทอร์เน็ตโดยอาศัยโปรโตคอล TCP/IP ในรูปแบบของผู้ให้และผู้รับบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Temperature Monitoring via Internet Network

Kriangkrai thawornwasu

Natapope Sarakhette

Assoc. Prof. Itthichai Arungsrisangchai

Academic Year 2000

### Abstract :

Temperature Monitoring via Internet Network is the system that capable to measure and observe physical quantities through internet network. In this project, the temperature measurement was the case study. The system is composed of the temperature measurement that can sense temperature from  $0^{\circ}\text{C}$  to  $100^{\circ}\text{C}$ , the interface circuit that capable to select ports, baud rates, data bit and programs are developed for user to observe the result of measurement through internet by using TCP/IP as a protocol in the manner of Server/Client.

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญภาพ	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการเบื้องต้น	3
2.1 TEMPERRATURE SENSOR	3
2.1.1 THERMISTOR	3
2.1.2 ไอซีที่ตรวจวัดอุณหภูมิ	5
2.1.2.1 ไอซี LM34CZ / LM35CZ	6
2.1.2.2 ไอซี LM335Z	7
2.1.2.3 ไอซี LM3911	8
2.1.3 RTD TRANSDUCER	8
2.1.3.1 หลักการของ RTD	9
2.1.3.2 ชนิดของ RTD	9
2.1.3.3 ความผิดพลาดอันเกิดจากกระแสเลี้ยงอาร์ทีดี	11
2.1.4 THERMOCOUPLE	11
2.2 วงจรขยายสัญญาณเทอร์โมคัปเปิ้ล โดยใช้ AD-595	15
2.2.1 คุณสมบัติของ AD-595	15
2.2.2 หลักการทำงานของ AD-595	15
2.2.3 การชดเชยที่รอยต่อเย็น	16
2.3 วงจรส่วนแสดงผลด้วย ICL 7106	16
2.4 วงจร ANALOG TO DIGITAL CONVERTERS	19
2.4.1 คุณสมบัติของ ไอซี	19
2.4.2 ส่วนการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล	20
2.4.3 ความสัมพันธ์ของอินพุตและเอาต์พุตของ ไอซี ADC	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
2.4.4 แรงดันอ้างอิงของวงจร	22
บทที่ 3 เครือข่ายอินเทอร์เน็ต	23
3.1 เครือข่ายคอมพิวเตอร์	23
3.1.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์	23
3.1.1.1 เครือข่ายย่อยส่วนของผู้ใช้	23
3.1.1.2 เครือข่ายย่อยส่วนของการสื่อสาร	25
3.2 มาตรฐานอ้างอิงในการสื่อสารข้อมูล	27
3.3 โพรโทคอล TCP/IP	33
3.3.1 สถาปัตยกรรมของ TCP/IP	34
3.3.2 IP ADDRESS	35
3.3.3 กลไกการทำงานของโปรโตคอล TCP	38
3.3.4 กลไกการทำงานของโปรโตคอล IP	40
3.3.5 การส่งข้อมูลโดยอาศัย Router และ Routing Algorithm	42
บทที่ 4 การรับส่งข้อมูลแบบอนุกรมด้วย MCS-51	47
4.1 พื้นฐานการรับส่งข้อมูล	47
4.1.1 การรับส่งข้อมูลแบบซิงโครนัส	48
4.1.2 การรับส่งข้อมูลแบบอะซิงโครนัส	48
4.2 รูปแบบของการรับส่งข้อมูลแบบอนุกรม	48
4.3 MCS-51 กับการรับส่งข้อมูลแบบอนุกรม	50
4.4 Serial Port Control Register	51
4.5 Mode Of Operation	52
4.5.1 8-Bit Shift Register ( Mode 0 )	52
4.5.2 8-Bit UART With Variable Baud Rate ( Mode 1 )	54
4.5.3 9-Bit UART With Fixed Baud Rate ( Mode 2 )	55
4.5.4 9-Bit UART With Variable Baud Rate ( Mode 3 )	55
4.6 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล	55
4.7 อัตราการส่งข้อมูลของพอร์ทอนุกรม	57
บทที่ 5 การออกแบบและการสร้าง	60
5.1 การออกแบบส่วนวงจร	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
5.1.1 วงจรขยายสัญญาณเทอร์โมคัปเปิล	60
5.1.1.1 วงจรขยายสัญญาณจากเทอร์โมคัปเปิลโดยใช้อปแอมป์	60
5.1.1.2 วงจรขยายสัญญาณเทอร์โมคัปเปิลโดยใช้ ADC-595	61
5.1.2 วงจรส่วนแสดงผลด้วย LCD	62
5.1.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลโดยใช้ ADC 0804	63
5.2 การออกแบบส่วนของโปรแกรม	64
5.2.1 โปรแกรมส่วนควบคุมการส่งสัญญาณด้วยมาตรฐาน RS232	64
5.2.2 โปรแกรมส่วนทำการติดต่อกับคอมพิวเตอร์	66
5.2.3 โปรแกรมส่วนควบคุมการเชื่อมต่อผ่านอินเทอร์เน็ต	67
5.2.3.1 ส่วนของผู้ให้บริการ	67
5.2.3.2 ส่วนของผู้ใช้บริการ	67
บทที่ 6 การทดลองและผลการทดลอง	70
6.1 ส่วนของวงจรวัดอุณหภูมิ	70
6.2 ส่วนของการแปลงสัญญาณอนาลอกไปเป็นดิจิทัล	71
6.3 ส่วนของโปรแกรม	73
บทที่ 7 สรุปและวิจารณ์ผลการทดลอง	79
7.1 สรุปผลการทดลอง	79
7.1.1 ส่วนของวงจรวัดอุณหภูมิ	79
7.1.2 ส่วนของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล	79
7.1.3 ส่วนของโปรแกรม	79
7.2 คุณสมบัติของโครงการและโปรแกรม Server/Client	80
7.3 ข้อจำกัดของโครงการและแนวทางพัฒนาแก้ไข	80
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

	หน้า
รูปที่ 1.1 ส่วนประกอบของระบบ Temperature Monitoring via Network	2
รูปที่ 2.1 คุณลักษณะของค่าความต้านทานกับอุณหภูมิ	4
รูปที่ 2.2 ลักษณะของเทอร์มิสเตอร์ในแบบต่าง ๆ	4
รูปที่ 2.3 วงจรใช้งานของเทอร์มิสเตอร์	4
รูปที่ 2.4 รูปร่างและวงจรใช้งานพื้นฐานของไอซี LM34CZ และ LM35CZ	6
รูปที่ 2.5 วงจรตรวจวัดอุณหภูมิของ LM34CZ และ LM35CZ	6
รูปที่ 2.6 วงจรตรวจวัดอุณหภูมิใช้แหล่งจ่ายไฟตัวเดียว	6
รูปที่ 2.7 การต่อไอซีเข้ากับ โหลดของตัวเก็บประจุสูง	7
รูปที่ 2.8 รูปร่างและการใช้งานพื้นฐานของไอซี LM335Z	7
รูปที่ 2.9 รูปร่างและบล็อกไดอะแกรมของไอซี LM3911	8
รูปที่ 2.10 อัตราส่วนของความต้านทานของโลหะที่ใช้อาร์ทีดีเทียบกับค่าความต้านทาน ของแต่ละชนิดที่ 0 องศาเซลเซียส	10
รูปที่ 2.11 รอยต่อของเทอร์โมคัปเปิ้ล	11
รูปที่ 2.12 การใช้วงจรบริดจ์เพื่อชดเชยค่าผิดพลาดของเทอร์โมคัปเปิ้ล	14
รูปที่ 2.13 โครงสร้างภายในของ AD-595	15
รูปที่ 2.14 วงจรส่วนอนาล็อก ภายใน ไอซี ICL 7106	17
รูปที่ 2.15 วงจรส่วนดิจิตอลภายใน ICL 7106	18
รูปที่ 2.16 เวลาการทำงานของ ADC 0804	20
รูปที่ 2.17 การทำงานแบบ Successive Approximation ( SAR )	21
รูปที่ 3.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์	23
รูปที่ 3.2 เครือข่ายในส่วนของเครือข่ายย่อยผู้ใช้บริการ	24
รูปที่ 3.3 เครือข่ายในส่วนของเครือข่ายย่อยของการสื่อสาร	25
รูปที่ 3.4 รูปแบบต่าง ๆ ของเครือข่ายแบบจุดต่อจุด	26
รูปที่ 3.5 รูปแบบต่าง ๆ ของเครือข่ายแบบแพร่กระจาย	27
รูปที่ 3.6 มาตรฐานการติดต่อแบบ OSI 7 Layer	28
รูปที่ 3.7 การแบ่งกลุ่มของ OSI 7 Layer Model	29
รูปที่ 3.8 การรับส่งข้อมูลแต่ละชั้นของ TCP/IP ใน OSI 7 Layer Model ซึ่งทั้งข้อมูลและ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
TCP Header จะถูกผนึกรวมกันเป็นข้อมูลที่เรียกว่า IP Datagram	33
รูปที่ 3.9 ส่วนประกอบของ TCP/IP Stack	34
รูปที่ 3.10 เครื่องข่าย 3 เครื่องข่ายที่แต่ละเครื่องข่ายมี IP Address ของตัวเอง	35
รูปที่ 3.11 ตัวอย่างการทำ Encapsulation ของข้อมูล FTP เทียบกับ TCP/IP Layer	36
รูปที่ 3.12 รูปแบบของ IP Datagram ที่ประกอบด้วย Header และ Payload	37
รูปที่ 3.13 รูปแบบของ TCP Packet จะเห็นว่ามี Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล	39
รูปที่ 3.14 รูปแบบของ UDP Packet จะเห็นว่ามีส่วน Header ที่มีขนาดเล็กมาก จึงใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP	39
รูปที่ 3.15 แอปพลิเคชันต่าง ๆ ใน Transport Layer ที่เรียกใช้โปรโตคอล TCP และ UDP	40
รูปที่ 3.16 โปรโตคอล TCP และ UDP อาศัยโปรโตคอล IP ในชั้น Internetwork Layer ในการส่งผ่านข้อมูล และแสดงความสัมพันธ์ของโปรโตคอล IP, ICMP และ ARP	40
รูปที่ 3.17 การส่งผ่านข้อมูลผ่านเครื่องข่าย	42
รูปที่ 3.18 หน้าที่ของ Router ในการพิจารณาเส้นทางและการแลกเปลี่ยนข้อมูล Routing Table ระหว่าง Router ด้วยกัน	43
รูปที่ 3.19 Distance-Vector Routing Protocol	44
รูปที่ 3.20 Intradomain / Interdomain ในการติดต่อระหว่างอุปกรณ์ Router	45
รูปที่ 3.21 แผนผังการเชื่อมต่อของบริษัทผู้ให้บริการ ISP กับเครื่องข่ายต่างประเทศโดยการใช้ Router หลักที่มี AS Number เป็นหมายเลขประจำตัวในการติดต่อ	46
รูปที่ 4.1 การรับส่งข้อมูลแบบขนาน	47
รูปที่ 4.2 การส่งข้อมูลแบบซิงโครนัส	48
รูปที่ 4.3 การส่งข้อมูลแบบอนุกรม	49
รูปที่ 4.4 บิตต่าง ๆ ของข้อมูลที่ส่งแบบอนุกรม	49
รูปที่ 4.5 การรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัสภายใน	50
รูปที่ 4.6 ไคอะแกรมเวลาการส่งข้อมูล	53
รูปที่ 4.7 ไคอะแกรมเวลาการรับข้อมูล	53
รูปที่ 4.8 การส่งข้อมูลออกโดยใช้พอร์ตรีจิสเตอร์	54
รูปที่ 4.9 การรับส่งข้อมูลในโหมด 1	54
รูปที่ 4.10 การกำหนด Baud Rate ในโหมดต่าง ๆ	58

	หน้า
รูปที่ 5.1 วงจรขยายที่สร้างด้วยออปแอมป์	60
รูปที่ 5.2 วงจรขยายสัญญาณเทอร์โมคัปเปิ้ล โดยใช้ AD-595	61
รูปที่ 5.3 วงจรส่วนที่ใช้แสดงผล	62
รูปที่ 5.4 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	63
รูปที่ 5.5 รูปแบบมาตรฐานที่ใช้ในการติดต่อผ่านพอร์ตอนุกรมในโรงงาน	64
รูปที่ 5.6 แผนผังการทำงานของโปรแกรม MCS-51	65
รูปที่ 5.7 แผนผังการทำงานของส่วนรับข้อมูล	66
รูปที่ 5.8 แผนผังการทำงานของส่วนผู้ให้บริการ ( Server )	68
รูปที่ 5.9 แผนผังการทำงานของส่วนผู้ใช้บริการ ( Client )	69
รูปที่ 6.1 กราฟเปรียบเทียบค่าที่ได้จากวงจรวัดกับค่าที่ได้จากเทอร์โมมิเตอร์	71
รูปที่ 6.2 กราฟเปรียบเทียบค่าที่วัดได้กับค่าอินพุทที่ป้อนให้วงจรแปลงสัญญาณ	73
รูปที่ 6.3 หน้าต่างของโปรแกรม Telemetry Server	74
รูปที่ 6.4 หน้าต่างของโปรแกรม Telemetry Client	75
รูปที่ 6.5 การทดสอบการใช้งานจริง	77
รูปที่ 6.6 ผลการทดลองของโปรแกรม Telemetry Server	77
รูปที่ 6.7 ผลการทดลองของโปรแกรม Telemetry Client	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 การเลือกใช้งานไอซีตรวจวัดอุณหภูมิ	5
ตารางที่ 2.2 ค่าแรงดันต่ออุณหภูมิที่เกิดขึ้นระหว่างรอยต่อของโลหะต่าง ๆ เทียบกับ Pt ที่ 0 °C	12
ตารางที่ 2.3 ค่าแรงดันที่เกิดขึ้นจากคู่โลหะต่าง ๆ เทียบที่อุณหภูมิ 0 °C	13
ตารางที่ 4.1 บิตต่าง ๆ ของรีจิสเตอร์ SCON	51
ตารางที่ 4.2 โหมดต่าง ๆ ของการรับส่งแบบอนุกรม	52
ตารางที่ 4.3 ความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่าง ๆ	59
ตารางที่ 6.1 ค่าอุณหภูมิของเทอร์โมมิเตอร์เทียบกับวงจรวัด	70
ตารางที่ 6.2 ผลการทดลองของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	72
ตารางที่ 6.3 ค่าที่ได้จากการเชื่อมต่อพอร์ตอนุกรม	74
ตารางที่ 6.4 ค่าที่ได้จากการติดต่อส่งข้อมูลผ่านเครือข่าย	76

## บทที่ 1

### บทนำ

โครงการเรื่อง การวัดอุณหภูมิผ่านเครือข่ายอินเทอร์เน็ต ( Temperature Monitoring via Internet Network ) เป็นการศึกษาถึงเรื่องของการวัดสัญญาณต่าง ๆ แล้วทำการส่งข้อมูลผ่านเข้าไปในระบบอินเทอร์เน็ต เพื่อที่จะสามารถสังเกตและควบคุมผ่านเครือข่ายอินเทอร์เน็ตได้ โดยจะเห็นได้ว่าในปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทที่สำคัญในโลกของการสื่อสาร มีโครงข่ายโยงใยไปทั่วทุกมุมโลก ผู้ใช้บริการสามารถเข้าถึงได้อย่างสะดวกและรวดเร็ว

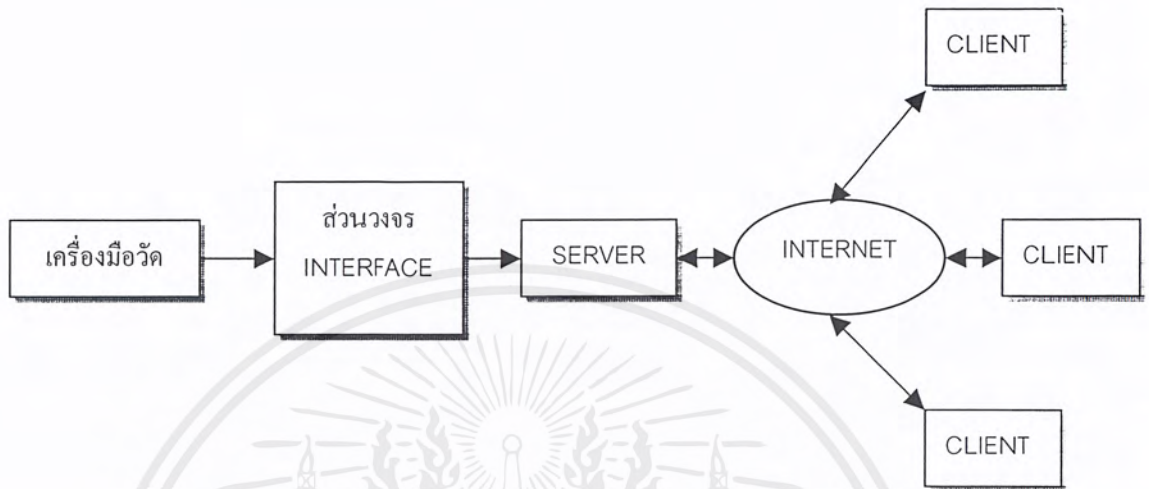
#### วัตถุประสงค์ของการทำโครงการ

1. เพื่อทำการสร้างระบบที่สามารถสามารถสังเกตการเปลี่ยนแปลงของข้อมูลที่วัดจากเครื่องมือวัด โดยไม่จำเป็นต้องอยู่บริเวณพื้นที่ปฏิบัติงาน
2. เพื่อทำการประยุกต์ใช้งานเครือข่ายอินเทอร์เน็ตกับระบบการวัดสัญญาณต่างๆ
3. เพื่อสร้างระบบที่ง่ายต่อการรวบรวมผลการวัดต่างๆเพื่อใช้ในการศึกษาและพัฒนา ระบบการวัด
4. เพื่อสร้างระบบการวัดที่ช่วยให้ผู้ปฏิบัติงานสามารถหลีกเลี่ยงอันตรายและลดความเสี่ยงในการปฏิบัติงานจากสภาพแวดล้อมที่เป็นอันตราย เช่น บริเวณที่มีอุณหภูมิสูงหรือต่ำมาก , บริเวณที่มีสารพิษหรือฝุ่นมาก , บริเวณที่มีความดันสูง เป็นต้น
5. เพื่อสร้างระบบที่ง่ายต่อการใช้งานทำให้ผู้ปฏิบัติงานไม่จำเป็นต้องมีความรู้เกี่ยวกับเครื่องมือวัดมากก็สามารถทำการวัดได้
6. เพื่อนำผลที่วัดได้ใช้ในการควบคุมและพัฒนาระบบต่อไป

#### ส่วนประกอบของระบบ Telemetry

โดยในโครงการนี้จะทำการศึกษาเกี่ยวกับส่วนต่าง ๆ ของระบบ Telemetry ไม่ว่าจะเป็นการตรวจวัดสัญญาณ ซึ่งในที่นี้เราจะใช้การตรวจสอบวัดอุณหภูมิเป็นกรณีศึกษา การเชื่อมต่อเครื่องมือวัดเข้ากับเครื่องคอมพิวเตอร์ส่วนบุคคล ( PC ) และการติดต่อผ่านเครือข่ายอินเทอร์เน็ต ในลักษณะของผู้ให้บริการและผู้รับบริการ ( Server / Client ) โดยส่วนประกอบของระบบแสดงไว้ในรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 ส่วนประกอบของระบบ Temperature Monitoring via Internet Network

ส่วนที่ 1 เป็นส่วนของเครื่องมือวัดโดยในโรงงานนี้ใช้อุณหภูมิเป็นกรณีศึกษา ส่วนที่เป็นตัวตรวจจับอุณหภูมิจึงเลือกใช้ Thermocouple เป็นตัวตรวจสอบ

ส่วนที่ 2 เป็นส่วนของวงจรที่ใช้เชื่อมต่อเครื่องมือวัดเข้าสู่ระบบการวัดทางไกล ซึ่งจะประกอบไปด้วย ส่วนการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัล ( ADC 0804 ) และส่วนที่ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์ด้วยมาตรฐาน RS 232 โดยการทำในส่วนนี้จะควบคุมด้วยไมโครคอนโทรลเลอร์ ( MCS-51 )

ส่วนที่ 3 เป็นส่วนการติดต่อเพื่อทำการสังเกตผลการวัดผ่านเครือข่ายอินเทอร์เน็ต ในส่วนนี้จะประกอบไปด้วยส่วนผู้ให้บริการ ( Server ) และส่วนของผู้ใช้บริการ ( Client ) ซึ่งทั้งสองส่วนได้ทำการพัฒนาขึ้นด้วยภาษาเดลไฟล์ ( Delphi ) มีรายละเอียดดังต่อไปนี้

- ส่วนของผู้ให้บริการ ( Server ) จะมีหน้าที่รองรับการติดต่อจากผู้ใช้บริการ ( Client ) และทำการติดต่อกับพอร์ตอนุกรมเพื่อทำการรับข้อมูลจากพอร์ตอนุกรมมาประมวลผลก่อนส่งให้ผู้ใช้บริการ ( Client ) รวมทั้งยังสามารถแสดงผลเองได้ด้วย
- ส่วนของผู้ใช้บริการ ( Client ) มีหน้าที่ร้องขอและขงเลิกการติดต่อกับผู้ให้บริการ และรับค่าเมื่อมีการติดต่อแล้วนำมาประมวลผลก่อนที่จะแสดงผลด้วยกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการเบื้องต้น

#### 2.1 TEMPERATURE SENSOR

ตัวตรวจจับอุณหภูมิทำงานโดยใช้หลักการการเปลี่ยนค่าของคุณสมบัติต่างๆในตัวเองไม่ว่าจะเป็นในรูปของความต้านทาน แรงดัน หรือกระแส ตามค่าอุณหภูมิที่เปลี่ยนแปลงไป ซึ่งมีด้วยกันหลายชนิดดังต่อไปนี้

##### 2.1.1 THERMISTOR

Thermistor หรือ Thermal Resistor เป็นสิ่งประดิษฐ์สารกึ่งตัวนำที่มีค่าสัมประสิทธิ์ของความต้านทานต่ออุณหภูมิเป็นลบ นั่นคือถ้าอุณหภูมิสูงขึ้น ค่าความต้านทานของ Thermistor จะลดลง ใน Thermistor บางแบบนั้นค่าความต้านทานของมันที่อุณหภูมิห้องอาจจะลดลงถึง 6 % ต่ออุณหภูมิแต่ละองศาเซลเซียสที่เพิ่มขึ้น ซึ่งจะเห็นได้ว่า Thermistor มีความไวต่อการเปลี่ยนแปลงอุณหภูมิดีมาก ดังนั้นจึงนิยมใช้ Thermistor ในการวัดและควบคุมตลอดจนการชดเชยอุณหภูมิในช่วงอุณหภูมิต่ำ ๆ ตั้งแต่ 100 ถึง 300 องศาเซลเซียส ซึ่งสามารถแสดงสมการความสัมพันธ์ของความสัมพันธ์ระหว่างค่าความต้านทานกับอุณหภูมิได้ดังนี้

$$R = R_0 e^{\beta/T} \quad (2.1)$$

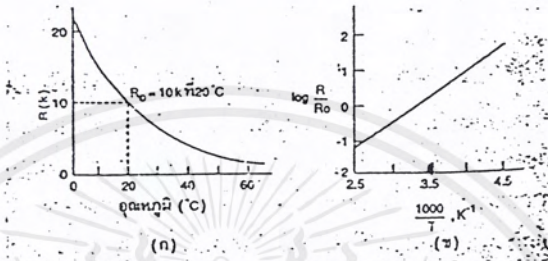
โดยที่  $R_0$  คือค่าคงที่ของเทอร์มิสเตอร์แต่ละตัวที่ได้จากผู้ผลิต

$T$  คือ อุณหภูมิสัมพัทธ์ หน่วยเป็นองศาเคลวิน (K)

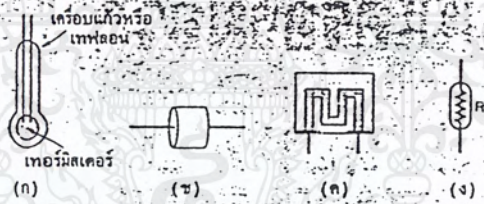
โลหะออกไซด์ที่ใช้ทำเป็น Thermistor นั้นได้แก่ Manganese, Nickel, Cobalt, Uranium, ทองแดงและเหล็ก เป็นต้น ค่าความต้านทานของ Thermistor จะอยู่ในช่วง 0.5 โอห์ม ถึง 75 เมกะโอห์ม ทั้งนี้จะขึ้นอยู่กับขนาดและรูปร่างของมัน Thermistor ที่มีขนาดเล็กที่สุดจะเป็นแบบลูกบิดหรือลูกประคำ ที่มีเส้นผ่าศูนย์กลางตั้งแต่ 0.15 มม. ถึง 1.25 มม. Thermistor แบบนี้จะติดตั้งอยู่ที่ปลายของแท่งแก้วแข็งเพื่อสะดวกแก่การใช้งาน ส่วน Thermistor แบบอื่น ๆ นั้นได้แก่ Thermistor แบบกลม (disk) และแบบวงแหวน (washer) ซึ่งมีขนาดเส้นผ่าศูนย์กลางตั้งแต่ 2.5 มม. ถึง 25 มม.

Thermistor มีคุณลักษณะที่สำคัญประการ ซึ่งทำให้ Thermistor สามารถใช้ได้อย่างกว้างขวางในการวัดและควบคุมอุณหภูมิ คุณลักษณะทั้งสามประการได้แก่ คุณลักษณะของค่าความเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นเป็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

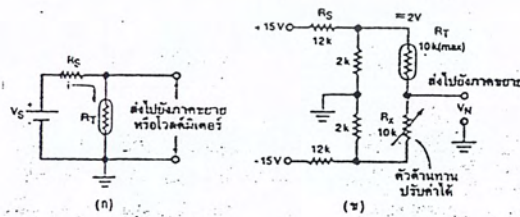
ต้านทานกับอุณหภูมิ , คุณลักษณะของแรงดันกับกระแสไฟฟ้า และ คุณลักษณะของกระแสกับเวลา ซึ่งสามารถแสดงได้ดังกราฟแสดงความสัมพันธ์ระหว่างค่าความต้านทานกับอุณหภูมิดังต่อไปนี้



รูปที่ 2.1 คุณลักษณะของค่าความต้านทานกับอุณหภูมิ



รูปที่ 2.2 ลักษณะของเทอร์มิสเตอร์ในแบบต่างๆ



รูปที่ 2.3 วงจรใช้งานของเทอร์มิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thermistor ยังสามารถประยุกต์ไปใช้ในการวัดหรือวิเคราะห์คาร์บอนไดออกไซด์ได้โดยใช้ Thermistor สองตัวต่อร่วมกัน หรือใช้ในงานวัดและควบคุมอุณหภูมิต่าง ๆ ได้อีกมาก

### 2.1.2 ไอซีตรวจวัดอุณหภูมิ

ไอซีตรวจวัดอุณหภูมิเป็นอุปกรณ์ที่เปลี่ยนค่าอุณหภูมิเป็นแรงดันไฟฟ้าได้โดยตรง บริษัท เนชั่นเนลเซมิคอนดักเตอร์เป็นผู้ผลิตอุปกรณ์เหล่านี้สู่ตลาด โดยมีตัวอย่างดังตารางที่ 2.1

ตาราง 2.1 ตารางเลือกใช้งาน ไอซีตรวจวัดอุณหภูมิ

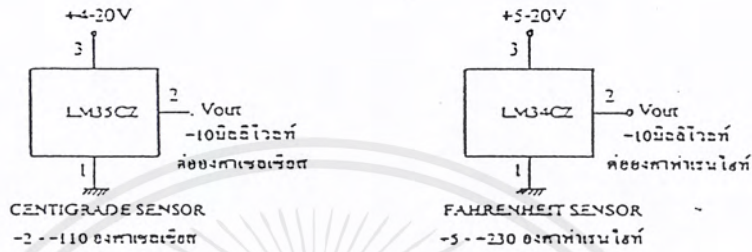
IC NUMBER	LM34CZ	LM35CZ	LM335Z	LM3911N
TEMP. RANGE	-40°F to +230°F	-40°C to +110°C	-40°C to +100°C	-25°C to +85°C
SPOT ACCURACY (TYP)	±0.8°F at +77°F	±0.4°C at +25°C	±2°C at +25°C	NA
RANGE ACCURACY (TYP)	±1.8°F	±0.8°C	±4.0°C	±10°C
OUTPUT SCALE	10 mV/°C	10 mV/°C	10 mV/K	10 mV/K
SUPPLY VOLTAGE	5V – 30V	4V – 30V	NA	8V – 35V
QUIESCENT CURRENT	70 uA	60 uA	0.4 mA – 5 mA	1 mA

จากรูปจะเป็นไอซีที่ใช้กันในงานอุตสาหกรรมและมีราคาถูก โดยที่ไอซีเบอร์ LM34CZ และ LM35CZ จะกินกำลังงานต่ำมาก ๆ และมีความเที่ยงตรงสูงให้อาท์พุทเป็นสัดส่วนกับองศาฟาเรนไฮน์และองศาเซลเซียสตามลำดับ ส่วน LM335Z เป็นไอซีที่สามารถปรับค่าเอาท์พุทได้ และเอาท์พุท จะเป็นสัดส่วนกับองศาเคลวินและไอซี LM3911N จะรวมวงจรจ่ายแรงดันอ้างอิงตัว

ตรวจสอบอุณหภูมิหรือนำไปใช้เป็นวัดอุณหภูมิที่มีความเที่ยงตรงสูงก็ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

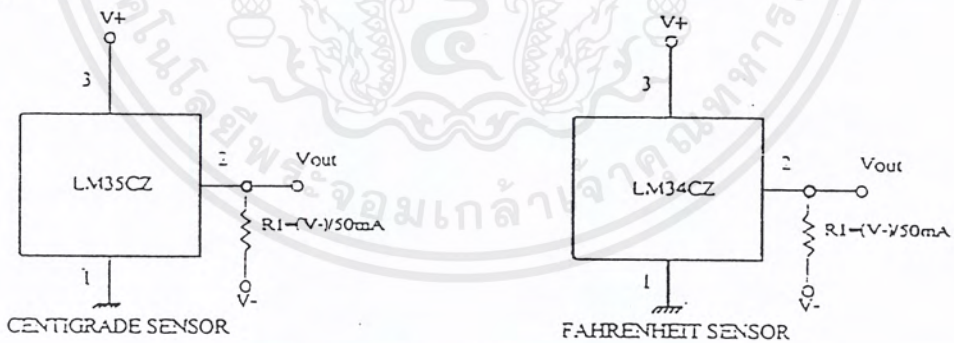
### 2.1.2.1 ไอซี LM34CZ / LM35CZ

ทั้งไอซี LM34CZ และ LM35CZ จะมีตัวถังแบบ TO-92 ซึ่งมี 3 ขาและได้รับการออกแบบให้กินกระแสขั้วไม่กี่ปีกี่สิบไมโครแอมป์ซึ่งจะช่วยลดผลกระทบจากอุณหภูมิภายในตัวไอซีด้วย

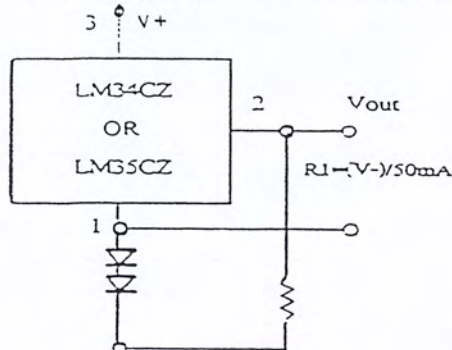


รูปที่ 2.4 รูปร่างและวงจรใช้งานพื้นฐานของไอซี LM34CZ และ LM35CZ

จากรูปแสดงให้เห็นถึงรูปร่างและวงจรใช้งานของไอซีสองตัวนี้ ถ้าสังเกตจะพบว่าวงจรพื้นฐานเหล่านี้จะอ่านค่าอุณหภูมิค่าสุดที่ +5 และ +2 องศาฟาเรนไฮน์แต่ถ้าต้องการให้ไอซีสองตัวนี้วัดค่าได้มากกว่านี้จะต้องต่อวงจรโดยใช้แหล่งจ่ายไฟสองชุดดังรูปที่ 2.5 ซึ่งจะทำให้วัดค่าได้ถึง -40 องศาฟาเรนไฮน์ หรือ -40 องศาเซลเซียส หรืออาจจะใช้แหล่งจ่ายไฟตัวเดียวแต่ทำการสร้างแรงดันลบ ดังรูปที่ 2.6



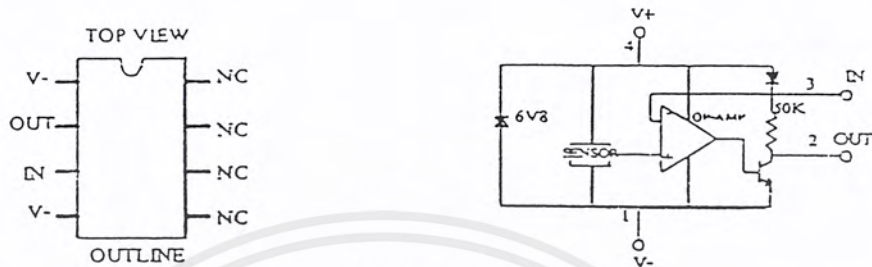
รูปที่ 2.5 วงจรตรวจวัดอุณหภูมิของ LM34CZ และ LM35CZ



เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 2.6 วงจรตรวจวัดอุณหภูมิใช้แหล่งจ่ายไฟตัวเดียวว่าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 2.1.2.3 ไอซี LM3911



รูปที่ 2.9 รูปร่างและบล็อกไดอะแกรมของไอซี LM3911

ในรูปที่ 2.9 แสดงให้เห็นถึงรูปร่างและบล็อกไดอะแกรมของไอซีควบคุมอุณหภูมิ LM3911 ซึ่งภายในประกอบด้วยซีเนอร์ไดโอดจ่ายแรงดันอ้างอิง ตัวตรวจวัดอุณหภูมิค่าผิดพลาดต่ำที่แปรตามองศาเคลวินและออปแอมป์

### 2.1.3 RTD TRANSDUCER

RTD (Resistance Temperature Detector) เป็น Temperature Transducer ชนิดหนึ่งที่สามารถเปลี่ยนแปลงคุณสมบัติทางไฟฟ้าตามอุณหภูมิ คุณสมบัติที่เปลี่ยนแปลง ได้แก่ ค่าความนำหรือค่าความต้านทานนั่นเอง การวัดอุณหภูมิด้วยการวัดค่าความต้านทานของ RTD บางครั้งจึงเรียกว่า Resistance Thermometer RTD ส่วนใหญ่เป็นโลหะ Platinum (P) เพราะให้ความถูกต้องสูงและมีคุณสมบัติเกือบเป็นเส้นตรง PRTD ที่ใช้ในปัจจุบันมี Standard การวัดในย่าน 220 ถึง 850 องศาเซลเซียสโดยที่ PRTD สามารถทนอยู่ในสภาพอุณหภูมิคงที่ได้โดยให้ความถูกต้องแม่นยำตลอดย่านการวัดนี้ โลหะทุกชนิดจะมีความต้านทานสูงขึ้นเมื่ออุณหภูมิสูงขึ้นหรือ เรียกว่ามีค่าสัมประสิทธิ์อุณหภูมิเป็นบวก ( PTC : Positive Temperature Coefficient ) ซึ่งเป็นคุณสมบัติของ RTD และคุณสมบัติที่สำคัญอีกข้อหนึ่งของ RTD คือความสามารถในการคงสภาพเดิมหลังจากที่อยู่ภายใต้การเปลี่ยนแปลงของอุณหภูมิมาก ๆ อันหมายถึง การมีเสถียรภาพในการใช้งานทำให้การวัดอุณหภูมิได้ค่าที่ถูกต้องเสมอ การเปลี่ยนแปลงค่าความต้านทานของ RTD จะคิดเป็นอัตราส่วนกับค่าความต้านทานที่อุณหภูมิ ดังนั้นค่า Error จากการวัดจะมีค่าน้อย ถ้าค่าความต้านทานที่อุณหภูมิปกติของ RTD มีค่ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3.1 หลักการของ RTD

ความต้านทานไฟฟ้าในเส้นลวดโลหะจะเปลี่ยนค่าไปตามสมการที่ 2.2

$$R_t = R_o(1 + \alpha T) \quad \text{หรือ} \quad dR_t/dT = \alpha R_o \quad (2.2)$$

เมื่อ  $R_t$  คือค่าความต้านทานของลวดโลหะที่อุณหภูมิ  $t$  องศาเซลเซียส

$R_o$  คือค่าความต้านทานของลวดโลหะที่อุณหภูมิ 0 องศาเซลเซียส

$\alpha$  คือสัมประสิทธิ์การเปลี่ยนแปลงค่าความต้านทานไฟฟ้าต่ออุณหภูมิ 1 องศาเซลเซียส

( $\Omega/\Omega/^\circ\text{C}$ ) ( Temperature Coefficient Of Resistance ) ซึ่งค่า  $\alpha$  มีค่าเปลี่ยนไปตามชนิดของโลหะเช่นพลาตินัม 0.00392 ( $\Omega/\Omega/^\circ\text{C}$ ) จากย่านอุณหภูมิ 0  $^\circ\text{C}$  ในงานปฏิบัติค่า  $\alpha$  ของการเปลี่ยนแปลงอุณหภูมิในแต่ละช่วงจะไม่แปรผันเป็นเส้นตรง ( Nonlinearity ) ในงานปฏิบัติการมาตรฐานที่ต้องการค่าแน่นอนสามารถทำได้โดยใช้สมการที่ 2.3

$$R_t = R_o(1 + \alpha T + \beta T^2 + \gamma T^4) \quad (2.3)$$

ค่า  $\alpha, \beta$  และ  $\gamma$  ได้จากการทดลอง ( Empirical Quantity ) ซึ่งทางบริษัทผู้ผลิตจะกำหนดมา เช่น พลาตินัม

$$\alpha = 3.985 \times 10^{-3}$$

$$\beta = -5.856 \times 10^{-7}$$

$$\gamma = 4.330 \times 10^{-10}$$

สูตรนี้สามารถใช้ได้ตลอดทั้งย่านการใช้งานที่ต้องการความถูกต้องแม่นยำสูง ( Accurary ) แต่โดยทั่ว ๆ ไปการคำนวณจะใช้สูตร  $R_t = R_o(1 + \alpha T)$  โดยเลือกใช้ค่า  $\alpha$  ตามตาราง

### 2.1.3.2 ชนิดของ RTD

ในช่วง 90 ปีที่ผ่านมา สถาบันและสมาคมหลายแห่งที่เกี่ยวข้องกับการผลิตอาร์ทีดีได้กำหนดมาตรฐานต่างกันไป สมาคมเหล่านี้ได้แก่สมาคมผู้ผลิตเครื่องมือทางวิทยาศาสตร์ ( SAMA ) แห่งสหรัฐ สถาบันมาตรฐานของอังกฤษ ( British Standard Insitte ) สถาบันมาตรฐานแห่งเยอรมัน ( Detsches Institute fur Normung e . V . 's ) แต่ต่อมากลุ่มกำหนดมาตรฐาน ICE ( IBS Standard Groups ) ได้กำหนด Calibration Curve ของอาร์ทีดีแบบพลาตินัมแบบใหม่และเป็นที่ยอมรับของสมาคมทั้งสามดังกล่าวช่วงอุณหภูมิต่ำกว่า 400  $^\circ\text{C}$  ถึง 600  $^\circ\text{C}$  ค่าความผิดพลาดไม่เกิน  $\pm 0.5$   $^\circ\text{C}$

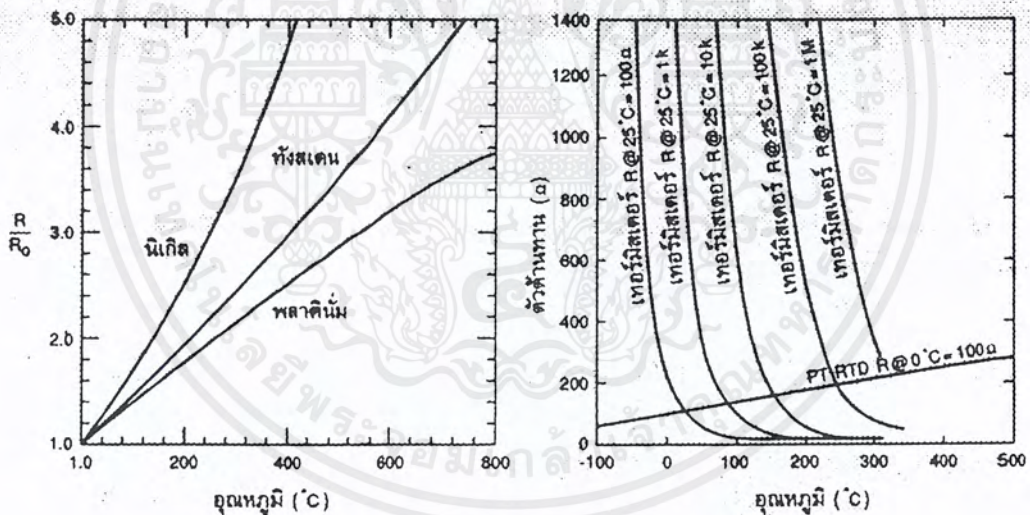
จากรูปที่ 2.10 กราฟแสดงการเปลี่ยนแปลงค่าของ RTD ที่ทำจากโลหะต่าง ๆ แกนในแนวตั้งจะบอกความต้านทานที่อุณหภูมิ  $T(R_t)$  จะมีค่าเป็นกี่เท่าของความต้านทานที่ 0  $^\circ\text{C}$  ซึ่งอาจเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงได้ถึง 5 เท่า หรือมากกว่าอาร์ทีดีแบบพลาตินัมเป็นแบบที่ใช้ในห้องปฏิบัติการทั่วไปหรือเรียกว่าแบบ Standard Platinum Resistance Thermometer (SPRTs) ในช่วงอุณหภูมิสูงถึง 600 °C ควรใช้อาร์ทีดีแบบพลาตินัมแบบ 10 โอห์ม เพราะจะให้เสถียรภาพดีกว่า 100 โอห์ม

นิเกิลอาร์ทีดี ไม่ใช่แบบมาตรฐานเหมือนพลาตินัม เพราะต่างบริษัทก็กำหนดค่าความต้านทานที่ 0 °C ต่างกันออกไป ทำให้การเลือกใช้อุปกรณ์ประกอบอื่น ๆ เช่น ทรานสมิตเตอร์, เครื่องควบคุม, เครื่องบันทึกเฉพาะของบริษัทนั้น ๆ บางบริษัทกำหนดค่าความต้านทาน 135.116 โอห์มที่ 0 °C นิเกิลอาร์ทีดีวัดอุณหภูมิได้ไม่สูงเท่ากับพลาตินัม ย่านการใช้งานอยู่ในช่วง 95 °C ถึง 360 °C ค่าความผิดพลาดต่ำกว่าแบบพลาตินัมแต่ให้ลิเนียร์ลิตีดีกว่าแบบพลาตินัม

อาร์ทีดี แบบทองแดงเป็นแบบที่ให้ลิเนียร์ลิตีในการวัดดีที่สุด แต่ย่านการใช้งานแคบ ประมาณ 200 °C ถึง 150 °C และมีค่าความต้านทานจำเพาะต่ำ ปกติจะมีค่าความต้านทาน 10 โอห์มที่ 0 °C ค่าความผิดพลาด 0.25 °C



รูปที่ 2.10 อัตราส่วนของความต้านทานของโลหะที่ใช้อาร์ทีดีเทียบกับค่าความต้านทานของแต่ละชนิดที่ 0 °C

อาร์ทีดีแบบทังสเตนก็มีที่ใช้อยู่บ้างแต่ก็ไม่แพร่หลาย เพราะมีเสถียรภาพไม่ดี ( คุณสมบัติเปลี่ยนไปเมื่อผ่านการใช้งานแล้ว ) แต่มีข้อดีคือความแข็งแรง ( Strength ) สูงกว่าแบบอื่น ๆ สามารถทนอุณหภูมิได้สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

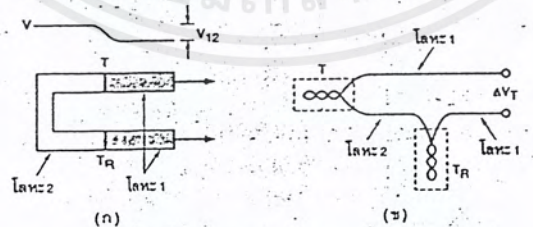
### 2.1.3.3 ความผิดพลาดอันเกิดจากกระแสเลี้ยงอาร์ทีดี

อาร์ทีดีจำเป็นต้องมีกระแสไฟฟ้าเลี้ยงตัวมันเองจำนวนหนึ่ง ถ้ามีค่ามากขึ้นเอาที่พหุโวลต์เตจก็จะสูงขึ้น แต่กระแสจำนวนนี้ก็สร้างความร้อนขึ้นในตัวอาร์ทีดีเองด้วยออกไซด์ หรืออะลูมิเนียมออกไซด์ ในข้อ 2.4 จะกล่าวโครงสร้างของอาร์ทีดี พลังงานความร้อนที่เกิดขึ้นเท่ากับค่ากระแสกำลังสองหารด้วยค่าความต้านทาน

ความร้อนที่เกิดขึ้นนี้จะทำให้เกิดค่าความผิดพลาดทำให้ความต้านทานสูงขึ้น ดังนั้นจึงจำเป็นต้องจำกัดค่ากระแสไม่ให้สูงเกินไป แต่ในสภาวะการใช้งานที่ต้องจุ่มลงไปของเหลวค่าความร้อนที่เกิดขึ้นนี้จะกระจายลงสารของเหลวทำให้ค่าความผิดพลาดนี้ต่ำลง ทั้งนี้ขึ้นอยู่กับความเร็วในการไหลความหนาแน่นของตัวกลางอาจลดลงถึง 1/5 ของค่าความผิดพลาดบรรยากาศปกติ

### 2.1.4 THERMOCOUPLE

เทอร์โมคัปเปิลเป็นทรานสดิวเซอร์ที่ใช้หลักการของเทอร์โมอิเล็กทริกซึ่งก็คือปรากฏการณ์ที่โลหะสองชนิดที่มีความหนาแน่นของอิเล็กตรอนภายในแตกต่างกันถูกนำมาต่อเข้าด้วยกันและเมื่อรอยต่อได้รับความร้อนจะเกิดแรงดันขึ้นระหว่างรอยต่อนั้น ดังในรูปที่ 2.11 (ก) ทรานสดิวเซอร์นี้จึงถูกเรียกว่า เทอร์โมคัปเปิล (Thermocouple)



รูปที่ 2.11 รอยต่อของเทอร์โมคัปเปิล

(ก) แสดงแรงดันที่เกิดขึ้นระหว่างรอยต่อ

(ข) แสดงการต่อใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัติ คู่โลหะมีมากมายหลายชนิด จึงต้องกำหนดโลหะอ้างอิง วัสดุที่ใช้ประกอบกันเป็น Thermocouple นั้นส่วนมากจะได้แก่ Iron-Constantan, Copper-Constantan, Chromel-Alumel และ Platinum/Rhodium-Platinum เพื่อความเป็นสากลในการบอกสเปกของโลหะต่าง ๆ ว่า จะเกิดค่าแรงดันต่ออุณหภูมิเท่าไร โดยโลหะที่เหมาะสมกับหน้าที่นี้คือ พลาตินัม ( Pt ) และจากการทดลองพบว่า โลหะแต่ละชนิดให้กำเนิดแรงดันเมื่อเทียบกับ Pt เป็นดังตารางที่ 2.2

ตารางที่ 2.2 ค่าแรงดันต่ออุณหภูมิที่เกิดขึ้นระหว่างรอยต่อของโลหะต่าง ๆ เทียบกับ Pt ที่ 0 °C

วัสดุ	ความไว ( $\mu\text{V}/^{\circ}\text{C}$ )	วัสดุ	ความไว ( $\mu\text{V}/^{\circ}\text{C}$ )
คอนสแตนแตน	- 35.0	ทองแดง	+ 6.5
นิกเกิล	- 15.0	ทอง	+ 6.5
อะลูเมล	- 13.6	ทังสแตน	+ 7.5
คาร์บอน	+ 3.0	เหล็ก	+ 18.5
อะลูมินัม	+ 3.5	โครเมล	+ 25.8
เงิน	+ 6.5	ซิลิกอน	+ 440.0

จากตารางที่ 1 สามารถนำไปหาแรงดันที่จะเกิดขึ้นระหว่างโลหะสองชนิดได้โดยใช้สมการที่ 2.4

$$V_{12} = V_1 - V_2 \quad (2.4)$$

โดยที่  $V_{12}$  คือ แรงดันต่ออุณหภูมิที่เกิดขึ้นระหว่างโลหะที่ 1 เทียบกับโลหะที่ 2 (  $\mu\text{V}/^{\circ}\text{C}$  )

$V_1$  คือ แรงดันต่ออุณหภูมิที่เกิดขึ้นระหว่างโลหะที่ 1 เทียบกับ Pt (  $\mu\text{V}/^{\circ}\text{C}$  )

$V_2$  คือ แรงดันต่ออุณหภูมิที่เกิดขึ้นระหว่างโลหะที่ 2 เทียบกับ Pt (  $\mu\text{V}/^{\circ}\text{C}$  )

ตัวอย่าง เมื่อเอา Cr กับ Al มาต่อกันจะเกิดแรงดันต่ออุณหภูมิที่รอยต่อเท่ากับ  $V_{\text{Cr/Al}} = V_{\text{Cr/Pt}} - V_{\text{Al/Pt}}$  แทนค่าจากตารางที่ 1 จะได้ = +25.8 - (-13.6) นั่นคือ 39.4  $\mu\text{V}/^{\circ}\text{C}$  ถ้าเอารอยต่อนี้ มาไว้ที่อุณหภูมิ 25 °C จะเกิดแรงดัน 0.985 mV

จากตัวอย่างพบว่า ถ้าเทอร์โมคัปเปิลมีเพียงรอยต่อเดียวจะเกิดแรงดันขึ้นที่อุณหภูมิห้องแน่นอน เพราะแรงดันต่ออุณหภูมิถูกอ้างอิงที่ 0 °C วิธีการที่ช่วยให้แรงดันที่อุณหภูมิใช้งานปกติหรือยังไม่ได้วัดอุณหภูมิมีค่าเท่ากับ 0 V ทำได้โดยการเพิ่มอีก 1 รอยต่อเป็นเทอร์โมคัปเปิล 2 รอยต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังแสดงในรูปที่ 2.11 ( ข ) ค่า  $\square V_T = V_{12} - V_{21} = 0$  เมื่อทั้งสองรอยต่อมีอุณหภูมิเท่ากัน ถ้ารอยต่อที่ 1 มีอุณหภูมิเพิ่มขึ้นหรือลดลง จะสามารถหาแรงดันทางเอาต์พุตได้ดังนี้คือ

$$V_0 = V_{12} (T - T_R) \quad (2.5)$$

โดยที่  $T$  คือ อุณหภูมิที่รอยต่อที่ 1 ซึ่งใช้วัดอุณหภูมิ ( $^{\circ}\text{C}$ )

$T_R$  คือ อุณหภูมิที่รอยต่อที่ 2 หรือรอยต่ออ้างอิง ( $^{\circ}\text{C}$ )

ตารางที่ 2.3 ค่าแรงดันที่เกิดขึ้นจากคูโลหะต่าง ๆ เช่น Cu-Con , Pt-Pt-10Rh , Cr-Al โดยแรงดันมีหน่วยเป็นมิลลิโวลต์ เทียบที่อุณหภูมิ  $0^{\circ}\text{C}$

อุณหภูมิ ( $^{\circ}\text{C}$ )	Cu-Con	Pt-Pt- 10Rh	Cr-Al	อุณหภูมิ ( $^{\circ}\text{C}$ )	Cu-Con	Pt-Pt- 10Rh	Cr-Al
- 60	2.14	.....	- 2.20	40	1.61	0.235	1.61
- 40	- 1.46	.....	- 1.50	60	2.47	0.364	2.43
- 20	- 0.75	.....	- 0.77	80	3.36	0.500	3.26
0.00	0.00	0.000	0.00	100	4.28	0.643	4.10
20	0.78	0.113	0.80	120	5.23	0.792	4.92

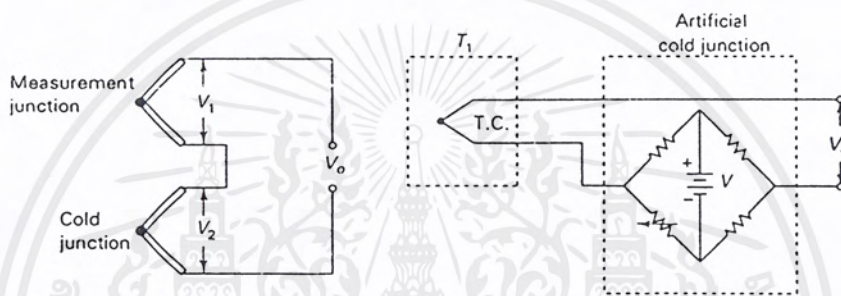
ในตารางที่ 2.3 แสดงค่าแรงดันที่เกิดขึ้นในเทอร์โมคัปเปิลที่ใช้คูโลหะต่าง ๆ ซึ่งมาจากสมการที่ได้กล่าวไปข้างหน้า เพื่อสะดวกต่อการใช้งาน กล่าวคือเมื่อรู้ชนิดของคูโลหะและทราบแรงดันที่เกิดขึ้น สามารถใช้ตารางนี้เทียบเป็นอุณหภูมิได้เลย

การนำไปใช้งานจริง รอยต่ออ้างอิงจะอยู่ที่ขั้วต่อใช้งานดังในรูปที่ 2.12 จะเห็นได้ว่าถ้าจะให้เกิด 2 รอยต่อ ต้องใช้ตัวนำชนิดเดียวกับโลหะที่ 1 จากรูปที่ 2.11 ( ข ) แต่ในทางปฏิบัติอาจหาตัวนำที่เป็นโลหะชนิดดังกล่าวไม่ได้ ต้องใช้สายทองแดงแทน แรงดันที่เกิดตรงขั้วต่อใช้งานทั้งสองขั้วต่อใช้งานทั้งสองขั้วเป็นส่วนที่ไม่ต้องการ ดังนั้นเพื่อชดเชยข้อผิดพลาด ตรงนี้จึงใช้วงจรบริดจ์ต่ออนุกรมกับเทอร์โมคัปเปิลดังรูปที่ 2.12 พร้อมกับดีคเทอร์มิสเตอร์ไว้ที่ขั้วต่อด้านหนึ่งเพื่อวัดอุณหภูมิอ้างอิงที่ขั้วต่อแล้วนำมาแปลงค่าเป็นแรงดันไปหักลบกับค่าแรงดันที่เกิดขึ้นระหว่างขั้วต่อที่ไม่ต้องการ ทั้งนี้  $\alpha$  ค่า ของเทอร์มิสเตอร์จะต้องสอดคล้องกับค่า ของเทอร์โมคัปเปิลด้วย และก่อนที่จะนำไปวัดอุณหภูมิจริง ๆ ต้องปรับค่า  $R_x$  จนแรงดัน  $\square V = 0$  เมื่อเทอร์โมคัปเปิลอยู่ ณ อุณหภูมิ  $0^{\circ}\text{C}$  ก่อนค่าวัดได้จึงจะถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการใช้เทอร์โมคัปเปิลมาตรฐานวัดอุณหภูมิคือ ทนทาน ใช้วัดอุณหภูมิสูง ๆ ได้ ย่านการวัดกว้าง แรงดันต่ออุณหภูมิมีความเป็นเชิงเส้นสูง

ส่วนข้อเสีย คือ ราคาแพง ขนาดใหญ่ต้องมีการปรับแต่งก่อนใช้งานยุ่งยากกว่าทรานสดิวเซอร์ที่กล่าวไปแล้วและแรงดันที่ได้ต่ำ



รูปที่ 2.12 การใช้งานบริดจ์เพื่อชดเชยค่าผิดพลาดของเทอร์โมคัปเปิล

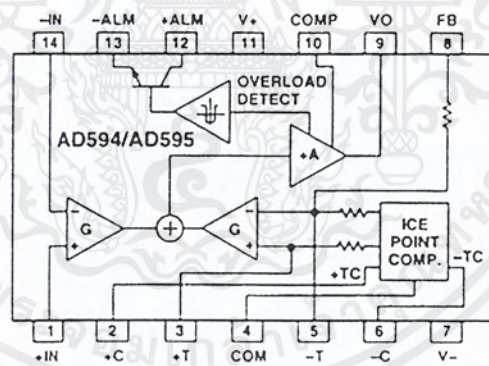
โดยในรายงานนี้จะใช้เทอร์โมคัปเปิล Type K ที่เป็น NiCr-Ni โดยเลือกชนิดที่ออกแบบมาสำหรับใช้วัดอุณหภูมิของของเหลวและก๊าซซึ่งเป็นชนิดที่นิยมใช้งานแพร่หลายที่สุด และมีย่านการใช้งานที่ค่อนข้างกว้างประมาณ  $\pm 1000$  °C โดยมีค่าแรงดันแสดงไว้ในภาคผนวก แต่เนื่องจากเทอร์โมคัปเปิลมีค่าแรงดันที่ต่ำมากเราจึงต้องการวงจรที่จะทำการขยายสัญญาณที่ได้ต่อไป

## 2.2 วงจรขยายสัญญาณเทอร์โมคัปเปิล

### 2.2.1 คุณสมบัติของ AD-595

AD-595 เป็น Monolithic Thermocouple Amplifiers with Cold Junction Compensation ซึ่งมีคุณสมบัติดังต่อไปนี้

- เป็นวงจขยายสำหรับ K-type Thermocouple
- เอาท์พุทอิมพีแดนซ์ต่ำ และให้สัญญาณเอาท์พุทขนาด  $10 \text{ mV} / ^\circ\text{C}$
- มีการชดเชยรอยต่อความเย็น ( มี Ice Point Compensation )
- มีการปรับแต่งด้วยระบบ Laser Wafer Trimming ( LWT )
- กินกำลังไฟต่ำ ( น้อยกว่า  $1 \text{ mW}$  ) และสามารถใช้กับแหล่งจ่ายได้ตั้งแต่  $+5$  ถึง  $\pm 15 \text{ V}$
- อินพุทเป็นแบบ Differential Input ช่วยลดสัญญาณรบกวนที่เกิดขึ้นจากตัวเทอร์โมคัปเปิลและสายวัด



รูปที่ 2.13 โครงสร้างภายในของ AD-595

### 2.2.2 หลักการทำงานของ AD-595

จากรูปที่ 2.13 จะเห็นว่า AD-595 จะมีลักษณะเป็น Differential Amplifier 2 ตัว ที่นำเอาท์พุทมารวมกันแล้วนำไปควบคุมกำลังขยายหลักของวงจร โดยค่ากำลังหลักนี้มีค่า 247.3 เนื่องจาก AD-595 ต้องการแรงดันเอาท์พุทเป็น  $10 \text{ mV}/^\circ\text{C}$  แต่ K Type Thermocouple ให้ค่าแรงดันเอาท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พุทประมาณ  $40.4 \mu\text{V}/^{\circ}\text{C}$  เพราะฉะนั้นค่ากำลังขยายจะเป็น  $10 \text{ mV}/^{\circ}\text{C}$  หากด้วย  $40.44 \mu\text{V}/^{\circ}\text{C}$  ( $V_{\text{out}} / V_{\text{in}}$ ) จะได้เป็นกำลังขยายหลักเป็น 247.3

ในการทำงานปกติขา 9 จะต่อกับขา 8 ซึ่งเป็นการป้อนกลับ โดยสัญญาณจากเทอร์โมคัปเปิลจะป้อนเข้าสู่ขา 1 และ ขา 14 และถูกขยายด้วยค่า G ซึ่งเป็นค่ากำลังขยายของ Differential Amplifier ตัวแรก และถูกขยายต่อด้วยกำลังขยายหลัก A เอาท์พุทที่ได้จะถูกป้อนกลับไปที่ Differential Amplifier ตัวที่สองที่ขา Inverting ที่ส่วนนี้จะมีการชดเชยค่ารอยต่อเย็นด้วย จากนั้นสัญญาณที่ได้จะถูกส่งไปรวมกับ (Summing) สัญญาณจากส่วนแรกแล้วผ่านกำลังขยายหลัก A ได้เป็นสัญญาณเอาท์พุทที่มีค่าเป็น  $10 \text{ mV}/^{\circ}\text{C}$  เมื่อมีการกระตุ้นเทอร์โมคัปเปิล

### 2.2.3 การชดเชยที่รอยต่อเย็น

ในส่วนของการป้อนกลับที่ Differential Amplifier ตัวที่ 2 จะมีการชดเชยค่าของรอยต่อเย็น ซึ่งคือค่าอัตราส่วนโดยตรงของผลต่างของอุณหภูมิ  $0^{\circ}\text{C}$  กับอุณหภูมิของตัว AD-595 ในหน่วย องศาเซลเซียส โดยค่าอุณหภูมินี้จะมีผลกับค่าเอาท์พุทของวงจรทำให้เกิดความผิดพลาดได้ ดังนั้นส่วน Ice Point Compensation จึงทำการชดเชยในลักษณะของค่าแรงดันผ่านทางอัตราส่วนของค่าความต้านทานที่ต่อเข้าสู่ขา Inverting ของ Differential Amplifier ตัวที่ 2 และเนื่องจาก Differential Amplifier ทั้งสองตัวถูกออกแบบมาให้มีค่ากำลังขยายเท่ากัน เมื่อนำมารวมกันจึงทำให้ผลของอุณหภูมิของตัว AD-595 หายไป

โดยส่วนของ Ice Point Compensation จะมีส่วนของ Temperature Coefficient (TC) ที่ขา 2 และขา 6 ทั้งค่าบวกและค่าลบ โดยแรงดันส่วนนี้สามารถนำมาต่อกับความต้านทานภายนอกเพื่อทำการปรับแต่งใหม่ (Recalibration) เพื่อสำหรับใช้กับเทอร์โมคัปเปิลชนิดอื่นๆได้ นอกจากนี้เรายังสามารถต่อความต้านทานภายนอกระหว่างขา 5 และขา 9 เพื่อทำการปรับค่ากำลังขยายหลักได้

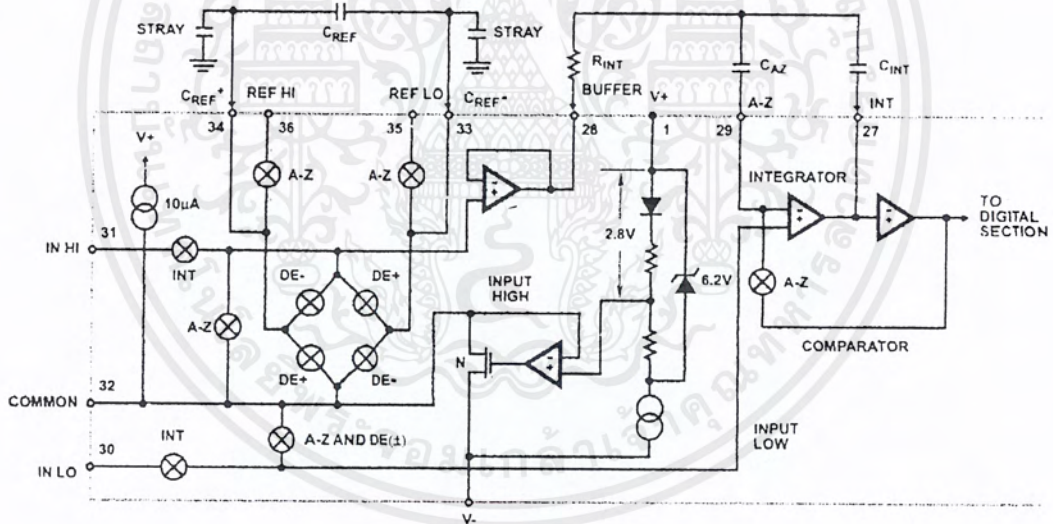
### 2.3 วงจรส่วนแสดงผลด้วย ICL

เราจะแสดงผลของค่าที่วัดได้จากส่วนของวงจรขยายสัญญาณด้วยจอแบบ LCD โดยจะใช้ ICL 7106 เป็นวงจรที่ร่วมกับ LCD โดย ICL 7106 เป็น 3 1/2 Digit LCD Display A/D Converter ซึ่งคือวงจรรวมที่ใช้ในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลที่มีคุณภาพสูงและกินกำลังไฟต่ำ โดยจะประกอบไปด้วย วงจรจับดีโค๊ดเตอร์ 7 เซกเมนต์ วงจรขับจอแสดงผล LCD ขนาด 3

1/2 หลัก วงจรสร้างแรงดันอ้างอิง และวงจรสร้างสัญญาณนาฬิกา โดย ICL 7106 ถูกออกแบบมาให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

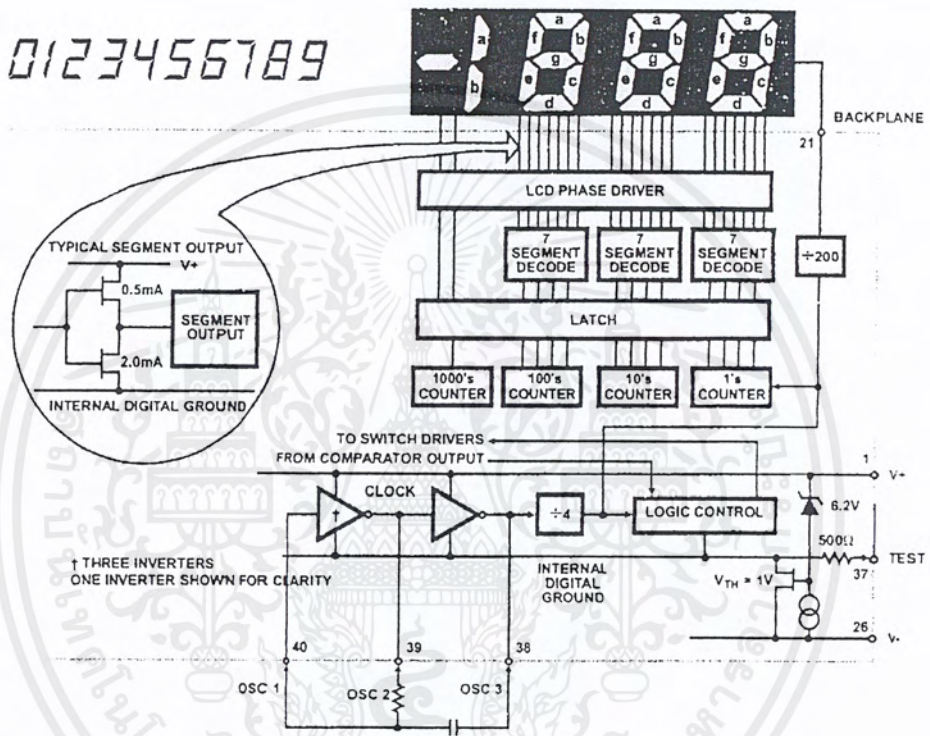
เชื่อมต่อกับจอแสดงผลแบบ LCD และมีวงจรจับแบ็กเฟลนอยู่ภายในดังแสดงในรูปที่ 2.15 มีคุณสมบัติเด่นๆ ดังนี้คือ

- มั่นใจได้ว่าค่าที่อ่านได้จะเป็นศูนย์ เมื่อค่าแรงดันอินพุตที่ป้อนให้มีค่าเท่ากับ 0 V ในทุกสเกลของการวัด
- กินกระแสอินพุตต่ำเพียง 1 pA
- สัญญาณรบกวนน้อยกว่า 15 uVp-p
- มีวงจรสร้างสัญญาณนาฬิกาและแรงดันอ้างอิงอยู่ภายใน
- กินกำลังงานต่ำเพียง 10 mV
- ไม่ต้องการวงจรจับหรือวงจรขยายจากภายนอก



รูปที่ 2.14 วงจรส่วนอนาลอกภายใน ICL 7106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 วงจรส่วนดิจิทัลภายใน ICL 7106

จากรูปที่ 2.14 จะเห็นว่าการทำงานของ 7106 ใช้หลักการของ ‘ Dual Slope ’ โดยในส่วน  
ของวงจรถอดในในแต่ละรอบของการทำงานจะแบ่งเป็น 3 ส่วนย่อยคือ

1. Auto-Zero Phase (A-Z) ในช่วงนี้จะเกิดเหตุการณ์ 3 อย่าง เริ่มจากขาอินพุท(ขา 31 และ  
ขา 32 ) จะถูกตัดจากภายนอกแล้วลัดวงจรเข้าไปที่ขา Analog Common จากนั้น  $C_{REF}$  จะถูกประจุ  
ด้วยค่าแรงดันอ้างอิง จากนั้นจะทำการต่อส่วนป้อนกลับของ A-Z ทุกส่วน เพื่อทำการประจุ  $C_{AZ}$   
เพื่อชดเชยค่าแรงดันออฟเซต ( Offset ) ของวงจรถอด Buffer Amplifier , Integrator และ Comparator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Signal Integrate Phase (INT) ในช่วงนี้รูป A-Z จะถูกเปิดออก ขาอินพุตต่อกับภายนอก จากนั้นวงจรจะทำการอินทิเกรตสัญญาณจนหมดช่วงเวลา

3. De-Integrate Phase (DE) หรือ Reference integrate เริ่มจากขา IN LO จะต่อกับขา COMMON ส่วน IN HI จะข้ามไปต่อกับ C<sub>REF</sub> ทำให้เอาต์พุตของวงจรอินทิเกรตตกลงเป็นศูนย์

เวลาที่ใช้ในการทำให้เอาต์พุตกลับลงมาเป็นศูนย์จะเป็นอัตราส่วนกับช่วงของอินพุต ซึ่งค่าที่ได้จะถูกส่งไปที่ส่วนของวงจรดิจิทัลในรูปที่ 2.15 เพื่อทำการนับค่า โดยค่าที่ได้จะถูกนำไปแสดงผลที่ LCD ต่อไป โดยมีความสัมพันธ์ดังสมการที่ 2.6

$$\text{Display Count} = 1000 \left( \frac{V_{in}}{V_{ref}} \right) \tag{2.6}$$

## 2.4 วงจร ANALOG TO DIGITAL CONVERTERS

การทำงานของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล ในโครงงานนี้ใช้ไอซี ADC0804 (ADC 8 บิต) ซึ่งไอซี ADC นี้จะเป็นส่วนที่แปลงสัญญาณเอาต์พุต ที่เป็นแรงดันไฟฟ้าของส่วนที่วัดอุณหภูมิซึ่งเป็นสัญญาณอนาล็อก ให้เป็นสัญญาณดิจิทัล ซึ่งสามารถรับสัญญาณอินพุตแบบอนาล็อกได้โดยตรงสูงถึง +5 โวลต์ ไอซี ADC จะรับสัญญาณเข้ามาเปรียบเทียบกับค่าแรงดันอ้างอิงของวงจรแล้วจึงเปลี่ยนค่าข้อมูลแบบดิจิทัล ซึ่งความเร็วในการสุ่มสัญญาณ สามารถกำหนดได้จากความถี่สัญญาณนาฬิกาของวงจร ค่าความถี่ของสัญญาณสามารถกำหนดได้ โดยวงจร R และ C โดยมีความสัมพันธ์ดังสมการที่ 2.7

$$F = 1/(1.1RC) \tag{2.7}$$

F คือ ค่าความถี่ของการสุ่มขนาดสัญญาณของไอซี

R คือ ค่าความต้านทานของ R ในวงจร

C คือ ค่าความจุของ C ในวงจร

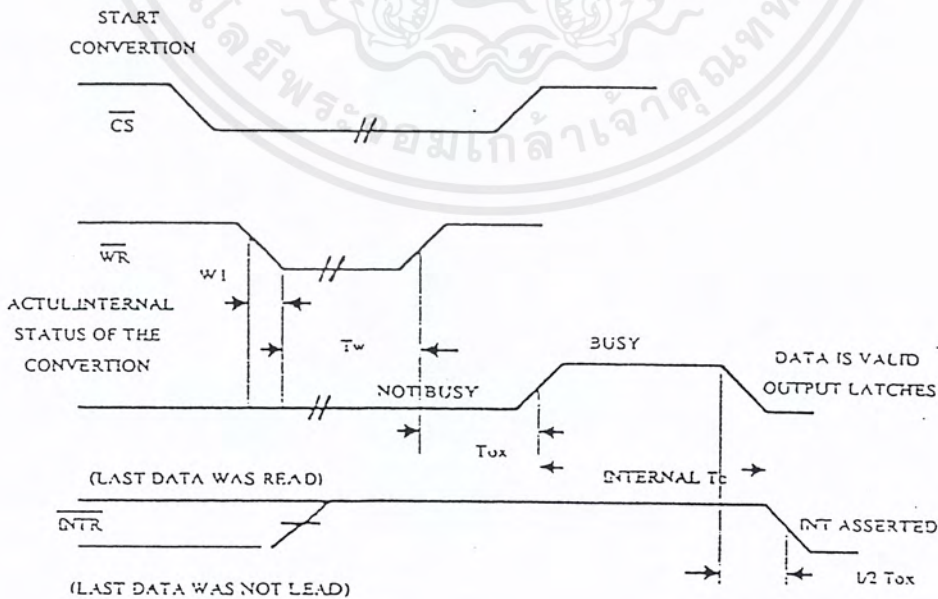
### 2.4.1 คุณสมบัติของไอซี

- ความผิดพลาดโดยรวมเกิดจากการปรับค่าไม่เกิน 1 บิต
- ความละเอียดขนาด 8 บิต = 256 ระดับ
- ใช้สัญญาณนาฬิกา 640 Hz
- มีช่วงเวลาการทำงานในการแปลงแต่ละบิต เท่ากับ 100 ไมโครวินาที
- ใช้แหล่งจ่ายไฟตรง 5 โวลต์
- กินกำลังไฟต่ำ 15 มิลลิวัตต์

2.4.2 ส่วนการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

ประกอบด้วยส่วนประมาณค่าซ้ำๆ (Successive) ส่วนของเอาท์พุทมัลติเพล็กซ์ (Multiplex) และส่วนของสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

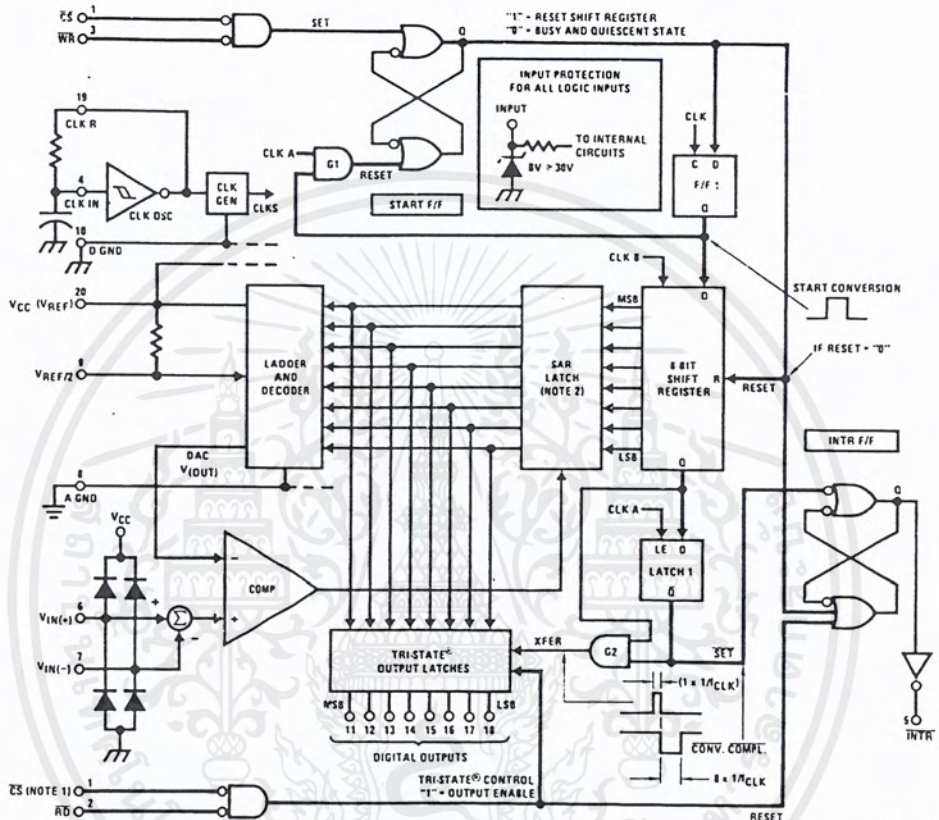
ไอซี ADC แบบนี้จะมีการเก็บสัญญาณเอาท์พุทเพิ่มขึ้นมาด้วยเพื่อให้ค่าเอาท์พุทยังคงมีอยู่แม้ว่าวงจรจะถูกปลดออกจากบัสในช่วงของเวลาก็ตาม ขณะวงจรทำการแปลงสัญญาณอยู่ ADC จะติดต่อกับไมโครคอนโทรลเลอร์ได้นั้นต้องมีสัญญาณเลือกเป็นลอจิก 0 จากระบบไมโครคอนโทรลเลอร์มาควบคุม ADC เมื่อขา WR เป็นลอจิก 0 ADC จะแปลงค่า VI เป็นสัญญาณดิจิทัลเก็บไว้ในบัพเฟอร์เอาท์พุท และไมโครคอนโทรลเลอร์จะอ่านข้อมูลจาก ADC เมื่อสัญญาณเลือก CS เป็นลอจิก 0 พร้อมกับในแต่ละรอบของการแปลงค่าดังรูปที่ 2.16



รูปที่ 2.16 เวลาการทำงานของ ADC 0804

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล ภายในตัวไอซี ADC เป็นแบบ Successive Approximation เป็นการประมาณค่าอย่างต่อเนื่องดังรูปที่ 2.17



Note 13:  $\overline{CS}$  shown twice for clarity.  
 Note 14: SAR = Successive Approximation Register.

รูปที่ 2.17 การทำงานแบบ Successive Approximation (SAR)

2.4.3 ความสัมพันธ์ของอินพุตและเอาต์พุตของไอซี ADC

ไอซี ADC จะใช้การเปรียบเทียบขนาดของสัญญาณอินพุตกับแรงดันอ้างอิงของวงจร แล้วจึงเปลี่ยนข้อมูลแบบดิจิตอล ซึ่งมีความสัมพันธ์ดังนี้

- ถ้าค่าของขนาดสัญญาณอนาลอกอินพุต เท่ากับขนาดของแรงดันอ้างอิง จะได้ข้อมูลมีค่าเป็นครึ่งหนึ่งของข้อมูลสูงสุด
- ถ้าขนาดของสัญญาณอนาลอกอินพุตมีค่าเป็นสองเท่าของแรงดันอ้างอิงจะได้ข้อมูลมีค่าสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.4 แรงดันอ้างอิงของวงจร

แรงดันอ้างอิงของวงจรนี้จะเป็จุดอ้างอิงมาตรฐานของวงจร เป็นตัวกำหนดความเที่ยงตรงในการเปลี่ยนขนาดของสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัล ซึ่งวงจรต้องการแรงดันอ้างอิงที่มีความเที่ยงตรงสูง เพราะถ้าหากแรงดันอ้างอิงของวงจรไม่มีความเที่ยงตรงแน่นอน มีการเปลี่ยนแปลงแล้ว การทำงานของวงจรก็จะมีเปลี่ยนแปลงตามไปด้วย นั่นก็คือจะส่งผลให้ค่าของสัญญาณข้อมูลดิจิทัลที่ได้ก็จะเปลี่ยนแปลงตามไปด้วย ถึงขนาดอินพุตมีค่าคงที่ และแรงดันอ้างอิงของวงจรจะถูกกำหนดให้มีค่าสูงสุดประมาณ 2 เท่า โดยมีจุดเริ่มต้นที่สัญญาณอินพุตลบของวงจรเสมอ

สัญญาณอินพุตลบเป็นตัวกำหนดค่าอินพุตด้านต่ำของวงจรมันเอง หรือจุดเริ่มต้นของการตรวจรับสัญญาณอินพุต ของวงจรมันเอง สามารถกำหนดให้มีแรงดันไม่เกินแรงดันอ้างอิงสูงสุดของวงจร

สัญญาณอินพุตบวก คือค่าแรงดันอินพุตของวงจรมันเอง ซึ่งต้องกำหนดให้มีขนาดสูงสุดไม่เกินค่าที่วงจร ADC รับได้ ต้องเป็นแรงดันช่วงบวกเท่านั้น

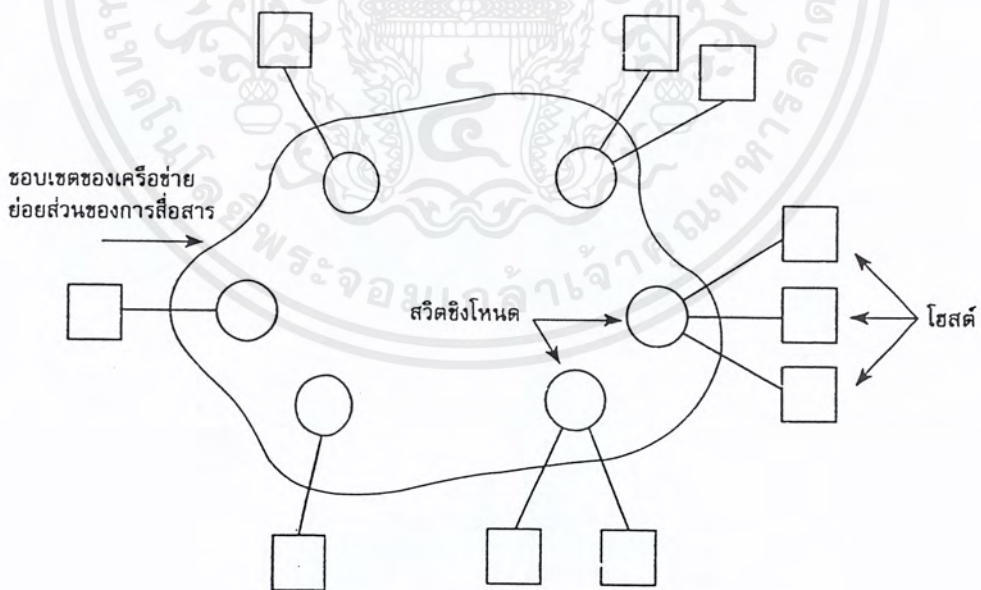
### บทที่ 3

#### เครือข่ายอินเทอร์เน็ต

#### 3.1 เครือข่ายคอมพิวเตอร์ ( Computer Network )

##### 3.1.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์ ( Network Components )

เครือข่ายคอมพิวเตอร์จะประกอบด้วยโหนด ( node ) 2 ชนิดคือ โหนดที่ทำหน้าที่ประมวลข้อมูล หรือ Computing node และ โหนดที่ทำหน้าที่ส่งข้อมูล หรือ Switching node ผ่านสายส่ง จากต้นทางไปยังปลายทางได้ ดังแสดงในรูปที่ 3.1 ปกติแล้วเมื่อผู้ใช้เครือข่ายต้องการประมวลผล ข้อมูลจะอาศัยเทอร์มินัลในการส่งข้อมูลไปประมวลผลที่คอมพิวเตอร์ประมวลผล เช่น โฮสต์ ( host ) โดยที่ข้อมูลเหล่านั้นจะผ่านโหนดส่งข้อมูลต่าง ๆ ซึ่งจะส่งข้อมูลให้ถึงปลายทาง และหลังจากที่ โฮสต์ประมวลผลเสร็จก็จะส่งผลผ่านสวิตชิงโหนดต่าง ๆ กลับมาให้ผู้ใช้ได้ ดังนั้นจึงมักแบ่งเครือข่ายคอมพิวเตอร์ออกเป็น 2 ส่วนใหญ่ ๆ ตามหน้าที่ของโหนดนั้น ๆ ก็คือแบ่งเป็น เครือข่ายย่อยส่วนของผู้ใช้ ( User Subnetwork ) และเครือข่ายย่อยส่วนของการสื่อสาร ( Communication Subnetwork )

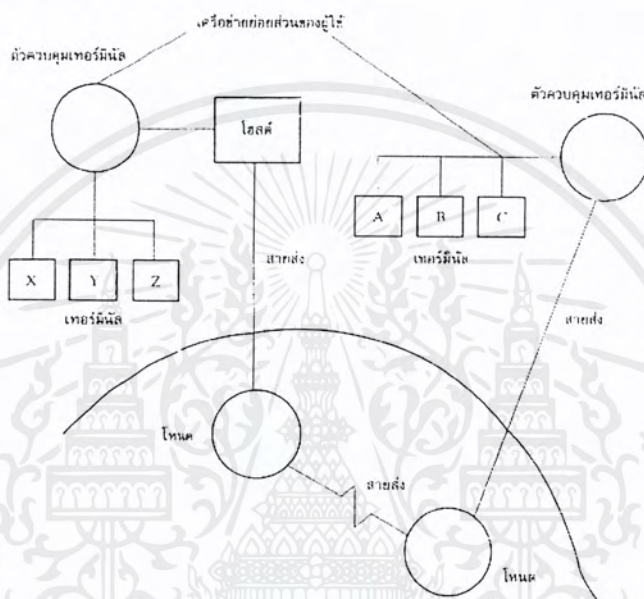


รูปที่ 3.1 องค์ประกอบของเครือข่ายคอมพิวเตอร์

##### 3.1.1.1 เครือข่ายย่อยส่วนของผู้ใช้ ( User Subnetwork ) คือส่วนที่ผู้ใช้ต้องการส่งข้อมูลไป

ประมวลผลแล้วแสดงผลออกมาให้ทราบ เครือข่ายส่วนนี้ประกอบด้วยโฮสต์คอมพิวเตอร์ ( Host เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

computer ) เทอร์มินัล ( Terminal ) และตัวควบคุมเทอร์มินัล ( Terminal controllers ) โดยที่โฮสต์ จะมีทรัพยากรต่าง ๆ เช่น ฐานข้อมูล โปรแกรมประยุกต์ที่ผู้ใช้สามารถใช้งานได้เป็นต้น ผู้ใช้จะ อาศัยเทอร์มินัลในการป้อนข้อมูลและแสดงผลที่ได้จากการประมวลผลนั่นเอง โดยผ่านตัวควบคุม เทอร์มินัลหรือตัวควบคุมระบบการสื่อสาร ( Communication controller ) ซึ่งทำหน้าที่ควบคุมการ ทำงานของกลุ่มของเทอร์มินัลดังแสดงในรูปที่ 3.2



รูปที่ 3.2 เครื่องข่ายในส่วนของเครือข่ายย่อยผู้ใช้บริการ

การใช้งานคอมพิวเตอร์ในลักษณะที่ผู้ใช้ป้อนข้อมูลผ่านเทอร์มินัลแล้ว การประมวลผลเกิด ที่โฮสต์คอมพิวเตอร์ศูนย์กลาง มักจะถูกเรียกว่า ระบบโฮสต์-เทอร์มินัล ( Host-Terminal ) ซึ่งใน กรณีที่มีผู้ใช้หลายคนประมวลผลคอมพิวเตอร์พร้อม ๆ กัน จะทำให้การทำงานของระบบ คอมพิวเตอร์ช้ามาก ดังนั้นในปัจจุบันจึงมีการใช้งานคอมพิวเตอร์ในลักษณะของผู้ใช้บริการ / ผู้ให้ บริการ ( Client / Server ) มากขึ้น ซึ่งเป็นรูปแบบหนึ่งของการประมวลผลแบบกระจาย กล่าวคือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์จะให้บริการ เช่น บริการเก็บแฟ้มข้อมูล ( File Server ) บริการพิมพ์งานออกเครื่องพิมพ์ ( Print Server ) บริการประมวลผลงานเฉพาะอย่าง ( Computing Server ) หรือบริการการติดต่อสื่อสารกับเครือข่าย ( Communication Server ) เป็นต้น ส่วนเครื่อง คอมพิวเตอร์ของผู้ใช้บริการ ( Client ) นั้นก็จะมีส่วนประมวลผลเป็นของตัวเองโดยจะส่งคำขอใ้ งาน ( Request ) ไป เมื่อเครื่องผู้ให้บริการ ( Server ) ทำงานตามคำขอ ก็จะส่งผลลัพธ์ตอบกลับมา เครื่องของผู้ใช้บริการก็จะสามารถประมวลผลได้ด้วยตัวเอง ซึ่งการประมวลผลลักษณะนี้ถึงแม้ว่าจะมี ผู้ใช้งานจำนวนมากก็ไม่ทำให้การทำงานของระบบช้าแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 เครือข่ายย่อยส่วนของการสื่อสาร ( Communication Subnetwork ) จะประกอบด้วยโหนดหลายโหนดติดต่อสื่อสารกันผ่านสายส่ง ( Transmission Link ) เพื่อจะส่งข้อมูลจากต้นทางถึงปลายทางได้อย่างถูกต้องและรวดเร็ว ตัวอย่างของเครือข่ายย่อยส่วนนี้ ได้แก่ เครือข่ายโทรศัพท์ เครือข่ายดาวเทียม เครือข่ายวิทยุ เป็นต้น และจากรูปที่ 3.3 จะแสดงให้เห็นว่าเครือข่ายย่อยส่วนของการสื่อสารจะประกอบด้วยโหนดเครือข่ายสายส่ง ( Network node ) และตัวแปลงสัญญาณ ( Signal Converters )



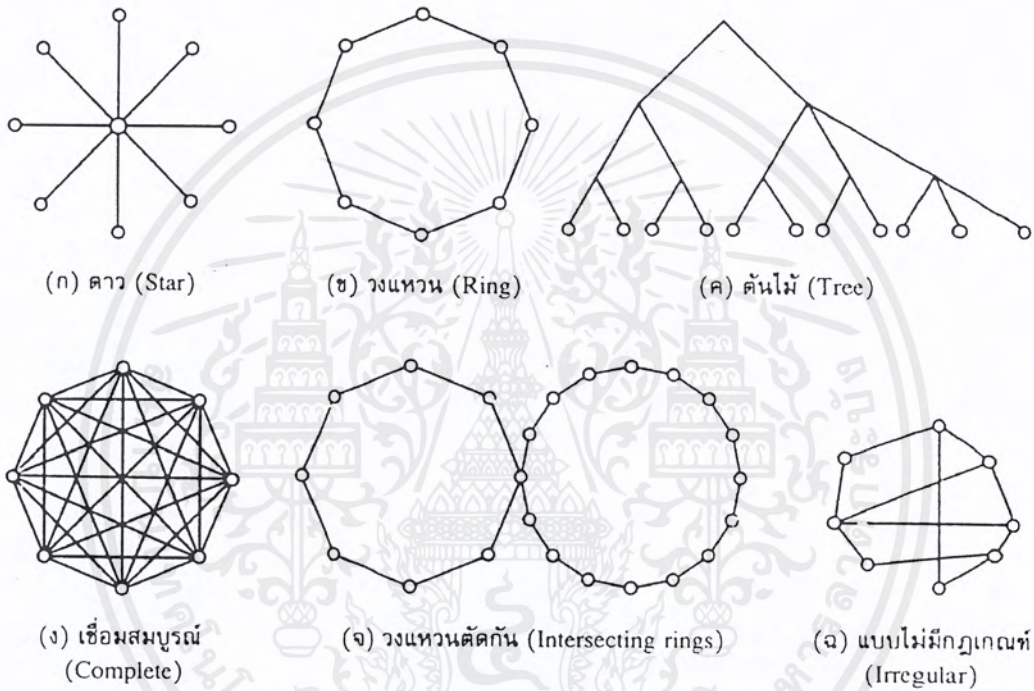
รูปที่ 3.3 เครือข่ายในส่วนของเครือข่ายย่อยของการสื่อสาร

สายส่งจะเป็นช่องทางการสื่อสาร ( Communication Channels ) หรือตัวกลางที่ใช้ในการส่งข้อมูล สายส่งอาจจะเป็นสายเคเบิลโทรศัพท์ โทรเลข วิทยุ หรือดาวเทียม ซึ่งอัตราการส่งข้อมูลมีตั้งแต่ 50 bps จนถึงระดับ  $10^{14}$  bps ปกติแล้วมักจะมีการใช้สายส่งมากกว่าหนึ่งชนิดสำหรับโหนดที่ติดกัน เพื่อเพิ่มความเชื่อถือได้ของระบบ สำหรับตัวแปลงสัญญาณนั้นจะทำหน้าที่ ในการเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณที่เหมาะสมสำหรับการส่งข้อมูลผ่านสายส่ง เช่น เปลี่ยนสัญญาณดิจิทัล เป็นสัญญาณแสงเมื่อส่งข้อมูลผ่านเส้นใยนำแสง ตัวอย่างของตัวแปลงสัญญาณ เช่น โมเด็ม ( Modem ) ซึ่งใช้ในการเปลี่ยนสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก ( Analog ) เพื่อส่งข้อมูลผ่านวงจรโทรศัพท์ ส่วนโหนดเครือข่ายนั้นจะทำหน้าที่เป็นทั้งตัวเชื่อมประสานกับเครือข่ายย่อยส่วนของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทำหน้าที่ส่งข้อมูลผ่านเครือข่ายจากต้นทางไปยังปลายทางได้อย่างถูกต้องรวดเร็วและมีประสิทธิภาพ ซึ่งบางทีจะเรียกโหนดเครือข่ายนี้ว่า Router โดยทั่วไปจะแบ่งชนิดเครือข่ายย่อยส่วนของการสื่อสารออกตามลักษณะการส่งข้อมูลระหว่างโหนดเครือข่ายโดยผ่านสายส่ง ออกเป็น 2 ชนิด คือ

1. แบบจุดต่อจุด (Point - to - point channels )
2. แบบแพร่กระจายข้อมูล ( Broadcast channels )



รูปที่ 3.4 รูปแบบต่างๆ ของเครือข่ายแบบจุดต่อจุด

จากรูปที่ 3.4 แสดงรูปแบบของเครือข่ายแบบจุดต่อจุด ซึ่งจะเห็นว่าแต่ละสายส่งจะถูกใช้เชื่อม

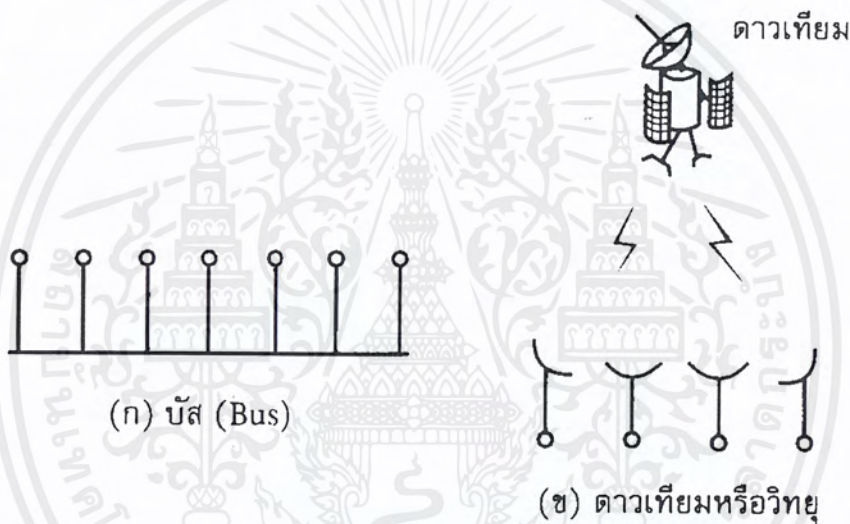
โยงเพื่อส่งข้อมูลระหว่างโหนด 2 โหนด และหากโหนดคู่ใดไม่มีสายเชื่อมโยงจะต้องส่งข้อมูลผ่านโหนดอื่น สำหรับเทคนิคในการส่งข้อมูลผ่านโหนดต่าง ๆ ของเครือข่ายแบบนี้ จะทำในลักษณะต่างๆได้แก่

- Circuit Switching เช่น ระบบโทรศัพท์ปัจจุบัน ซึ่งเมื่อมีการขอสร้างการติดต่อระหว่างต้นทางและปลายทาง เครือข่ายจะหาเส้นทางผ่านโหนดต่าง ๆ แล้วจะเชื่อมต่อสวิตช์ภายในโหนดเหล่านั้น เพื่อสร้างช่องสัญญาณให้ข้อมูลผ่านจากต้นทางถึงปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Packet switching ซึ่งจะทำงานในลักษณะของการเก็บรอและส่งออก ( Store and Forward ) คือ เมื่อโหนดใดได้รับข้อมูลซึ่งเป็นลักษณะบล็อกเล็ก ๆ ที่เรียกว่า Packet เข้ามาแล้ว โหนดนั้นจะพิจารณาเลือกสายส่งเพื่อส่งข้อมูลออกให้ถึงปลายทางได้เร็วที่สุด แล้วจะนำไปเก็บรอไว้จนกว่าสายส่งนั้นว่างจึงส่งข้อมูลออกไปยังโหนดอื่นต่อไป

สำหรับเครือข่ายย่อยส่วนของการสื่อสารแบบกระจายนั้น เป็นลักษณะที่โหนดเครือข่ายทั้งหมดจะใช้สายส่งหรือช่องสัญญาณหนึ่งร่วมกัน ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 รูปแบบต่างๆ ของเครือข่ายแบบแพร่กระจาย

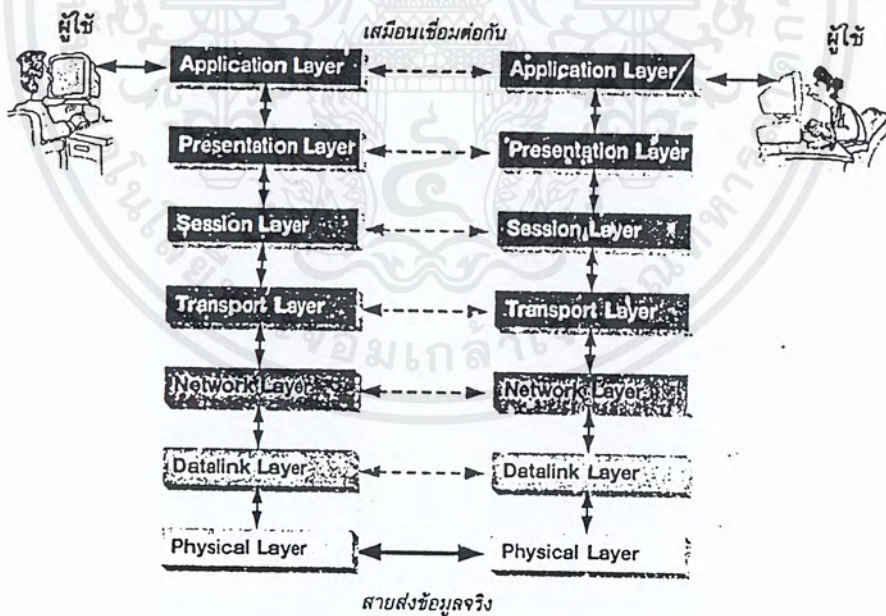
เมื่อโหนดใดส่งข้อมูลลงในช่องสัญญาณ ข้อมูลจะถูกแพร่ออกไป และถูกพบเห็น โดยทุก ๆ โหนด ดังนั้นในกรณีที่ต้องการส่งข้อมูลให้แก่โหนดหนึ่งจะต้องใส่แอดเดรส ( Address ) ของโหนดนั้นลงในข้อมูลด้วย นอกจากนี้ในระบบแพร่กระจายข้อมูลนี้ยังสามารถส่งข้อมูลให้แก่ทุกโหนดพร้อมกัน หรือกลุ่มของโหนดบางโหนดก็ได้ เนื่องจากระบบแพร่กระจายข้อมูลนี้มีการใช้ช่องสัญญาณร่วมกันระหว่างหลาย ๆ โหนด ดังนั้นจึงต้องมีวิธีกควบคุมการใช้ช่องสัญญาณ เพื่อป้องกันไม่ให้โหนดต่างๆ ส่งข้อมูลออกมาชนกันเอง สำหรับวิธีการควบคุมการแบ่งใช้ช่องสัญญาณนี้ได้แก่ การจัดสรรช่องสัญญาณ ( Allocation )

### 3.2 OSI Model : มาตรฐานอ้างอิงในการสื่อสารข้อมูล

เมื่อคอมพิวเตอร์ของเรามีการรับส่งข้อมูลกับคอมพิวเตอร์เครื่องอื่นๆ การเชื่อมต่อระหว่างคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกันเป็นระบบเครือข่ายก็เกิดขึ้น อย่างไรก็ตามการเชื่อมต่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์คนละระบบหรือคนละยี่ห้อเป็นสิ่งที่ทำได้จากในยุคแรก ๆ ของการสื่อสารข้อมูล เนื่องจากขาดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล ส่วนมากแต่ละบริษัทผู้จัดทำก็จะมีมาตรฐานของตนเองซึ่งเข้ากันไม่ได้กับรายอื่นๆ ทำให้ผู้ใช้ต้องผูกติดอยู่กับผู้ผลิตแต่ละราย และเป็นขีดจำกัดในการเชื่อมต่อคอมพิวเตอร์คนละชนิดไม่ให้นำส่งข้อมูลกันได้ ระบบคอมพิวเตอร์ในยุคนี้จึงเป็นที่ถือว่าเป็น ระบบปิด ( Closed System ) นั่นเอง

ปัญหานี้ทำให้หน่วยงานกำหนดมาตรฐานสากล คือ International Standards Organization หรือ ISO จัดการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลจากคอมพิวเตอร์ระบบหนึ่งไปยังคอมพิวเตอร์อีกระบบหนึ่งขึ้น จุดมุ่งหมายก็เพื่อเปิดช่องทางให้ข้อมูลที่เก็บอยู่ในระบบคอมพิวเตอร์หนึ่ง ๆ รับส่งไปยังคอมพิวเตอร์ที่เป็นระบบเดียวกันหรือต่างระบบได้อย่างอิสระโดยไม่ขึ้นกับผู้ผลิตอย่างที่เป็นอยู่ในอดีตซึ่งเป็นการทำงานที่เรียกว่า ระบบเปิด ( Open System ) และเรียกโครงสร้างของมาตรฐานการรับส่งข้อมูลนี้ว่า Open System Interconnection หรือ OSI ซึ่งจัดทำขึ้นราวกลางปี ค.ศ. 1970 และใช้อ้างอิงกันมาจนถึงในยุคปัจจุบัน

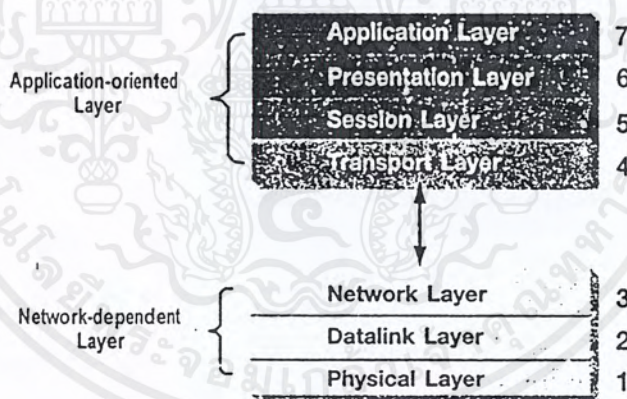


รูปที่ 3.6 มาตรฐานการติดต่อแบบ OSI 7 Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OSI กำหนดให้การสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบหนึ่งแบ่งออกเป็น 7 ชั้นตอนย่อย ๆ ซึ่งคอมพิวเตอร์ทั้งสองระบบจะมีชั้นตอนทั้ง 7 นี้เหมือนกันทั้งสองฝั่ง เราเรียกการสื่อสารข้อมูลนี้ว่า OSI 7 – Layer Reference Model ดังแสดงในรูปที่ 3.6 โดยโครงสร้างการสื่อสารข้อมูลที่กำหนดขึ้นมีคุณสมบัติคือ

แต่ละชั้น ( Layer ) ของการสื่อสารข้อมูล จะเสมือนเชื่อมต่อกับชั้นที่เทียบเท่ากันของคอมพิวเตอร์อีกด้านหนึ่ง ส่วนการเชื่อมต่อกันจริง ๆ จะมีเพียงชั้นที่ 1 หรือ Layer 1 ซึ่งเป็นชั้นล่างสุดเท่านั้นที่มีการรับส่งข้อมูลเกิดขึ้นผ่านสายส่งข้อมูลระหว่างคอมพิวเตอร์ทั้งสองระบบ ส่วนชั้นอื่น ๆ จะไม่ได้เชื่อมต่อกันจริง เพียงแต่ทำงานเสมือนกับว่ามีการติดต่อรับส่งข้อมูลโดยใช้กลไกที่เรียกว่า Peer-to-Peer Protocol และแต่ละชั้นที่ทำหน้าที่รับส่งข้อมูลจะมีการติดต่อรับส่งข้อมูลกับชั้นที่อยู่ติดกับตัวเองเท่านั้นโดยอาศัย Interface Protocol จะติดต่อรับส่งข้อมูลข้ามกระโดดไปชั้นอื่น ๆ ในคอมพิวเตอร์ของตัวเองไม่ได้ โดยผู้ใช้ หรือ User จะติดต่อรับส่งข้อมูลผ่านทางชั้นที่ 7 ซึ่งอยู่ด้านบนสุดของ OSI 7 – Layer Model เท่านั้น



รูปที่ 3.7 การแบ่งกลุ่มของ OSI 7-Layer Model

ในทางปฏิบัตินั้น OSI 7 – Layer Model จะแบ่งออกเป็น 2 กลุ่มใหญ่ ๆ ดังรูปที่ 3.7 คือกลุ่มแรกได้แก่ 4 ชั้นด้านบนคือชั้นที่ 7 , 6 , 5 และ 4 ทำหน้าที่เชื่อมต่อรับส่งข้อมูลระหว่างผู้ใช้กับซอฟต์แวร์โปรแกรมประยุกต์ ให้รับส่งข้อมูลกับฮาร์ดแวร์ที่อยู่ชั้นล่างได้ถูกต้อง เรียกว่า Application Oriented Layers ซึ่งจะเกี่ยวข้องกับซอฟต์แวร์เป็นหลัก โดย 4 ชั้นด้านบนนี้มีก็จะเป็นซอฟต์แวร์ของบริษัทหนึ่งรวมอยู่อย่างเบ็ดเสร็จในโปรแกรมเดียว จะแยกออกจากกันเป็นชั้น ๆ เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้โปรแกรมของบริษัทอื่นได้ลำบาก หรือในบางกรณีก็อาจทำไม่ได้เลย ส่วนกลุ่มที่สองจะเป็นชั้นล่าง ได้แก่ชั้นที่ 3 , 2 และ 1 ทำหน้าที่เกี่ยวกับการรับส่งข้อมูลผ่านสายส่ง และควบคุมการรับส่งข้อมูล ตรวจสอบข้อผิดพลาด รวมทั้งเลือกเส้นทางที่ใช้ในการรับส่งข้อมูลซึ่งจะเกี่ยวข้องกับฮาร์ดแวร์ เป็นหลักเรียกว่า Network Dependent Layers สามารถแยกแต่ละชั้นออกจากกันได้โดยง่าย และใช้ผลิตภัณฑ์ของต่างบริษัทกันในแต่ละชั้นได้อย่างไม่มีปัญหา OSI 7 – Layer Model ที่แบ่งการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบออกเป็น 7 ชั้นนั้น แต่ละชั้นมีชื่อเรียกและทำหน้าที่การทำงานดังนี้ คือ

### ชั้นที่ 7 Application Layer

เป็นชั้นที่อยู่บนสุดของขบวนการรับส่งข้อมูลทำหน้าที่เชื่อมต่อผู้ใช้เข้ากับระบบคอมพิวเตอร์ โดยรับคำสั่งต่าง ๆ จากผู้ใช้ให้ระบบคอมพิวเตอร์แปลความหมาย และทำงานตามคำสั่งที่ได้รับในระดับโปรแกรมประยุกต์ เช่น แปลความหมายของการกดปุ่มบนเมาส์ให้ เป็นต้น ซึ่งการแปลความหมายของการกดปุ่มบนเมาส์ให้เป็นคำสั่งในการคัดลอกข้อมูล หรือดึงข้อมูลมาแสดงผลบนจอภาพ เป็นต้น ซึ่งการแปลออกมาถูกต้องตามกฎที่ใช้ในระบบปฏิบัติการของคอมพิวเตอร์นั้น ๆ ตัวอย่าง เช่น ถ้ามีการคัดลอกข้อมูลเกิดขึ้นในระบบ คำสั่งที่ใช้จะต้องสร้างไฟล์ได้ถูกต้อง มีชื่อไฟล์ยาวไม่เกินจำนวนที่ระบบปฏิบัติการใช้อยู่ และชื่อไฟล์ต้องประกอบด้วยตัวอักษรตามที่กำหนด ไม่มีตัวอักษรห้ามมาตั้งเป็นชื่อไฟล์ เป็นต้น สิ่งต่าง ๆ เหล่านี้จะเกิดขึ้นในชั้นที่ 7 ของการสื่อสารข้อมูล รวมทั้งฟังก์ชันในการเชื่อมต่อรับส่งข้อมูลระหว่างชั้นที่ 7 กับชั้นที่ 6 ด้วย

### ชั้นที่ 6 Presentation Layer

เป็นชั้นที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งว่า การรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร ข้อมูลที่ทำการรับส่งกันในชั้นที่ 6 นี้จะอยู่ในรูปแบบของข้อมูลขั้นสูง ซึ่งอยู่ในรูปแบบของคำสั่งที่มีกฎ ( Syntax ) บังคับอย่างแน่นอน เช่น ในการคัดลอกข้อมูลก็จะมีขั้นตอนย่อยประกอบกัน คือสร้างไฟล์ที่กำหนดขึ้นมาเสียก่อน จากนั้นจึงเปิดไฟล์แล้วทำการรับข้อมูลจากปลายทางมาเก็บลงในไฟล์ที่สร้างขึ้นใหม่โดยเนื้อหาของข้อมูลที่ทำการรับส่งระหว่างกัน ก็คือคำสั่งของขั้นตอนย่อย ๆ ข้างต้นนั่นเอง คำสั่งเหล่านี้จะต้องหมายถึงว่าจะให้ทำอะไรบ้างและถูกต้องตามกฎด้วย นอกจากนี้ในชั้นที่ 6 ยังทำหน้าที่แปลความหมายของคำสั่งที่ได้รับจากชั้นที่ 7 ให้เป็นคำสั่งระดับปฏิบัติการส่งให้ชั้นที่ 5 ต่อไปอีกด้วย

### ชั้นที่ 5 Session Layer

ทำหน้าที่ควบคุมจังหวะในการรับส่งข้อมูลของคอมพิวเตอร์ทั้งสองด้านที่รับส่งแลกเปลี่ยนข้อมูลกันให้มีความสอดคล้องกัน ( Synchronization ) และกำหนดวิธีที่ใช้รับส่งข้อมูล เช่น อาจจะเป็นในลักษณะ Half Duplex หรือ Full Duplex ซึ่งในชั้นที่ 5 นี้จะเป็นชั้นที่ใช้ควบคุมการรับส่งข้อมูล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลในลักษณะดังกล่าว ข้อมูลที่รับส่งกันในชั้นที่ 5 นี้จะอยู่ในรูปของ Dialog หรือประโยคของข้อมูลที่สนทนาโต้ตอบกันระหว่างด้านรับและด้านที่ส่งข้อมูล ไม่ได้มองเป็นคำสั่งอย่างในชั้นที่ 6 เช่น เมื่อผู้รับได้รับข้อมูลส่วนแรกจากผู้ส่ง ก็จะโต้ตอบกลับไปให้ผู้ส่งรู้ว่าได้รับข้อมูลส่วนแรกเรียบร้อยแล้ว และพร้อมที่จะรับข้อมูลส่วนที่ส่งต่อไป คล้ายกับการสนทนาโต้ตอบกันระหว่างผู้รับและผู้ส่งนั่นเอง

#### ชั้นที่ 4 Transport Layer

ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลระดับสูงของชั้นที่ 5 มาเป็นข้อมูลที่รับส่งในระดับฮาร์ดแวร์ เช่น แปลงค่าหรือชื่อของคอมพิวเตอร์ในเครือข่ายให้เป็น Network Address พร้อมทั้งเป็นชั้นที่ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล ให้ข้อมูลมีการไหลลื่นตลอดเส้นทางตามจังหวะที่ควบคุมจากชั้นที่ 5 โดยในชั้นที่ 4 นี้ จะเป็นรอยต่อระหว่างการรับส่งข้อมูลของซอฟต์แวร์กับฮาร์ดแวร์ การรับส่งข้อมูลของระดับสูงจะถูกแยกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลที่ชั้นที่ 4 นี้ และจะไม่มีส่วนใดผูกติดกับฮาร์ดแวร์ที่ใช้รับส่งข้อมูลในระดับล่าง ดังนั้นฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลในระดับล่างลงไปจากชั้นที่ 4 สามารถสับเปลี่ยนและใช้ข้ามไปมากับซอฟต์แวร์รับส่งข้อมูลในระดับสูงที่อยู่ข้างบนได้ง่าย หน้าที่อีกประการหนึ่งของชั้นที่ 4 คือ การควบคุมคุณภาพของการรับส่งข้อมูลให้มีมาตรฐานในระดับที่ตกลงกันของทั้งสองฝ่าย และการตัดข้อมูลออกเป็นส่วนย่อย ๆ ให้เหมาะกับลักษณะการทำงานของฮาร์ดแวร์ที่ใช้ในเน็ตเวิร์ค เช่น หากชั้นที่ 5 ต้องการส่งข้อมูลที่มีความยาวมากเกินกว่าที่ระบบเครือข่ายจะส่งได้ ชั้นที่ 4 ก็จะทำหน้าที่ตัดข้อมูลออกเป็นส่วนย่อย ๆ แล้วส่งไปให้ผู้รับ ข้อมูลที่ได้รับปลายทางก็จะถูกนำมาต่อกันที่ชั้นที่ 4 ของด้านผู้รับ และส่งให้ชั้นที่ 5 ต่อไป

#### ชั้นที่ 3 Network Layer

ทำหน้าที่เชื่อมต่อคอมพิวเตอร์ของด้านรับและด้านส่งเข้าหากันผ่านระบบเครือข่าย พร้อมทั้งเลือกหรือกำหนดเส้นทางที่จะใช้ในการรับส่งข้อมูลระหว่างกัน และส่งผ่านข้อมูลที่ได้รับไปยังอุปกรณ์ในเครือข่ายต่าง ๆ จนกระทั่งถึงปลายทาง ในชั้นที่ 3 นี้ข้อมูลที่รับส่งกันจะอยู่ในรูปแบบของกลุ่มข้อมูลที่เรียกว่า Packet หรือ Frame ข้อมูลที่ชั้นที่ 4 , 5 , 6 และ 7 มองเห็นเป็นคำสั่งและ Dialog ต่าง ๆ นั้น จะถูกแปลงและผนึกรวมอยู่ในรูปของ Packet หรือ Frame ที่มีเพียงแอดเดรสของผู้รับ , ผู้ส่ง , ลำดับการรับส่งและส่วนของข้อมูลเท่านั้น ตัวเนื้อหาของข้อมูลจะไม่มีผลใด ๆ ในการรับส่งข้อมูลเลย ไม่ว่าข้อมูลในระดับสูงจะเป็น วิดีโอ , ภาพ , เสียง หรือข้อมูลอื่นใดก็ตาม แต่ในชั้นที่ 3 จะมองข้อมูลทั้งหมดเป็น Packet หรือ Frame เท่านั้น หน้าที่อีกประการหนึ่งของชั้นที่ 3 นี้คือการทำ Call Setup หรือเรียกติดต่อกับคอมพิวเตอร์ปลายทางก่อนการรับส่งข้อมูล แลการทำ Call

Clearing หรือยกเลิกการติดต่อเมื่อการรับส่งข้อมูลจบลงแล้ว ในกรณีที่มีการรับส่งข้อมูลนั้นต้องมีการติดต่อกันก่อน

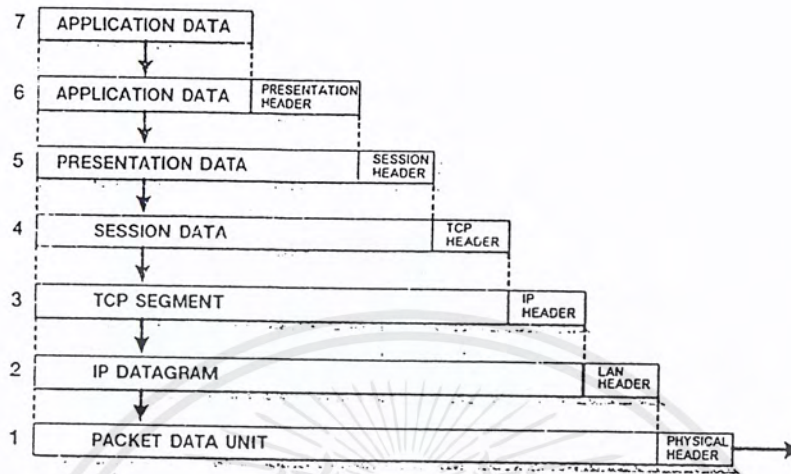
## ชั้นที่ 2 Datalink Layer

เป็นชั้นที่ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลในระดับฮาร์ดแวร์ โดยเมื่อมีการสั่งให้รับข้อมูลจากในชั้นที่ 3 ลงมา ชั้นที่ 2 จะทำหน้าที่แปลคำสั่งนั้นให้เป็นคำสั่งควบคุมฮาร์ดแวร์ที่ใช้รับส่งข้อมูล ทำการตรวจสอบข้อผิดพลาดในการรับส่งข้อมูลของระดับฮาร์ดแวร์ และแก้ไขข้อผิดพลาดที่ตรวจพบนั้น ข้อมูลที่อยู่ในชั้นที่ 2 นี้จะอยู่ในรูปของ Frame คือกลุ่มของข้อมูลที่มีรูปร่างตามข้อบังคับของฮาร์ดแวร์ที่ใช้ในการรับส่งข้อมูล เช่น ถ้าฮาร์ดแวร์ที่ใช้เป็น Ethernet LAN ข้อมูลก็จะมีรูปร่างของ Frame ตามที่ระบุไว้ในมาตรฐานของ Ethernet หากว่าฮาร์ดแวร์ที่ใช้รับส่งข้อมูลเป็นชนิดอื่น เช่น Token Ring LAN

## ชั้นที่ 1 Physical Layer

เป็นชั้นล่างสุดของชั้นตอนในการรับส่งข้อมูลของ OSI 7 – Layer Reference Model ซึ่งเป็นชั้นเดียวที่มีการเชื่อมต่อกันทางกายภาพระหว่างคอมพิวเตอร์สองระบบที่ทำการรับส่งข้อมูลกัน ในชั้นที่ 1 นี้จะกำหนดคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้งสองระบบ เช่น สายที่ใช้รับส่งข้อมูลจะเป็นแบบไหน , ความเร็วในการรับส่งข้อมูลเป็นเท่าไร เป็นต้น ข้อมูลในชั้นที่ 1 นี้จะมองเห็นเป็นการรับส่งข้อมูลที่ระบิตเรียงกันต่อกันไป โดยไม่มีการพิจารณาเรื่องความหมายของข้อมูลเลย การรับส่งจะส่งข้อมูล “0” หรือ “1” ไปให้คอมพิวเตอร์ด้านรับส่งข้อมูลในระดับฮาร์ดแวร์เท่านั้น หากการรับส่งข้อมูลมีปัญหาเนื่องจากฮาร์ดแวร์จะเป็นหน้าที่ของชั้นที่ 1 นี้ในการตรวจสอบและแจ้งข้อผิดพลาดนั้นให้ชั้นอื่น ๆ ที่อยู่เหนือขึ้นไปทราบ

ในการรับส่งข้อมูลใน OSI 7 – Layer Model นั้น ข้อมูลจากชั้นบนสุด คือชั้นที่ 7 เมื่อถูกส่งลงไปชั้นถัดลงไป ข้อมูลเดิมก็จะถูกผนวกเข้ากับข้อมูลที่ใช้ควบคุมของแต่ละชั้นซ้อน ๆ กันเป็นลำดับเท่ากับจำนวนชั้นที่ผ่านลงไป ตัวอย่างเช่น Application Data เมื่อถูกส่งลงไปยังชั้นถัดไปก็จะถูกผนวกด้วย Application Header และทั้ง Application Header และ Application Data จะรวมกันเป็นข้อมูลของชั้นที่อยู่ถัดลงไปอีกเรื่อย ๆ เป็นเช่นนี้จนกระทั่งถึงชั้นล่างสุดซึ่งเป็น Physical Layer ซึ่งเมื่อข้อมูลถูกส่งไปถึงปลายทาง ข้อมูลที่ได้รับจะถูกแยก Header ที่เพิ่มเข้ามานี้ออกทีละชั้น ซึ่งเป็นขบวนการย้อนกลับกับด้านส่ง จนกระทั่งถึงชั้นบนสุด จึงจะเป็นข้อมูลของ Application Data ให้ผู้รับตามต้องการ ดังแสดงในรูปที่ 3.8



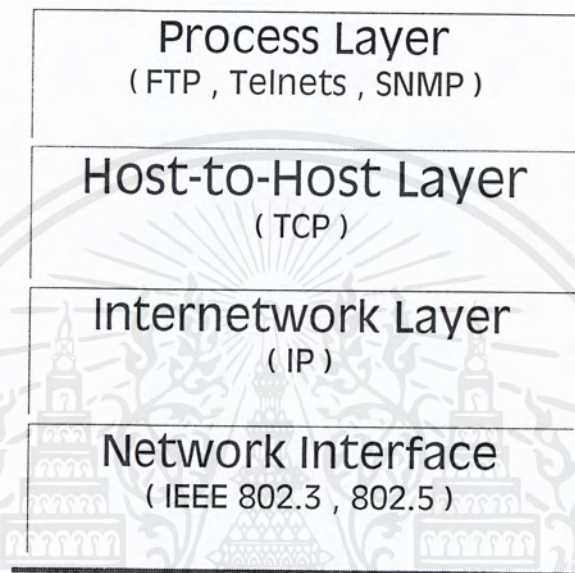
รูปที่ 3.8 การรับส่งข้อมูลแต่ละชั้นของ TCP/IP ใน OSI 7 Layer Model ซึ่งทั้งข้อมูลและ TCP Header จะถูกผนึกรวมกันเป็นข้อมูลที่เรียกว่า IP Datagram

### 3.3 โพรโทคอล TCP/IP

โพรโทคอล TCP/IP เป็นโพรโทคอลที่สำคัญที่มีการใช้กันอย่างแพร่หลายตามการขยายตัวของอินเทอร์เน็ตและอินทราเน็ต โดยความจริงแล้ว TCP/IP เป็นกลุ่มของโพรโทคอลหลายตัวที่ประกอบกันเป็นชุดให้ใช้งาน โดยมีชื่อเรียกว่า Transmission Control Protocol / Internet Protocol ตัวอย่างของโพรโทคอลในกลุ่มของ TCP/IP ที่เราพบและใช้งานบ่อยๆ โดยผ่านทางแอปพลิเคชันต่างๆ เช่น internet protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Simple Mail Transfer Protocol (SMTP) และ Domain Name System (DNS) เป็นต้น โดยโพรโทคอลที่มีบทบาทสำคัญในการทำงานในเครือข่ายอินเทอร์เน็ต คือ IP (Internet Protocol) เนื่องจากเมื่อโพรโทคอลอื่นๆต้องการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตนั้น จำเป็นต้องอาศัยการผนึกข้อมูล (Encapsulation) ไปกับโพรโทคอล IP ที่มีกลไกการระบุเส้นทาง (Route Service) ผ่าน Gateway เพื่อนำข้อมูลไปยังเครือข่ายหรือเครื่องปลายทางที่ต้องการ โดยกลไกการระบุเส้นทางจะทำงานที่โพรโทคอล IP เท่านั้น หรืออาจเรียกว่า IP เป็นโพรโทคอลที่มีความสามารถในการระบุเส้นทางของการส่งผ่านข้อมูลได้ (Routable)

### 3.3.1 สถาปัตยกรรมของ TCP/IP ( TCP/IP Reference Model )

TCP / IP จะมีการแบ่งจำนวนชั้นตอนที่ไว้รับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองระบบออกเป็น 4 ชั้นเท่านั้น หรือเรียกว่าเป็น TCP / IP Stack โดยมีชื่อเรียกแตกต่างกันดังแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 ส่วนประกอบของ TCP/IP Stack

- **Process Layer** จะเป็น Application Protocol ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้และให้บริการต่าง ๆ เช่น FTP , Telnet เป็นต้น

- **Host-to-Host Layer** จะเป็น TCP หรือ UDP ที่ทำหน้าที่คล้ายกับชั้นที่ 4 ของ OSI 7 — Layer Model คือ ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล และตัดข้อมูลออกเป็นส่วนย่อยให้เหมาะกับเครือข่ายที่ใช้รับส่งข้อมูล รวมทั้งประกอบข้อมูลส่วนย่อย ๆ นี้เข้าด้วยกันเมื่อถึงปลายทาง

- **Internetwork Layer** ได้แก่ส่วนของโปรโตคอล IP ซึ่งทำหน้าที่คล้ายกับชั้นที่ 3 ของ OSI 7 — Layer Model คือเชื่อมต่อกับคอมพิวเตอร์เข้าระบบเครือข่ายที่อยู่ชั้นล่างลงไป และทำหน้าที่เลือกเส้นทางการรับส่งข้อมูลในลักษณะเครือข่ายต่าง ๆ จนไปถึงผู้รับข้อมูล ในชั้นนี้จะจัดการกับกลุ่มข้อมูลในลักษณะที่เรียกว่า Frame ในรูปแบบของ TCP / IP ที่เรารู้จักกันนั่นเอง

- **Network Interface** คือชั้นที่ควบคุมฮาร์ดแวร์รับส่งข้อมูลผ่านเครือข่าย ซึ่งเทียบได้กับชั้นที่ 1 และ 2 ของ OSI 7 — Layer Model ในชั้นนี้จะทำหน้าที่เชื่อมต่อกับฮาร์ดแวร์ และควบคุมการ

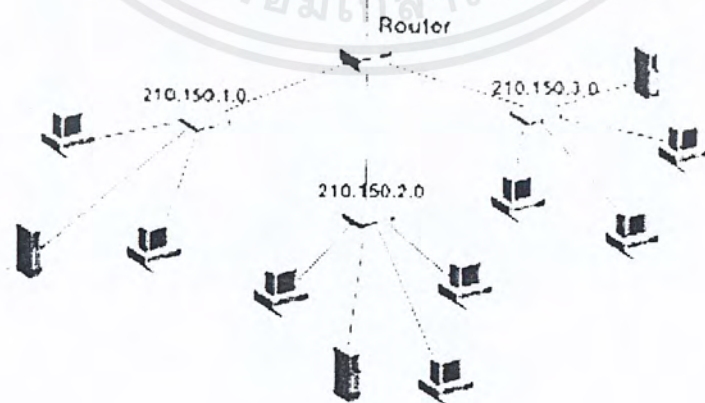
รับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย ซึ่งที่ใช้กันอยู่จะเป็นตามมาตรฐานของ IEEE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้ใช้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 IP ADDRESS

IP Address เป็นหมายเลขที่ถูกกำหนดขึ้นมาเป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่างๆที่เชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต โดยอุปกรณ์แต่ละชิ้นจะมี IP Address ไม่ซ้ำกัน แต่อย่างไรก็ตามหมายเลขนี้จะไม่ถูกผูกติดกับตัวอุปกรณ์ สามารถกำหนดหรือแก้ไขเปลี่ยนแปลงได้ ทั้งนี้เนื่องจากการกำหนดด้วยซอฟต์แวร์ แตกต่างกับหมายเลข MAC Address ( Media Access Control Address ) ซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์ที่ต่ออยู่ในเครือข่าย ค่า MAC Address จะถูกกำหนดมาจากบริษัทผู้ผลิตอุปกรณ์ตั้งแต่ต้น โดยจะมีค่าไม่ซ้ำกันและไม่สามารถแก้ไขได้ ค่า MAC Address เป็นการระบุค่าอ้างอิงของอุปกรณ์ในระดับล่างสุด ( Physical Layer ) ของกลไกการรับส่งข้อมูลผ่านเครือข่าย

การทำงานของ Internet Protocol (IP) จำเป็นต้องอาศัย IP Address ในการระบุและอ้างถึงอุปกรณ์ต่างๆที่ต่ออยู่ในเครือข่าย หมายเลข IP Address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วนและคั่นด้วยเครื่องหมายจุด ( . ) ดังนั้นตัวเลขแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (  $2^8$  ) โดยค่าของ IP Address จะถูกกำหนดออกเป็น 2 ความหมายคือ ค่าของหมายเลขอุปกรณ์เครือข่าย ( Host Address ) และค่าของหมายเลขเครือข่าย ( Network Address ) ตัวอย่างเช่น มีเครื่องเว็บเซิร์ฟเวอร์เชื่อมต่ออยู่ในเครือข่าย 2 เครื่อง โดยแต่ละเครื่องมี IP Address ประจำตัวคือ 205.144.78.2 และ 205.144.78.3 จะเห็นได้ว่าเครื่องทั้งสองมีค่า 205.144.78 เหมือนกันคือมีค่าของหมายเลขเครือข่ายเหมือนกัน แสดงว่าต่ออยู่บนเครือข่ายเดียวกัน บนสายสัญญาณที่เชื่อมโยงเส้นเดียวกันแต่มีหมายเลขประจำเครื่องที่แตกต่างกันคือ 2 และ 3 ตามลำดับ

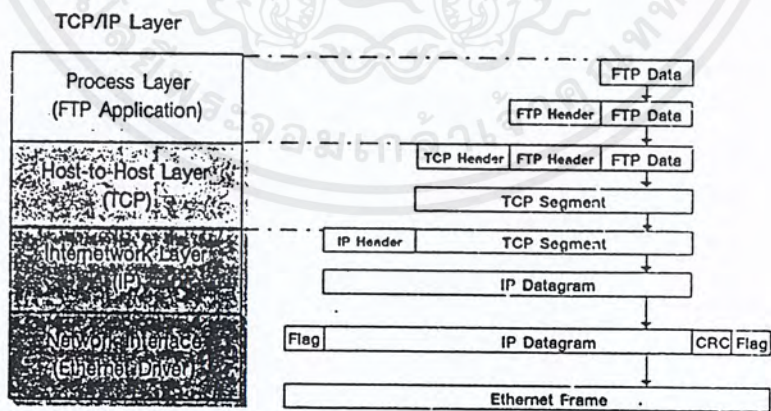


รูปที่ 3.10 เครื่องข่าย 3 เครื่องข่ายที่แต่ละเครื่องข่ายมี IP Address ของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตนั้น ตัวข้อมูลที่จะส่งจะถูกทำให้มีขนาดเล็กลง โดยแบ่งออกเป็นส่วนย่อยๆ เรียกว่า Data Packet หรือ Datagram ซึ่งการแบ่งข้อมูลเป็นส่วนย่อยนี้ จะช่วยให้เครือข่ายสามารถส่งข้อมูลได้อย่างราบรื่น ไม่ติดขัด อีกทั้งยังทำให้สามารถทำการตรวจทาน ความถูกต้องของข้อมูลที่ปลายทางและแก้ไขข้อผิดพลาดได้อย่างมีประสิทธิภาพด้วย และที่สำคัญ ช่องสัญญาณที่ใช้ในการเชื่อมต่อในระบบเครือข่ายมีจำกัดและเป็นสิ่งที่มีต้องใช้ร่วมกัน เมื่อมีอุปกรณ์ ใดส่งข้อมูล อุปกรณ์อื่นก็ต้องรอให้การส่งข้อมูลนั้นเสร็จสิ้นก่อน ดังนั้นการแบ่งข้อมูลเป็นส่วน ย่อยจะเป็นการแบ่งเวลาให้กับผู้ใช้งานรายอื่นๆด้วย

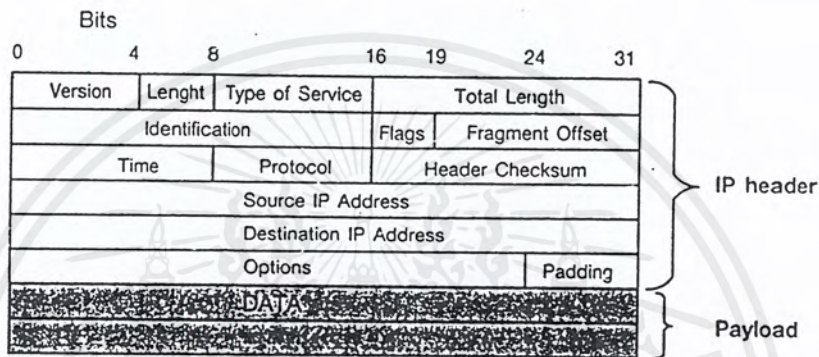
โดยตัวของข้อมูลที่ถูกแยกออกเป็น Data Packet หรือ Datagram จะมีลักษณะเป็นข้อมูล แบบต่อเนื่อง (Stream Byte) คือมีการลำดับก่อนหลังของข้อมูล เพื่อให้ประกอบข้อมูลย่อยคืนสู่ สภาพเดิมได้อย่างถูกต้อง และมีรูปแบบที่แน่นอน คือ Datagram จะประกอบด้วยส่วนของ Header และส่วนของตัวข้อมูล (Body) โดยในส่วนหัวของข้อมูลจะประกอบด้วยข้อมูลต่างๆที่ระบุที่อยู่ ปลายทางที่จะต้องส่งข้อมูลไป , หมายเลขเส้นทางที่ส่งข้อมูลมา , ค่าความยาวของ Datagram นี้ และ ข้อมูลอื่นๆ โดย Datagram นี้จะถูกผนึก (Encapsulation) เข้าไปกับโปรโตคอล IP เรียกรวมกันเป็น IP Datagram นับว่าเป็นกลไกสำคัญของการส่งข้อมูลด้วย TCP/IP โดยขบวนการที่ใช้จะมีขั้นตอน คร่าวๆ ดังรูปที่ 3.11 ซึ่งเป็นตัวอย่างของการเรียกใช้โปรแกรมแอปพลิเคชัน FTP ( File Transfer Protocol ) ร่วมกับ TCP/IP ในการส่งผ่านข้อมูลไปในเครือข่ายแบบ Ethernet



รูปที่ 3.11 ตัวอย่างการทำ Encapsulation ของข้อมูล FTP เทียบกับ TCP/IP Layer

หลังจากขบวนการ Encapsulation นั้น เราจะพบว่าข้อมูลในรูปของ Datagram จะถูกแปลง เป็น Ethernet Frame หรือเฟรมข้อมูลในรูปแบบอื่นๆตามลักษณะการเชื่อมต่อทางกายภาพ เช่น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ethernet หรือ Token-Ring เป็นต้น และเพื่อให้ข้อมูลสามารถส่งออกสู่เครือข่ายและข้ามเครือข่ายไปสู่อินเทอร์เน็ตได้ ตัวข้อมูลที่ถูกแปลงมาเป็น IP Datagram นี้จะประกอบด้วย 2 ส่วน IP Header ที่มีขนาด 32 ไบต์ และส่วนเนื้อข้อมูลที่เรียกว่า Payload โดยจะมีขนาดขึ้นกับข้อมูล ทำให้ IP Datagram มีขนาดไม่แน่นอน และมีลักษณะตามรูปที่ 3.12



ส่วนของ IP Header มีการแบ่งย่อยเพื่อระบุพารามิเตอร์ในการใช้งานต่างๆ ดังนี้

- Version มีขนาด 4 บิต กำหนดเป็น 4 เมื่อใช้มาตรฐาน IPv4 และเป็น 6 เมื่อใช้เป็น IPv6 (มาตรฐานที่จะใช้งานในอนาคต)
- Length มีขนาด 4 บิต กำหนดความยาวของ IP Header
- Type of Service มีขนาด 8 บิต บอกให้ทราบว่าจะดำเนินการกับข้อมูลนี้อย่างไร เช่น Low delay , High Throughout เป็นต้น
- Total Length มีขนาด 16 บิต กำหนดความยาวสุทธิของ IP Datagram เป็นจำนวนไบต์ ซึ่งมีค่าไม่เกิน  $2^{16}$  หรือ 65,535 ไบต์
- Identification เป็นข้อมูลที่บอกให้ทราบว่า IP Datagram นี้มาจากที่ใด
- Flag และ Fragment Offset เป็นส่วนข้อมูลที่ใช้ระบุการแยกและรวมข้อมูล เพื่อให้ข้อมูลที่ถูกแยกออกเป็นข้อมูลย่อย ( Fragment ) สามารถกลับมารวมกันใหม่ตามลำดับได้อย่างถูกต้อง
- Time เป็นข้อมูลแสดงจำนวนเวลามากที่สุดของ IP Datagram นี้สามารถส่งผ่านเครือข่ายไปยังปลายทางได้ มีหน่วยเป็นวินาที ปกติเป็น 32 โดยเมื่อ IP Datagram ถูกส่งผ่าน Router ตัวหนึ่งก็จะถูกลดค่า Time ลงไปหนึ่ง ทำให้สามารถใช้ค่านี้นับจำนวนเครือข่ายที่ IP Datagram นี้ถูกส่งผ่านมาได้ เรียกว่า Hop Count

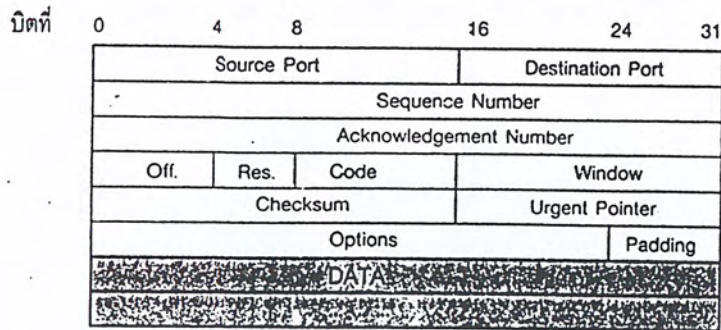
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Protocol เป็นการระบุโปรโตคอลที่ทำงานในชั้น ( layer ) ข้างบนที่ผ่นึกลงมาใน IP Datagram
- Header Checksum เป็นส่วนของข้อมูลที่ใช้ตรวจสอบความถูกต้องของข้อมูลในส่วนของ IP Header เท่านั้น โดยเมื่อเกิดความผิดพลาดของข้อมูล IP Datagram นั้นจะถูกยกเลิกไปไม่นำมาใช้งาน
- Source / Destination IP Address เป็นส่วนที่เก็บหมายเลข IP Address ของต้นทางและปลายทางตามลำดับ
- Option มีขนาดไม่แน่นอนใช้เก็บค่าพารามิเตอร์ส่วนประกอบปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการใช้งาน
- Padding ทำหน้าที่เป็นส่วนเติมข้อมูลให้ IP Header ครบ 32 ไบต์

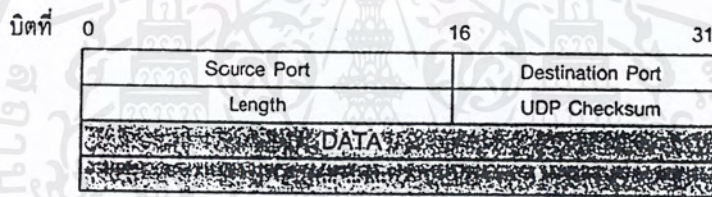
### 3.3.3 กลไกการทำงานของโปรโตคอล TCP

โปรโตคอล TCP ( Transmission Control Protocol ) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ Stream Oriented Protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆก่อน ( Datagram ) แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูลย่อยประกอบกลับเป็นข้อมูลทั้งหมดได้ และจะแยกข้อมูลที่ไม่ถูกต้องออก การติดต่อระหว่างกันต้องเป็นแบบ Connection-Oriented คือ ต้องมีการสร้างช่องทางการติดต่อ ( Session ) ทั้ง 2 ด้านเสียก่อน แล้วจึงทำการรับส่งข้อมูลไปได้พร้อมกัน ( Full Duplex ) คล้ายกับการใช้โทรศัพท์

เมื่อมีแอปพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้นของ Process Layer ติดต่อไปจากนั้นจะมีการสร้างช่องส่งข้อมูลผ่านพอร์ต ( Port ) ที่กำหนดเพื่อส่งข้อมูลไปยังโปรโตคอล TCP ที่ทำงานอยู่ในชั้น Host-to-Host Layer ในระหว่างการรับส่งข้อมูลนี้โปรโตคอล TCP จะเพิ่มกระบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณตรวจสอบทานข้อมูล ( Acknowledgement ) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น

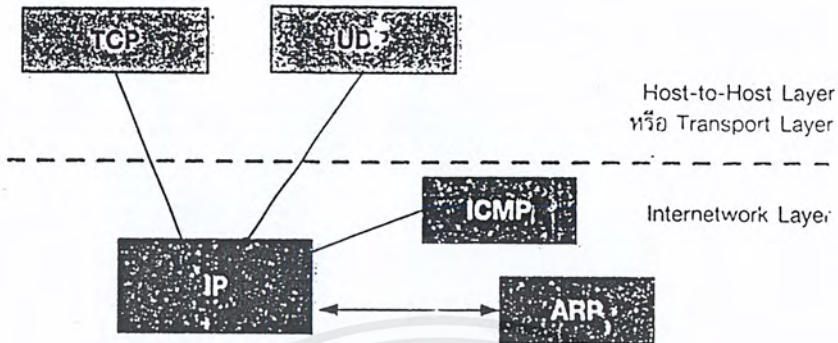


รูปที่ 3.13 รูปแบบของ TCP Packet จะเห็นว่ามี Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล

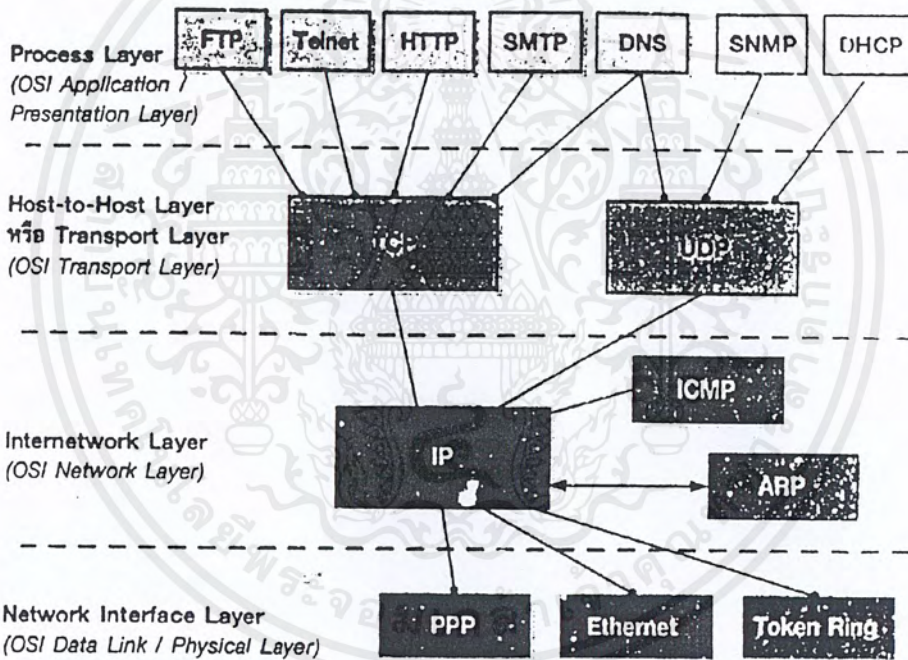


รูปที่ 3.14 รูปแบบของ UDP Packet จะเห็นว่ามีส่วน Header ที่มีขนาดเล็กมาก จึงใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP

โดยนอกจากโปรโตคอล TCP แล้วยังมีโปรโตคอล UDP ( User Datagram Protocol ) ที่ทำงานในลักษณะที่แตกต่างจากโปรโตคอล TCP คือจะทำการติดต่อสื่อสารแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน ( Connectionless-Oriented ) และไม่มีการตรวจสอบความถูกต้องครบถ้วนของข้อมูลในการรับส่งด้วย จึงทำให้โปรโตคอล UDP ใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP เนื่องจากการทำงานที่ต่างกันทำให้โปรโตคอลทั้งสองแบบมีแอปพลิเคชันที่เรียกใช้งานต่างกันดังรูปที่ 3.15



รูปที่ 3.15 แอปพลิเคชันต่างๆใน Transport Layer ที่เรียกใช้โปรโตคอล TCP และ UDP



รูปที่ 3.16 โปรโตคอล TCP และ UDP อาศัยโปรโตคอล IP ในชั้น Internetwork Layer ในการส่งผ่านข้อมูล และแสดงความสัมพันธ์ของโปรโตคอล IP , ICMP และ ARS

### 3.3.4 กลไกการทำงานของโปรโตคอล IP

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งข้ามไปยังเครือข่ายใดๆได้อย่างถูกต้อง โดยจะทำงานแบบ Packet Switching คือมีการส่งผ่านข้อมูลผ่านอุปกรณ์ Router หรือ Gateway ในระบบเครือข่ายจนกว่าจะถึงปลายทางที่กำหนดโดย IP Address เมื่อข้อมูลไปถึงปลายทางที่กำหนดโดย IP Address แล้วจะมีกลไกการแปลงหมายเลข IP Address เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เป็นหมายเลขประจำตัวอุปกรณ์ ( MAC Address ) ที่ถูกต้องอีกทีหนึ่งด้วยโปรโตคอล ARP ( Address Resolution Protocol ) นอกจากนี้ยังมีการใช้งานโปรโตคอล ICMP ( Internet Control Message Protocol ) ควบคู่ไปด้วย โดยจะมีหน้าที่ในการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งข้อความที่โปรโตคอล ICMP ส่งนั้นแบ่งออกได้เป็น 2 แบบ คือ ICMP Error Message หรือข้อความแจ้งความผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูลเพิ่มเติม ดังรูปที่ 3.16 แสดงความสัมพันธ์ของโปรโตคอลที่ใช้ร่วมกับ IP

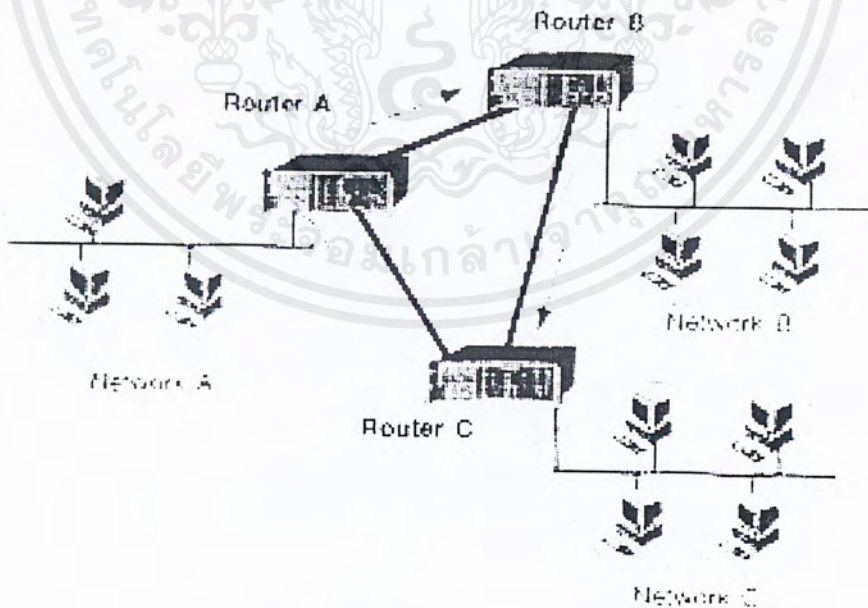
ในการส่งผ่านข้อมูล หรือ IP Datagram ไปยังเครือข่ายอินเทอร์เน็ตนั้น โปรโตคอล IP จะทำการพิจารณาว่าปลายทางในการส่งนั้นเป็นภายในเครือข่ายของตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยตรวจสอบค่าหมายเลขเครือข่าย ( Network Address ) ถ้าหมายเลขเครือข่ายของ IP Address เหมือนกับของต้นทางหรือไม่ ถ้าตรงกันแสดงว่าเป็นการส่งข้อมูลภายในเครือข่ายเดียวกัน แต่ถ้าต่างกันแสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน ซึ่งจะมีกลไกที่ต่างกัน ดังนี้

- การส่งข้อมูลภายในเครือข่ายเดียวกัน มีกลไกดังต่อไปนี้
  1. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP เพื่อแปลงหมายเลข IP Address ปลายทางให้เป็นค่าหมายเลขประจำตัวของอุปกรณ์ เช่น MAC Address เป็นต้น
  2. เมื่อโปรโตคอล IP ได้รับหมายเลขประจำตัวอุปกรณ์แล้ว ก็จะทำการส่งข้อมูลนั้นไปยังอุปกรณ์ที่ระบุไว้
- การส่งข้อมูลข้ามเครือข่าย มีกลไกดังต่อไปนี้
  1. โปรโตคอล IP ตรวจสอบพบว่าหมายเลข IP Address ปลายทางอยู่คนละเครือข่ายกัน โปรโตคอล IP ก็จะทำการอ่านค่า IP Address ของ Router เพื่อเตรียมส่งข้อมูลไปที่ Router แทน ( กำหนดเป็น Default Router )
  2. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP เพื่อแปลงหมายเลข IP Address ปลายทางให้เป็นค่าหมายเลขประจำตัวของอุปกรณ์
  3. โปรโตคอล IP ส่งข้อมูล IP Datagram ไปยัง Router ที่กำหนดไว้ จากนั้น Router ก็จะทำการส่งผ่านข้อมูลข้ามเครือข่ายต่อไป

### 3.3.5 การส่งข้อมูลโดยอาศัย Router และ Routing Algorithm

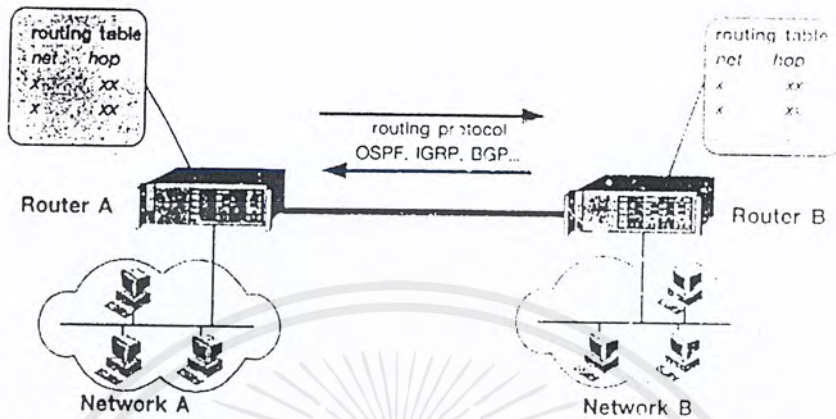
อุปกรณ์ Router ทำหน้าที่ในการเชื่อมโยงเครือข่ายที่อยู่ห่างไกลเข้าด้วยกัน ไม่ว่าเครือข่ายนั้นจะต่างกันหรือเหมือนกันทางด้านกายภาพก็ตาม การเชื่อมโยงนี้มีได้หลายลักษณะ โดยส่วนใหญ่แล้วจะทำการเชื่อมโยงกันโดยผ่านบริการ Wide Area Network Services ( WAN Services ) เช่น ISDN ( Integrated Services Digital Network ) , Frame Relay , Point-to-Point Leased Circuit , X.25 เป็นต้น โดยมีบทบาทการทำงานอย่างคร่าวๆดังต่อไปนี้

- เมื่ออุปกรณ์ Router ได้รับ IP Datagram มาจะทำการตรวจสอบความถูกต้องของข้อมูลก่อน ด้วยขบวนการ Checksum จนกว่าจะได้ข้อมูลที่ถูกต้อง
- ต่อมา Router จะทำการพิจารณาเส้นทางในการส่งผ่านข้อมูล ด้วยขบวนการ Router Algorithm
- เมื่อทราบเส้นทางที่จะทำการส่งข้อมูลแล้ว ก็จะทำให้นำข้อมูลใส่ลงในลำดับ ( Queue ) เพื่อรอการส่งต่อไป
- นอกจากนี้ Router จะทำการรับส่งข้อมูล Routing Table ระหว่าง Router ด้วยกันเอง เพื่อเป็นการแลกเปลี่ยนข้อมูลเกี่ยวกับเส้นทางในการส่งผ่านข้อมูลอยู่เป็นระยะๆตามเวลาที่กำหนด โดยในการแลกเปลี่ยนนั้นจะอาศัยโปรโตคอลที่เรียกว่า Routing Protocol



รูปที่ 3.17 การส่งผ่านข้อมูลผ่านเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 หน้าทีของ Router ในการพิจารณาเส้นทางและการแลกเปลี่ยนข้อมูล Routing Table ระหว่าง Router ด้วยกัน

ขบวนการพิจารณาเส้นทางของการส่งผ่านข้อมูล หรือที่เรียกว่า Routing Algorithm มีบาทในการปรับปรุงข้อมูลใน Routing Table ให้ใหม่อยู่เสมอ ตามสถานะที่มีการเปลี่ยนแปลงไปของเครือข่าย รวมทั้งยังมีบทบาทในการพิจารณาว่า การส่งข้อมูลนั้นจะเป็นไปอย่างมีประสิทธิภาพตามเงื่อนไขที่กำหนดไว้หรือไม่ ช่วยให้ไม่พบกับปัญหาการส่งข้อมูลไม่ถึงปลายทาง ดังนั้นในการติดต่อระหว่าง Router รวมถึงวิธีคิดพิจารณาเส้นทางของการส่งผ่านข้อมูลที่สามารถทำได้หลายแบบ ซึ่งจะกำหนดได้เป็นโปรโตคอลสำหรับการพิจารณาเส้นทางของการส่งผ่านข้อมูลหรือ Routing Protocol ซึ่งจะทำงานบนโปรโตคอล IP อีกทีหนึ่ง โดยโปรโตคอล IP สามารถผนึกข้อมูลหรือโปรโตคอลที่จะระบุเส้นทางส่งผ่านข้อมูลเข้าไปได้

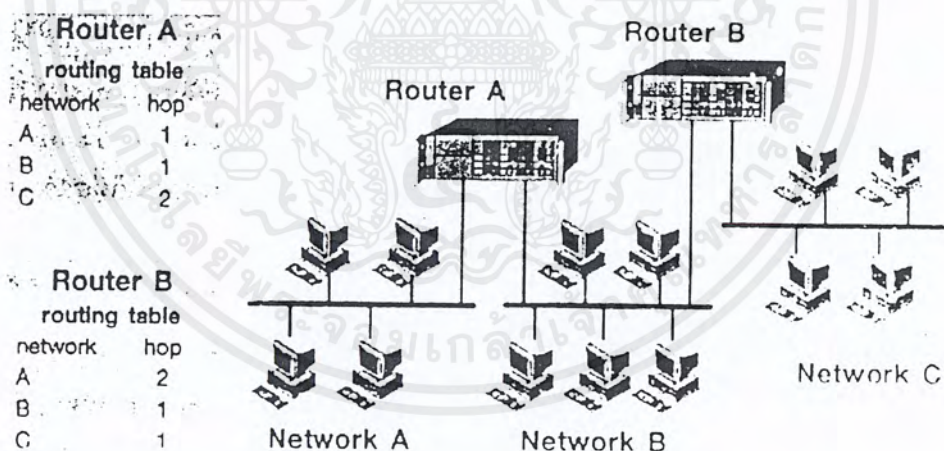
Routing Algorithm มีหลายประเภท ซึ่งแต่ละประเภทมีวิธีการคำนวณพิจารณาหาเส้นทางและวิธีปรับปรุงข้อมูลระหว่าง Router ด้วยกันที่แตกต่างกันด้วย ดังนั้นถ้าต้องการให้สามารถเชื่อมโยงเครือข่ายระหว่างกันได้ และสามารถส่งข้อมูลข้ามเครือข่ายได้อย่างถูกต้องแล้วละก็ อุปกรณ์ Router ที่ใช้เชื่อมต่อกันนั้นจำเป็นต้องใช้งาน Routing Protocol ที่เหมือนกันด้วย โดยสามารถแบ่งประเภทของ Routing Algorithm ได้เป็น 2 ประเภทใหญ่ๆ คือ

**1. Interior Routing Protocol** มักใช้งานกับเครือข่ายขนาดเล็กที่มีเครือข่ายย่อยเชื่อมต่อเป็นสมาชิกอยู่ โดยใช้เป็นเส้นทางติดต่อแลกเปลี่ยนข้อมูลภายในกลุ่มสมาชิกด้วยกัน เช่น บริษัทผู้ให้บริการอินเทอร์เน็ต (ISP) จะมีการเชื่อมต่อเครือข่ายย่อยๆ คือเครือข่ายของลูกค้าแบบองค์กรที่เชื่อมต่อเครือข่ายของตนเข้ากับเครือข่ายของ ISP ทำให้เครือข่ายของลูกค้าอยู่ในเครือข่ายอินเทอร์เน็ตได้ โดยในการติดต่อส่งผ่านข้อมูลอุปกรณ์ Router จะทำการแลกเปลี่ยนข้อมูลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Routing Table เพื่อให้ทราบว่าเส้นทางใดเป็นเส้นทางในการส่งผ่านข้อมูล ตัว Router ของบริษัทผู้ให้บริการ ( ISP ) จะทำหน้าที่เป็น Router หลักอยู่ในระบบที่เรียกว่า Autonomous System ( AS ) ที่ จะทำหน้าที่เชื่อมโยงไปสู่ระบบภายนอกและออกสู่ระบบอินเทอร์เน็ตต่อไป โดย AS นี้จะทำการ เชื่อมโยงระหว่าง Router หลักของผู้ให้บริการเข้าด้วยกัน Router หลักของ ISP จะมีหมายเลข ประจำตัวหรือ AS Number ที่ไม่ซ้ำกัน และมีโปรโตคอลบางชนิดที่ใช้งานค่าหมายเลข AS Number ในการแลกเปลี่ยนสถานะของเครือข่ายและข้อมูล Routing Table โดยมีแบ่งย่อยอีกได้เป็นอีกหลาย แบบ ตัวอย่างเช่น

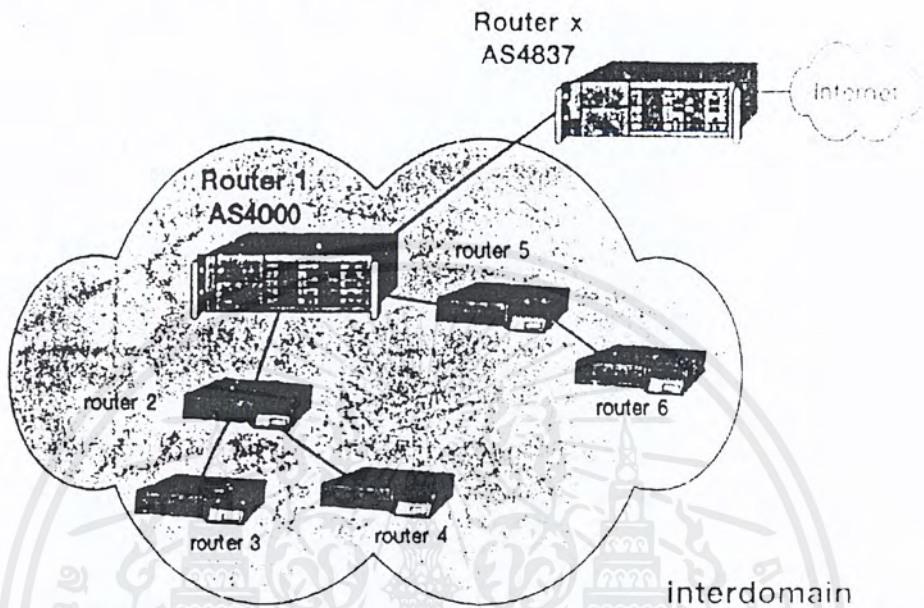
**1.1 ) Distance-Vector Routing Protocol** มีลักษณะที่สำคัญคือ ในแต่ละ Router จะมีข้อมูล Routing Table เอาไว้พิจารณาเส้นทางในการส่งข้อมูล โดยพิจารณาจากระยะทางที่ข้อมูลจะไปถึง ปลายทางเป็นหลักโดยจะนับเป็นจำนวน Hop ดังรูปที่ 3.19 โดยการส่งในลักษณะของ Distance-Vector Routing Protocol จะเลือกหาเส้นทางที่ดีที่สุดและมีการคำนวณตาม Routing Algorithm เพื่อให้ได้ผลลัพธ์ออกมา ซึ่งมักจะเป็นเส้นทางที่มีจำนวน Hop น้อยที่สุด อุปกรณ์ Router ที่เชื่อมต่อกัน จะทำการปรับปรุงข้อมูลใน Routing Table อยู่เป็นระยะๆ ด้วยการ Broadcast ข้อมูลทั้งหมดใน Routing Table ไปในเครือข่ายตามระยะเวลาที่กำหนดไว้



รูปที่ 3.19 Distance-Vector Routing Protocol

**1.2 ) Link-State Routing Protocol** มีกลไกการทำงาน คือ อุปกรณ์ Router จะทำการ Broadcast ข้อมูลการเชื่อมต่อของเครือข่ายตนเองไปให้ตัวอุปกรณ์อื่นๆทราบ ข้อมูลนี้เรียกว่า Link-State ซึ่งเกิดจากการคำนวณโดยตัวอุปกรณ์ที่ทำการคำนวณ จะทำการคำนวณค่าการเชื่อมต่อโดยการพิจารณาให้ Router ของตนเองเป็นหลักในการสร้าง Routing Table ขึ้นมา ดังนั้นข้อมูล Link-

State ที่ส่งออกไปในเครือข่ายจะเป็นข้อมูลที่บอกเส้นทางที่ดีที่สุดของตนเอง โดยไม่สนใจอุปกรณ์อื่น บางครั้งเรียกว่าเป็น Intradomain Routing Protocol ดังรูปที่ 3.20



รูปที่ 3.20 Intradomain / Interdomain ในการติดต่อระหว่างอุปกรณ์ Router

2 . **Exterior Routing Protocol** เป็นการเชื่อมต่อเครือข่ายภายใน เช่น ISP เข้ากับเครือข่ายอินเทอร์เน็ตนั้น จำเป็นต้องมีการเชื่อมต่อแบบ Exterior Routing Protocol และอาศัยหมายเลข AS Number ในการติดต่อ ถ้าอาศัยหลักการของ Interior Routing Protocol คือให้ Router ในเครือข่ายทุกตัวในอินเทอร์เน็ตส่งค่า Routing Table ออกไปในเครือข่ายนั้นไม่สามารถทำได้เพราะช่องสัญญาณที่มีอยู่ไม่เพียงพอ และส่งผลให้เกิดการติดขัดได้ บางครั้งเราเรียกโปรโตคอล Exterior Routing Protocol ว่าเป็น Interdomain Routing Protocol ซึ่งมีลักษณะดังรูปที่ โดยหลักการการทำงานจะใช้ในลักษณะของ Distance-Vector Routing Protocol และ Link-State Roting Protocol เช่นกัน แต่จะทำการแลกเปลี่ยนข้อมูลระหว่าง Router หลักที่มีหมายเลข AS Number แทน ประกอบกับต้องมีการพิจารณาปัจจัยอื่นๆประกอบด้วย เช่น ขนาดของช่องสัญญาณ ( Bandwidth ) , จำนวน Delay ที่เกิดขึ้นในช่องสัญญาณ , ความหนาแน่นของช่องสัญญาณ และความน่าเชื่อถือของช่องสัญญาณ ( Reliable ) เป็นต้น ตัวอย่างโปรโตคอลที่ใช้งานได้แก่ RIP ( Routing Information Protocol ) , IGRP ( Interior Gateway Routing Protocol ) , OSPF ( Open Shortest Path First ) , EGP ( Exterior Gateway Protocol ) และ BGP ( Border Gateway Protocol ) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 4

### การรับส่งข้อมูลแบบอนุกรมด้วย MCS-51

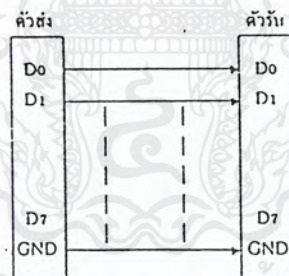
#### 4.1 พื้นฐานการรับส่งข้อมูล

การรับส่งข้อมูลในระบบคอมพิวเตอร์โดยทั่วไปจะหมายถึง การรับส่งข้อมูลเป็นจำนวน ไบต์ ๆ ให้กับอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์ ซึ่งอาจแบ่งประเภทของการรับส่งข้อมูลได้ 2 แบบ

1) การรับส่งข้อมูลแบบขนาน ( Parallel )

2) การรับส่งข้อมูลแบบอนุกรม ( Serial )

การรับส่งข้อมูลแบบขนานจะเป็นการรับส่งข้อมูลจำนวน 1 ไบต์ ออกไปทางพอร์ทในเวลาเดียวกันในระบบคอมพิวเตอร์ 1 ไบต์ จะมีจำนวน 8 บิต คือ  $D_0 - D_7$  ถ้ามีการส่งข้อมูลแบบขนานจะใช้สายสัญญาณอย่างน้อย 9 เส้น คือสาย Data 8 เส้นและสายกราวด์ 1 เส้นดังรูป



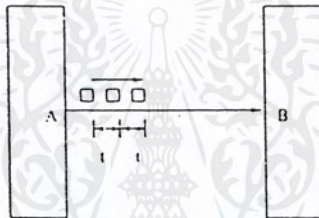
รูปที่ 4.1 การรับส่งข้อมูลแบบขนาน

การรับส่งข้อมูลแบบอนุกรม คือ การรับส่งข้อมูลที่ละบิต จนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์ คือ  $D_0 - D_7$  อาจส่งบิต  $D_0$  ออกไปก่อนแล้วตามด้วย  $D_1$  ไปเรื่อย ๆ จนถึง  $D_7$  การส่งข้อมูลทั้งสองแบบมีข้อดีข้อเสียต่างกันคือการส่งข้อมูลแบบขนาน สามารถส่งข้อมูลได้เร็วคือส่งทีเดียวจะได้ข้อมูลครบ 1 ไบต์ แต่ถ้าต้องส่งเป็นระยะไกล ๆ จะสิ้นเปลืองสายสัญญาณมาก ถ้าเป็นการส่งแบบอนุกรม เมื่อต้องการส่งข้อมูลเป็นระยะไกล ๆ จะช่วยประหยัดสายสัญญาณเนื่องจากจะใช้สายอย่างน้อยเพียง 2 เส้น คือสายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลานาน

เนื่องจากการส่งที่ละบิต ในบทนี้จะกล่าวถึงพื้นฐานการรับส่งข้อมูลแบบอนุกรม โดยจะเน้นที่ตัว MCS-51 เป็นสำคัญ

#### 4.1.1 การรับส่งข้อมูลแบบซิงโครนัส ( Synchronous Input / Output )

การรับส่งข้อมูลแบบนี้ไม่ว่าจะเป็นการส่งแบบอนุกรมหรือ ขนานข้อมูลแต่ละไบต์ที่ถูกส่งออกไปจะมีช่วงเวลาห่างกันแน่นอน เช่นการส่งข้อมูลจาก A ไป B ดังรูปที่ 4.2 Data 1 จะห่างจาก Data 2 เป็นเวลา  $t$  และ Data 3 เป็นเวลา  $t$  เช่นกัน ระบบนี้เหมาะกับงานที่ไม่มีความยุ่งยากมาก



รูปที่ 4.2 การส่งข้อมูลแบบซิงโครนัส

#### 4.1.2 การรับส่งข้อมูลแบบอะซิงโครนัส ( Asynchronous Input / Output )

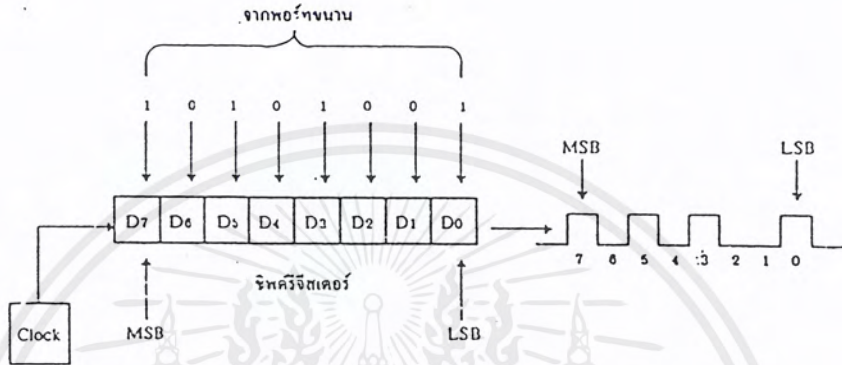
การรับส่งข้อมูลแบบนี้ไม่ว่าจะเป็นการส่งแบบอนุกรมหรือขนานข้อมูลที่ส่งออกไปจะไม่มีเวลาห่างกันแน่นอน ซึ่งจะขึ้นอยู่กับความพร้อมของผู้ส่งและผู้รับ โดยจะมีสายสัญญาณตรวจสอบความพร้อมที่จะติดต่อกันหรือไม่ โดยสัญญาณที่เพิ่มขึ้นมาจากระบบแบบซิงโครนัสเรียกว่า Status Line

#### 4.2 รูปแบบของการรับส่งข้อมูลแบบอนุกรม

เมื่อไมโครคอมพิวเตอร์ต้องการจะส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะส่งข้อมูลออกไปทางพอร์ทซึ่งเป็นพอร์ทแบบขนานก่อน จากนั้นจะมีอุปกรณ์มาต่อที่พอร์ท เพื่อแปลงข้อมูลแบบขนานให้เป็นแบบอนุกรมอีกทีหนึ่ง ( Parallel - To - Serial Conversion ) ตัวแปลงข้อมูลนี้อาจพิจารณาว่า เป็น Shift Register ดังรูปที่ 4.3 เมื่อข้อมูลที่จะส่งอยู่ใน Shift Register แล้ว ตัวสัญญาณนาฬิกาจะเป็นตัวกระตุ้นให้ส่งข้อมูลบิตตัวออกไปในเวลาแรก จากนั้นจะส่งบิตต่อไปตามออกมา จากรูปที่ 4.3 จะเป็นการส่งข้อมูล A9H ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวรับข้อมูลแบบอนุกรมเมื่อตัวรับรับข้อมูลจะเป็นการรับเข้ามาใน Shift Register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนานอีกทีหนึ่ง ( Serial – to – Parallel ) ระบบคอมพิวเตอร์ในปัจจุบันจะมีตัวแปลง Parallel – to – Serial และ Serial – to – Parallel อยู่ในชิพ



รูปที่ 4.3 การส่งข้อมูลแบบอนุกรม

ไอซี เรียกว่า Universal Asynchronous Receiver Transmitter ( UART ) การส่งข้อมูลแบบอนุกรมนั้นจะต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้การรับส่งข้อมูลสามารถทำงานได้ถูกต้องมากขึ้น โดยมีการเติมค่าบิตต่างๆ ลงไปตามรูปที่ 4.4

Stop	P	D7	D6	D5	D4	D3	D2	D1	D0	Start
------	---	----	----	----	----	----	----	----	----	-------

รูปที่ 4.4 บิตต่างๆของข้อมูลที่ส่งแบบอนุกรม

ถ้ามีการส่งข้อมูลแบบ 8 บิต จะต้องส่งบิตแรกออกไปก่อน เรียกว่าบิตเริ่มต้น ( start Bit ) ถ้ามีการส่งข้อมูลหลาย ๆ ไบต์ออกมาบิตนี้จะเป็นตัวบอกว่า มีข้อมูลใหม่มาแล้ว โดยทั่วไปบิตเริ่มต้นมักมีระดับลอจิกเป็น " 0 " ต่อจากบิตเริ่มต้นจะเป็นข้อมูลบิต D<sub>0</sub> ถึง D<sub>7</sub> จากนั้นจะตามด้วยบิตตรวจสอบความถูกต้อง ( Parity Bit ) ถ้าข้อมูล 8 บิตที่ส่งออกมา จำนวนของบิตที่มีค่าเป็น " 1 " เป็นจำนวนคู่ บิตนี้จะมีค่าเป็น 0 แต่ถ้าจำนวนของบิตที่มีค่าเป็น " 1 " เป็นคี่ บิตนี้จะมีค่าเป็น " 1 " จากนั้น ข้อมูลที่ส่งออกไปจะตามด้วยบิตสิ้นสุดข้อมูล ( Stop Bit ) เพื่อเป็นการบอกว่าข้อมูลบิตที่ส่งมา 8 บิตนั้นหมดแล้ว ตัวบิต Stop อาจมีจำนวนมากกว่า 1 บิต ก็ได้ เช่น 1.5 บิต, 2 บิต

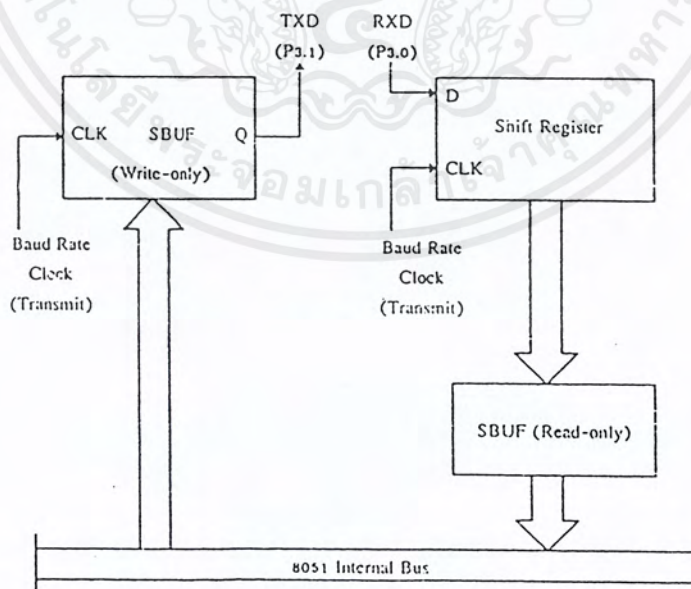
การส่งข้อมูลแบบอนุกรมนั้นความเร็วของการส่งจะมีค่าเป็น บิตต่อวินาที เรียกว่า “ Baud Rate “

#### 4.3 MCS – 51 กับการรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้นภายในชิพ MCS-51 จะมี UART อยู่ในตัว ซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ ถ้าเป็นไมโครโปรเซสเซอร์ เช่น เบอร์ Z-80 ถ้าต้องการรับส่งข้อมูลแบบอนุกรมจะต้องนำชิพ UART มาประกอบด้วย

พอร์ทอนุกรมของ MCS – 51 จะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองจะอยู่ในพอร์ท 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ทอนุกรมของ MCS – 51 สามารถทำงานแบบ Full Duplex ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Register โดยรีจิสเตอร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลไปที่ตำแหน่งนี้ จะเป็นการส่งข้อมูลออกทางพอร์ทอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้จะเป็นการรับข้อมูลจากพอร์ทอนุกรม โดยใน SBUF จะประกอบด้วยบัฟเฟอร์ 2 ตัว สำหรับส่งและรับข้อมูล ดังรูปที่4.5



รูปที่4.5 การรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัฟเฟอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ที่ตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่าง ๆ ของการรับส่งข้อมูลแบบอนุกรม

สำหรับความเร็วของการส่งข้อมูล ( Baud Rate ) สามารถหาได้จากตารางหารสัญญาณนาฬิกาที่ใช้กับ MCS – 51

#### 4.4 Serial Port Control Register

MCS – 51 มีโหมดการทำงานของพอร์ทอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมโหมดการทำงานได้โดยเขียนข้อมูลไปยังรีจิสเตอร์ SCON ความหมายของแต่ละบิตแสดงดังตารางที่ 4.1 และ 4.2

ตารางที่ 4.1 บิตต่าง ๆ ของรีจิสเตอร์ SCON

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตแฟลคกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับส่งข้อมูลในโหมด 2 และ 3 สามารถ Set และ Clear ได้โดย Software
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	TI	99H	บิตแฟลคแสดงการอินเตอร์รัพท์ภายหลังการส่งข้อมูลออกไปโดยจะ Set เมื่อส่งข้อมูลออกไปหมดแล้วและสามารถ Clear ได้โดย Software
SCON.0	RI	98H	บิตแฟลคแสดงการอินเตอร์รัพท์ภายหลังการรับข้อมูลเข้ามาและสามารถ Clear ได้โดย Software

ตารางที่ 4.2 โหมดต่าง ๆ ของการรับส่งแบบอนุกรม

SM0	SM1	MODE	ความหมาย	BAUD RATE
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้ (Oscillator Frequency /12 )
0	1	1	8- bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer
1	0	2	9- bit UART	เปลี่ยนแปลงไม่ได้ (Oscillator Frequency /12 or 64)
1	1	3	9- bit UART	สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer

ก่อนที่จะใช้พอร์ทอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อนเพื่อกำหนดโหมดการทำงานและลักษณะต่าง ๆ เช่น

```
MOV SCON, #01010010 B
```

เป็นการกำหนดให้พอร์ทอนุกรมทำงานในโหมด 1 และอีนานเบิลให้มีการรับข้อมูล พร้อมกับกำหนดให้ TI เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้วบิต TI จะถูกเซตเป็น " 1 " ในการส่งข้อมูล จะต้องคอยตรวจสอบบิต TI เพราะว่าถ้า TI ยังไม่เป็น " 1 " แสดงว่าข้อมูลยังส่งไปไม่หมด ถ้าหากมีการเขียนข้อมูลไปต่อก็จะไปไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับในการรับข้อมูลบิต REN จะต้องเซตให้เป็น " 1 " ยกเว้นโหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้ว บิต RI จะถูกเซตเป็น " 1 "

#### 4.5 Mode Of Operation

ใน MCS – 51 การสื่อสารทางพอร์ทอนุกรมจะมีอยู่ 4 ประเภท หรือ 4 โหมด ซึ่งจะกำหนดได้ที่บิต SM0 และ SM1 ใน SCON โดยจะมี 3 โหมด เป็นการสื่อสารแบบ Asynchronous โดยลักษณะของข้อมูลที่ส่งจะมีบิตเริ่มต้น ( Start Bit ) และบิตจบ ( Stop Bit ) คล้ายกับการสื่อสารแบบ RS – 232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ทอนุกรมในลักษณะซีพรีซีเตอร์

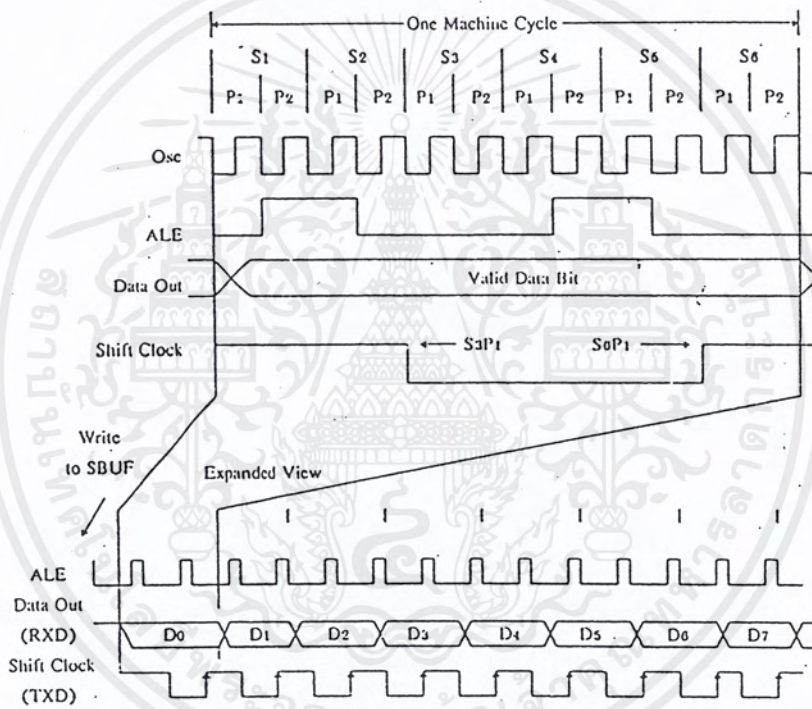
##### 4.5.1 8 – Bit Shift Register ( Mode 0 )

การทำงานในโหมดนี้จะใช้ขา RXD ในการรับส่งข้อมูลโดยต่อกับ Shift Register ภายนอก ส่วนขา TXD จะเป็น Out Put Clock เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิตถ้ามีการส่งข้อมูลออก ส่วน RXD จะทำหน้าที่รับข้อมูลเข้ารีจิสเตอร์ภายนอก เมื่อรีจิสเตอร์เต็มแล้วจะส่งข้อมูลเข้าพอร์ทอนุกรมต่อไป

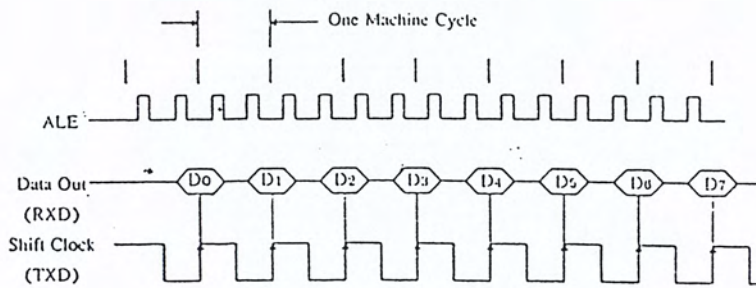
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่า Baud Rate เท่ากับ 1/12 ของความถี่ที่ใช้บนชิพ

ในการส่งข้อมูลจะทำโดย เขียนข้อมูลไปที่รีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD (P3.0) โดยจะสอดคล้องกับสัญญาณที่ออกมาทางขา TXD ซึ่งสัญญาณของขา TXD จะถูกส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก "0" ใน S3P1 และจะกลับเป็นลอจิก "1" ใน S6P1 ซึ่งแสดงได้ดังรูปที่ 4.6



รูปที่ 4.6 ไคอะแกรมเวลาการส่งข้อมูล

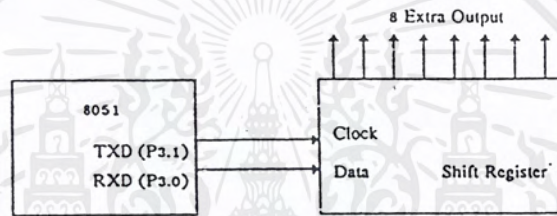


รูปที่ 4.7 ไคอะแกรมเวลาการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการรับข้อมูลจะรับได้เมื่อเซตขา Receiver Enable (REN) เป็น " 1 " และเคลียร์ขา Receiver Interrupt Bit (RI) เป็น " 0 " ข้อมูลจะเข้าสู่ MCS – 51 เมื่อ Clock Shift ถูกส่งออกไปทาง TXD ที่ขอบขาขึ้นของ Clock Shift บิตค่าจะถูกส่งเข้ามาก่อนดังรูปที่ 4.7

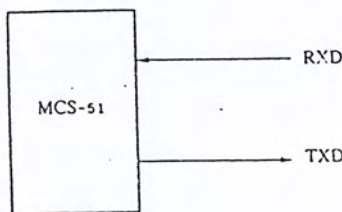
ในการประยุกต์ใช้งานโหมคนี้อาจต้องมีไอซีพรีจิสเตอร์มาต่อภายนอก เช่น ถ้าหากต้องการส่งข้อมูลออกมาทางพอร์ทอนุกรม อาจต่อวงจรได้ดังรูป 4.8 โดยใช้ ไอซี Serial – To – Parallel Shift Register โดยส่งข้อมูลออกมาทาง RXD และใช้ TXD เป็น Clock



รูปที่ 4.8 การส่งข้อมูลออกโดยใช้ชิพรีจิสเตอร์

**4.5.2 8 – Bit UART With Variable Baud Rate ( Mode 1 )**

ในโหมคนี้อจะเป็นการรับส่งข้อมูลแบบ 10 บิตซึ่งประกอบด้วยบิตเริ่มต้น ( เป็น "0" ) ข้อมูล 8 บิตและบิตจบ ( เป็น " 1 " ) นอกจากนี้ยังสามารถกำหนดค่า Baud Rate ได้โดยค่า Baud Rate นี้จะแปรตามตัวจับเวลาตัวที่ 1 ในโหมคนี้อ จะส่งข้อมูลเข้าทาง TXD และรับข้อมูลเข้าทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัว Stop Bit จะเข้ามายังบิต RB8 ใน SCON



รูปที่ 4.9 การรับส่งข้อมูลในโหมค 1

ค่า Baud Rate ที่ใช้ในการรับส่งข้อมูลจะกำหนดโดย Timer 1 หลังจากโปรแกรมไปใน Timer 1 แล้วสามารถเลือกค่า Baud Rate ได้อีกสองค่าคือ ค่าจาก Timer 1 Overflow หาร 32 กับ ค่าจาก Timer 1 Overflow หาร 16

การส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิตไปที่ SBUF โดยบิตที่ 9 ( Stop Bit ) ให้เขียนลงใน TB8 ใน SCON จากนั้นข้อมูลจะถูกส่งออกมาทางขา TXD โดยส่ง Start Bit ออกมาก่อนตามด้วยข้อมูล 8 บิต และจบด้วย Stop Bit เมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag ( TI ) จะเป็น "1" ดังนั้นในการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

ในการรับข้อมูล จะเริ่มจากเมื่อมีการเปลี่ยนแปลงลอจิกจาก 1 เป็น 0 ทางขา RXD หมายความว่าเริ่มรับบิตเริ่มต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SBUF และ Stop Bit จะถูกเก็บในบิต RB8 ของรีจิสเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้วบิต Interrupt Flag ( RI ) จะถูกเซต ดังนั้นในการอ่านข้อมูลจะอ่านได้เมื่อบิต RI ถูกเซตแล้ว เมื่ออ่านข้อมูลไปแล้วจะต้องเคลียร์บิตนี้

#### 4.5.3 9 – Bit UART With Fixed Baud Rate ( Mode 2 )

การทำงานในโหมดนี้ไม่สามารถกำหนดค่า Baud Rate ได้ ซึ่งค่า Baud Rate จะมีสองค่าคือ  $1/64$  และ  $1/32$  ของสัญญาณนาฬิกาบนชิพ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต บิตเริ่มต้นบิตหยุด รวมเป็น 11 บิต โดยข้อมูล 9 บิตจะเป็นจำนวนข้อมูล 8 บิต และบิตที่โปรแกรมได้อีก 1 บิต โดยบิตนี้จะเป็นบิตที่ 9 ซึ่งจะใช้เป็น Parity บิต ในการส่งข้อมูลจะต้องเขียนไปที่ บิต TB8 ในรีจิสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

#### 4.5.4 9 – Bit UART With Variable Baud Rate ( Mode 3 )

การทำงานในโหมดนี้จะคล้ายกับโหมด 2 แต่สามารถกำหนดค่า Baud Rate ได้โดยการโปรแกรมไปที่ Timer 1 หลังจากโปรแกรมแล้วยังสามารถเลือกได้อีก 2 ค่าคือ ความถี่การ Overflow ของ Timer 1 หารด้วย 16 และหารด้วย 32

### 4.6 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล

การรับข้อมูล ถ้าจะให้ MCS –51 รับข้อมูลทางพอร์ทอนุกรมจะต้องโปรแกรมไปที่บิต Receiver Enable ( REN ) ในรีจิสเตอร์ SCON ให้เป็นลอจิก " 1 " ซึ่งอาจทำได้สองวิธีดังนี้

```
SETB REN
```

เป็นการเซตบิต REN ให้เป็น " 1 " หรืออาจทำโดยใช้คำสั่ง

```
MOV SCON ,#xx1xxxxB
```

ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า x หมายความว่า เป็นอะไรก็ได้ขึ้น

กับการใช้งานในโหมดต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแบบ 9 บิต ในการรับส่งข้อมูลที่มีบิตข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และโหมด 3 การส่งข้อมูลบิตที่ 9 จะถูกเขียนในบิต TB8 โดยการเขียนโปรแกรม สำหรับการรับข้อมูลเมื่อข้อมูลเข้าถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

การเพิ่มบิต Parity การส่งข้อมูลแบบ 9 บิต สามารถใช้บิตที่ 9 เป็นบิต Parity ได้ซึ่งบิต Parity จะอยู่ใน Program Status Word (PSW) โดยจะถูกเซตหรือเคลียร์ทุก ๆ แมชชีน ไซเคิลที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะส่งข้อมูลแบบ 8 บิต ตามด้วยบิต Even Parity เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

```
MOV C, P ; อ่านค่าบิต P มาเก็บใน C
MOV TB8, C ; นำค่าบิต Parity เขียนลงใน TB8
MOV SBUF, A ; ส่งข้อมูลไปทางพอร์ทอนุกรม
```

ถ้าเป็นแบบ Odd Parity ให้แก้ไขข้อมูลที่อ่านได้จากบิต Parity เสียก่อนที่จะส่งออกไปซึ่งเขียนโปรแกรมได้ดังนี้

```
MOV C, P ; อ่านค่าบิต Parity มาเก็บใน C
CPL C ; กลับค่าให้เป็น Odd Parity
MOV TB8, C ; เขียนค่าลงใน TB8
MOV SBUF, A ; ส่งข้อมูลออกไปทางพอร์ทอนุกรม
```

การส่งข้อมูลแบบมี Parity บิตด้วยไมโซจะส่งได้แบบ 9 บิต หรือโหมด 2 และ 3 เท่านั้น ในโหมด 1 ซึ่งส่งข้อมูลแบบ 8 บิตก็สามารถทำได้ อย่างเช่นการส่งรหัส ASCII จะใช้บิตข้อมูล 7 บิต สำหรับบิตที่เหลืออีกหนึ่งบิตจะเป็นบิต Parity รวมเป็น 8 บิต ซึ่งสามารถเขียนโปรแกรมได้ดังนี้

```
CLR ACC.7 ; เคลียร์ค่าบิต 7 เพื่อใช้เป็น Parity บิต
MOV C, P ; นำบิต Parity มาเก็บใน C
MOV ACC.7, C ; เขียนค่าบิต Parity ลงในรีจิสเตอร์ A
MOV SBUF, A ; ส่งข้อมูลออกไปทางพอร์ทอนุกรม
```

แฟลกอินเทอร์รัพท์เมื่อมีการรับส่งข้อมูลเสร็จสิ้นจะมีผลต่อแฟลกอินเทอร์รัพท์ (RI และ TI) ในรีจิสเตอร์ SCON ซึ่งบิตเหล่านี้จะถูกเซตโดย Hardware แต่ต้องเคลียร์ด้วย Software

บิต RI ถ้าถูกเซตหมายความว่าบัฟเฟอร์ที่ใช้รับข้อมูลเต็มให้อ่านไปได้แล้ว และบิตนี้สามารถใช้อินเทอร์รัพท์ MCS - 51 แต่ถ้าเขียนโปรแกรมจะใช้วิธีตรวจเช็คบิตนี้ ถ้าเป็น " 1 " หมายความว่าให้อ่านข้อมูลมาเก็บใน รีจิสเตอร์ A ได้ แต่ก่อนอ่านจะต้องเคลียร์ RI เสียก่อนเพื่อจะได้รับข้อมูลถัดไปได้ ซึ่งเขียนโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WAIT : JNB RI , WAIT ; ถ้าบิตนี้ไม่เป็น " 1 " จะทำงานอยู่ที่เดิม
      CLR RI ; เคลียร์ RI
      MOV A ; อ่านค่ามาเก็บใน A

```

บิต TI เมื่อส่งข้อมูลออกไปแล้วบิตนี้จะถูกเซตเป็นการบอกว่าบัพเฟอร์ส่งข้อมูลว่างแล้วให้ส่งข้อมูลใหม่เข้าไปได้ ซึ่งสามารถใช้บิตนี้อินเทอร์รัพท์ MCS - 51 ได้เช่นกัน แต่ถ้าเขียนโปรแกรมคอยตรวจเช็คอาจเขียนได้ดังนี้

```

WAIT : JNB TI , WAIT ; ตรวจบิต TI ว่าเป็น " 1 " หรือยัง
      CLR TI ; เคลียร์ TI
      MOV SBUF , A ; เขียนข้อมูลลงไป

```

#### 4.7 อัตราการส่งข้อมูลของพอร์ทอนุกรม

จากการศึกษาการรับส่งข้อมูลในโหมดต่าง ๆ พบว่าในโหมด 0 และ โหมด 2 ไม่สามารถกำหนด Baud Rate เองได้ โดยในโหมด 0 ค่า Baud Rate จะมีค่าเท่ากับความถี่ของ Oscillator หารด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ Oscillator หารด้วย 32 และหารด้วย 64 สองค่านี้เรียกว่า SMOD 0 และ SMOD 1 ซึ่งสามารถกำหนดได้ในรีจิสเตอร์ PCON บิตที่ 7 ในรีจิสเตอร์ PCON นี้ ไม่สามารถเข้าถึงข้อมูลระดับบิตได้ การเขียนข้อมูลลงไปทีละบิตจะต้องใช้วิธีที่เรียกว่า " Read - Modify - Write " คืออ่านค่าขึ้นมาแก้ไขแล้วเขียนลงไปใหม่ ตัวอย่างเช่น

```

MOV A , PCON ; อ่านค่าจาก PCON มาเก็บในรีจิสเตอร์ A
SETB ACC.7 ; เซตบิต 7 ( SMOD )
MOV PCON , A ; เขียนค่าลงไปใหม่ใน PCON

```

สำหรับโหมด 1 และโหมด 3 สามารถกำหนดค่า Baud Rate ได้โดยการโปรแกรมลงใน Timer 1 ในการโปรแกรมแต่ละครั้งจะมี SMOD สองค่าเช่นกัน ค่า Baud Rate ของโหมดต่าง ๆ แสดงได้ดังรูปที่ 4.10

การใช้ Timer 1 กำหนด Baud Rate Clock

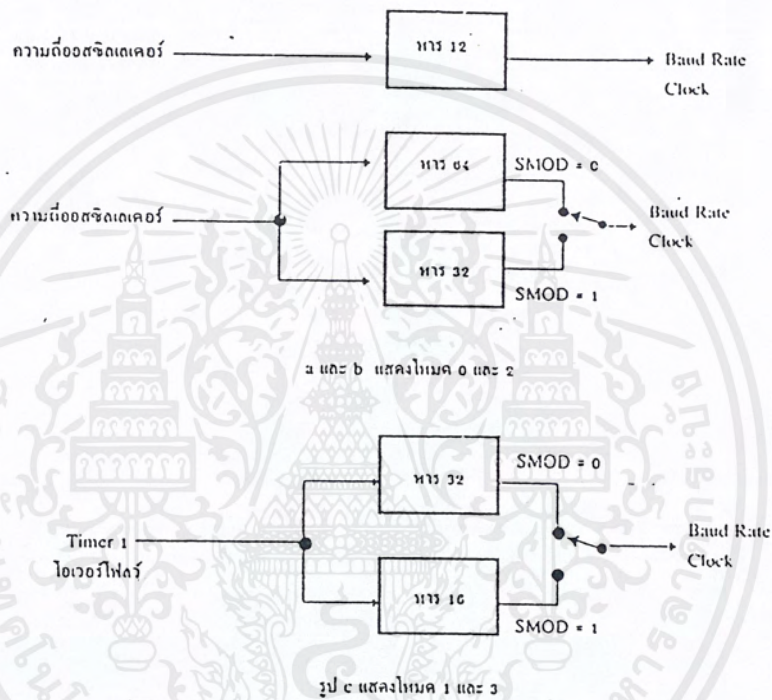
การกำหนดค่าลงใน Timer 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8 - Bit Auto Reload Mode ( โหมด 2 ) โดยเขียนค่าไปที่ TH1 ซึ่งโปรแกรมที่รีจิสเตอร์ TMOD ได้ดังนี้

```
MOD TMOD , #0010xxxxB
```

ค่า x หมายความว่า เป็นอะไรก็ได้เพราะบิตเหล่านี้ใช้ใน Timer 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการ Baud Rate ต่ำ ๆ สามารถใช้ 16-Bit Mode ได้ โดยโปรแกรมเป็น TMOD = 0001xxxxB ค่า Baud Rate ที่ส่งออกมาจะมีค่าเท่ากับ ความถี่ของ Timer 1 เกิด Overflow หาร ด้วย 32 (หรือหารด้วย 16 ถ้าเป็น SMOD = 1)



รูปที่ 4.10 แสดงการกำหนด Baud Rate ในโหมดต่าง ๆ

รูปแบบทั่วไปของการหาค่า Baud Rate ในโหมด 1 และ 3 สามารถหาได้ดังนี้

$$\text{Baud Rate} = \text{Timer 1 Overflow Rate} / 32 \quad (4.1)$$

ถ้าต้องการค่า Baud Rate เท่ากับ 1,200 สามารถคำนวณค่าความถี่ Overflow ของ Timer 1 ได้ดังนี้

$$1,200 = \text{Timer 1 Overflow Rate} / 32$$

จะได้ Timer 1 Overflow Rate เท่ากับ 38.4 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าระบบ MCS – 51 ใช้ความถี่สัญญาณนาฬิกาจาก Crystal เท่ากับ 12 MHz ตัว Timer จะได้รับ Clock เท่ากับ 1 MHz หรือ 1,000 kHz ถ้าเราต้องการ Timer 1 Overflow เท่ากับ 38.4 kHz ดังนั้นค่าอัตรา Overflow มีค่าเท่ากับ  $1,000 / 38.4 = 26.04$  Clock โดยค่า Overflow จะเกิดเมื่อเกิดการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องให้ Timer 1 นับไป 26 Count ดังนั้นค่าที่จะให้รีจิสเตอร์ TH1 มีค่าเท่ากับ -26 ซึ่งใช้เป็นค่า Reload ดังนั้นเขียนคำสั่งได้ดังนี้

```
MOV TH1, #-26
```

ตัวโปรแกรมแอสเซมเบอร์ทั่วไปจะแปลงค่า -26 เป็น 0E6H เอง จากที่ผ่านมาจะเห็นว่าความถี่ Baud Rate จะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาที่ใช้จาก Crystal ในตารางที่ 4.3 จะเป็นค่าที่ต้องกำหนดใน Timer 1 เมื่อต้องการค่า Baud Rate ต่าง ๆ

ตารางที่ 4.3 ความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่าง ๆ

ค่า Baud Rate	Crystal	SMOD โหมด	ค่า TH1	ค่า Baud Rate ที่ได้	ผิดพลาด
9600	12.000	1	-7 (F9H)	8923	7%
2400	12.000	0	-13 (F3H)	2404	0.16%
1200	12.000	0	-16 (E6H)	1202	0.16%
19200	11.059	1	-3 (FDH)	19200	0
9600	11.059	0	-3 (FDH)	9600	0
2400	11.059	0	-12 (F4H)	2400	0
1200	11.059	0	-24 (E8H)	1200	0

## บทที่ 5

### การออกแบบและการสร้าง

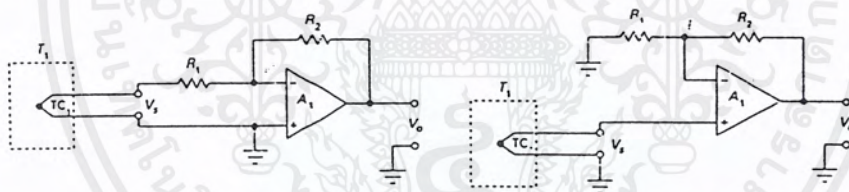
#### 5.1 การออกแบบส่วนของวงจร

##### 5.1.1 วงจรขยายสัญญาณเทอร์โมคัปเปิล

จากทฤษฎีและหลักการของการวัดอุณหภูมิเทอร์โมคัปเปิลชนิด K ซึ่งทำมาจากโครเมิล (CHROMEL) กับ อลูเมล (ALOMEL) ซึ่งให้แรงเคลื่อนไฟฟ้าออกมาประมาณ  $40.44 \mu\text{V} / ^\circ\text{C}$  จะเห็นได้ว่ามีสัญญาณมีขนาดเล็กมาก ดังนั้นเราจึงต้องทำการขยายสัญญาณดังกล่าว ซึ่งสามารถกระทำได้หลายวิธีดังนี้อาจทำได้หลายลักษณะดังนี้

##### 5.1.1.1 วงจรขยายสัญญาณจากเทอร์คัปเปิลโดยใช้ออปแอมป์

สามารถสร้างได้ทั้งวงจรขยายแบบกลับเฟส ( Inverting Amplifier ) หรือวงจรขยายแบบไม่กลับเฟส ( Non Inverting Amplifier ) ก็ได้ ดังรูปที่ 5.1

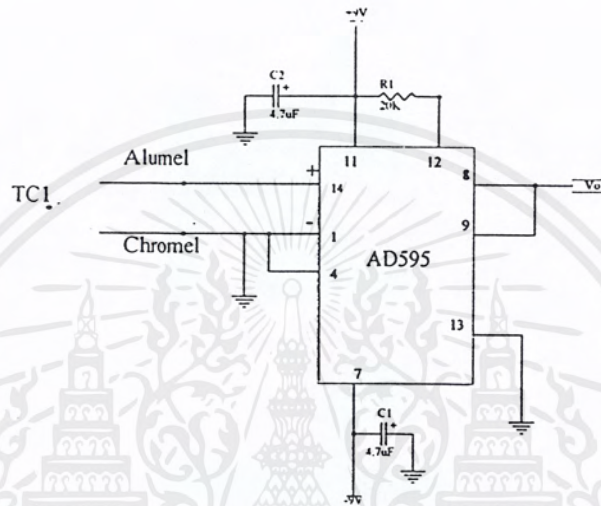


รูปที่ 5.1 วงจรขยายที่สร้างด้วยออปแอมป์

เนื่องจากเทอร์โมคัปเปิลมีค่าความต้านทานต่ำมาก ดังนั้นค่าความต้านทานขาเข้าของวงจร ( $R_1$ ) จึงไม่จำเป็นต้องมีค่าสูงมาก ( $R_1 > 100 \Omega$ ) โดยเราสามารถกำหนดอัตราขยายของวงจรได้ด้วยอัตราส่วนของ  $R_1/R_2$  สำหรับวงจรขยายแบบกลับเฟส และ  $[(R_1/R_2) + 1]$  แต่การใช้วงจรขยายแบบนี้สัญญาณที่ได้ต้องการการชดเชยรอยต่อเย็น ( Cold Junction Compensation ) และสัญญาณที่ได้อาจจะมี ค่าผิดพลาดเนื่องจากค่าแรงดันออฟเซต ( Offset ) ซึ่งเป็นค่าแรงดันที่มีผลมาจากตัวของออปแอมป์ได้ ดังนั้นใน โครงการนี้เราจึงไม่เลือกใช้งานวงจรขยายในลักษณะนี้

### 5.1.1.2 วงจรขยายสัญญาณเทอร์โมคัปเปิลโดยใช้ AD-595

AD-595 เป็น Monolithic Thermocouple Amplifiers with Cold Junction Compensation ซึ่งจะให้แรงดันเอาต์พุตมีค่า  $10 \text{ mV}/^{\circ}\text{C}$  โดยการต่อใช้งานแสดงดังรูปที่ 5.2



รูปที่ 5.2 วงจรขยายสัญญาณเทอร์โมคัปเปิลโดยใช้ AD-595

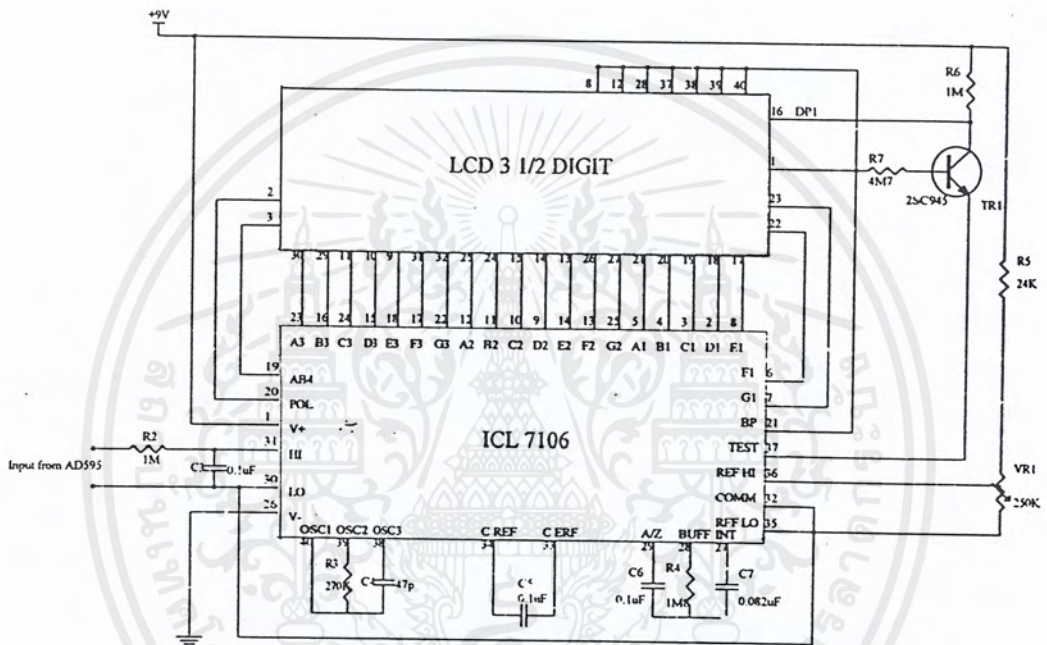
จากรูปเป็นวงจรใช้งานของเทอร์โมคัปเปิลที่ใช้ AD-595 ซึ่ง AD-595 มีหลักการทำงานของวงจรดังที่ได้กล่าวไว้ข้างต้น (2.2.2) โดยค่าแรงดันเอาต์พุตที่ได้สามารถเขียนเป็นความสัมพันธ์ได้ดังนี้

$$\text{AD-595 Output} = (\text{Type K Voltage} + 11 \mu\text{V}) \times 247.3$$

โดย ค่าแรงดัน  $11 \mu\text{V}$  เป็นค่าแรงดันออฟเซตเนื่องจากวงจรถ่าย และ  $247.3$  เป็นค่ากำลังขยายหลักของวงจรซึ่งอธิบายไว้ในหัวข้อ 2.2.2 จากความสัมพันธ์ข้างต้นอาจแสดงได้เป็นค่าเอาต์พุตประมาณ  $10 \text{ mV}/^{\circ}\text{C}$  แต่ความสัมพันธ์นี้เป็นเพียงการประมาณค่า (Interpreting) โดยค่าแรงดันที่ถูกต้องสามารถอ่านได้เป็นความสัมพันธ์ระหว่างแรงดันเอาต์พุตกับอุณหภูมิในหน่วยองศาเซลเซียส

5.1.2 วงจรส่วนแสดงผลด้วย LCD

จากสัญญาณที่ได้มาจากส่วนของวงจรรขยายสัญญาณเทอร์โมคัปเปิล จะถูกนำมาผ่าน ICL 7016 เพื่อทำการแปลงจากสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอลและแสดงผลด้วย LCD เพื่อที่จะสามารถอ่านค่าได้ทันทีในกรณีที่ผู้ปฏิบัติอยู่ในบริเวณนั้น โดยวงจรที่ใช้แสดงไว้ดังรูปที่ 5.3



รูปที่ 5.3 วงจรส่วนที่ใช้แสดงผล

จากรูปสามารถอธิบายหลักการทำงานของวงจรได้คือ สัญญาณเอาต์พุตที่ได้จากวงจรรขยาย จะถูกนำมาป้อนเข้าขา 31 (IN HI) และขา 30 (IN LO) โดยผ่านทาง R<sub>2</sub> เพื่อป้องกันอินพุตของวงจร และ C<sub>3</sub> เป็นตัวกรองแรงดันไฟฟ้าที่เข้ามากรณีที่เกิดการเปลี่ยนแปลงอย่างรวดเร็วและช่วยลดสัญญาณรบกวนที่จะเข้ามาทางอินพุต สัญญาณที่เข้ามาจะถูกแปลงจากอนาลอกเป็นดิจิตอลและส่งไปยัง LCD โดยจะแสดงผลด้วยความถี่ของสัญญาณนาฬิกาที่กำหนดโดยค่า R<sub>3</sub> และ C<sub>4</sub> โดยในที่นี้กำหนดไว้ที่ความถี่ประมาณ 45 KHz ซึ่งทำให้ตัวเลขบน LCD กระพริบประมาณ 3 ครั้งต่อวินาที ส่วนค่า C<sub>5</sub> จะเป็นตัวปรับศูนย์อัตโนมัติ (Auto-Zero) หมายถึง ขณะที่แรงดันอินพุตเป็นศูนย์ เช่น ทำการลัดวงจรขา 31 และ 30 เข้าด้วยกัน ตัวเลขจะแสดงค่า 0 V ส่วนค่า R<sub>5</sub> และ VR<sub>5</sub> จะทำหน้าที่กำหนดแรงดันอ้างอิงให้กับวงจร โดยจะทำการปรับ VR<sub>5</sub> เพื่อให้เครื่องแสดงค่าแรงดันที่ต้องการ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากรูปจะเห็นได้ว่า ที่ขา 19 และขา 4 เราทำการต่อ C และ R เพื่อกำหนดความถี่ของ สัญญาณนาฬิกาให้กับวงจร ในที่นี้เท่ากับ KHz ส่วนขา 1 และ 2 ซึ่งเป็นขา CS และ RD ทำการ ต่อลงกราวด์เพื่อให้วงจรทำงานอยู่ตลอด เราจะเริ่มต้นการทำงานของวงจรด้วยการกดสวิตช์ เพื่อให้ขา WR เปลี่ยนเป็น logic 1 ( ขอบขาลง ) โดยวงจรจะทำงานในลักษณะของ Successive Approximation ที่ละบิต โดยเริ่มจากบิต MSB ( Most Significant Bit ) ก่อน ส่วนการสร้างแรงดัน อ้างอิงให้กับวงจร ในที่นี้เลือกใช้ LM336 ที่ให้ค่าแรงดันคงที่ 2.5 V ป้อนเข้าที่ขา  $V_{REF2}$  (ขา 9)

โดยสัญญาณดิจิทัลที่ได้ (ขา 11 ถึงขา 18) จะถูกส่งต่อไปที่พอร์ต 1 ของ MCS-51 เพื่อทำ แปลงลักษณะการส่งสัญญาณจากการส่งสัญญาณแบบขนาน( Parallel) ไปเป็นการส่งสัญญาณแบบอนุกรม( Serial ) เพื่อใช้ในการติดต่อกับเครื่องคอมพิวเตอร์ด้วยมาตรฐาน RS 232 ต่อไป

## 5.2 การออกแบบส่วนของโปรแกรม

### 5.2.1 โปรแกรมส่วนควบคุมการส่งสัญญาณด้วยมาตรฐาน RS232

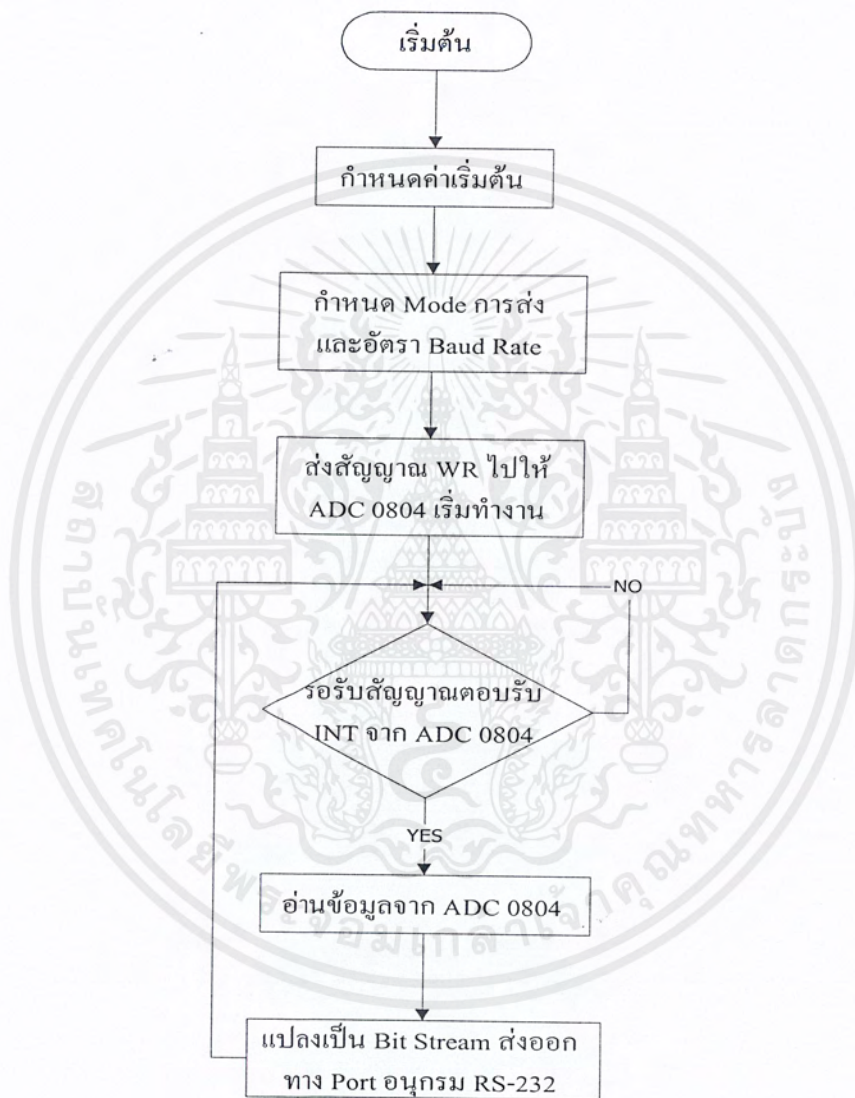
การทำงานของวงจรในส่วนของการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัล และการควบคุมการส่งสัญญาณแบบอนุกรม ทำงานโดยใช้ไมโครคอนโทรลเลอร์ 89C51 ในการประมวลผล เพื่อกำหนดหน้าที่และลำดับการทำงาน โดยการทำงานของโปรแกรมแสดงได้โดยแผนผังการทำงาน ( Flow Chart ) ดังรูปที่ 5.6

โดยในโครงงานนี้เราทำการกำหนดเลือกการส่งในโหมดที่ 1 ซึ่งกำหนดค่า Baud Rate เป็น 9600 บิต/วินาที โดยทำการกำหนดที่ TMOD Register ซึ่งเป็นการกำหนดตามค่าของอัตราการ Overflow ของ Timer และไม่มีการตรวจสอบความถูกต้องด้วย Parity Bit ดังที่ได้กล่าวถึงไว้ในบทที่ 4 โดยรูปแบบของ Bit Stream ที่ใช้เป็นมาตรฐานในการส่งข้อมูลของโครงงานนี้จะมีลักษณะดังรูปที่ 5.5

เริ่มต้น	ข้อมูล0	ข้อมูล1	ข้อมูล2	ข้อมูล3	ข้อมูล4	ข้อมูล5	ข้อมูล6	ข้อมูล7	หยุด
----------	---------	---------	---------	---------	---------	---------	---------	---------	------

รูปที่ 5.5 รูปแบบมาตรฐานที่ใช้ในการติดต่อผ่านพอร์ตอนุกรมในโครงงาน

จากรูปที่ 5.5 จะเห็นว่าประกอบด้วยบิตเริ่มต้น ( Start Bit ) เป็นตัวบอกจุดเริ่มต้นของข้อมูล มีค่าลอจิกเป็น 0 หรือเป็น Space ต่อด้วยบิตข้อมูล 8 บิต และลงท้ายด้วยบิตหยุด ( Stop Bits ) ที่มีค่าลอจิกเป็น 1 หรือ Mark

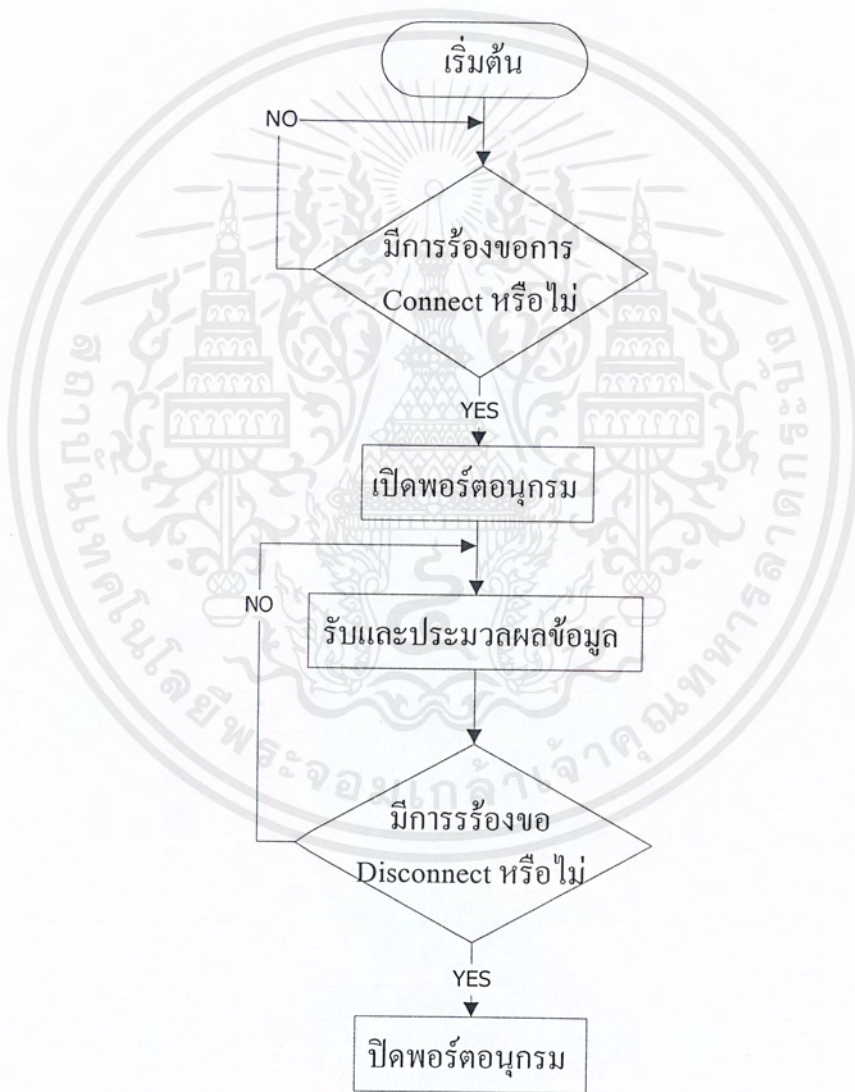


รูปที่ 5.6 แผนผังการทำงานของโปรแกรม MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.2.2 โปรแกรมส่วนทำการติดต่อกับคอมพิวเตอร์

หลังจากที่ข้อมูลจาก MCS-51 ถูกส่งผ่านมาที่พอร์ตอนุกรม เราจะทำการเขียนโปรแกรมเพื่อเชื่อมการติดต่อระหว่างวงจรถับคอมพิวเตอร์ เพื่อทำการรับค่าเข้ามาประมวลผลในคอมพิวเตอร์ โดยในโครงการนี้เราเลือกใช้โปรแกรมภาษา Delphi ที่มี Component ช่วยสนับสนุนการติดต่อแบบอนุกรมด้วยมาตรฐาน RS 232 โดยส่วนการรับค่ามีแผนผังการทำงานดังรูปที่ 5.7



รูปที่ 5.7 แผนผังการทำงานของส่วนรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรมจะทำการอ่านข้อมูลจากพอร์ตอนุกรมในรูปของรหัส ASCII จากนั้นก็ทำการแปลงข้อมูลกลับไปเป็นค่าตัวเลขของ ASCII ค่านั้น แล้วจึงนำค่าที่ได้ไปคำนวณออกเป็นค่าของ อุณหภูมิโดย

$$\text{ค่าอุณหภูมิ} = (\text{ค่าของรหัส ASCII} * 0.1960) / 0.01 \quad (5.1)$$

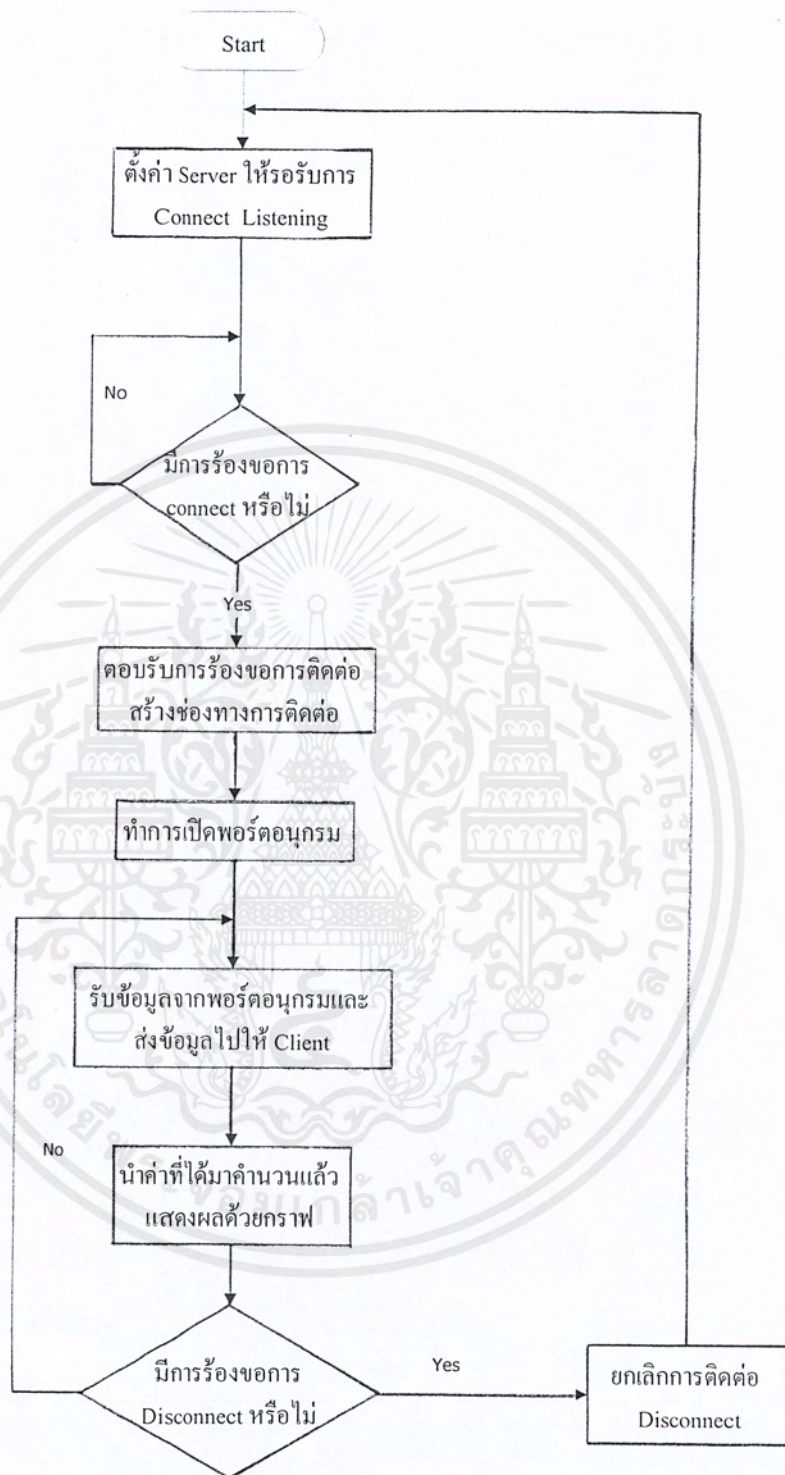
### 5.2.3 โปรแกรมส่วนควบคุมการเชื่อมต่อผ่านอินเทอร์เน็ต

เมื่อทำการเชื่อมต่อกับวงจรและสามารถรับค่าที่ได้จากเครื่องมือวัดได้แล้ว ก็จะทำาการเขียนโปรแกรมเพื่อทำการติดต่อกับเครือข่ายอินเทอร์เน็ตด้วยโปรแกรมภาษาเซลไฟล์ โดยทำการเชื่อมต่อเข้าสู่ระบบเครือข่ายอินเทอร์เน็ตด้วยโปรโตคอล TCP/IP ซึ่งมีหลักการคือจะทำการเชื่อมต่อในลักษณะของ Connection-Oriented คือทำการสร้างช่องทางติดต่อกันระหว่างผู้ใช้บริการ ( Client ) กับ ผู้ให้บริการ ( Server ) ก่อนที่จะทำการถ่ายโอนข้อมูลระหว่างกัน จุดมุ่งหมายของโปรแกรมส่วนนี้ สร้างขึ้นเพื่อให้ผู้ใช้บริการสามารถทำการตรวจสอบผลของเครื่องมือวัดต่างๆที่ต่ออยู่ที่ผู้ให้บริการ ได้โดยผู้ใช้บริการเพียงแต่ระบุ IP Address ของผู้ให้บริการเท่านั้น

โดยโปรแกรมในส่วนนี้จะประกอบด้วยโปรแกรม 2 ส่วนด้วยกัน คือ ส่วนของผู้ให้บริการ ( Server ) และส่วนของผู้ใช้บริการ ( Client ) โดยมีรายละเอียดดังต่อไปนี้

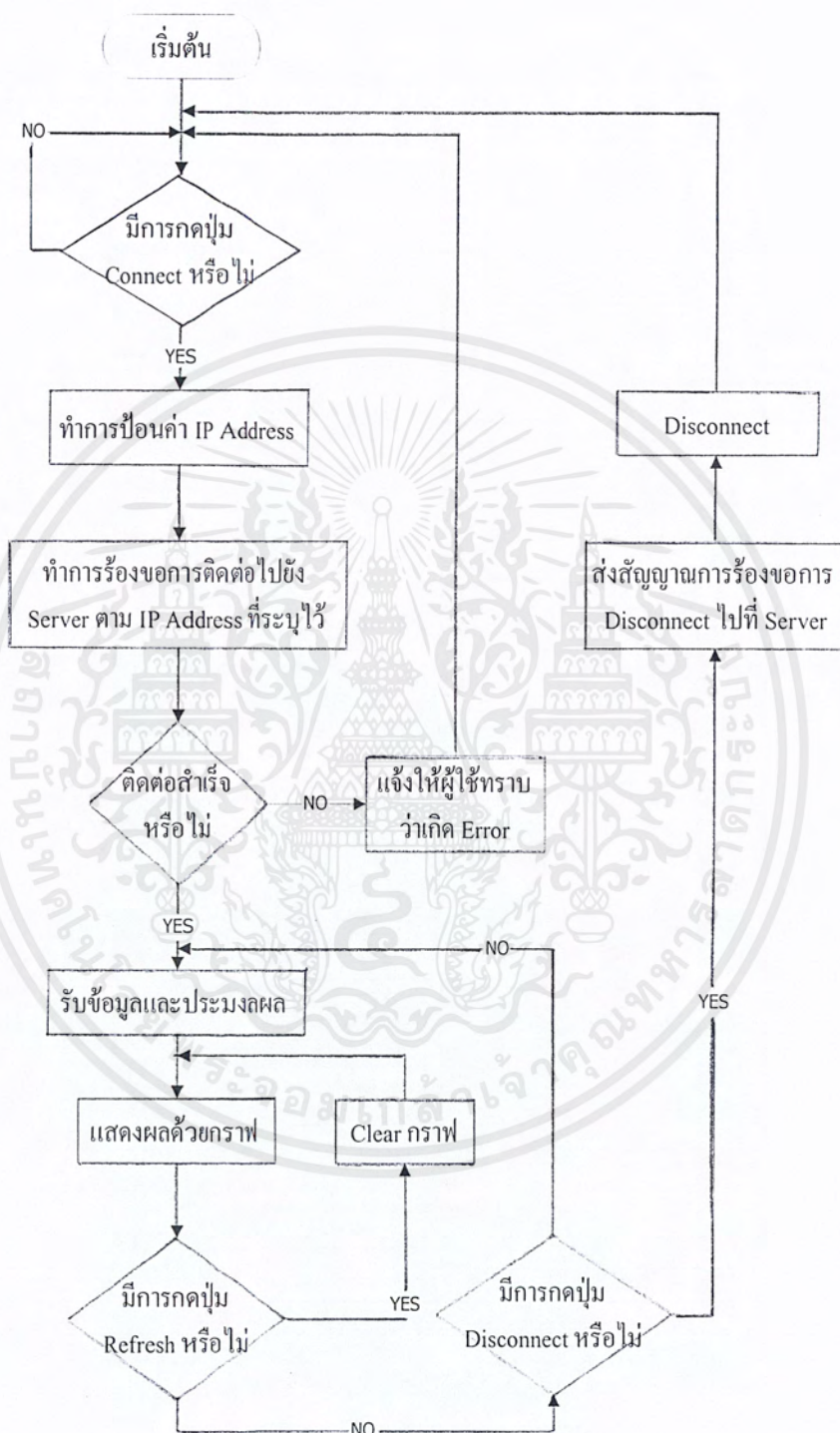
5.2.3.1 ) ส่วนของผู้ให้บริการ ( Server ) โปรแกรมในส่วนนี้จะประกอบไปด้วยโปรแกรม ส่วนที่ทำการเชื่อมต่อกับผู้ใช้บริการ และส่วนที่ทำการเชื่อมต่อกับวงจรซึ่งกล่าวถึงไปแล้วข้างต้น ส่วนของผู้ให้บริการ ( Server ) จะทำการเตรียมพร้อมรอรับการร้องขอติดต่อกับผู้ใช้บริการ(Client) อยู่ตลอดเวลา เมื่อมีการร้องขอการติดต่อกับผู้ใช้บริการ ผู้ให้บริการก็จะทำการสร้างช่องทางติดต่อกับและส่งสัญญาณตอบรับการติดต่อกลับไปให้ผู้ใช้บริการ จากนั้นก็จะเริ่มทำการส่งข้อมูลที่ได้จากเครื่องมือวัดพอร์ตอนุกรม จนกว่าจะมีการขอยกเลิกการติดต่อ จึงค่อยทำการปิดช่องทางติดต่อกับผู้ใช้บริการ โดยแผนผังแสดงการทำงานของส่วนผู้ให้บริการเป็นดังรูปที่ 5.8

5.2.3.2 ) ส่วนของผู้ใช้บริการ ( Client ) การทำงานในส่วนของผู้ใช้บริการจะมีหน้าที่หลักๆ 2 ส่วนคือ ส่วนของการส่งสัญญาณร้องขอการติดต่อกับผู้ให้บริการ และการส่งสัญญาณขอยกเลิกการติดต่อ กับอีกส่วนคือส่วนของการแสดงผล โดยในโครงงานนี้ได้เลือกการแสดงผลด้วยการเขียนกราฟเนื่องจากเห็นว่าสะดวกต่อการสังเกตเปรียบเทียบผลในช่วงเวลาต่างๆ โดยโปรแกรม จะทำการรับค่าจากผู้ให้บริการในรูปของ TEXT FILE จึงต้องนำข้อมูลที่ได้นำมาแปลงเป็นตัวเลข แล้วนำมาคำนวณผลตามสมการข้างต้นและนำไปเขียนกราฟต่อไป โดยแผนผังแสดงการทำงานของส่วนผู้ให้บริการเป็นดังรูปที่ 5.9



รูปที่ 5.8 แผนผังการทำงานของส่วนผู้ให้บริการ ( Server )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แผนผังการทำงานของส่วนผู้ใช้บริการ ( Client )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การทดลองและผลการทดลอง

#### 6.1 ส่วนของวงจรวัดอุณหภูมิ

ในส่วนนี้เราเลือกใช้เทอร์โมคัปเปิลและไอซีเบอร์ AD 595 ในการสร้างวงจรวัดอุณหภูมิ จากนั้นทำการทดลองวัดค่าอุณหภูมิของน้ำที่  $0^{\circ}\text{C}$  ถึง  $100^{\circ}\text{C}$  แล้วทำการเปรียบเทียบค่าระหว่างที่ได้จากวงจรวัดที่ทำการออกแบบกับค่าที่วัดได้จากเทอร์โมมิเตอร์แบบปรอทได้ผลดังตารางที่ 6.1

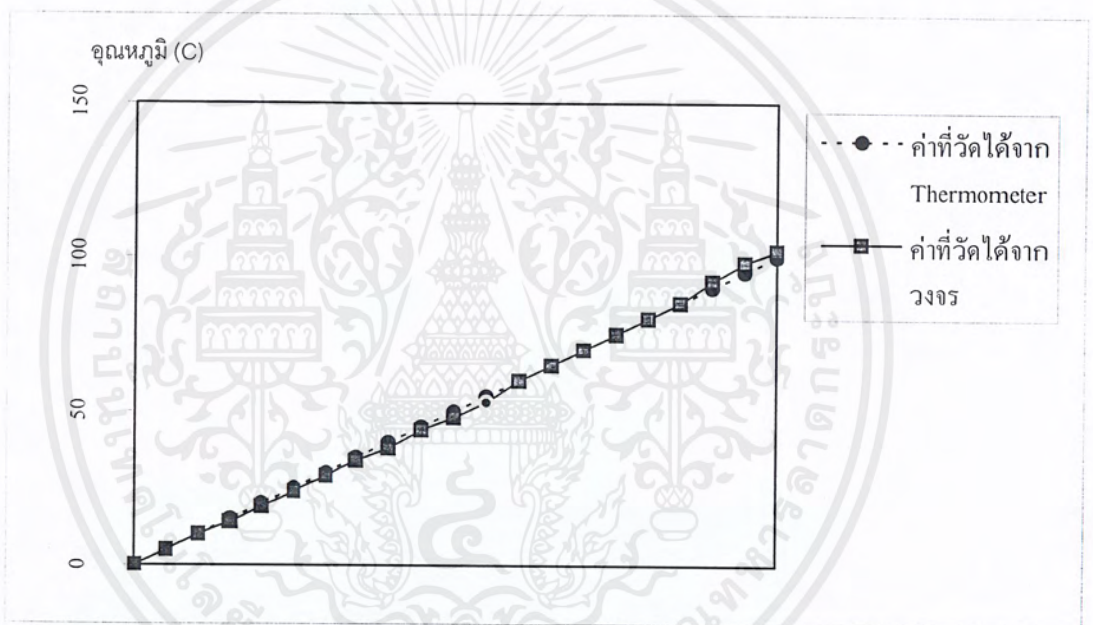
ตารางที่ 6.1 ค่าอุณหภูมิของเทอร์โมมิเตอร์เทียบกับวงจรวัด

ค่าอุณหภูมิจาก Thermometer ( องศาเซลเซียส )	ค่าอุณหภูมิจากวงจรวัด ( องศาเซลเซียส )	ค่าความผิดพลาด %
0	0	0.00
5	5	0.00
10	10	0.00
15	14	6.67
20	19	5.00
25	24	4.00
30	29	3.33
35	34	2.86
40	38	5.00
45	44	2.22
50	48	4.00
55	53	3.64
60	60	0.00
65	65	0.00
70	70	0.00
75	75	0.00
80	80	0.00
85	85	0.00
90	92	2.22

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าอุณหภูมิจาก Thermometer ( องศาเซลเซียส )	ค่าอุณหภูมิจากวงจรวัด ( องศาเซลเซียส )	ค่าความผิดพลาด %
95	98	2.11
100	102	2.00

จากค่าที่ได้นำมาแสดงผลเปรียบเทียบในรูปของกราฟได้ดังรูปที่ 6.1 และจากผลการทดลองจะเห็นว่ามีความผิดพลาดสูงสุด 6.67 เปอร์เซ็นต์



รูปที่ 6.1 กราฟเปรียบเทียบค่าที่ได้จากวงจรวัดกับค่าที่ได้จากเทอร์โมมิเตอร์

## 6.2 ส่วนของการแปลงสัญญาณอนาล็อกไปเป็นดิจิทัล

ในโครงงานนี้เลือกใช้ไอซีเบอร์ ADC 0804 ซึ่งจะทำการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัลขนาด 8 บิต โดยอาศัยการควบคุมจากไมโครคอนโทรลเลอร์ MCS-51 โดย ADC 0804 มีความสามารถในการแปลงแรงดันขนาด 0 ถึง 5 V ไปเป็นสัญญาณดิจิทัล 0 ถึง 255 ระดับ ในการทดลองเราทำการป้อนระดับแรงดันค่าต่างๆเข้าไปในคัวอุปกรณ์และอ่านค่าสัญญาณเอาท์พุทจาก LED ที่ต่ออยู่กับ MCS-51 แล้วเปรียบเทียบกับค่าที่ได้จากการคำนวณได้ผลดังตารางที่ 6.2

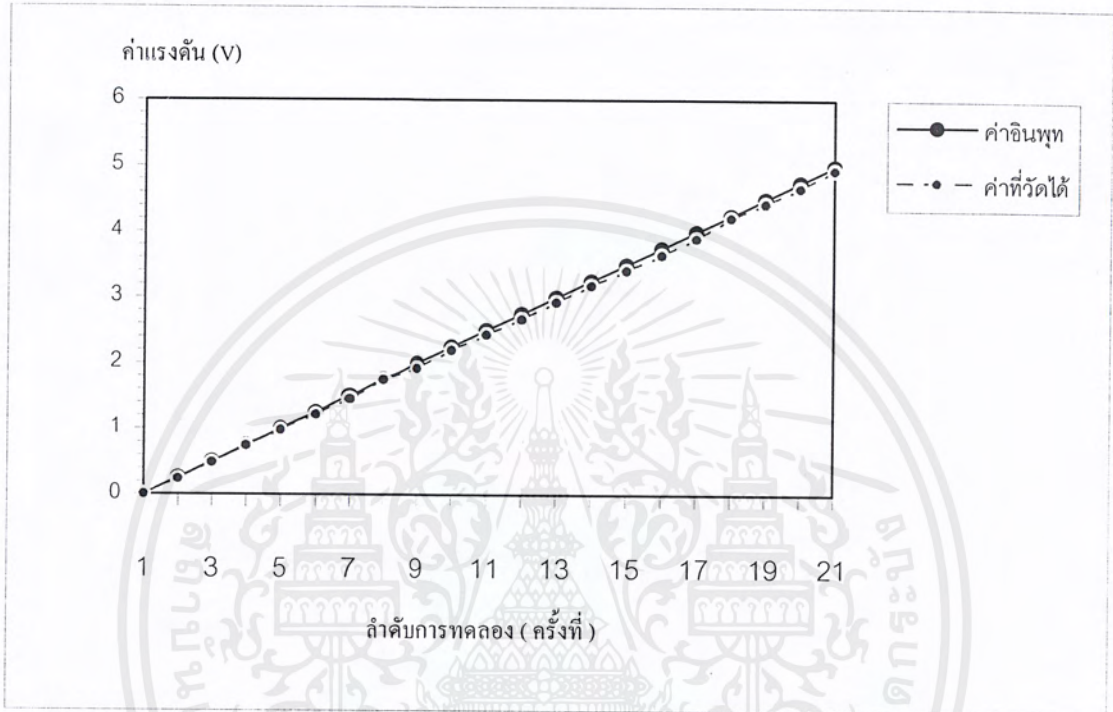
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.2 ผลการทดลองของวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล

Input Voltage (V)	ผลการทดลอง	ค่าที่ได้จากการคำนวณ	ค่าผิดพลาด (%)
0.00	0000 0000	0.000	0
0.25	0000 1100	0.235	5.87
0.50	0001 1001	0.490	1.95
0.75	0010 0110	0.745	0.64
1.00	0011 0010	0.981	1.95
1.25	0011 1110	1.216	2.73
1.50	0100 1010	1.451	3.26
1.75	0101 1001	1.745	0.27
2.00	0110 0010	1.922	3.91
2.25	0111 0000	2.196	2.36
2.50	0111 1100	2.432	2.73
2.75	1000 1000	2.667	3.02
3.00	1001 0101	2.929	2.06
3.25	1010 0010	3.177	2.25
3.50	1010 1110	3.412	2.51
3.75	1011 1010	3.647	2.73
4.00	1100 0111	3.902	2.44
4.25	1101 0111	4.216	0.80
4.50	1110 0010	4.432	1.51
4.75	1110 1110	4.667	1.74
5.00	1111 1100	4.942	1.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองสามารถนำมาแสดงผลในรูปของกราฟได้ดังรูปที่ 6.2 และจะพบว่าวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลที่ใช้ไอซี ADC 0804 มีค่าความผิดพลาดสูงสุดประมาณ 5 เปอร์เซ็นต์



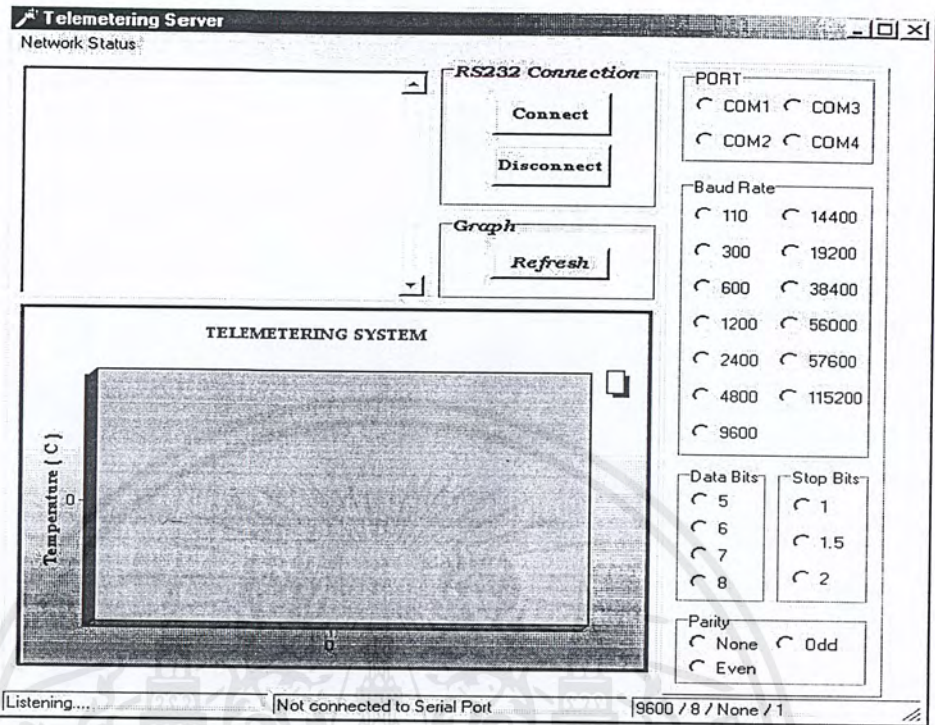
รูปที่ 6.2 กราฟเปรียบเทียบค่าที่วัดได้กับค่าอินพุทที่ป้อนให้วงจรแปลงสัญญาณ

### 6.3 ส่วนของโปรแกรม

เราจะทำการตรวจสอบการทำงานของโปรแกรม Telemetry Server / Client ซึ่งพัฒนาขึ้นด้วยภาษาเดลไฟต์ซึ่งมีความสามารถในการเชื่อมต่อกับโครงงานผ่านพอร์ตอนุกรมและสามารถโอนถ่ายข้อมูลที่ได้จากโครงงานผ่านเครื่องข่ายอินเทอร์เน็ตได้ โดยการใช้งานมีขั้นตอนดังต่อไปนี้

1. นำโครงงานมาเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ส่วนบุคคล ( PC ) ซึ่งส่วนใหญ่จะมีพอร์ตอนุกรม 2 พอร์ต คือ COM1 และ COM2 โดยในโครงงานนี้ใช้พอร์ต COM1
2. ทำการเรียกโปรแกรม Telemetry Server ขึ้นมาด้วยการรันไฟล์ Server.exe บนระบบปฏิบัติการวินโดวส์ 95 / 98 / NT จะได้หน้าต่างที่มีลักษณะดังรูปที่ 6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 หน้าต่างของโปรแกรม Telemetering Server

3. ทำการเลือกคุณสมบัติการเชื่อมต่อกับพอร์ตอนุกรม ไม่ว่าจะเป็น อัตราความเร็ว ( Baud Rate ) , บิตพาริตี , จำนวนบิตข้อมูล , จำนวนบิตหยุดและพอร์ตที่จะติดต่อ โดยในโครงการจะใช้เป็นพอร์ต COM1 , อัตราความเร็ว 9600 บิตต่อวินาที , ไม่มีพาริตีบิต , บิตข้อมูล 8 บิต และบิตหยุด 1 บิต

4. ทำการทดลองในการเชื่อมต่อและรับค่าจากพอร์ตอนุกรม โดยทำการส่งสัญญาณดิจิตอล 8 บิต ค่าต่างๆไปที่ MSC-51 แล้วสังเกตค่าที่อ่านได้จากโปรแกรมนำมาเปรียบกับค่าอินพุตได้ผลดังตารางที่ 6.3

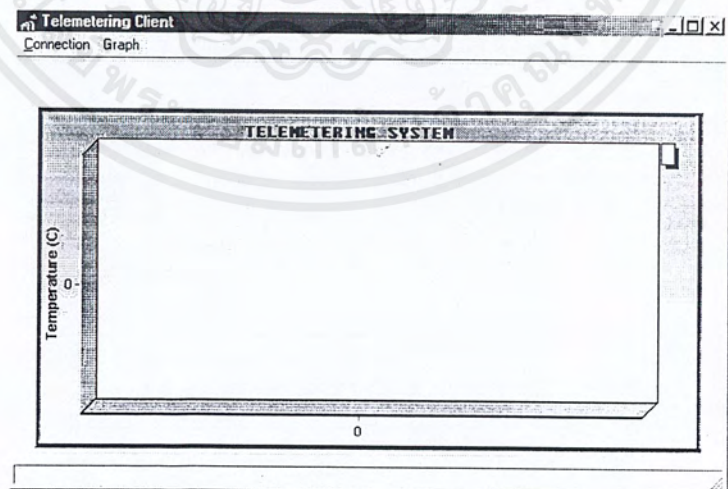
ตารางที่ 6.3 ค่าที่ได้จากการเชื่อมต่อพอร์ตอนุกรม

ค่าดิจิตอลอินพุต	ค่าที่อ่านได้ที่โปรแกรม	ค่าได้จากการคำนวณ	ค่าผิดพลาด (%)
0000 0000	0	0	0
0000 0001	1	1	0
0000 0010	2	2	0
0000 1000	8	8	0
0010 0000	32	32	0

ค่าดิจิตอลอินพุท	ค่าที่อ่านได้ที่โปรแกรม	ค่าได้จากการคำนวณ	ค่าผิดพลาด (%)
1000 0000	128	128	0
0111 1111	127	127	0
1011 1111	191	191	0
1101 1111	223	223	0
1110 1111	239	239	0
1111 0111	247	247	0
1111 1011	251	251	0
1111 1101	253	253	0
1111 1110	254	254	0
1111 1111	255	255	0

จากผลการทดลองจะเห็นว่าการรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ MCS-51 กับโปรแกรมผ่านพอร์ตอนุกรมไม่มีความผิดพลาด

5. ทำการทดสอบโปรแกรมการเชื่อมต่อผ่านเครื่องข่ายอินเทอร์เน็ตโดยการเรียกโปรแกรม Telemetry Client ด้วยการรัน Client.exe บนคอมพิวเตอร์ที่ทำการเชื่อมต่ออยู่ในระบบเครือข่าย จะได้น้ำต่างดังรูปที่ 6.4



รูปที่ 6.4 หน้าต่างของโปรแกรม Telemetry Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทำการติดต่อไปยังโปรแกรม Telemetry Server ที่รันอยู่บนเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายอินเทอร์เน็ต และอยู่ในสถานะ Listening.. การติดต่อทำได้โดยการกดปุ่ม Connect แล้วใส่ค่า IP Address

8. ทำการทดลองเช่นเดียวกับโปรแกรม Telemetry Server จะ ได้ผลดังตารางที่ 6.4

ตารางที่ 6.4 ค่าที่ได้จากการติดต่อส่งข้อมูลผ่านเครือข่าย

ค่าดิจิตอลอินพุท	ค่าที่อ่านได้ที่โปรแกรม	ค่าได้จากการคำนวณ	ค่าผิดพลาด (%)
0000 0000	0	0	0
0000 0001	1	1	0
0000 0010	2	2	0
0000 0100	4	4	0
0000 1000	8	8	0
0001 0000	16	16	0
0010 0000	32	32	0
0100 0000	64	64	0
1000 0000	128	128	0
0111 1111	127	127	0
1011 1111	191	191	0
1101 1111	223	223	0
1110 1111	239	239	0
1111 0111	247	247	0
1111 1011	251	251	0
1111 1101	253	253	0
1111 1110	254	254	0
1111 1111	255	255	0

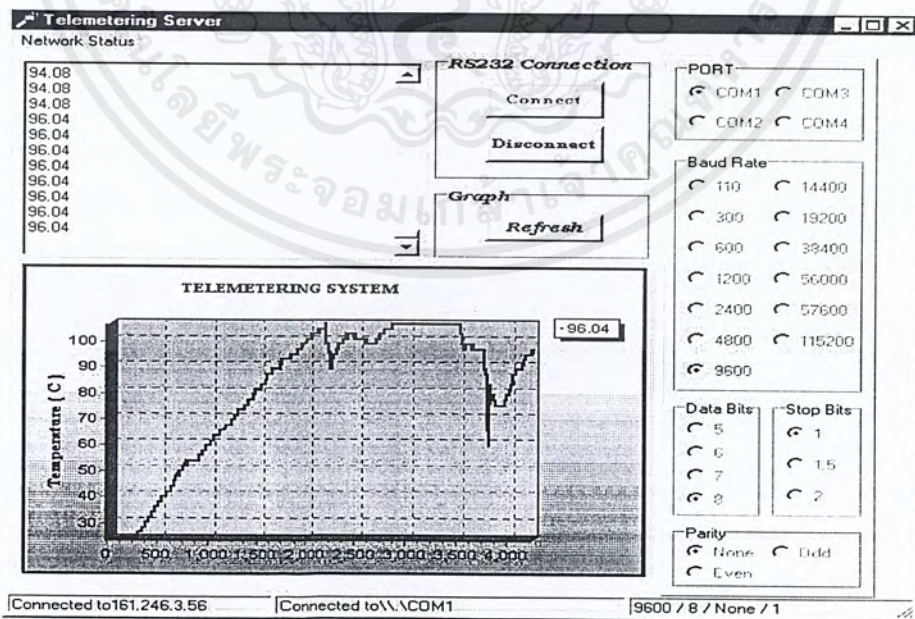
จะเห็นได้ว่าการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตด้วยโปรแกรม Telemetry Server/Client สามารถรับค่าได้อย่างครบถ้วน ไม่มีข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทำการทดสอบการใช้งานจริงของโครงการ โดยทำการวัดค่าของน้ำในหม้อต้มคุณภาพสูงที่สามารถทำอุณหภูมิได้ประมาณ  $125^{\circ}\text{C}$  ดังแสดงดังรูปที่ 6.5 เป็นระยะเวลา 3 ชั่วโมง โดยทำการปรับเปลี่ยนอุณหภูมิของน้ำในหม้อต้มด้วยการเติมน้ำเย็นทุก 20 นาที ผลที่ได้แสดงดังรูปที่ 6.6 และ 6.7 ตามลำดับ

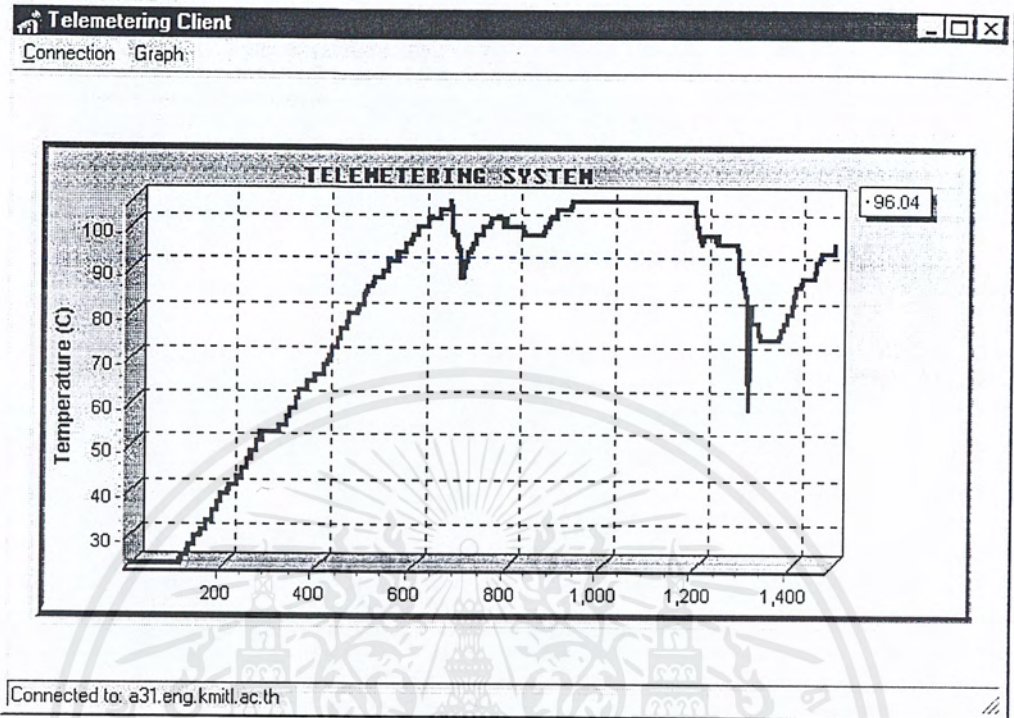


รูปที่ 6.5 การทดสอบการใช้งานจริง



รูปที่ 6.6 ผลการทดลองของโปรแกรม Telemetering Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 ผลการทดลองของ โปรแกรม Telemetering Client

จากรูปที่ 6.6 และ 6.7 จะเห็นว่าทั้งฝ่ายผู้ให้บริการและฝ่ายผู้ให้บริการแสดงผลผ่านการเขียนกราฟได้กราฟที่มีลักษณะเดียวกันและอ่านได้ค่าเดียวกันที่เวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### สรุปและวิจารณ์ผลการทดลอง

#### 7.1 สรุปผลการทดลอง

##### 7.1.1 ส่วนของวงจรวัดอุณหภูมิ

จากผลการทดลองในตารางที่ 6.1 จะเห็นได้ว่าเมื่อทำการเปรียบเทียบค่าที่ได้จากวงจรกับค่าที่ได้จากเทอร์โมมิเตอร์แบบปรอทมีค่าใกล้เคียงกันมากมีความผิดพลาดสูงสุดเพียง 6.67 % แต่ยังมีข้อจำกัดเกิดขึ้นกับวงจรวัดคือการวัดไม่เป็นเชิงเส้นในช่วงอุณหภูมิสูงๆดังจะเห็นได้จากรูปที่ 6.6 และ 6.7 ซึ่งเป็นผลมาจากคุณสมบัติของเทอร์โมคัปเปิลเอง ซึ่งควรทำการปรับแต่งและพัฒนาต่อไป

##### 7.1.2 ส่วนของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล

จากผลการทดลองในตารางที่ 6.2 จะเห็นได้ว่าความผิดพลาดสูงสุด คือ 5.87 % ที่ค่าแรงดันอินพุต 0.25 V การที่มีค่าความผิดพลาดสูงเนื่องจากการทำงานของ ADC 0804 จะแปลงสัญญาณอนาล็อกขนาด 0 ถึง 5 V ไปเป็นสัญญาณดิจิตอล 0 ถึง 255 ( 0000 0000 ถึง 1111 1111 ) จะเห็นได้ว่าหนึ่งระดับของสัญญาณดิจิตอลจะมีค่าที่เทียบกับการเปลี่ยนแปลงของสัญญาณอนาล็อกประมาณ 19.608 mV ดังนั้นที่ค่าอินพุต 0.25 V สัญญาณดิจิตอลต้องเปลี่ยนไปเท่ากับ 12.75 ระดับ จึงได้ค่าที่ออกมาเป็น 0000 1100 ( 12 ) เมื่อแปลงกลับเป็นสัญญาณจะเท่ากับ 0.235 V ซึ่งตรงกับค่าที่วัดได้นั่นเอง แสดงว่าการทำงานของวงจรไม่มีข้อผิดพลาด ค่าที่ผิดพลาดเกิดจากข้อจำกัดของตัวอุปกรณ์ ACD 0804 เอง อย่างไรก็ตามความผิดพลาดก็อยู่ในระดับที่ยอมรับได้ แนวทางแก้ไขถ้าต้องการความแม่นยำมากก็ต้องเลือกใช้ไอซีที่มีความละเอียดมากกว่า 8 บิต

##### 7.1.3 ส่วนของโปรแกรม

1. ส่วนของโปรแกรมการรับค่าจากพอร์ตอนุกรม จากผลการทดลองในตารางที่ 6.3 จะเห็นได้ว่าการรับค่าจากพอร์ตอนุกรมโดยรับค่ามาในรูปแบบของรหัส ASCII ตามมาตรฐาน UART แล้วนำมาแปลงเป็นค่าตัวเลขประจำของอักขระ ASCII นั้นๆ ให้ผลที่ถูกต้องแม่นยำมากสามารถรับค่าได้ครบทุกค่าตั้งแต่ 0 ถึง 255

2. ส่วนของโปรแกรมการติดต่อและถ่ายโอนข้อมูลระหว่างผู้ให้บริการ ( Server ) กับผู้ใช้บริการ ( Client ) หลังจากทำการทดสอบโดยทำการเชื่อมต่อ Server เข้ากับเครื่องคอมพิวเตอร์หลายๆเครื่องที่มีค่า IP Address ต่างๆกัน แล้วทดสอบการติดต่อจาก Client ที่ต่างๆ พบว่าการติดต่อ

สามารถทำได้อย่างสมบูรณ์ทุกค่า IP Address ส่วนการถ่ายโอนข้อมูลซึ่งได้ผลดังตารางที่ 6.4 จะเห็นได้ว่าทำได้อย่างถูกต้องเช่นเดียวกัน

3. หลังจากการทดสอบด้วยการให้ทำการวัดค่าของอุณหภูมิของน้ำในหม้อต้มอย่างต่อเนื่องเป็นเวลา 3 ชั่วโมงพบว่าสามารถแสดงผลได้อย่างต่อเนื่อง โดยค่าที่วัดได้ทั้งที่ส่วน Server และ Client มีค่าตรงกันโดยตลอดดังแสดงในรูปที่ 6.6 และ 6.7 แสดงว่าโครงการสามารถใช้งานในการสังเกตผลการวัดในการใช้งานจริงได้อย่างต่อเนื่องเป็นเวลานาน

## 7.2 คุณสมบัติของโครงการและโปรแกรม Server / Client

1. โครงการสามารถเชื่อมต่อกับเครื่องมือวัดที่ให้ค่าเอาต์พุตเป็นค่าแรงดันได้หลายชนิด ไม่ว่าจะเป็น แรงดัน , มลพิษ , ความชื้น เป็นต้น
2. โปรแกรมสามารถปฏิบัติการได้บนระบบวินโดวส์ 98 / NT / ME ทำให้สามารถทำงานโปรแกรมอื่นๆได้ในขณะที่ทำการสังเกตผลการวัด
3. โปรแกรมมีขนาดไม่ใหญ่จึงไม่ต้องการทรัพยากรจำนวนมากมาสนับสนุน จึงสามารถทำงานได้บนคอมพิวเตอร์รุ่น 486 ขึ้นไป
4. สามารถวัดค่าอุณหภูมิได้ตั้งแต่  $0^{\circ}\text{C}$  ถึง  $100^{\circ}\text{C}$
5. โปรแกรมสามารถเลือกการเชื่อมต่อผ่านพอร์ตอนุกรมได้ 4 พอร์ต คือ COM1 , COM2 , COM3 และ COM4
6. โปรแกรมสามารถเลือกอัตราความเร็ว ( Baud Rate ) ได้
7. โปรแกรมสามารถเลือกบิตข้อมูลได้ตั้งแต่ 5 , 6 , 7 จนถึง 8 บิต และบิตหยุดได้ตั้งแต่ 1 , 1.5 จนถึง 2 บิต
8. โปรแกรมสามารถเลือกการตรวจสอบพาริตีได้ 3 แบบ คือ None ( ไม่มีการตรวจสอบ ) Even ( พาริตีคู่ ) , Odd ( พาริตีคี่ )
9. สามารถใช้งานโปรแกรม Telemetry Server เป็นโปรแกรมในการรับค่าจากพอร์ตอนุกรมได้
10. สามารถทำการทำการเชื่อมต่อและยกเลิกการเชื่อมต่อกับพอร์ตอนุกรมผ่านทางผู้ให้บริการได้

## 7.3 ข้อจำกัดของโครงการและแนวทางพัฒนาแก้ไข

1. โครงการไม่สามารถทำการวัดค่าอุณหภูมิที่มีค่าต่ำกว่า  $0^{\circ}\text{C}$  ได้ มีแนวทางแก้ไขโดยเมื่อมีค่าแรงดันเอาต์พุตที่เป็นลบออกมาจากเครื่องมือวัดให้ทำการกลับเป็นค่าบวกแล้วส่งสัญญาณให้ไมโครคอนโทรลเลอร์ทราบ จากนั้นอาจมีการเพิ่มบิตพิเศษอีก 1 บิตในการติดต่อแบบอนุกรมเพื่อแจ้งให้โปรแกรมทราบเพื่อแสดงผลได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โครงการงานทำการเชื่อมต่อกับเครื่องมือวัดได้เพียงตัวเดียว มีแนวทางแก้ไขโดยใช้วงจร Multiplex แล้วใช้ไมโครคอนโทรลเลอร์ควบคุมโดยการให้แต่ละ Channel ผลัดกันส่งข้อมูลตามลำดับและเวลาที่กำหนด
3. การเชื่อมต่อกับพอร์ตอนุกรมโดยใช้บิตข้อมูล 6 , 7 หรือ 8 บิต กับบิตหยุด 1.5 บิตไม่สามารถทำได้เนื่องจากข้อจำกัดของComponent CommPort Driver ที่ใช้ในการพัฒนาโปรแกรม อาจแก้ไขโดยใช้ Componet ตัวอื่นๆ
4. ความเร็วของการติดต่อผ่านเครือข่ายในการโอนย้ายข้อมูลจะขึ้นอยู่กับเครือข่ายที่ใช้ งาน และความสามารถของเครื่องคอมพิวเตอร์ที่ใช้ปฏิบัติงาน

การที่จะปรับปรุงข้อจำกัดเหล่านี้ก็จะพบปัญหาทางด้านความจำกัดของเวลาเนื่องจากการหาข้อมูลในบางเรื่องทำได้ยากต้องอาศัยการทดลองแล้วนำผลที่ได้มีพัฒนาและประยุกต์ใช้ใน โครงการงาน อีกประการหนึ่งคือข้อจำกัดด้านค่าใช้จ่ายเนื่องจากอุปกรณ์ที่ใช้งานมีราคาแพงมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงค่าแรงดันต่ออุณหภูมิของเทอร์โมคัปเปิลชนิด K ที่เป็น NiCr-Ni

°C	0	-10	-20	-30	-40	-50	-60	-70	-80	-90
-200	-5.891	-6.035	-6.158	-6.262	-6.344	-6.404	-6.441	-6.458		
-100	-3.553	-3.852	-4.138	-4.410	-4.669	-4.912	-5.141	-5.354	-5.550	-5.730
0	0	-0.392	-0.777	-1.156	-1.527	-1.889	-2.243	-2.586	-2.920	-3.242
°C	0	10	20	30	40	50	60	70	80	90
0	0	0.397	0.798	1.203	1.611	2.022	2.436	2.850	3.266	3.681
100	4.095	4.508	4.919	5.327	5.733	6.137	6.539	6.939	7.338	7.737
200	8.137	8.537	8.938	9.341	9.745	10.151	10.560	10.969	11.381	11.793
300	12.207	12.623	13.039	13.456	13.874	14.292	14.712	15.132	15.552	15.974
400	16.395	16.818	17.241	17.664	18.088	18.513	18.938	19.363	19.788	20.214
500	20.640	21.066	21.493	21.919	22.346	22.772	23.198	23.624	24.050	24.476
600	24.902	25.327	25.751	26.176	26.599	27.022	27.445	27.867	28.288	28.709
700	29.128	29.547	29.965	30.383	30.799	31.214	31.629	32.042	32.455	32.866
800	33.277	33.686	34.095	34.502	34.909	35.314	35.718	36.121	36.524	36.925
900	37.325	37.724	38.122	38.519	38.915	39.310	39.703	40.096	40.488	40.879
1000	41.269	41.657	42.045	42.432	42.817	43.202	43.585	43.968	44.349	44.729
1100	45.108	45.486	45.863	46.238	46.612	46.985	47.356	47.726	48.095	48.462
1200	48.828	49.192	49.555	49.916	50.276	50.633	50.990	51.344	51.697	52.049
1300	52.398	52.747	53.093	53.439	53.782	54.125	54.466	54.807		

The Nickel-Chromium/Nickel reference tables were developed from the following polynomials:

temperature range  $-270^{\circ}\text{C}$  to  $0^{\circ}\text{C}$

temperature range  $0^{\circ}\text{C}$  to  $1372^{\circ}\text{C}$

$$E = \sum_{i=0}^{10} a_i t_{68}^i \mu\text{V}$$

$$E = \sum_{i=0}^8 b_i t_{68}^i + 125 \exp \left[ -1/2 \left( \frac{t_{68} - 127}{65} \right)^2 \right] \mu\text{V}$$

$$\begin{aligned} a_0 &= 0 \\ a_1 &= 3.9475433139 \times 10^{-1} \\ a_2 &= 2.7465251138 \times 10^{-2} \\ a_3 &= -1.6565406716 \times 10^{-4} \\ a_4 &= -1.5190912392 \times 10^{-6} \\ a_5 &= -2.4581670924 \times 10^{-8} \\ a_6 &= -2.4757917816 \times 10^{-10} \\ a_7 &= -1.5585276173 \times 10^{-12} \\ a_8 &= -5.9729921255 \times 10^{-15} \\ a_9 &= -1.2688801216 \times 10^{-17} \\ a_{10} &= -1.1382797374 \times 10^{-20} \end{aligned}$$

$$\begin{aligned} b_0 &= -1.8533063273 \times 10^{-1} \\ b_1 &= 3.8918344612 \times 10^{-1} \\ b_2 &= 1.6645154356 \times 10^{-2} \\ b_3 &= -7.8702374448 \times 10^{-5} \\ b_4 &= 2.2835785557 \times 10^{-7} \\ b_5 &= -3.5700231258 \times 10^{-10} \\ b_6 &= 2.9932909136 \times 10^{-13} \\ b_7 &= -1.2849848798 \times 10^{-16} \\ b_8 &= 2.2239974336 \times 10^{-20} \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



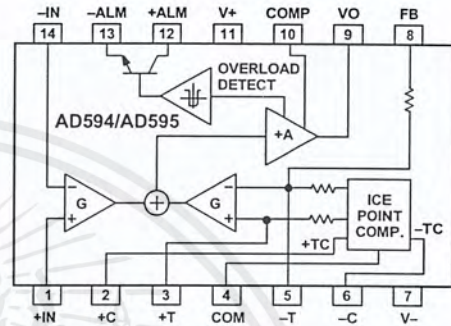
# Monolithic Thermocouple Amplifiers with Cold Junction Compensation

## AD594/AD595

### FEATURES

- Pretrimmed for Type J (AD594) or Type K (AD595) Thermocouples
- Can Be Used with Type T Thermocouple Inputs
- Low Impedance Voltage Output: 10 mV/°C
- Built-In Ice Point Compensation
- Wide Power Supply Range: +5 V to ±15 V
- Low Power: <1 mW typical
- Thermocouple Failure Alarm
- Laser Wafer Trimmed to 1°C Calibration Accuracy
- Setpoint Mode Operation
- Self-Contained Celsius Thermometer Operation
- High Impedance Differential Input
- Side-Brazed DIP or Low Cost Cerdip

### FUNCTIONAL BLOCK DIAGRAM



### PRODUCT DESCRIPTION

The AD594/AD595 is a complete instrumentation amplifier and thermocouple cold junction compensator on a monolithic chip. It combines an ice point reference with a precalibrated amplifier to produce a high level (10 mV/°C) output directly from a thermocouple signal. Pin-strapping options allow it to be used as a linear amplifier-compensator or as a switched output setpoint controller using either fixed or remote setpoint control. It can be used to amplify its compensation voltage directly, thereby converting it to a stand-alone Celsius transducer with a low impedance voltage output.

The AD594/AD595 includes a thermocouple failure alarm that indicates if one or both thermocouple leads become open. The alarm output has a flexible format which includes TTL drive capability.

The AD594/AD595 can be powered from a single ended supply (including +5 V) and by including a negative supply, temperatures below 0°C can be measured. To minimize self-heating, an unloaded AD594/AD595 will typically operate with a total supply current 160 µA, but is also capable of delivering in excess of ±5 mA to a load.

The AD594 is precalibrated by laser wafer trimming to match the characteristic of type J (iron-constantan) thermocouples and the AD595 is laser trimmed for type K (chromel-alumel) inputs. The temperature transducer voltages and gain control resistors

are available at the package pins so that the circuit can be recalibrated for the thermocouple types by the addition of two or three resistors. These terminals also allow more precise calibration for both thermocouple and thermometer applications.

The AD594/AD595 is available in two performance grades. The C and the A versions have calibration accuracies of ±1°C and ±3°C, respectively. Both are designed to be used from 0°C to +50°C, and are available in 14-pin, hermetically sealed, side-brazed ceramic DIPs as well as low cost cerdip packages.

### PRODUCT HIGHLIGHTS

1. The AD594/AD595 provides cold junction compensation, amplification, and an output buffer in a single IC package.
2. Compensation, zero, and scale factor are all precalibrated by laser wafer trimming (LWT) of each IC chip.
3. Flexible pinout provides for operation as a setpoint controller or a stand-alone temperature transducer calibrated in degrees Celsius.
4. Operation at remote application sites is facilitated by low quiescent current and a wide supply voltage range +5 V to dual supplies spanning 30 V.
5. Differential input rejects common-mode noise voltage on the thermocouple leads.

### REV. C

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>  
Fax: 781/326-8703 © Analog Devices, Inc., 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AD594/AD595—SPECIFICATIONS (@ +25°C and $V_S = 5\text{ V}$ , Type J (AD594), Type K (AD595) Thermocouple, unless otherwise noted)

Model	AD594A			AD594C			AD595A			AD595C			Units
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
<b>ABSOLUTE MAXIMUM RATING</b>													
+ $V_S$ to $-V_S$		36			36			36			36		Volts
Common-Mode Input Voltage	$-V_S - 0.15$	$+V_S$		$-V_S - 0.15$	$+V_S$		$-V_S - 0.15$	$+V_S$		$-V_S - 0.15$	$+V_S$		Volts
Differential Input Voltage	$-V_S$	$+V_S$		$-V_S$	$+V_S$		$-V_S$	$+V_S$		$-V_S$	$+V_S$		Volts
Alarm Voltages													
+ALM	$-V_S$	$-V_S + 36$		$-V_S$	$-V_S + 36$		$-V_S$	$-V_S + 36$		$-V_S$	$-V_S + 36$		Volts
-ALM	$-V_S$	$+V_S$		$-V_S$	$+V_S$		$-V_S$	$+V_S$		$-V_S$	$+V_S$		Volts
Operating Temperature Range	-55	+125		-55	+125		-55	+125		-55	+125		°C
Output Short Circuit to Common	Indefinite			Indefinite			Indefinite			Indefinite			
<b>TEMPERATURE MEASUREMENT</b> (Specified Temperature Range 0°C to +50°C)													
Calibration Error at +25°C <sup>1</sup>		±3			±1			±3			±1		°C
Stability vs. Temperature <sup>2</sup>		±0.05			±0.025			±0.05			±0.025		°C/°C
Gain Error		±1.5			±0.75			±1.5			±0.75		%
Nominal Transfer Function		10			10			10			10		mV/°C
<b>AMPLIFIER CHARACTERISTICS</b>													
Closed Loop Gain <sup>3</sup>		193.4			193.4			247.3			247.3		
Input Offset Voltage		(Temperature in °C) × 51.70 μV/°C			(Temperature in °C) × 51.70 μV/°C			(Temperature in °C) × 40.44 μV/°C			(Temperature in °C) × 40.44 μV/°C		μV
Input Bias Current		0.1			0.1			0.1			0.1		μA
Differential Input Range	-10	+50		-10	+50		-10	+50		-10	+50		mV
Common-Mode Range	$-V_S - 0.15$	$-V_S - 4$		$-V_S - 0.15$	$-V_S - 4$		$-V_S - 0.15$	$-V_S - 4$		$-V_S - 0.15$	$-V_S - 4$		Volts
Common-Mode Sensitivity - RTO		10			10			10			10		mV/V
Power Supply Sensitivity - RTO		10			10			10			10		mV/V
Output Voltage Range													
Dual Supply	$-V_S + 2.5$	$+V_S - 2$		$-V_S + 2.5$	$+V_S - 2$		$-V_S + 2.5$	$+V_S - 2$		$-V_S + 2.5$	$+V_S - 2$		Volts
Single Supply	0	$+V_S - 2$		0	$-V_S - 2$		0	$+V_S + 2$		0	$+V_S - 2$		Volts
Usable Output Current <sup>4</sup>		±5			±5			±5			±5		mA
3 dB Bandwidth		15			15			15			15		kHz
<b>ALARM CHARACTERISTICS</b>													
$V_{CE(SAT)}$ at 2 mA		0.3			0.3			0.3			0.3		Volts
Leakage Current		±1			±1			±1			±1		μA max
Operating Voltage at -ALM		$+V_S - 4$			$+V_S - 4$			$+V_S - 4$			$+V_S - 4$		Volts
Short Circuit Current		20			20			20			20		mA
<b>POWER REQUIREMENTS</b>													
Specified Performance		$+V_S = 5, -V_S = 0$			$+V_S = 5, -V_S = 0$			$+V_S = 5, -V_S = 0$			$+V_S = 5, -V_S = 0$		Volts
Operating <sup>5</sup>		$+V_S$ to $-V_S \leq 30$			$+V_S$ to $-V_S \leq 30$			$+V_S$ to $-V_S \leq 30$			$+V_S$ to $-V_S \leq 30$		Volts
Quiescent Current (No Load)													
+ $V_S$		160	300		160	300		160	300		160	300	μA
- $V_S$		100			100			100			100		μA
<b>PACKAGE OPTION</b>													
TO-116 (D-14)		AD594AD			AD594CD			AD595AD			AD595CD		
CerDip (Q-14)		AD594AQ			AD594CQ			AD595AQ			AD595CQ		

## NOTES

<sup>1</sup>Calibrated for minimum error at +25°C using a thermocouple sensitivity of 51.7 μV/°C. Since a J type thermocouple deviates from this straight line approximation, the AD594 will normally read 3.1 mV when the measuring junction is at 0°C. The AD595 will similarly read 2.7 mV at 0°C.

<sup>2</sup>Defined as the slope of the line connecting the AD594/AD595 errors measured at 0°C and 50°C ambient temperature.

<sup>3</sup>Pin 8 shorted to Pin 9.

<sup>4</sup>Current Sink Capability in single supply configuration is limited to current drawn to ground through a 50 kΩ resistor at output voltages below 2.5 V.

<sup>5</sup>- $V_S$  must not exceed -16.5 V.

Specifications shown in boldface are tested on all production units at final electrical test. Results from those tests are used to calculate outgoing quality levels. All min and max specifications are guaranteed, although only those shown in boldface are tested on all production units.

Specifications subject to change without notice.

## INTERPRETING AD594/AD595 OUTPUT VOLTAGES

To achieve a temperature proportional output of 10 mV/°C and accurately compensate for the reference junction over the rated operating range of the circuit, the AD594/AD595 is gain trimmed to match the transfer characteristic of J and K type thermocouples at 25°C. For a type J output in this temperature range the TC is 51.70 μV/°C, while for a type K it is 40.44 μV/°C. The resulting gain for the AD594 is 193.4 (10 mV/°C divided by 51.7 μV/°C) and for the AD595 is 247.3 (10 mV/°C divided by 40.44 μV/°C). In addition, an absolute accuracy trim induces an input offset to the output amplifier characteristic of 16 μV for the AD594 and 11 μV for the AD595. This offset arises because the AD594/AD595 is trimmed for a 250 mV output while applying a 25°C thermocouple input.

Because a thermocouple output voltage is nonlinear with respect to temperature, and the AD594/AD595 linearly amplifies the

compensated signal, the following transfer functions should be used to determine the actual output voltages:

$$AD594 \text{ output} = (\text{Type J Voltage} + 16 \mu\text{V}) \times 193.4$$

$$AD595 \text{ output} = (\text{Type K Voltage} + 11 \mu\text{V}) \times 247.3 \text{ or conversely:}$$

$$\text{Type J voltage} = (AD594 \text{ output}/193.4) - 16 \mu\text{V}$$

$$\text{Type K voltage} = (AD595 \text{ output}/247.3) - 11 \mu\text{V}$$

Table I lists the ideal AD594/AD595 output voltages as a function of Celsius temperature for type J and K ANSI standard thermocouples, with the package and reference junction at 25°C. As is normally the case, these outputs are subject to calibration, gain and temperature sensitivity errors. Output values for intermediate temperatures can be interpolated, or calculated using the output equations and ANSI thermocouple voltage tables referred to zero degrees Celsius. Due to a slight variation in alloy content between ANSI type J and DIN FE-CuNi

# AD594/AD595

**Table I. Output Voltage vs. Thermocouple Temperature (Ambient +25°C,  $V_S = -5\text{ V}, +15\text{ V}$ )**

Thermocouple Temperature °C	Type J Voltage mV	AD594 Output mV	Type K Voltage mV	AD595 Output mV
-200	-7.890	-1523	-5.891	-1454
-180	-7.402	-1428	-5.550	-1370
-160	-6.821	-1316	-5.141	-1269
-140	-6.159	-1188	-4.669	-1152
-120	-5.426	-1046	-4.138	-1021
-100	-4.632	-893	-3.553	-876
-80	-3.785	-729	-2.920	-719
-60	-2.892	-556	-2.243	-552
-40	-1.960	-376	-1.527	-375
-20	-0.995	-189	-0.777	-189
-10	-0.501	-94	-0.392	-94
0	0	3.1	0	2.7
10	.507	101	.397	101
20	1.019	200	.798	200
25	1.277	250	1.000	250
30	1.536	300	1.203	300
40	2.058	401	1.611	401
50	2.585	503	2.022	503
60	3.115	606	2.436	605
80	4.186	813	3.266	810
100	5.268	1022	4.095	1015
120	6.359	1233	4.919	1219
140	7.457	1445	5.733	1420
160	8.560	1659	6.539	1620
180	9.667	1873	7.338	1817
200	10.777	2087	8.137	2015
220	11.887	2302	8.938	2213
240	12.998	2517	9.745	2413
260	14.108	2732	10.560	2614
280	15.217	2946	11.381	2817
300	16.325	3160	12.207	3022
320	17.432	3374	13.039	3227
340	18.537	3588	13.874	3434
360	19.640	3801	14.712	3641
380	20.743	4015	15.552	3849
400	21.846	4228	16.395	4057
420	22.949	4441	17.241	4266
440	24.054	4655	18.088	4476
460	25.161	4869	18.938	4686
480	26.272	5084	19.788	4896
500	27.388	5300	20.640	5107
520	28.511	5517	21.493	5318
540	29.642	5736	22.346	5529
560	30.782	5956	23.198	5740
580	31.933	6179	24.050	5950
600	33.096	6404	24.902	6161
620	34.273	6632	25.751	6371
640	35.464	6862	26.599	6581
660	36.671	7095	27.445	6790
680	37.893	7332	28.288	6998
700	39.130	7571	29.128	7206
720	40.382	7813	29.965	7413
740	41.647	8058	30.799	7619
750	42.283	8181	31.214	7722
760	-	-	31.629	7825
780	-	-	32.455	8029
800	-	-	33.277	8232
820	-	-	34.095	8434
840	-	-	34.909	8636
860	-	-	35.718	8836
880	-	-	36.524	9035
900	-	-	37.325	9233
920	-	-	38.122	9430
940	-	-	38.915	9626
960	-	-	39.703	9821
980	-	-	40.488	10015
1000	-	-	41.269	10209
1020	-	-	42.045	10400
1040	-	-	42.817	10591
1060	-	-	43.585	10781
1080	-	-	44.439	10970
1100	-	-	45.108	11158
1120	-	-	45.863	11345
1140	-	-	46.612	11530
1160	-	-	47.356	11714
1180	-	-	48.095	11897
1200	-	-	48.828	12078
1220	-	-	49.555	12258
1240	-	-	50.276	12436
1250	-	-	50.633	12524

thermocouples Table I should not be used in conjunction with European standard thermocouples. Instead the transfer function given previously and a DIN thermocouple table should be used. ANSI type K and DIN NiCr-Ni thermocouples are composed

### SINGLE AND DUAL SUPPLY CONNECTIONS

The AD594/AD595 is a completely self-contained thermocouple conditioner. Using a single +5 V supply the interconnections shown in Figure 1 will provide a direct output from a type J thermocouple (AD594) or type K thermocouple (AD595) measuring from 0°C to +300°C.

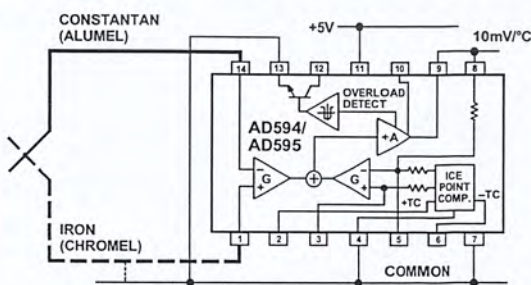


Figure 1. Basic Connection, Single Supply Operation

of identical alloys and exhibit similar behavior. The upper temperature limits in Table I are those recommended for type J and type K thermocouples by the majority of vendors.

Any convenient supply voltage from +5 V to +30 V may be used, with self-heating errors being minimized at lower supply levels. In the single supply configuration the +5 V supply connects to Pin 11 with the V- connection at Pin 7 strapped to power and signal common at Pin 4. The thermocouple wire inputs connect to Pins 1 and 14 either directly from the measuring point or through intervening connections of similar thermocouple wire type. When the alarm output at Pin 13 is not used it should be connected to common or -V. The precalibrated feedback network at Pin 8 is tied to the output at Pin 9 to provide a 10 mV/°C nominal temperature transfer characteristic.

By using a wider ranging dual supply, as shown in Figure 2, the AD594/AD595 can be interfaced to thermocouples measuring both negative and extended positive temperatures.

# AD594/AD595

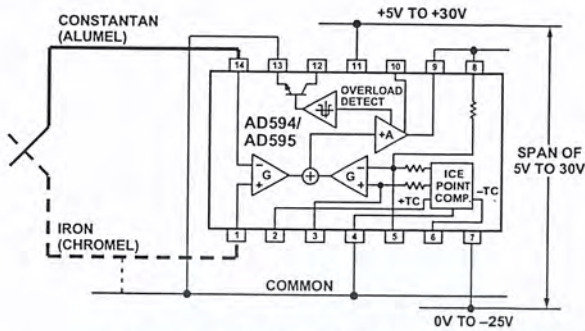


Figure 2. Dual Supply Operation

With a negative supply the output can indicate negative temperatures and drive grounded loads or loads returned to positive voltages. Increasing the positive supply from 5 V to 15 V extends the output voltage range well beyond the 750°C temperature limit recommended for type J thermocouples (AD594) and the 1250°C for type K thermocouples (AD595).

Common-mode voltages on the thermocouple inputs must remain within the common-mode range of the AD594/AD595, with a return path provided for the bias currents. If the thermocouple is not remotely grounded, then the dotted line connections in Figures 1 and 2 are recommended. A resistor may be needed in this connection to assure that common-mode voltages induced in the thermocouple loop are not converted to normal mode.

## THERMOCOUPLE CONNECTIONS

The isothermal terminating connections of a pair of thermocouple wires forms an effective reference junction. This junction must be kept at the same temperature as the AD594/AD595 for the internal cold junction compensation to be effective.

A method that provides for thermal equilibrium is the printed circuit board connection layout illustrated in Figure 3.

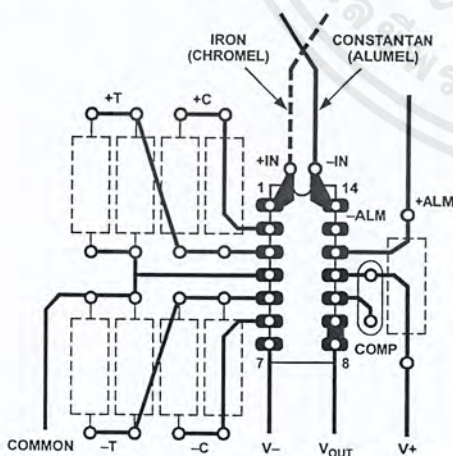


Figure 3. PCB Connections

Here the AD594/AD595 package temperature and circuit board are thermally contacted in the copper printed circuit board tracks under Pins 1 and 14. The reference junction is now composed of a copper-constantan (or copper-alumel) connection and copper-iron (or copper-chromel) connection, both of which are at the same temperature as the AD594/AD595.

The printed circuit board layout shown also provides for placement of optional alarm load resistors, recalibration resistors and a compensation capacitor to limit bandwidth.

To ensure secure bonding the thermocouple wire should be cleaned to remove oxidation prior to soldering. Noncorrosive rosin flux is effective with iron, constantan, chromel and alumel and the following solders: 95% tin-5% antimony, 95% tin-5% silver or 90% tin-10% lead.

## FUNCTIONAL DESCRIPTION

The AD594 behaves like two differential amplifiers. The outputs are summed and used to control a high gain amplifier, as shown in Figure 4.

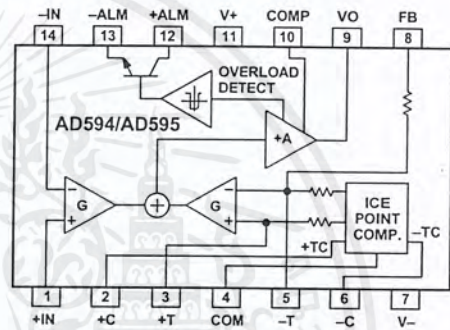


Figure 4. AD594/AD595 Block Diagram

In normal operation the main amplifier output, at Pin 9, is connected to the feedback network, at Pin 8. Thermocouple signals applied to the floating input stage, at Pins 1 and 14, are amplified by gain G of the differential amplifier and are then further amplified by gain A in the main amplifier. The output of the main amplifier is fed back to a second differential stage in an inverting connection. The feedback signal is amplified by this stage and is also applied to the main amplifier input through a summing circuit. Because of the inversion, the amplifier causes the feedback to be driven to reduce this difference signal to a small value. The two differential amplifiers are made to match and have identical gains, G. As a result, the feedback signal that must be applied to the right-hand differential amplifier will precisely match the thermocouple input signal when the difference signal has been reduced to zero. The feedback network is trimmed so that the effective gain to the output, at Pins 8 and 9, results in a voltage of 10 mV/°C of thermocouple excitation.

In addition to the feedback signal, a cold junction compensation voltage is applied to the right-hand differential amplifier. The compensation is a differential voltage proportional to the Celsius temperature of the AD594/AD595. This signal disturbs the differential input so that the amplifier output must adjust to restore the input to equal the applied thermocouple voltage.

The compensation is applied through the gain scaling resistors so that its effect on the main output is also 10 mV/°C. As a result, the compensation voltage adds to the effect of the thermocouple voltage a signal directly proportional to the difference between 0°C and the AD594/AD595 temperature. If the thermocouple reference junction is maintained at the AD594/AD595 temperature, the output of the AD594/AD595 will correspond to the reading that would have been obtained from amplification of a signal from a thermocouple referenced to an ice bath.

The AD594/AD595 also includes an input open circuit detector that switches on an alarm transistor. This transistor is actually a current-limited output buffer, but can be used up to the limit as a switch transistor for either pull-up or pull-down operation of external alarms.

The ice point compensation network has voltages available with positive and negative temperature coefficients. These voltages may be used with external resistors to modify the ice point compensation and recalibrate the AD594/AD595 as described in the next column.

The feedback resistor is separately pinned out so that its value can be padded with a series resistor, or replaced with an external resistor between Pins 5 and 9. External availability of the feedback resistor allows gain to be adjusted, and also permits the AD594/AD595 to operate in a switching mode for setpoint operation.

**CAUTIONS:**

The temperature compensation terminals (+C and -C) at Pins 2 and 6 are provided to supply small calibration currents only. The AD594/AD595 may be permanently damaged if they are grounded or connected to a low impedance.

The AD594/AD595 is internally frequency compensated for feedback ratios (corresponding to normal signal gain) of 75 or more. If a lower gain is desired, additional frequency compensation should be added in the form of a 300 pF capacitor from Pin 10 to the output at Pin 9. As shown in Figure 5 an additional 0.01 μF capacitor between Pins 10 and 11 is recommended.

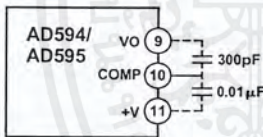


Figure 5. Low Gain Frequency Compensation

**RECALIBRATION PRINCIPLES AND LIMITATIONS**

The ice point compensation network of the AD594/AD595 produces a differential signal which is zero at 0°C and corresponds to the output of an ice referenced thermocouple at the temperature of the chip. The positive TC output of the circuit is proportional to Kelvin temperature and appears as a voltage at +T. It is possible to decrease this signal by loading it with a resistor from +T to COM, or increase it with a pull-up resistor from +T to the larger positive TC voltage at +C. Note that adjustments to +T should be made by measuring the voltage which tracks it at -T. To avoid destabilizing the feedback amplifier the measuring instrument should be isolated by a few thousand ohms in series with the lead connected to -T.

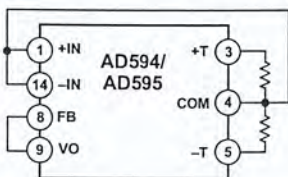


Figure 6. Decreased Sensitivity Adjustment

Changing the positive TC half of the differential output of the compensation scheme shifts the zero point away from 0°C. The zero can be restored by adjusting the current flow into the negative input of the feedback amplifier, the -T pin. A current into

this terminal can be produced with a resistor between -C and -T to balance an increase in +T, or a resistor from -T to COM to offset a decrease in +T.

If the compensation is adjusted substantially to accommodate a different thermocouple type, its effect on the final output voltage will increase or decrease in proportion. To restore the nominal output to 10 mV/°C the gain may be adjusted to match the new compensation and thermocouple input characteristics. When reducing the compensation the resistance between -T and COM automatically increases the gain to within 0.5% of the correct value. If a smaller gain is required, however, the nominal 47 kΩ internal feedback resistor can be paralleled or replaced with an external resistor.

Fine calibration adjustments will require temperature response measurements of individual devices to assure accuracy. Major reconfigurations for other thermocouple types can be achieved without seriously compromising initial calibration accuracy, so long as the procedure is done at a fixed temperature using the factory calibration as a reference. It should be noted that intermediate recalibration conditions may require the use of a negative supply.

**EXAMPLE: TYPE E RECALIBRATION—AD594/AD595**

Both the AD594 and AD595 can be configured to condition the output of a type E (chromel-constantan) thermocouple. Temperature characteristics of type E thermocouples differ less from type J, than from type K, therefore the AD594 is preferred for recalibration.

While maintaining the device at a constant temperature follow the recalibration steps given here. First, measure the device temperature by tying both inputs to common (or a selected common-mode potential) and connecting FB to VO. The AD594 is now in the stand alone Celsius thermometer mode. For this example assume the ambient is 24°C and the initial output VO is 240 mV. Check the output at VO to verify that it corresponds to the temperature of the device.

Next, measure the voltage -T at Pin 5 with a high impedance DVM (capacitance should be isolated by a few thousand ohms of resistance at the measured terminals). At 24°C the -T voltage will be about 8.3 mV. To adjust the compensation of an AD594 to a type E thermocouple a resistor, R1, should be connected between +T and +C, Pins 2 and 3, to raise the voltage at -T by the ratio of thermocouple sensitivities. The ratio for converting a type J device to a type E characteristic is:

$$r(AD594) = (60.9 \mu V/^\circ C) / (51.7 \mu V/^\circ C) = 1.18$$

Thus, multiply the initial voltage measured at -T by r and experimentally determine the R1 value required to raise -T to that level. For the example the new -T voltage should be about 9.8 mV. The resistance value should be approximately 1.8 kΩ.

The zero differential point must now be shifted back to 0°C. This is accomplished by multiplying the original output voltage VO by r and adjusting the measured output voltage to this value by experimentally adding a resistor, R2, between -C and -T, Pins 5 and 6. The target output value in this case should be about 283 mV. The resistance value of R2 should be approximately 240 kΩ.

Finally, the gain must be recalibrated such that the output VO indicates the device's temperature once again. Do this by adding a third resistor, R3, between FB and -T, Pins 8 and 5. VO should now be back to the initial 240 mV reading. The resistance value

REV. C านี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AD594/AD595

of R3 should be approximately 280 kΩ. The final connection diagram is shown in Figure 7. An approximate verification of the effectiveness of recalibration is to measure the differential gain to the output. For type E it should be 164.2.

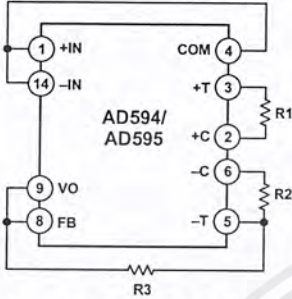


Figure 7. Type E Recalibration

When implementing a similar recalibration procedure for the AD595 the values for R1, R2, R3 and r will be approximately 650 Ω, 84 kΩ, 93 kΩ and 1.51, respectively. Power consumption will increase by about 50% when using the AD595 with type E inputs.

Note that during this procedure it is crucial to maintain the AD594/AD595 at a stable temperature because it is used as the temperature reference. Contact with fingers or any tools not at ambient temperature will quickly produce errors. Radiational heating from a change in lighting or approach of a soldering iron must also be guarded against.

### USING TYPE T THERMOCOUPLES WITH THE AD595

Because of the similarity of thermal EMFs in the 0°C to +50°C range between type K and type T thermocouples, the AD595 can be directly used with both types of inputs. Within this ambient temperature range the AD595 should exhibit no more than an additional 0.2°C output calibration error when used with type T inputs. The error arises because the ice point compensator is trimmed to type K characteristics at 25°C. To calculate the AD595 output values over the recommended -200°C to +350°C range for type T thermocouples, simply use the ANSI thermocouple voltages referred to 0°C and the output equation given on page 2 for the AD595. Because of the relatively large nonlinearities associated with type T thermocouples the output will deviate widely from the nominal 10 mV/°C. However, cold junction compensation over the rated 0°C to +50°C ambient will remain accurate.

### STABILITY OVER TEMPERATURE

Each AD594/AD595 is tested for error over temperature with the measuring thermocouple at 0°C. The combined effects of cold junction compensation error, amplifier offset drift and gain error determine the stability of the AD594/AD595 output over the rated ambient temperature range. Figure 8 shows an AD594/AD595 drift error envelope. The slope of this figure has units of °C/°C.

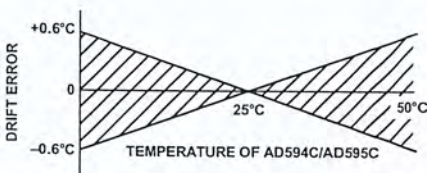


Figure 8. Drift Error vs. Temperature

### THERMAL ENVIRONMENT EFFECTS

The inherent low power dissipation of the AD594/AD595 and the low thermal resistance of the package make self-heating errors almost negligible. For example, in still air the chip to ambient thermal resistance is about 80°C/watt (for the D package). At the nominal dissipation of 800 μW the self-heating in free air is less than 0.065°C. Submerged in fluorinert liquid (unstirred) the thermal resistance is about 40°C/watt, resulting in a self-heating error of about 0.032°C.

### SETPOINT CONTROLLER

The AD594/AD595 can readily be connected as a setpoint controller as shown in Figure 9.

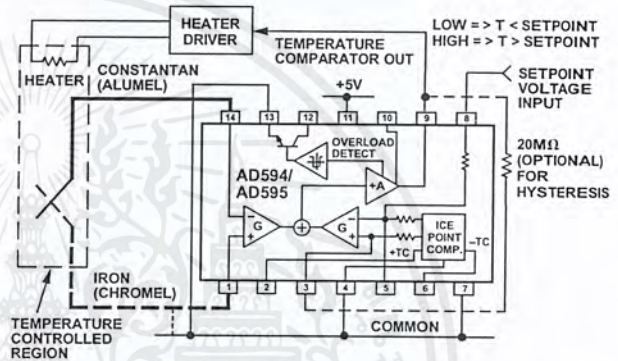


Figure 9. Setpoint Controller

The thermocouple is used to sense the unknown temperature and provide a thermal EMF to the input of the AD594/AD595. The signal is cold junction compensated, amplified to 10 mV/°C and compared to an external setpoint voltage applied by the user to the feedback at Pin 8. Table I lists the correspondence between setpoint voltage and temperature, accounting for the nonlinearity of the measurement thermocouple. If the setpoint temperature range is within the operating range (-55°C to +125°C) of the AD594/AD595, the chip can be used as the transducer for the circuit by shorting the inputs together and utilizing the nominal calibration of 10 mV/°C. This is the centigrade thermometer configuration as shown in Figure 13.

In operation if the setpoint voltage is above the voltage corresponding to the temperature being measured the output swings low to approximately zero volts. Conversely, when the temperature rises above the setpoint voltage the output switches to the positive limit of about 4 volts with a +5 V supply. Figure 9 shows the setpoint comparator configuration complete with a heater element driver circuit being controlled by the AD594/AD595 toggled output. Hysteresis can be introduced by injecting a current into the positive input of the feedback amplifier when the output is toggled high. With an AD594 about 200 nA into the +T terminal provides 1°C of hysteresis. When using a single 5 V supply with an AD594, a 20 MΩ resistor from V<sub>O</sub> to +T will supply the 200 nA of current when the output is forced high (about 4 V). To widen the hysteresis band decrease the resistance connected from V<sub>O</sub> to +T.

## ALARM CIRCUIT

In all applications of the AD594/AD595 the -ALM connection, Pin 13, should be constrained so that it is not more positive than  $(V+) - 4\text{ V}$ . This can be most easily achieved by connecting Pin 13 to either common at Pin 4 or  $V-$  at Pin 7. For most applications that use the alarm signal, Pin 13 will be grounded and the signal will be taken from +ALM on Pin 12. A typical application is shown in Figure 10.

In this configuration the alarm transistor will be off in normal operation and the 20 k pull up will cause the +ALM output on Pin 12 to go high. If one or both of the thermocouple leads are interrupted, the +ALM pin will be driven low. As shown in Figure 10 this signal is compatible with the input of a TTL gate which can be used as a buffer and/or inverter.

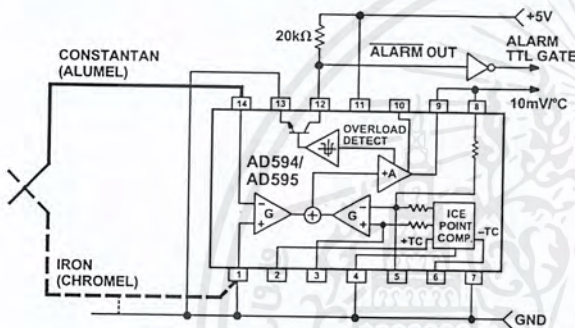


Figure 10. Using the Alarm to Drive a TTL Gate ("Grounded" Emitter Configuration)

Since the alarm is a high level output it may be used to directly drive an LED or other indicator as shown in Figure 11.

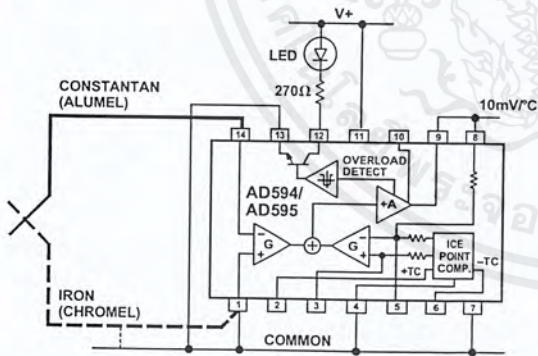


Figure 11. Alarm Directly Drives LED

A 270  $\Omega$  series resistor will limit current in the LED to 10 mA, but may be omitted since the alarm output transistor is current limited at about 20 mA. The transistor, however, will operate in a high dissipation mode and the temperature of the circuit will rise well above ambient. Note that the cold junction compensation will be affected whenever the alarm circuit is activated. The time required for the chip to return to ambient temperature will depend on the power dissipation of the alarm circuit, the nature of the thermal path to the environment and the alarm duration.

The alarm can be used with both single and dual supplies. It can be operated above or below ground. The collector and emitter of the output transistor can be used in any normal switch configuration. As an example a negative referenced load can be driven from -ALM as shown in Figure 12.

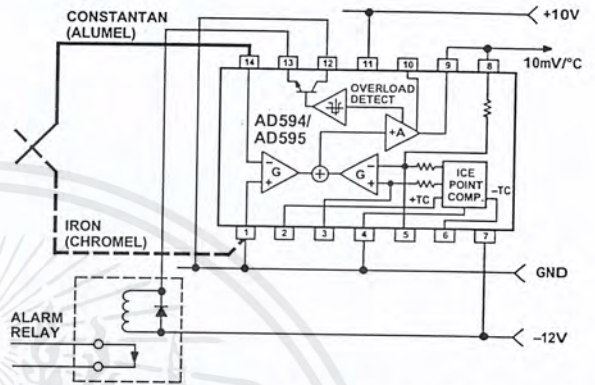


Figure 12. -ALM Driving A Negative Referenced Load

The collector (+ALM) should not be allowed to become more positive than  $(V-) + 36\text{ V}$ , however, it may be permitted to be more positive than  $V+$ . The emitter voltage (-ALM) should be constrained so that it does not become more positive than 4 volts below the  $V+$  applied to the circuit.

Additionally, the AD594/AD595 can be configured to produce an extreme upscale or downscale output in applications where an extra signal line for an alarm is inappropriate. By tying either of the thermocouple inputs to common most runaway control conditions can be automatically avoided. A +IN to common connection creates a downscale output if the thermocouple opens, while connecting -IN to common provides an upscale output.

## CELSIUS THERMOMETER

The AD594/AD595 may be configured as a stand-alone Celsius thermometer as shown in Figure 13.

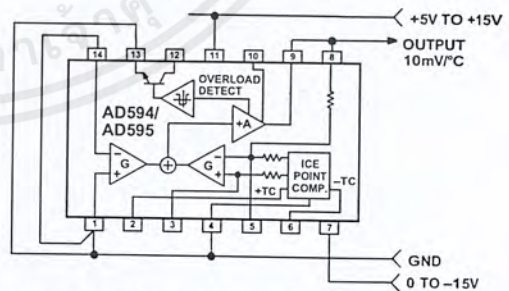


Figure 13. AD594/AD595 as a Stand-Alone Celsius Thermometer

Simply omit the thermocouple and connect the inputs (Pins 1 and 14) to common. The output now will reflect the compensation voltage and hence will indicate the AD594/AD595 temperature with a scale factor of 10 mV/ $^{\circ}\text{C}$ . In this three terminal, voltage output, temperature sensing mode, the AD594/AD595 will operate over the full military  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  temperature range.

# AD594/AD595

## THERMOCOUPLE BASICS

Thermocouples are economical and rugged; they have reasonably good long-term stability. Because of their small size, they respond quickly and are good choices where fast response is important. They function over temperature ranges from cryogenics to jet-engine exhaust and have reasonable linearity and accuracy.

Because the number of free electrons in a piece of metal depends on both temperature and composition of the metal, two pieces of dissimilar metal in isothermal and contact will exhibit a potential difference that is a repeatable function of temperature, as shown in Figure 14. The resulting voltage depends on the temperatures, T1 and T2, in a repeatable way.

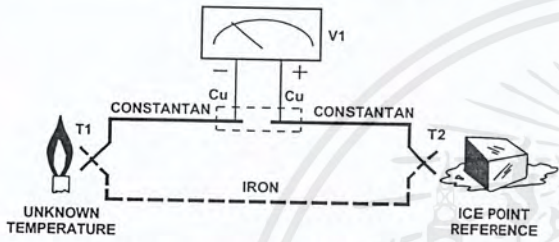


Figure 14. Thermocouple Voltage with 0°C Reference

Since the thermocouple is basically a differential rather than absolute measuring device, a known reference temperature is required for one of the junctions if the temperature of the other is to be inferred from the output voltage. Thermocouples made of specially selected materials have been exhaustively characterized in terms of voltage versus temperature compared to primary temperature standards. Most notably the water-ice point of 0°C is used for tables of standard thermocouple performance.

An alternative measurement technique, illustrated in Figure 15, is used in most practical applications where accuracy requirements do not warrant maintenance of primary standards. The reference junction temperature is allowed to change with the environment of the measurement system, but it is carefully measured by some type of absolute thermometer. A measurement of the thermocouple voltage combined with a knowledge of the reference temperature can be used to calculate the measurement junction temperature. Usual practice, however, is to use a convenient thermoelectric method to measure the reference temperature

and to arrange its output voltage so that it corresponds to a thermocouple referred to 0°C. This voltage is simply added to the thermocouple voltage and the sum then corresponds to the standard voltage tabulated for an ice-point referenced thermocouple.

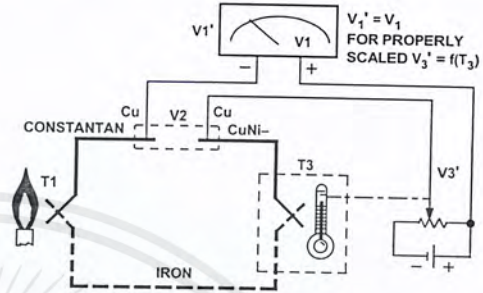


Figure 15. Substitution of Measured Reference Temperature for Ice Point Reference

The temperature sensitivity of silicon integrated circuit transistors is quite predictable and repeatable. This sensitivity is exploited in the AD594/AD595 to produce a temperature related voltage to compensate the reference of "cold" junction of a thermocouple as shown in Figure 16.

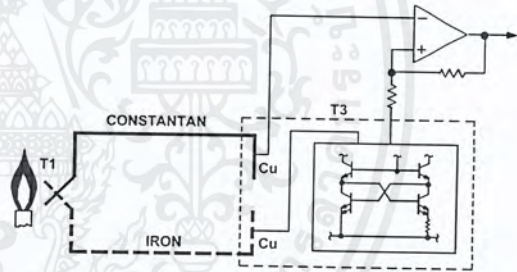


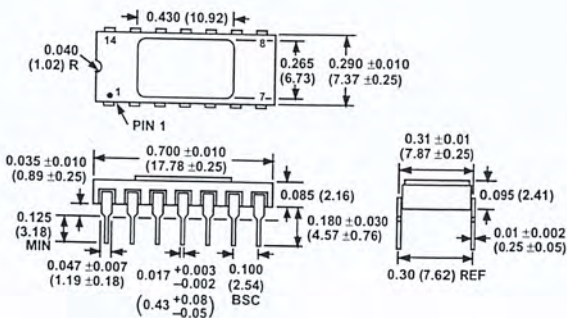
Figure 16. Connecting Isothermal Junctions

Since the compensation is at the reference junction temperature, it is often convenient to form the reference "junction" by connecting directly to the circuit wiring. So long as these connections and the compensation are at the same temperature no error will result.

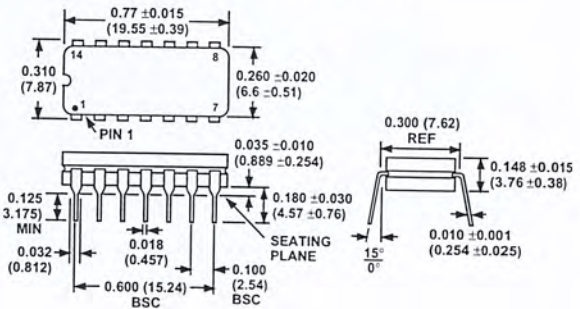
## OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

### TO-116 (D) Package



### Cerdip (Q) Package



โปรแกรมส่วนของผู้ให้บริการ ( Server )

unit TeleF3;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, Menus, ExtCtrls, TeeProcs, TeEngine, Chart, ScktComp, CPDrv,  
Series, ComCtrls;

type

TTelemetering = class(TForm)

Panel1: TPanel;

MainMenu: TMainMenu;

MenuConnection: TMenuItem;

Memo: TMemo;

TeleChart: TChart;

StatusBar: TStatusBar;

GroupRS232: TGroupBox;

Series1: TFastLineSeries;

ButtonConnect: TButton;

ButtonDisconnect: TButton;

cpDrv: TCommPortDriver;

ServerSocket: TServerSocket;

RadioPort: TRadioGroup;

RadioBaud: TRadioGroup;

RadioData: TRadioGroup;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RadioStop: TRadioGroup;
RadioParity: TRadioGroup;
GroupGraph: TGroupBox;
ButtonRefresh: TButton;
MenuListen: TMenuItem;
MenuExit: TMenuItem;
procedure ButtonConnectClick(Sender: TObject);
procedure ButtonDisconnectClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ButtonRefreshClick(Sender: TObject);
procedure MenuExitClick(Sender: TObject);
procedure cpDrvReceiveData(Sender: TObject; DataPtr: Pointer;
  DataSize: Cardinal);
procedure MenuListenClick(Sender: TObject);
procedure ServerSocketAccept(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ServerSocketClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ServerSocketClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure RadioBaudClick(Sender: TObject);
procedure RadioDataClick(Sender: TObject);
procedure RadioStopClick(Sender: TObject);
procedure RadioParityClick(Sender: TObject);
procedure RadioPortClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure ServerSocketClientRead(Sender: TObject;
    Socket: TCustomWinSocket);

protected
    IsServer : Boolean;

private
    // Updates the Status Bar //
    Procedure UpdateStatusPanels;
    // Open Port RS232 //
    Procedure RS232Connect;

var
    Telemetering: TTelemetering;
    x:integer;
    con:integer;

implementation
    {$R *.DFM}

procedure TTelemetering.FormCreate(Sender: TObject);
begin
    // Enable Setting And Button //
    ButtonConnect.Enabled := True;
    ButtonDisconnect.Enabled := True;
    RadioPort.Enabled := True;
    RadioData.Enabled := True;
    RadioStop.Enabled := True;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RadioParity.Enabled := True;
RadioBaud.Enabled := True;
StatusBar.SimplePanel := False;
UpdateStatusPanels;

// Listening..... //
MenuListenClick(nil);
con:=1;
x:=0;
end;

procedure TTelemetry.UpdateStatusPanels;
const _databits:array[TDataBits]of string = ('5','6','7','8');
      _parity:array[TParity]of string = ('None','Odd','Even','Mark','Space');
      _stopbits:array[TStopBits]of string = ('1','1.5','2');
begin
// Updates RS232 Connection Status //
if cpDrv.Connected then
    StatusBar.Panels[1].Text := 'Connected to'+ cpDrv.PortName
else
    StatusBar.Panels[1].Text := 'Not connected to Serial Port';

// Show Current Setting Status //
StatusBar.Panels[2].Text := IntToStr(cpDrv.BaudRateValue)+' / '+
    _databits[cpDrv.Databits]+' / '+
    _parity[cpDrv.Parity]+' / '+
    _stopbits[cpDrv.StopBits];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

Procedure TTelemetry.RS232Connect;
begin
// If already connected then do noting //
    if cpDrv.Connected then
        exit;
// Try to connect again //
    if not cpDrv.Connect then
        begin
            StatusBar.Panels[1].Text := 'Connection Error! Check COM port settings.';
            exit;
        end;

// Disable Setting //
        ButtonConnect.Enabled := False;
        ButtonDisconnect.Enabled := True;
        RadioPort.Enabled := False;
        RadioData.Enabled := False;
        RadioStop.Enabled := False;
        RadioBaud.Enabled := False;
        RadioParity.Enabled := False;
        UpdateStatusPanels;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TTelemetry.ButtonConnectClick(Sender: TObject);
begin
    RS232Connect;
end;

procedure TTelemetry.ButtonDisconnectClick(Sender: TObject);
begin
    // If already disconnect do noting //
    if not cpDrv.Connected then
        exit;
    // Disconnect //
    cpDrv.Disconnect;

    // Enable Setting & Connect Button //
    ButtonConnect.Enabled := True;
    ButtonDisconnect.Enabled := False;
    RadioPort.Enabled := True;
    RadioData.Enabled := True;
    RadioStop.Enabled := True;
    RadioBaud.Enabled := True;
    RadioParity.Enabled := True;
    UpdateStatusPanels;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TTelemetering.ButtonRefreshClick(Sender: TObject);
```

```
begin
```

```
    Series1.Clear;
```

```
    Memo.Clear;
```

```
end;
```

```
procedure TTelemetering.MenuExitClick(Sender: TObject);
```

```
begin
```

```
    Close;
```

```
end;
```

```
// Handler Incoming Data //
```

```
procedure TTelemetering.cpDrvReceiveData(Sender: TObject; DataPtr: Pointer;
```

```
    DataSize: Cardinal);
```

```
var s,ss,sss:string;
```

```
    c:integer;
```

```
    b:char;
```

```
    Data:real;
```

```
begin
```

```
    // Get Incoming Data //
```

```
    cpDrv.PausePolling;
```

```
    s := '    ';
```

```
    cpDrv.ReadData( @s[1], 8 );
```

```
// Convert ASCII Code into Value //
```

```
    c:= ord(s[1]);
```

```
    b:= char(c);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ss:=IntToStr(c);
Data:=c*1.96;
sss:=FloatToStr(Data);
Memo.Lines.Add( sss );

// Plot GRaph //
Series1.AddXY(x,Data,"Clred);
x:=x+1;

// Send Data to client //
if Con=0 then
    ServerSocket.Socket.Connections[0].SendText(b);
cpDrv.ContinuePolling;
end;

// Listening....//
procedure TTelemetry.MenuListenClick(Sender: TObject);
begin
    MenuListen.Checked := not MenuListen.Checked ;
    if MenuListen.Checked then
        begin
            ServerSocket.Active := True;
            StatusBar.Panels[0].Text := 'Listening....'
        end
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    if ServerSocket.Active then
        ServerSocket.Active := False;
        StatusBar.Panels[0].Text := 'Not Ready';
    end;
end;

Procedure TTelemetering.ServerSocketAccept(Sender: TObject;
Socket: TCustomWinSocket);
begin
    IsServer := true;
    StatusBar.Panels[0].Text := 'Connected to'+ Socket.RemoteAddress;
end;

// When Disconnect //
procedure TTelemetering.ServerSocketClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
begin
    // After Disconnect Disable Socket //
    ServerSocket.Active := False;
    con:=1;

    // Set Listen.... //
    MenuListen.Checked := not MenuListen.Checked;
    MenuListenClick(nil);
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// When Connect.... //
```

```
Procedure TTelemetering.ServerSocketClientConnect(Sender: TObject;  
Socket: TCustomWinSocket);
```

```
begin
```

```
    ServerSocket.Active := True;
```

```
    con := 0;
```

```
    RS232Connect;
```

```
end;
```

```
// Set Baud Rate //
```

```
procedure TTelemetering.RadioBaudClick(Sender: TObject);
```

```
begin
```

```
    case RadioBaud.ItemIndex of
```

```
        0 : CpDrv.BaudRateValue := 110;
```

```
        1 : CpDrv.BaudRateValue := 300;
```

```
        2 : CpDrv.BaudRateValue := 600;
```

```
        3 : CpDrv.BaudRateValue := 1200;
```

```
        4 : CpDrv.BaudRateValue := 2400;
```

```
        5 : CpDrv.BaudRateValue := 4800;
```

```
        6 : CpDrv.BaudRateValue := 9600;
```

```
        7 : CpDrv.BaudRateValue := 14400;
```

```
        8 : CpDrv.BaudRateValue := 19200;
```

```
        9 : CpDrv.BaudRateValue := 38400;
```

```
       10 : CpDrv.BaudRateValue := 56000;
```

```
       11 : CpDrv.BaudRateValue := 57600;
```

```
       12 : CpDrv.BaudRateValue := 115200
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
UpdateStatusPanels;
end;

// Set Data Bits //
procedure TTelemetry.RadioDataClick(Sender: TObject);
begin
    case RadioData.ItemIndex of
        0 : CpDrv.DataBits := db5BITS;
        1 : CpDrv.DataBits := db6BITS;
        2 : CpDrv.DataBits := db7BITS;
        3 : CpDrv.DataBits := db8BITS;
    end;
    UpdateStatusPanels;
end;

// Set Stop Bit //
procedure TTelemetry.RadioStopClick(Sender: TObject);
begin
    case RadioStop.ItemIndex of
        0 : cpDrv.StopBits := sb1BITS;
        1 : cpDrv.StopBits := sb1HALFBITS;
        2 : cpDrv.StopBits := sb2BITS;
    end;
    UpdateStatusPanels;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Set Parity Bit //
procedure TTelemetering.RadioParityClick(Sender: TObject);
begin
    case RadioParity.ItemIndex of
        0 : cpDrv.Parity := ptNONE;
        1 : cpDrv.Parity := ptEven;
        2 : cpDrv.Parity := ptOdd;
    end;
    UpdateStatusPanels;
end;

// Set Port //
procedure TTelemetering.RadioPortClick(Sender: TObject);
begin
    case RadioPort.ItemIndex of
        0 : cpDrv.Port := pnCOM1;
        1 : cpDrv.Port := pnCOM2;
        2 : cpDrv.Port := pnCOM3;
        3 : cpDrv.Port := pnCOM4;
    end;
end;

procedure TTelemetering.ServerSocketClientRead(Sender: TObject;
    Socket: TCustomWinSocket);
var R:String;
begin

```

```

// Receive the request signal for client disconnect //

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

R:=Socket.ReceiveText;
if R='a' then
begin
// Close Port RS232 //
cpDrv.Disconnect;

// Enable Setting RS232 //
ButtonConnect.Enabled := True;
ButtonDisconnect.Enabled := False;
RadioPort.Enabled := True;
RadioData.Enabled := True;
RadioStop.Enabled := True;
RadioBaud.Enabled := True;
RadioParity.Enabled := True;
UpdateStatusPanels;

// Disconnect //
ServerSocket.Close;

// Listen..for a new Connection
MenuListen.Checked :=False;
MenuListenClick(nil);
con:=1;
end;
end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมส่วนผู้ใช้บริการ ( Client )

unit Test2Form;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
Menus, StdCtrls, Buttons, ScktComp, ExtCtrls, ComCtrls, jpeg, TeEngine,  
Series, TeeProcs, Chart;

type

TChatForm = class(TForm)

MainMenu1: TMainMenu;

File1: TMenuItem;

Exit1: TMenuItem;

FileConnectItem: TMenuItem;

StatusBar1: TStatusBar;

Bevel1: TBevel;

Panel1: TPanel;

N1: TMenuItem;

Disconnect1: TMenuItem;

ClientSocket: TClientSocket;

Chart1: TChart;

Series1: TFastLineSeries;

Graph1: TMenuItem;

Refersh1: TMenuItem;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Memo1: TMemo;

procedure FileConnectItemClick(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Disconnect1Click(Sender: TObject);
procedure ClientSocketConnect(Sender: TObject;
    Socket: TCustomWinSocket);
procedure ClientSocketRead(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientSocketError(Sender: TObject; Socket: TCustomWinSocket;
    ErrorEvent: TErrorEvent; var ErrorCode: Integer);
procedure FormCreate(Sender: TObject);
procedure Refersh1Click(Sender: TObject);

protected
    IsServer: Boolean;
end;

var
    ChatForm: TChatForm;
    Server: String;
    m : integer;

implementation

{$R *.DFM}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TChatForm.FileConnectItemClick(Sender: TObject);
begin
    if ClientSocket.Active then ClientSocket.Active := False;
    if InputQuery('Computer to connect to', 'Address Name:', Server) then
        begin
            if Length(Server) > 0 then
                with ClientSocket do
                    begin
                        Host := Server;
                        Active := True;
                    end;
                end;
            end;
        end;

procedure TChatForm.Exit1Click(Sender: TObject);
begin
    ClientSocket.Close;
    Close;
end;

procedure TChatForm.Disconnect1Click(Sender: TObject);
begin
    ClientSocket.Socket.SendText('a');
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TChatForm.ClientSocketConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  StatusBar1.Panels[0].Text := 'Connected to: ' + Socket.RemoteHost;
end;

procedure TChatForm.ClientSocketRead(Sender: TObject;
  Socket: TCustomWinSocket);
var Data1,s : string;
    Data2 : Char;
    n : real;
begin
  Data1 := Socket.ReceiveText;
  Data2 := Data1[1];
  n := (ord(Data2)*1.96);
  m := m+1;
  S := FloatToStr(n);
  Memo1.Lines.Add(s);
  Series1.AddXY( m , n , " ,cITeeColor );
end;

procedure TChatForm.ClientSocketError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  ErrorCode := 0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TChatForm.FormCreate(Sender: TObject);
begin
    Series1.Clear;
end;

procedure TChatForm.Refersh1Click(Sender: TObject);
begin
    Series1.Clear;
    Memo1.Clear;
end;

initialization
    m := 0;
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

การที่รายงานฉบับนี้จะเสร็จสมบูรณ์แบบนี้ กระผมกราบขอขอบคุณท่านอาจารย์ รศ.อิทธิชัย อรุณศรีแสงไชย ซึ่งเป็นอาจารย์ที่ปรึกษา Project ท่านได้ให้คำแนะนำคอยดูแลและให้คำปรึกษาวางแผนการปฏิบัติงานต่าง ๆ เกี่ยวกับเรื่องการวัดทางไกลผ่านเครือข่ายอินเทอร์เน็ต และขอขอบคุณเพื่อน ๆ น้อง ๆ ทุกคนที่ให้กำลังใจ คอยช่วยเหลือจนทำให้รายงานฉบับนี้สำเร็จสมบูรณ์ลงได้

สุดท้ายนี้ขอกราบขอพระคุณ คุณพ่อ และคุณแม่ ผู้ซึ่งให้กำเนิด ให้การอบรมสั่งสอน และยังสนับสนุนการศึกษามาโดยตลอด

นาย เกรียงไกร ฉาวรส  
นาย ธีรภพ สารเชตต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. Joseph J Carr, “Sensor and Circuit” , Printice Hall Inc, 1993.
2. ชีรวัดน์ ประกอบผล, “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์”, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
3. อุทัย จิงภักดี, “แนวทางการออกแบบเครื่องมือวัดทางอิเล็กทรอนิกส์ ตอนที่ 2”, เซมิฯ, ฉบับที่ 138, 2537, หน้า 115-121.
4. เฉลิมพล ศรีอนันต์, “LCD ดิจิตอล เทอร์โมมิเตอร์”, Electronic World, ฉบับที่ 117, 2531, หน้า 94-97.
5. ไพโรจน์ ไววานิชกิจ, “เทคโนโลยีการวัดคุมผ่านเครือข่ายอินเทอร์เน็ต” เซมิฯ, ฉบับที่ 209, 2543, หน้า 237-244.
6. ชัยวัฒน์ ลิ้มพรจิตรวิไล, “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51”, บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
7. กมลมาศ คำจรกิจการ, “คู่มือพัฒนาโปรแกรมด้วย Delphi 4”, บริษัท เอช เอ็น กรุ๊ป จำกัด, 2542
8. สุวัฒน์ ปุณณชัยยะ, “เปิดโลกของ TCP / IP และ โพรโทคอลของอินเทอร์เน็ต”, บริษัท ค่านสุทธาการพิมพ์ จำกัด, 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้