

ระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน
AUTOMATIC DATA LOGGER SYSTEM VIA THE INTRANET SYSTEM



โดย

นายคะนองเดช พุทธรักษา
นายณรงค์ ปัทมโรจน์

เลขหม.....
เลขทะเบียน..... 46229
วัน, เดือน, ปี..... 21 ส.ค. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน
AUTOMATIC DATA LOGGER SYSTEM VIA THE INTRANET SYSTEM

โดย

นายคะนองเดช พุทธรักษา เลขประจำตัว 42015210
นายณรงค์ ปัทมโรจน์ เลขประจำตัว 42015247

อาจารย์ที่ปรึกษา

ดร.กิตติพล ชิตสกุล

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2544

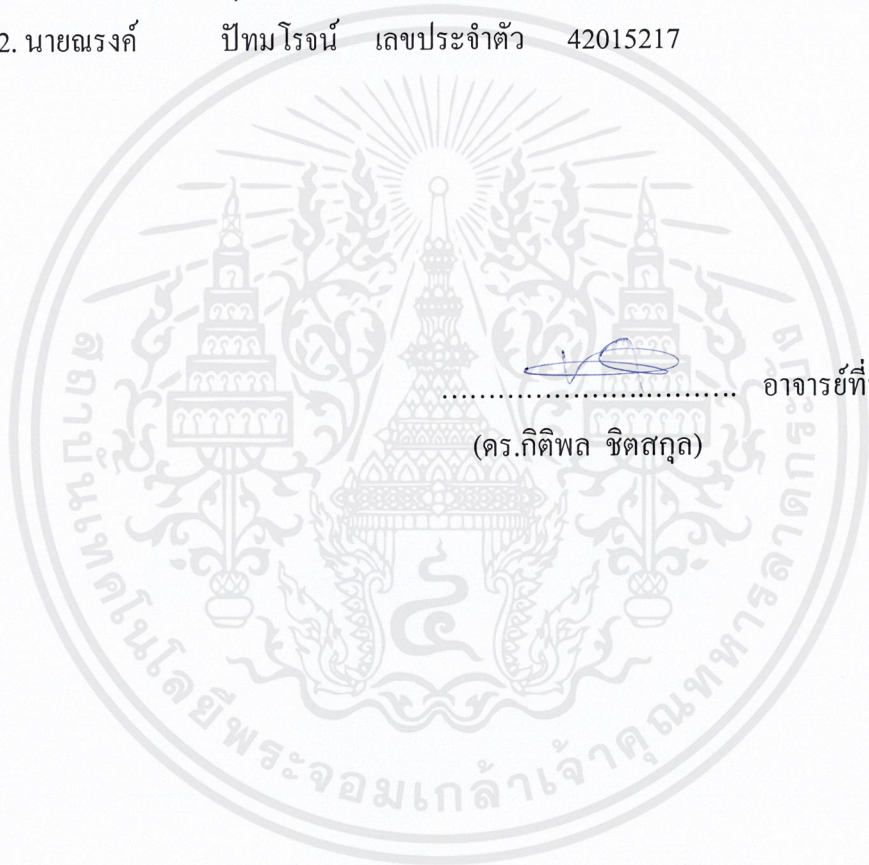
ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน

ผู้จัดทำ

1. นายคะนองเดช พุทธรักษา เลขประจำตัว 42015210
2. นายณรงค์ ปัทมโรจน์ เลขประจำตัว 42015217





..... อาจารย์ที่ปรึกษา

(ดร.กิตติพล ชิตสกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน

AUTOMATIC DATA LOGGER SYSTEM VIA THE INTRANET SYSTEM

นายคะนองเดช พุทธรักษา เลขประจำตัว 42015210

นายณรงค์ ปัทมโรจน์ เลขประจำตัว 42015217

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบ





..... อาจารย์ที่ปรึกษา

(ดร.กิติพล ชิตสกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน

นายคะนองเดช พุทธรักษา

นายณรงค์ ปัทมโรจน์

ดร.กิตติพล ชิตสกุล (อาจารย์ที่ปรึกษา)

ภาคเรียนที่ 2 ปีการศึกษา 2544

บทคัดย่อ

โครงการนี้เป็นการออกแบบสร้างระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายในแบบ 2 ช่องสัญญาณ สัญญาณอนาล็อกความถี่ต่ำจะถูกแปลงเป็นสัญญาณดิจิทัล 12 บิตโดยตัวแปลงสัญญาณ LTC1298 แล้วส่งไปยังไมโครคอนโทรลเลอร์ AT89S8252 จากนั้นข้อมูลส่งต่อไปแสดงผลและเก็บบันทึกผลบนไมโครคอมพิวเตอร์ ผ่านพอร์ตมาตรฐาน RS232 ฟังก์ชันการทำงานของระบบอยู่ภายใต้การควบคุมของไมโครคอมพิวเตอร์ผ่านไมโครคอนโทรลเลอร์ เช่น เลือกรหัสสัญญาณอินพุต ตั้งค่าเวลาในการวัด การบันทึกข้อมูล การแสดงผลของการวัดข้อมูลจะแสดงผลบนหน้าจคอมพิวเตอร์ในรูปแบบของตัวเลขและแสดงผลเป็นกราฟเปรียบเทียบกับเวลา ส่วนของโปรแกรมควบคุมและเชื่อมต่อกับผู้ใช้นไมโครคอมพิวเตอร์พัฒนาโดยใช้ภาษาแอสเซมบลี นอกจากนี้ยังสามารถสื่อสารข้อมูลผ่านระบบเครือข่ายภายในได้โดยใช้โปรโตคอล TCP/IP ในรูปแบบของ Server และ Client แบบ Multi User เพื่อทดสอบการทำงานของระบบ ได้ใช้วงจรวัดอุณหภูมิที่สร้างขึ้น เก็บบันทึกข้อมูลอุณหภูมิของอากาศ และเครื่องชั่งน้ำหนักขนาดเล็ก เก็บบันทึกข้อมูลการเปลี่ยนแปลงน้ำหนักของสสารหรือวัตถุขนาดเล็ก

AUTOMATIC DATA LOGGER SYSTEM VIA THE INTRANET SYSTEM

Mr. Kanongdej Puttharuksa

Mr. Narong Phuttamaroj

Dr. Kitiphol Chitsakul (Advisor)

2nd Semester, Academic Year 2001**ABSTRACT**

We present a design and construction of a 2-channels automatic data logger system via the intranet system designed for logging slow variation signals. The analog signal has been converted by an analog to digital converter, LTC1298, to 12-bits digital signal then transferred via the microcontroller, AT89S8252 to a microcomputer. The function of the logger such as selection of an input signal, setting times for measuring, stored signal is programmed on the microcomputer. The results of measure could be displayed in digits and graph on the monitor. The programs on the microcomputer such as the user interface, system controlling and graphic display are developed on Delphi™. In addition the logged data from the server could be transferred via Intranet using TCP/IP as protocol to multi user client. To evaluate the system performance, two analog sensing systems including electronic a thermometer and a micro balance have been added for logging the atmospheric temperature and weight changing of small objects. The results have been shown in this report.

กิตติกรรมประกาศ

ขอกราบขอบพระคุณท่านอาจารย์ ดร.กิตติพล ชิตสกุล อาจารย์ที่ปรึกษาที่ให้คำแนะนำ ดูแล และให้คำปรึกษาวางแผนการปฏิบัติโครงการ และขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ คอยช่วยเหลือจนทำให้ปริญญาบัตรฉบับนี้สำเร็จสมบูรณ์ลงได้

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ผู้ซึ่งให้กำเนิด ให้การอบรมสั่งสอนและยังสนับสนุนการศึกษามาโดยตลอด

นายคะนองเดช พุทธรักษา

นายณรงค์ ปัทมโรจน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและลักษณะของโครงการ	1
1.2 ขอบเขตของโครงการ	1
1.3 โครงร่างของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 การทำงานของ ADC ชนิด Successive Approximation	4
2.2 ไมโครคอนโทรลเลอร์	6
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	6
2.2.2 พอร์ตของ 8051	6
2.2.3 วงจรคีย์ของ MCS-51	11
2.2.4 การรับส่งข้อมูลอนุกรม (UART) ของ 8051 และ 8052	12
2.2.5 การอินเตอร์รัปต์	15
2.3 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C	18
2.3.1 คอนโทรล MSComm	19
2.4 โพรโตคอล TCP/IP	24
2.4.1 สถาปัตยกรรมของ TCP/IP (TCP/IP Reference Model)	24
2.4.2 IP ADDRESS	26
2.4.3 กลไกการทำงานของโปรโตคอล TCP	29
2.4.4 กลไกการทำงานของโปรโตคอล IP	32
2.4.5 การส่งข้อมูลโดยอาศัย Router และ Routing Algorithm	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

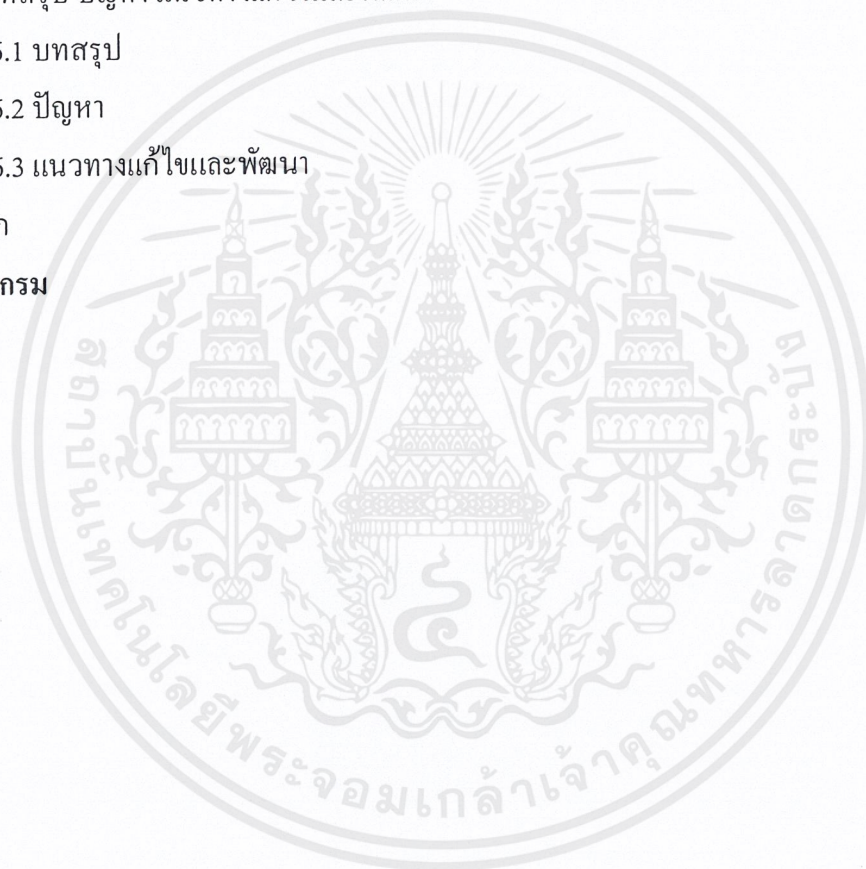
สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 3 การออกแบบ สร้างและการทำงาน	38
3.1 แนวคิดและองค์ประกอบของระบบ	38
3.2 วงจรภาค Sensor	39
3.2.1 วงจรภาควัดอุณหภูมิ	39
3.2.2 วงจรเครื่องชั่งน้ำหนักขนาดเล็ก	42
3.3 วงจรภาคไมโครคอนโทรลเลอร์	47
3.3.1 คุณสมบัติของ LTC1298	47
3.3.2 การทำงานของวงจรภาคไมโครคอนโทรลเลอร์	49
3.3.3 โปรแกรมส่วนไมโครคอนโทรลเลอร์	51
3.4 โปรแกรมติดต่อกับระบบเครือข่ายภายในโดยใช้โปรโตคอล TCP/IP	52
3.4.1 โปรแกรมส่วนของ Server	52
3.4.2 โปรแกรมส่วนของ Client	52
บทที่ 4 การทดลองและผลการทดลอง	55
4.1 ส่วนของวงจรวัดอุณหภูมิ	55
4.2 ส่วนของเครื่องชั่งน้ำหนักขนาดเล็ก	58
4.3 ส่วนของโปรแกรม Server	62
4.3.1 ส่วนของ Channel 1 ที่ใช้วัดอุณหภูมิ	62
4.3.2 ส่วนของ Channel 2 ที่ใช้ชั่งน้ำหนัก	63
4.4 ส่วนของโปรแกรม Client	64
4.4.1 โปรแกรม Client สำหรับการวัดอุณหภูมิ	64
4.4.2 โปรแกรม Client สำหรับการชั่งน้ำหนัก	65
บทที่ 5 การนำไปใช้งาน	68
5.1 การติดตั้งอุปกรณ์ที่เครื่องคอมพิวเตอร์ Server ที่ใช้บันทึกข้อมูล	68
5.1.1 การใช้งานโปรแกรมส่วนการวัดอุณหภูมิ	70
5.1.2 การใช้งานโปรแกรมส่วนการชั่งน้ำหนัก	74
5.2 การใช้เครื่องคอมพิวเตอร์ Client ที่ใช้ในการดูข้อมูล	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
5.2.1 การใช้โปรแกรม Client เพื่อดูข้อมูลการวัดอุณหภูมิ	75
5.2.2 การใช้โปรแกรม Client เพื่อดูข้อมูลการชั่งน้ำหนัก	79
บทที่ 6 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	82
6.1 บทสรุป	82
6.2 ปัญหา	83
6.3 แนวทางแก้ไขและพัฒนา	84
ภาคผนวก	85
บรรณานุกรม	



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 บล็อกไดอะแกรมของ Successive Approximation ADC	4
รูปที่ 2.2 Timing Diagram ของ SAR	5
รูปที่ 2.3 โครงสร้างของพอร์ท 0 (บิต)	7
รูปที่ 2.4 โครงสร้างของพอร์ท 1 (บิต)	8
รูปที่ 2.5 โครงสร้างของพอร์ท 2 (บิต)	8
รูปที่ 2.6 โครงสร้างของพอร์ท 3 (บิต)	10
รูปที่ 2.7 การต่อขาริเซ็ทให้กับ 8051	11
รูปที่ 2.8 วงจรสร้างค็ล็คของ MCS-51	12
รูปที่ 2.9 การรับส่งข้อมูลอนุกรมในโหมด 0	12
รูปที่ 2.10 การรับส่งข้อมูลอนุกรมในโหมด 1	13
รูปที่ 2.11 การรับส่งข้อมูลอนุกรมในโหมด 2	14
รูปที่ 2.12 การรับส่งข้อมูลอนุกรมในโหมด 3	14
รูปที่ 2.13 แหล่งที่มาของสัญญาณอินเตอร์รัปต์	15
รูปที่ 2.14 Interrupt Enable Register (IE)	16
รูปที่ 2.15 Interrupt Priority Register (IP)	17
รูปที่ 2.16 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C	19
รูปที่ 2.17 ส่วนประกอบของ TCP/IP Stack	25
รูปที่ 2.18 เครื่องข่าย 3 เครื่องข่ายที่แต่ละเครื่องข่ายมี IP Address ของตัวเอง	27
รูปที่ 2.19 ตัวอย่างการทำ Encapsulation ของข้อมูล FTP เทียบกับ TCP/IP Layer	28
รูปที่ 2.20 รูปแบบของ IP Datagram ที่ประกอบด้วย Header และ Payload	28
รูปที่ 2.21 รูปแบบของ TCP Packet จะเห็นว่ามี Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล	30
รูปที่ 2.22 รูปแบบของ UDP Packet จะเห็นว่ามีส่วน Header ที่มีขนาดเล็กมากจึงใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP	31
รูปที่ 2.23 แอปพลิเคชันต่างๆ ใน Transport Layer ที่เรียกใช้โปรโตคอล TCP และ UDP	31

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.24 โพรโทคอล TCP และ UDP อาศัยโปรโตคอล IP ในชั้น Internetwork Layer ในการส่งผ่านข้อมูลและแสดงความสัมพันธ์ของโปรโตคอล IP , ICMP และ ARS	32
รูปที่ 2.25 การส่งผ่านข้อมูลผ่านเครือข่าย	34
รูปที่ 2.26 หน้าที่ของ Router ในการพิจารณาเส้นทางและการแลกเปลี่ยนข้อมูล Routing Table ระหว่าง Router ด้วยกัน	34
รูปที่ 2.27 Distance-Vector Routing Protocol	36
รูปที่ 2.28 Intradomain / Interdomain ในการติดต่อระหว่างอุปกรณ์ Router	36
รูปที่ 3.1 ส่วนประกอบของระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน	38
รูปที่ 3.2 รูปร่างและการใช้งานพื้นฐานของ LM335	39
รูปที่ 3.3 วงจรภาควัดอุณหภูมิ	41
รูปที่ 3.4 เครื่องวัดอุณหภูมิ	42
รูปที่ 3.5 วงจรรับส่งโมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์	43
รูปที่ 3.6 วงจรปรับกระแสให้ขดลวดมิเตอร์	44
รูปที่ 3.7 เครื่องชั่งน้ำหนักขนาดเล็ก	45
รูปที่ 3.8 วงจรส่วนภาคขยายของเครื่องชั่งน้ำหนักขนาดเล็ก	46
รูปที่ 3.9 Diagram ของ LTC1298	47
รูปที่ 3.10 เงื่อนไขของ Channel Input	48
รูปที่ 3.11 เครื่องเก็บบันทึกข้อมูลอัตโนมัติและวงจรภาคไมโครคอนโทรลเลอร์	49
รูปที่ 3.12 วงจรส่วนภาคไมโครคอนโทรลเลอร์	50
รูปที่ 3.13 แผนผังการทำงานของไมโครคอนโทรลเลอร์	51
รูปที่ 3.14 แผนผังการทำงานของโปรแกรมส่วน Server	53
รูปที่ 3.15 แผนผังการทำงานของโปรแกรมส่วน Client	54
รูปที่ 4.1 กราฟแรงดันเอาต์พุตจากวงจรวัดอุณหภูมิเปรียบเทียบกับค่าอุณหภูมิจากเทอร์โมมิเตอร์แบบปรอท	57

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.2 กราฟเปรียบเทียบอุณหภูมิที่วัดได้จากเครื่องวัดอุณหภูมิกับอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์แบบปรอท	57
รูปที่ 4.3 เส้นลวดขนาดเล็กที่นำไปซึ่งหาน้ำหนักกับเครื่องชั่งน้ำหนักมาตรฐาน	58
รูปที่ 4.4 เครื่องชั่งน้ำหนักมาตรฐานยี่ห้อ Sartorius รุ่น BD210D	58
รูปที่ 4.5 กราฟความสัมพันธ์ระหว่างน้ำหนักและแรงดันเอาต์พุตของเครื่องชั่ง	60
รูปที่ 4.6 กราฟเปรียบเทียบค่าน้ำหนักที่วัดได้จากเครื่องชั่งน้ำหนักขนาดเล็กเปรียบเทียบกับน้ำหนักเฉลี่ยที่นำมาชั่ง	61
รูปที่ 4.7 ผลของการวัดค่าอุณหภูมิห้องปกติ	62
รูปที่ 4.8 ผลของการวัดค่าอุณหภูมิโดยให้ความร้อนแก่ LM335 และให้ LM335 เกิดการเย็นตัว	62
รูปที่ 4.9 ขณะที่ยังไม่ได้ทำการชั่งวัตถุใด ๆ	63
รูปที่ 4.10 ผลของนำการวัดมาชั่งน้ำหนัก	63
รูปที่ 4.11 เมื่อวัดมือน้ำหนักมากเกินไป 8 มิลลิกรัม	64
รูปที่ 4.12 แสดงผลของการวัดค่าอุณหภูมิห้องปกติจากเครื่อง Server	64
รูปที่ 4.13 แสดงผลของการเปลี่ยนแปลงค่าอุณหภูมิจากเครื่อง Server	65
รูปที่ 4.14 แสดงผลในขณะที่ยังไม่ได้นำเอาวัตถุใดๆ มาชั่งที่เครื่องชั่งจากเครื่อง Server	65
รูปที่ 4.15 แสดงผลของการนำวัตถุมาชั่งน้ำหนักจากเครื่อง Server	66
รูปที่ 4.16 แสดงผลเมื่อวัดมือน้ำหนักมากเกินไป 8 มิลลิกรัมจากเครื่อง Server	66
รูปที่ 5.1 การติดตั้งสาย RS-232 กับเครื่อง Data Logger และเครื่องคอมพิวเตอร์	68
รูปที่ 5.2 การติดตั้งสาย Port กับกล่องวัดอุณหภูมิและช่องสัญญาณที่ 1	69
รูปที่ 5.3 การติดตั้งสาย Port กับเครื่องชั่งน้ำหนักขนาดเล็กและช่องสัญญาณที่ 2	69
รูปที่ 5.4 โปรแกรมส่วน Data Logger Server	70
รูปที่ 5.5 หน้าจอในส่วนของการเลือกช่องสัญญาณอินพุตและตั้งค่าเวลาในการวัด	70
รูปที่ 5.6 หน้าจอในส่วนของ Channel 1	71

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 5.7 หน้าจอใช้ในการเก็บข้อมูล	71
รูปที่ 5.8 หน้าจอใช้ในการดูข้อมูลที่เก็บไว้	72
รูปที่ 5.9 หน้าจอแสดงข้อมูลที่บันทึกไว้เป็น Text File	72
รูปที่ 5.10 เกี่ยวกับ Data Logger Server	73
รูปที่ 5.11 เว็บไซต์ภาควิชาอิเล็กทรอนิกส์	73
รูปที่ 5.12 หน้าจอในส่วนของ Channel 2	74
รูปที่ 5.13 หน้าจอหลักของ Data logger Client สำหรับดูข้อมูลการวัดอุณหภูมิ	75
รูปที่ 5.14 การสร้างช่องทางการติดต่อกับเครื่อง Server	76
รูปที่ 5.15 หน้าจอที่ใช้ในการแสดงข้อมูลการวัดอุณหภูมิจากเครื่อง Server	76
รูปที่ 5.16 หน้าจอใช้ในการเก็บข้อมูลจากเครื่อง Server	77
รูปที่ 5.17 หน้าจอที่ใช้ในการดูข้อมูลจากเครื่อง Server ที่เก็บไว้	77
รูปที่ 5.18 หน้าจอแสดงข้อมูลที่บันทึกแสดงผลเป็น Text File	78
รูปที่ 5.19 เกี่ยวกับ Data Logger Client	78
รูปที่ 5.20 หน้าจอหลักของ Data logger Client ส่วนดูข้อมูลการชั่งน้ำหนัก	79
รูปที่ 5.21 การติดต่อกับเครื่อง Server	80
รูปที่ 5.22 หน้าจอที่ใช้ในการแสดงข้อมูลการชั่งน้ำหนักจากเครื่อง Server	80
รูปที่ 5.23 ระบบเก็บบันทึกข้อมูลอัตโนมัติ	81

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 อินเทอร์เน็ตเวิร์กเตอร์ของ MCS-51 และลำดับความสำคัญของการอินเทอร์เน็ตเวิร์ก	16
ตารางที่ 4.1 ผลของการวัดค่าแรงดันเอาต์พุตของวงจรภาควัดอุณหภูมิเปรียบเทียบกับเทอร์โมมิเตอร์แบบปรอท	55
ตารางที่ 4.2 ผลของการวัดค่าอุณหภูมิจากคอมพิวเตอร์เปรียบเทียบกับเทอร์โมมิเตอร์แบบปรอท	56
ตารางที่ 4.3 ตารางผลของการวัดค่าแรงดันเอาต์พุตของวงจรเครื่องชั่งน้ำหนัก กับค่าน้ำหนักโดยเฉลี่ย	59
ตารางที่ 4.4 ผลของการชั่งน้ำหนักจากคอมพิวเตอร์เปรียบเทียบกับเครื่องชั่งน้ำหนักมาตรฐาน	59

บทที่ 1

บทนำ

1.1 ความเป็นมาและลักษณะของโครงการ

ระบบเก็บข้อมูลอัตโนมัติมีความสำคัญอย่างยิ่งในกรณีที่ต้องเก็บข้อมูลต่อเนื่องระยะยาวหรือต้องเก็บข้อมูลในบริเวณทุรกันดารหรือเสี่ยงภัย ข้อจำกัดทั่วไปของระบบเก็บข้อมูลพื้นฐานคือจำกัดเวลาในการเก็บและการทำงานของระบบจำกัดที่ค่าที่ตั้งไว้แต่ต้นทำให้ในบางครั้งไม่เหมาะสมกับการเก็บข้อมูลที่ต้องใช้เวลานานๆ หรือมีจำนวนข้อมูลมาก รวมถึงระบบที่ต้องการความคล่องตัวในการปรับเปลี่ยนฟังก์ชันของการเก็บข้อมูลเมื่อต้องการทดลองในกรณีที่ต้องการมอนิเตอร์ข้อมูลอย่างต่อเนื่องในระยะไกล

เพื่อสนองต่อความต้องการดังกล่าว จึงได้พัฒนาระบบเก็บบันทึกข้อมูลอัตโนมัติที่สามารถส่งข้อมูลผ่านระบบเครือข่ายภายใน โดยสามารถวัดสัญญาณอินพุทอนาลอกมาประมวลผลแล้วแสดงผลเป็นตัวเลขและกราฟ จุดประสงค์ของการใช้งานจะเน้นการใช้งานภายในสถานที่เพราะการทำงานทั้งหมดจะขึ้นอยู่กับเครื่อง Server เป็นหลักและไม่ต้องเคลื่อนย้าย ฟังก์ชันการทำงานของระบบทั้งหมดนี้จะอยู่ภายใต้การควบคุมของไมโครคอมพิวเตอร์ผ่านไมโครคอนโทรลเลอร์ เช่น เลิกซ์ของสัญญาณอินพุท ตั้งค่าเวลาในการวัด การบันทึกข้อมูล การส่งข้อมูลไปยังเครื่อง Client การแสดงผลของการวัดข้อมูลจะแสดงผลบนหน้าจอคอมพิวเตอร์ในรูปแบบของตัวเลขและแสดงผลเป็นกราฟเปรียบเทียบกับเวลา โดยการแสดงผลจะแสดงในลักษณะของ Real-Time ส่วนของโปรแกรมควบคุมและเชื่อมต่อกับผู้ใช้บนไมโครคอมพิวเตอร์พัฒนาโดยใช้ภาษาแอสเซมบลี นอกจากนี้ยังสามารถสื่อสารข้อมูลผ่านระบบเครือข่ายภายในได้โดยใช้โปรโตคอล TCP/IP ในรูปแบบของ Server และ Client แบบ Multi User เพื่อทดสอบการทำงานของระบบ ได้ใช้วงจรวัดอุณหภูมิที่สร้างขึ้น เพื่อเก็บบันทึกข้อมูลอุณหภูมิของอากาศ เช่น การวัดอุณหภูมิของการเพาะเลี้ยงเนื้อเยื่อในห้องทดลอง และเครื่องชั่งน้ำหนักขนาดเล็ก เพื่อเก็บบันทึกข้อมูลการเปลี่ยนแปลงน้ำหนักของสสารในห้องทดลอง เป็นต้น

1.2 ขอบเขตของโครงการ

เนื่องจากข้อจำกัดในระยะเวลาในการทำโครงการ จึงกำหนดขีดความสามารถของระบบไว้

ดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สามารถวัดสัญญาณอินพุตและบันทึกผลโดยเครื่อง Server ได้
2. สามารถแสดงผลของการวัดข้อมูลเป็นกราฟเปรียบเทียบกับเวลาและตัวเลข ทางหน้าจอคอมพิวเตอร์ได้ทั้งเครื่อง Server และเครื่อง Client
3. สามารถส่งข้อมูลจากเครื่อง Server ผ่านระบบเครือข่ายภายใน ไปยังเครื่อง Client โดยใช้โปรโตคอล TCP/IP ได้
4. สามารถดูข้อมูลจากเครื่อง Server โดยเครื่อง Client ได้ โดยแสดงผลเป็นกราฟเปรียบเทียบกับเวลาและตัวเลขทางหน้าจอคอมพิวเตอร์เช่นเดียวกับเครื่อง Server
5. สามารถทำการวัดข้อมูล ได้ครั้งละ 1 ช่องสัญญาณเท่านั้น
6. ข้อมูลที่ทำการบันทึกแล้วสามารถเรียกดูได้เฉพาะข้อมูลที่เป็นตัวเลขหรือ Text File เท่านั้น ไม่สามารถดูผลของข้อมูลเป็นกราฟได้

1.3 โครงร่างของปฏิญานิพนธ์

ปฏิญานิพนธ์นี้ได้รวบรวมรายละเอียด แนวคิด การสร้างและทดสอบ โครงงานระบบเก็บบันทึกข้อมูลอัตโนมัติที่สามารถส่งข้อมูลผ่านระบบเครือข่ายภายใน โดยได้มีการแบ่งรายงานออกเป็นบทต่าง ๆ ทั้งนี้เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาสำคัญดังนี้

บทที่ 1 ได้กล่าวถึงความเป็นมาและลักษณะของโครงงาน ขอบเขตของโครงงาน และโครงร่างของปฏิญานิพนธ์

บทที่ 2 ทฤษฎีและหลักการทำงานของ Analog to Digital Converter ชนิด Successive Approximation , ไมโครคอนโทรลเลอร์ , การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C และโปรโตคอล TCP/IP

บทที่ 3 หลักการสร้างและออกแบบระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน ได้แก่ วงจรภาควัดอุณหภูมิ วงจรเครื่องชั่งน้ำหนักขนาดเล็ก วงจรภาคไมโครคอนโทรลเลอร์ หลักการพัฒนาโปรแกรมควบคุมไมโครคอนโทรลเลอร์ด้วยภาษาแอสเซมบลี และหลักการพัฒนาโปรแกรมแสดงผลในส่วนของ Server และ Client ด้วยภาษาแคลไฟล์

บทที่ 4 ผลการทดลองในการวัดค่าอุณหภูมิและผลการทดลองชั่งน้ำหนักของวัตถุขนาดเล็กลง โดยแสดงผลเป็นตัวเลขและกราฟเปรียบเทียบกับเวลา และผลการทดลองติดต่อข้อมูลระหว่างเครื่อง Server กับเครื่อง Client ผ่านระบบเครือข่ายภายใน โดยใช้โปรโตคอล TCP/IP

บทที่ 5 การนำไปประยุกต์ใช้งาน

บทที่ 6 สรุปปัญหาในการทำงาน แนวทางการแก้ไข และการพัฒนาต่อไปในอนาคต

ในภาคผนวก จะแสดงรายละเอียดและการทำงานของ LTC1298 ซึ่งทำหน้าที่แปลง
สัญญาณอนาลอกเป็นสัญญาณดิจิทัลขนาด 12 บิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

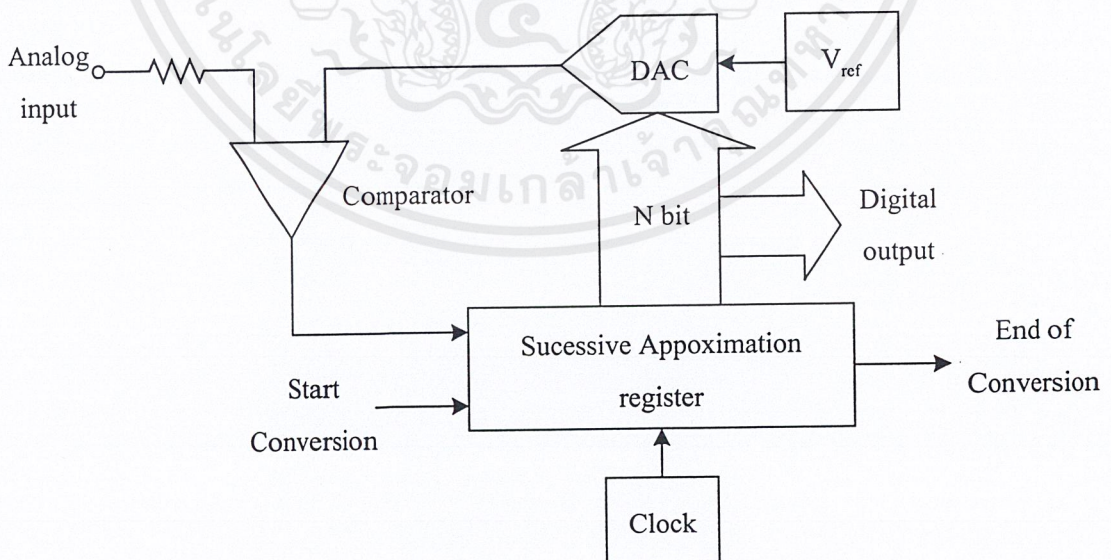
บทที่ 2

ทฤษฎีและหลักการ

หัวใจของระบบเก็บข้อมูลที่ได้พัฒนาขึ้นนี้ได้แก่ Analog to Digital Converter (ADC) ไมโครคอนโทรลเลอร์ และระบบเครือข่ายภายใน ในบทนี้จะกล่าวถึงรายละเอียดของส่วนดังกล่าวที่ได้สืบค้นและศึกษาเพื่อนำมาใช้ในการออกแบบโครงงาน ทั้งนี้เพื่อการทำ ความเข้าใจแนวคิดของระบบที่ได้พัฒนาขึ้น

2.1 การทำงานของ ADC ชนิด Successive Approximation

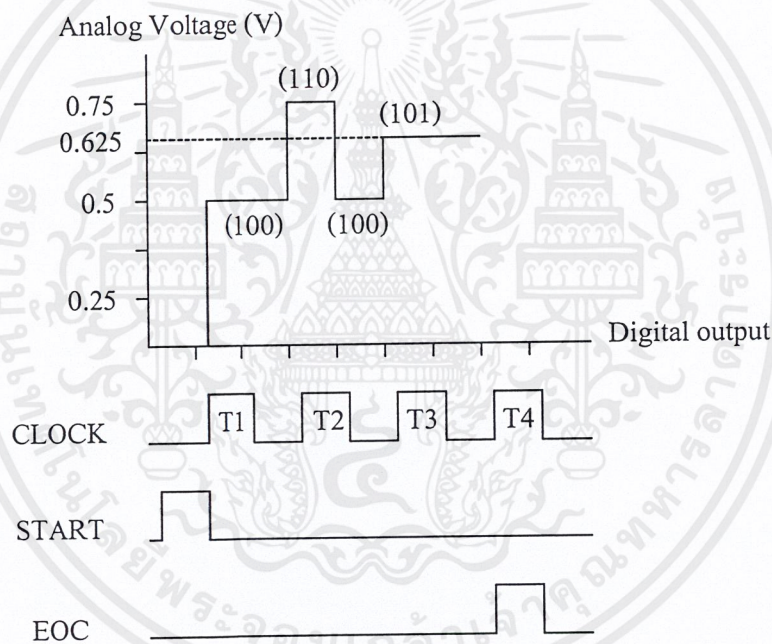
วงจร ADC ชนิดนี้ได้รับความนิยมในงานประยุกต์ที่ต้องการความเร็วปานกลางและค่อนข้างสูง การจัดวงจรจะคล้ายกันกับแบบเคาท์เตอร์ ที่ทำงานในลักษณะการป้อนกลับ ซึ่งบล็อกไดอะแกรมในรูปที่ 2.1 แสดงฟังก์ชันต่างๆ ใน ADC ชนิดนี้ คอมพาราเตอร์จะคอยเปรียบเทียบเอาต์พุตจาก DAC กับอนาลอกอินพุต V_{in} เอาต์พุตจะไปควบคุม Successive Approximation register (SAR) ซึ่งเป็นไอซี MSI (Medium Scale Integrated circuit) ที่ได้รับการออกแบบเป็นพิเศษเพื่อทำหน้าที่นี้โดยเฉพาะ



รูปที่ 2.1 บล็อกไดอะแกรมของ Successive Approximation ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.2 แสดงไทมิงไดอะแกรมของ ADC ที่มีระดับอนาล็อก 0.625V เมื่อ clock เข้าไป 1 ลูก จะทำให้ MSB (Most significant bit) (บิต 4) เป็น 1 ทุกบิตอื่นยังคงเป็นศูนย์ DAC จะเปลี่ยนเอาต์พุตของ SAR เป็นอนาล็อกเปรียบเทียบกับสัญญาณอนาล็อกอินพุต ถ้าผลการเปรียบเทียบที่คอมพาราเตอร์บอกว่าน้อยกว่าอินพุตก็ให้บิตนั้นเป็น 1 ไว้ แต่ถ้ามากกว่าจะให้บิตนั้นเป็น 0 จากนั้นทำการทดสอบบิตถัดไปโดยทำให้เป็น 1 หากผลรวมของสองบิตหรือบิตหลังมากกว่าก็จะทำให้บิตนั้นเป็น 0 แต่ถ้าน้อยกว่าให้คง 1 ไว้ แล้วทดสอบบิตถัดไปตามกรรมวิธีดังกล่าวจนครบทุกบิตหรือจนกว่าเอาต์พุตจะต่างจาก V_{in} ไม่เกิน 1 LSB



รูปที่ 2.2 Timing Diagram ของ SAR

การทำงานของ ADC แบบนี้เปรียบเทียบกับกับการใช้งานของตาชั่งสองแขน เมื่อวัตถุที่ต้องการทราบน้ำหนักเสมือนเป็นอินพุตของ ADC และเอาต์พุตที่เป็นดิจิตอลบิตเสมือนเป็นค้อนน้ำหนักมาตรฐานที่จะวางบนจานอีกข้างหนึ่ง เมื่อตาชั่งยังไม่สมดุลจะต้องมีการปรับค้อนน้ำหนักมาตรฐานจนกว่าจะเกิดสมดุล ในรูปที่ 2.1 คอมพาราเตอร์จะเป็นตัวตรวจสอบการสมดุลดังกล่าว และ SAR จะทำหน้าที่ปรับแต่งดิจิตอลบิต (ค้อนน้ำหนักมาตรฐาน)

มีข้อจำกัดประการหนึ่งสำหรับการ conversion คือสัญญาณอนาล็อกอินพุต จะต้องคงที่ในช่วงเวลาที่ทำการเปลี่ยนแปลงสัญญาณ โดยเปลี่ยนได้ไม่เกิน $\frac{1}{2}$ LSB ในช่วงสุดท้ายของการเปลี่ยน

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่บนสื่อออนไลน์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณดิจิทัลเอาต์พุตจะออกมาขนานกันทุกบิต แต่บางแบบจะให้เอาต์พุตออกมาในลักษณะอนุกรม วงจร ADC แบบนี้สามารถทำงานได้สองโหมด คือ โหมดที่ทำงานโดยอิสระ (Free run) และโหมดที่รอคำสั่ง start conversion จากภายนอก เวลาที่ใช้ในการเปลี่ยนสัญญาณใช้ (n+1) ลูกของพัลส์ clock โดย clock ลูกแรกจะใช้ในการรีเซ็ตรีจิสเตอร์ภายใน สุดท้ายคุณภาพของระบบจะขึ้นอยู่กับคุณภาพของ DAC ในระบบเป็นอย่างยิ่ง

2.2 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 นี้ผลิตโดยบริษัทอินเทลมีอยู่ด้วยกันหลายเบอร์ เช่น 8051, 8051AH, 8052AH, 80C51, 80C52, 83C51FA, 83C51FB เป็นต้น

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1. ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
2. มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
3. มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 802 ขึ้นไปมีถึง 256 ไบต์
4. หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
5. มีไทม์เมอร์ เคนต์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
6. รับอินเทอร์รัพท์ได้ 6 แหล่ง 5 เวกเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวกเตอร์
7. มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปได้ 4 โหมด
8. มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต

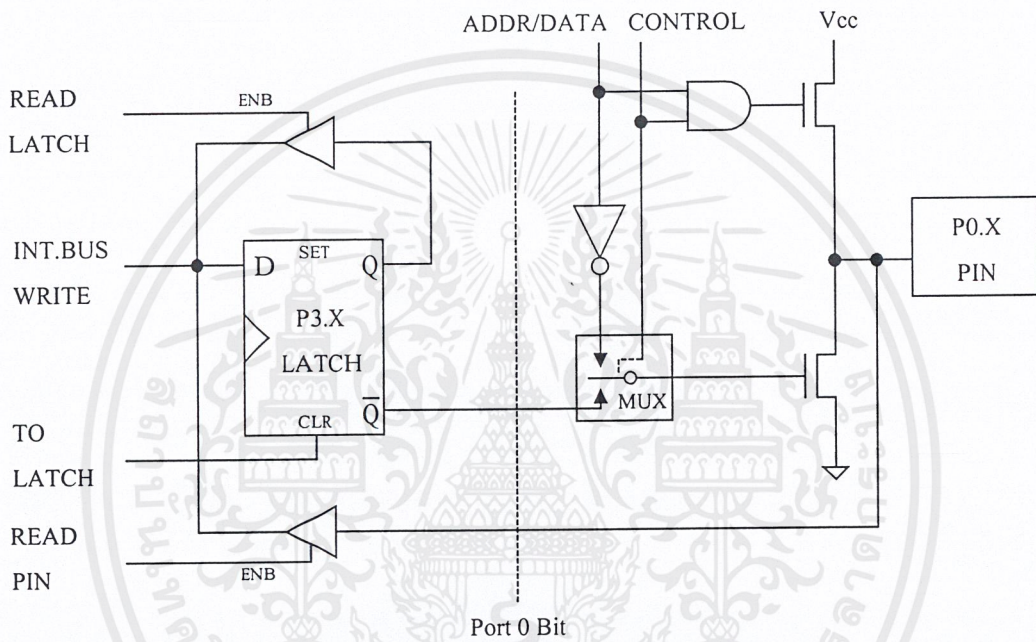
2.2.2 พอร์ตของ 8051

8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ ดังนี้

1. Vcc (ขา 40) ต่อกับ +5V
2. Vss (ขา 20) เป็นขา GND
3. พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ P0.7-P0.0 มีโครงสร้างแบบ Open-Drain Bi-directional ดังแสดงในรูปที่ 2.3 ใช้งานได้ 2 หน้าที่ คือแอสเตอเรสบัส (Address Bus) และดาต้าบัส

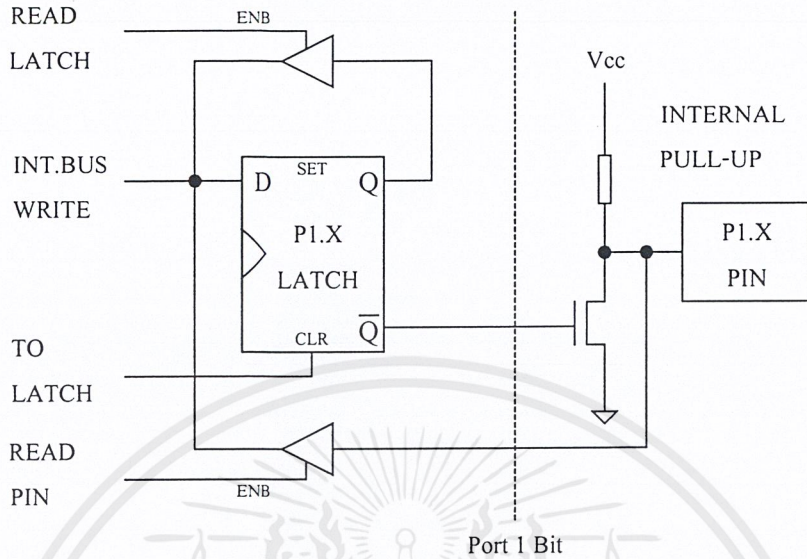
เอกสาร (เขียนโดย) ... เมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็น ไอโอพอร์ต (I/O Port) ถ้าต้องการไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ทำงานเป็นอินพุทพอร์ต (Input Port) ต้องส่งลอจิก “1” ไปยังพอร์ทนี้ จะมีผลให้ \bar{Q} ของ D-FF เป็น “0” ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ทแลทช์ (Input Port Latch) โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (pin) จะใช้สัญญาณ Read (pin)



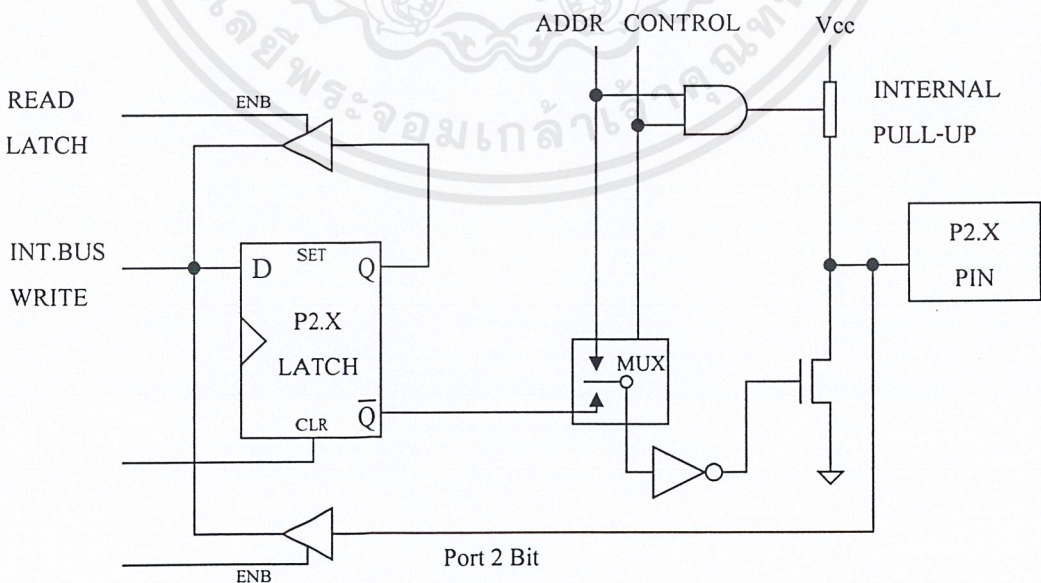
รูปที่ 2.3 โครงสร้างของพอร์ท 0 (บิต)

4. พอร์ท 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ P1.0-1.7 มีโครงสร้างคล้าย พอร์ท 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน (Internal Pull up Register) มีโครงสร้างดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างของพอร์ต 1 (บิต)

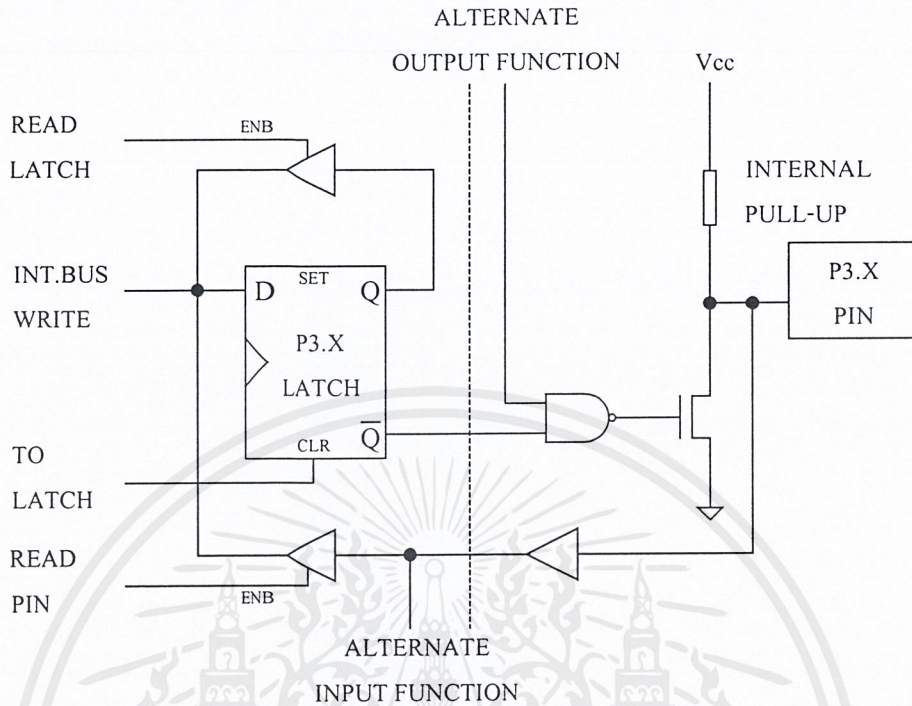
5. พอร์ต 2 (ขา21-28) มีทั้งหมด 8 บิต คือขา P2.7-2.0 มีโครงสร้างคล้ายพอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal pull up) พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอสครีบขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ตใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุทพอร์ตต้องส่งลอจิก “1” มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FET อยู่ในสถานะ OFF ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 โครงสร้างของพอร์ต 2 (บิต)

6. พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ ขา P3.7-3.0 มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าทีคือเป็นไอโอพอร์ต ถ้าจะโปรแกรมให้เป็นอินพุทพอร์ตต้องส่งลอจิก “1” มาที่พอร์ตนี้ก่อน และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมาและรับสัญญาณเข้าไป สัญญาณต่างๆ มีดังนี้

- P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)
 - P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)
 - P3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
 - P3.3/INT1 (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1
 - P3.4/TO (Counter 0 External Input) ขารับสัญญาณพัลส์อินพุทเข้าไปยังวงจร Counter 0 (เป็นอินพุทโหมดเคาท์เตอร์)
 - P3.5/T1 (Counter 1 External Input) ขารับสัญญาณพัลส์อินพุทเข้าไปยังวงจร Counter 1 (เป็นอินพุทโหมดเคาท์เตอร์)
 - P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก
 - P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก
- โครงสร้างของพอร์ต 3 ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 โครงสร้างของพอร์ท 3 (บิต)

7. ALE (ขา 30) เป็นขาส่งสไตรบสำหรับใช้ในการแลตซ์แอดเดรสไปต์ต่ำ (A7-A0) ที่ส่งออกมาจากพอร์ท 0 สัญญาณนี้จะแอกทีฟ (Active) ทุกๆ 2 ครั้ง ใน 1 เมกซ์ซินไซเคิล

8. PSEN (ขา 29) เป็นขาสไตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละเมกซ์ซินไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้

9. EA (ขา 30) ใช้เลือกหน่วยความจำโปรแกรมภายนอก

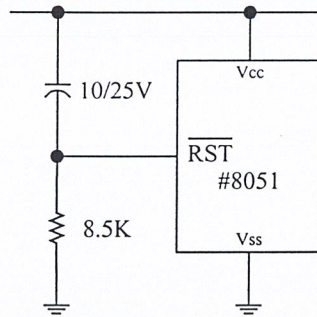
 ป้อน “0” จะอ่าน โปรแกรมจากภายนอกชิพ

 ป้อน “1” จะอ่าน โปรแกรมจากภายในชิพ

10. RST (ขา 9) ขารีสต จะรีเซ็ตได้ก็ต่อเมื่อป้อนลอจิก “1” เข้าที่ขานี้ นานอย่างน้อย 2 เมกซ์ซินไซเคิล

11. XTAL1 (ขา 19) ต่อคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรรอสซิติลเลเตอร์ภายใน

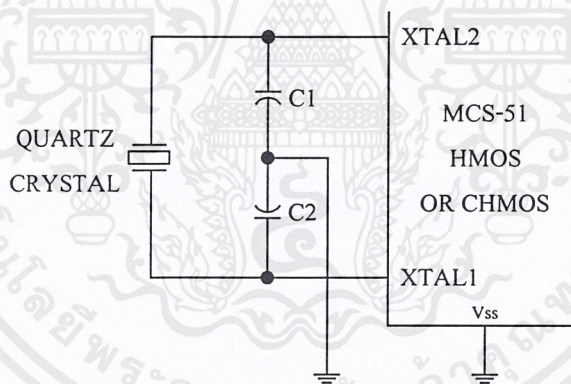
12. XTAL2 (ขา 18) ต่อคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรรอสซิติลเลเตอร์ภายใน



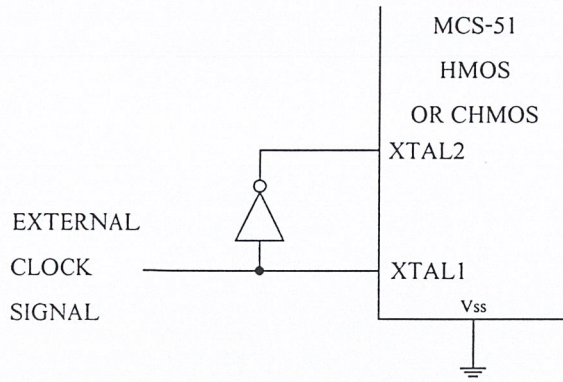
รูปที่ 2.7 การต่อขารี่เซ็ทให้กับ 8051

2.2.3 วงจรคล็อกของ MCS-51

การต่อมียู่ด้วยกัน 2 รูปแบบคือแบบใช้คล็อกภายในและคล็อกจากภายนอก มีรูปแบบการต่อดังรูปที่ 2.8



ก) ใช้คล็อกภายใน



ข) ใช้คล็อกจากภายนอก

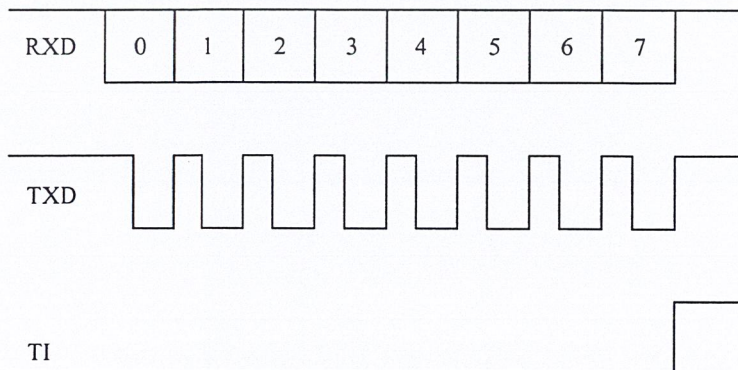
รูปที่ 2.8 วงจรสร้างคล็อกของ MCS-51

2.2.4 การรับส่งข้อมูลอนุกรม (UART) ของ 8051 และ 8052

พอร์ตสื่อสารอนุกรมของ 8051, 8052 มีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TXD (พอร์ต 3.1) ทางด้านรับใช้ขา RXD (พอร์ต 3.0) SBUF ให้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ตสื่อสารอนุกรมของ 8051 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรมมีดังนี้

1) โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทางขา RXD และไม่มีการส่ง start bit แต่จะส่ง shift clock ทางขา TXD (ความเร็ว 1/12 เท่าของ CPU Clock)



รูปที่ 2.9 การรับส่งข้อมูลอนุกรมในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

สำหรับ 8032, 8052 โดยใช้ Timer 1

$$\text{Baud Rate Mode 1} = \frac{K \times \text{Oscillator Freq.}}{32 \times 12 \times [256 - (\text{TH1})]} \quad (2.1)$$

สำหรับ 8032, 8052 โดยใช้ Timer 2

$$\text{Baud Rate Mode 1} = \frac{\text{Oscillator Freq.}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad (2.2)$$

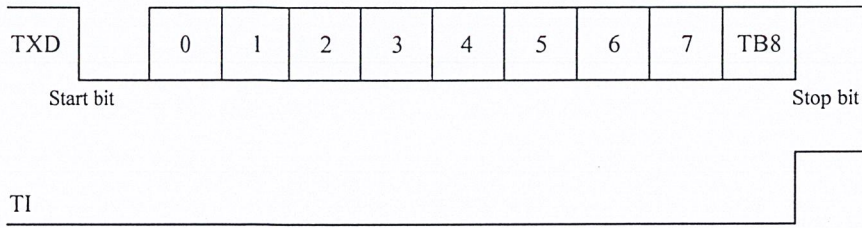


รูปที่ 2.10 การรับส่งข้อมูลอนุกรมในโหมด 1

3) โหมด 2 : พอร์ตสื่อสารอนุกรม 11 บิต ใช้บิตข้อมูล 9 บิต 1 start bit และ 1 stop bit (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยขึ้นกับบิต SMOD ใน PCON

$$\text{Baud Rate Mode 2} = 1/(32 \text{ Osc Freq}) \text{ เมื่อ SMOD} = 1$$

$$\text{Baud Rate Mode 2} = 1/(64 \text{ Osc Freq}) \text{ เมื่อ SMOD} = 0$$



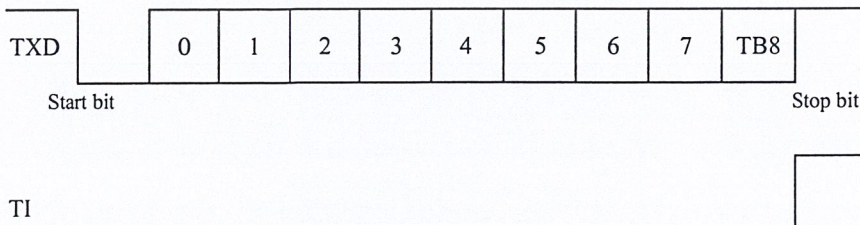
รูปที่ 2.11 การรับส่งข้อมูลอนุกรมในโหมด 2

4) โหมด 3 : พอร์ตสื่อสารอนุกรมแบบ 11 bit UART โดยส่งข้อมูล 9 บิต 1 start bit และ 1 stop bit เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1 สำหรับ 8051 หรือขึ้นกับอัตราโอเวอร์โพล์ของ Timer 2 สำหรับ 80C154D สำหรับ 8031, 8051 โดยใช้ Timer 1

$$\text{Baud Rate Mode 3} = \frac{K \times \text{Oscillator Freq.}}{32 \times 12 \times [256 - (\text{TH1})]} \tag{2.3}$$

สำหรับ 8032, 8052, 80154, 80154D โดยใช้ Timer 2

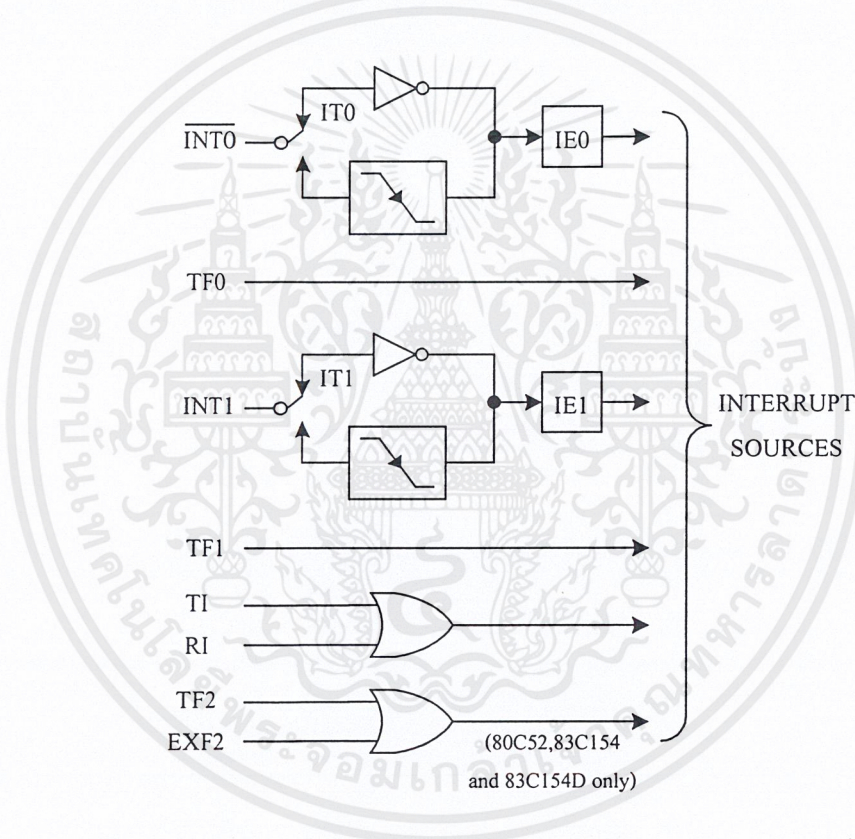
$$\text{Baud Rate Mode 3} = \frac{\text{Oscillator Freq.}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \tag{2.4}$$



รูปที่ 2.12 การรับส่งข้อมูลอนุกรมในโหมด 3

2.2.5 การอินเทอร์รัปต์

คือการขัดจังหวะโปรแกรมชั่วคราวแล้วมาทำโปรแกรมบริการอินเทอร์รัปต์ (Interrupt Service Routine : ISR) การตรวจสอบสัญญาณการร้องขออินเทอร์รัปต์จะตรวจสอบที่ตำแหน่ง S5P2 ของทุกๆ แมชชีนไซเคิลเมื่อพบแล้วในช่วงแมชชีนไซเคิลที่ 2 จะเป็นการตรวจสอบว่าเป็นของอุปกรณ์ใดและแมชชีนไซเคิลที่ 3 จะกระโดดไปทำโปรแกรมบริการอินเทอร์รัปต์ (Interrupt Service Routine : ISR) ดังรูปที่ 2.13 อินเทอร์รัปต์ของ MCS-51 ได้มาจาก 8 แหล่ง และมีเพียง 6 Vector (TI และ RI ใช้ตำแหน่งเดียวกัน) ตำแหน่งของเวกเตอร์ดังแสดงในตารางที่ 2.1



รูปที่ 2.13 แหล่งที่มาของสัญญาณอินเทอร์รัปต์

ตารางที่ 2.1 อินเทอร์รัปต์เวกเตอร์ของ MCS-51 และลำดับความสำคัญของการอินเทอร์รัปต์

ลำดับ	ชื่อสัญญาณอินเทอร์รัปต์	Vector Address	Priority
1	INT0	0003H	Highest ↑ ↓ Lowest
2	TF0	000BH	
3	INT1	0013H	
4	TF1	001BH	
5	TI+RI	0023H	
6	TF2+EXF2	002BH	

ถ้ามีอินเทอร์รัปต์เข้ามาพร้อมกัน INT0 ถือว่ามี Priority สูงสุด

1) Interrupt Enable Register (IE) อยู่ใน SFR ตำแหน่งที่ (0A8H)

ใช้ควบคุมอินเทอร์รัปต์ได้ 8 แหล่ง ดูตารางที่ 2.1 ประกอบ สามารถที่จะสั่งห้ามหรือไม่ห้ามการอินเทอร์รัปต์ได้จากกรีจิสเตอร์ชุดนี้ สามารถสั่งห้ามไม่ให้เกิดการขัดจังหวะทั้งหมดก็ได้เพียงแต่ไปรีเซ็ตบิต EA ใน Interrupt Enable Register (IE) ถ้าต้องการ Enable อินเทอร์รัปต์จากอุปกรณ์ตัวไหนก็เพียงแต่ไปเซ็ตบิตของอุปกรณ์ตัวนั้นไว้ แต่อย่าลืมเซ็ตบิต EA รายละเอียดของ IE ดังแสดงในรูปที่ 2.14

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

รูปที่ 2.14 Interrupt Enable Register (IE)

คำอธิบายความหมายในแต่ละบิตของรีจิสเตอร์ IE

- EA : 0 ไม่ยอมให้ทำอินเทอร์รัปต์จากแหล่งใด ๆ ทั้งสิ้น
 1 ยอมให้เลือกทำการอินเทอร์รัปต์จากอินเทอร์รัปต์แหล่งต่างๆ ได้
- X : ไม่ได้ใช้งาน
- ET2 : 0 ไม่ยอมให้ทำการอินเทอร์รัปต์เมื่อ TF2 เกิด โอเวอร์โฟลว์
 1 ยอมให้ทำอินเทอร์รัปต์เมื่อ TF2 เกิด โอเวอร์โฟลว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องปฏิบัติการเท่านั้น เมื่อผู้ยืมให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ES : 0 ไม่ยอมให้ทำอินเทอร์รัปต์จากพอร์ทสื่อสารอนุกรมได้
 1 ยอมให้ทำอินเทอร์รัปต์จากพอร์ทสื่อสารอนุกรมได้
- ET1 : 0 ไม่ยอมให้ทำการอินเทอร์รัปต์เมื่อ TF1 เกิดโอเวอร์โฟลว์
 1 ยอมให้ทำอินเทอร์รัปต์เมื่อ TF1 เกิดโอเวอร์โฟลว์
- EX1 : 0 ไม่ยอมให้ทำอินเทอร์รัปต์จากสัญญาณภายนอกหมายเลข 1
 1 ยอมให้ทำอินเทอร์รัปต์จากสัญญาณภายนอกหมายเลข 1
- ET0 : 0 ไม่ยอมให้ทำการอินเทอร์รัปต์เมื่อ TF0 เกิดโอเวอร์โฟลว์
 1 ยอมให้ทำการอินเทอร์รัปต์เมื่อ TF0 เกิดโอเวอร์โฟลว์
- EX0 : 0 ไม่ยอมให้ทำอินเทอร์รัปต์จากสัญญาณภายนอก (ขา INTO)
 1 ยอมให้ทำอินเทอร์รัปต์จากสัญญาณภายนอกได้ (ขา INTO)

2) Interrupt Priority Register (IP) อยู่ใน SFR ตำแหน่งที่ (0B8H)

ใช้กำหนดลำดับความสำคัญของการอินเทอร์รัปต์

(MSB)				(LSB)			
PCT	-	PT2	PS	PT1	PX1	PT0	PX0

รูปที่ 2.15 Interrupt Priority Register (IP)

คำอธิบายความหมายในแต่ละบิตของรีจิสเตอร์ IP

- PT2 : 0 Timer 2 มีลำดับความสำคัญต่ำสุด
 1 Timer 2 มีลำดับความสำคัญสูงสุด
- PS : 0 พอร์ทสื่อสารอนุกรม UART มีลำดับความสำคัญต่ำสุด
 1 พอร์ทสื่อสารอนุกรม มีลำดับความสำคัญสูงสุด
- PT1 : 0 Timer 1 มีลำดับความสำคัญต่ำสุด
 1 Timer 1 มีลำดับความสำคัญสูงสุด
- PT0 : 0 Timer 0 มีลำดับความสำคัญต่ำสุด
 1 Timer 0 มีลำดับความสำคัญสูงสุด
- PX0 : 0 อินเทอร์รัปต์ภายนอกชนิด 0 มีลำดับความสำคัญต่ำสุด
 1 อินเทอร์รัปต์ภายนอกชนิด 0 มีลำดับความสำคัญสูงสุด

PCT : 83C154/83C154D Only

- 0 ยอมให้มีการจัดลำดับความสำคัญของการอินเทอร์รัปต์ (Priority)
- 1 ไม่ยอมให้มีการจัดลำดับความสำคัญของการอินเทอร์รัปต์

2.3 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

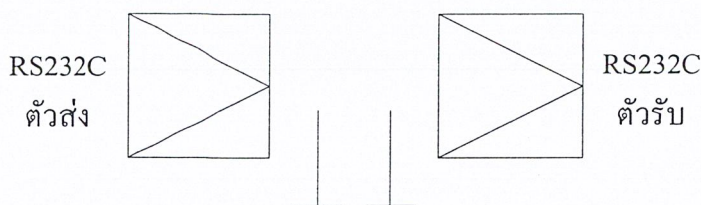
การสื่อสารข้อมูลแบบอนุกรมที่มีการใช้งานในปัจจุบันนี้นั้น ได้มีการกำหนดมาตรฐานในการรับส่งข้อมูลไว้หลายแบบด้วยกันแต่ที่ได้รับความนิยมอย่างกว้างขวางก็คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

ทั้งนี้เนื่องจากความต้องการในการสื่อสารข้อมูลผ่านทางเครือข่ายโทรศัพท์ที่มีมากขึ้นเรื่อยๆ ดังนั้นจึงได้มีการกำหนดมาตรฐานที่เรียกว่า RS-232C ขึ้นเพื่อเป็นมาตรฐานแก่อุปกรณ์ที่ถูกผลิตจากบริษัทต่างๆ ในสหรัฐอเมริกา Bell System Operating Telephone Companies เป็นบริษัทหลักบริษัทแรกที่เป็นผู้ผลิตและติดตั้งระบบสื่อสารข้อมูล และเป็นผู้ผลิตอุปกรณ์ต่างๆ ที่ใช้ในการอินเทอร์เฟซอุปกรณ์ดิจิทัลกับเครือข่ายโทรศัพท์รายใหญ่ อุปกรณ์นี้ก็คือ Bell Modem ซึ่งถูกพัฒนาโดย Bell Laboratories และถูกใช้เป็นมาตรฐานในงานอุตสาหกรรมจนถึงปัจจุบันนี้ ดังนั้นความต้องการข้อมูลเกี่ยวกับข้อกำหนดในการอินเทอร์เฟซกับโมเด็มจึงมีเพิ่มขึ้นเรื่อยๆ เพื่อตอบสนองความต้องการนี้ EIA , Bell System และผู้ผลิตโมเด็มรายอื่นๆ จึงได้ร่วมมือกันตั้งมาตรฐาน RS-232C ขึ้น

มาตรฐาน RS-232C ได้ถูกตีพิมพ์โดย EIA ในปี ค.ศ. 1969 โดยตัวอักษร RS แทน "Recommended Standard" 232 แทนหมายเลขของมาตรฐาน ส่วนอักษร C แสดงให้เห็นว่า มาตรฐานนั้นได้รับการแก้ไขก็ครั้ง

การที่มาตรฐานนี้เป็นที่นิยมใช้ก็เนื่องมาจากเป็นระบบการสื่อสารข้อมูลที่ใช้ในเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็นคอมพิวเตอร์ที่มีใช้อย่างแพร่หลายตั้งแต่อดีตถึงปัจจุบัน

มาตรฐาน RS-232C จะมีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุดเท่านั้น โดยมีลักษณะสมบัติทางไฟฟ้า และทางกายภาพ ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

2.3.1 คอนโทรล MSComm

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (Event-Driven Communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่นเมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้กรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต จะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอแดปเตอร์ของพอร์ตอนุกรมและแอแดปเตอร์ของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติมากมาย แต่สามารถทำความเข้าใจได้ไม่ยาก ดังนี้

1) CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อกันอยู่ (COM1 , COM2 , COM3 , COM4)

รูปแบบการใช้งานคำสั่ง

Object.Commport [= value]

โดย Value เป็นค่าของพอร์ตอนุกรมชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่อยู่ในระบบ MSComm จะสร้างสัญญาณแสดงค่าผิดพลาด Error68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

2) Setting

ใช้ในการกำหนดค่าอัตราบอด , พาราดี้ , จำนวนของบิตข้อมูล , จำนวนของบิตปิดท้าย รูปแบบการใช้งานคำสั่ง

Object.Setting [= value]

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

ค่าพอร์ตมาตรฐาน มีดังนี้

110	บิตต่อนาที
300	บิตต่อนาที
600	บิตต่อนาที
1200	บิตต่อนาที
2400	บิตต่อนาที
9600	บิตต่อนาที (ค่าปกติ)
14400	บิตต่อนาที
19200	บิตต่อนาที
28800	บิตต่อนาที
34800	บิตต่อนาที (สงวน)
56000	บิตต่อนาที (สงวน)
128000	บิตต่อนาที (สงวน)
256000	บิตต่อนาที (สงวน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่ามาตรฐานในการกำหนดค่าพารามิเตอร์มีดังนี้

สัญลักษณ์	รายละเอียด
E	พารามิเตอร์คู่ (Even)
M	ลอจิก "1" (Mark)
N	ไม่ใช่ (ค่าปกติ)
O	พารามิเตอร์คี่ (Odd)
S	ลอจิก "0" (Space)

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

ค่าที่ระบุจำนวนบิตท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ) , 1 , 5 และ 2

3) PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม
รูปแบบการใช้งานคำสั่ง

```
Object.PortOpen [= value]
```

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูล ด้วยคอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติ เมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ตแต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้วค่าที่กำหนดไว้จะเป็นค่าเดิม

4) Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ
รูปแบบการใช้งานคำสั่ง

Object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่าน โดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น ComInputModeText คุณสมบัตินี้ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น ComInputModeBinary คุณสมบัตินี้ Input จะส่งกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ Variant

5) InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ
รูปแบบการใช้งานคำสั่ง

Object.InBufferCount [=value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษรซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

6) InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์
รูปแบบการใช้งานคำสั่ง

Object.InBuffersize [=value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง `InBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

7) `InputLen`

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ
รูปแบบการใช้งานคำสั่ง

```
Object.InputLen [=value]
```

ค่าเริ่มต้นคุณสมบัติ `InputLen` มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้คำสั่ง `Input` ของ `MSComm` อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมดถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากกว่าจำนวน `InputLen` คำสั่ง `Input` จะส่งค่าว่าง (“ ”) กลับออกมาผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ `InBufferCount` โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

8) `InputMode`

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับ โดยคำสั่ง `Input`
รูปแบบการใช้งานคำสั่ง

```
Object.InputMode [=value]
```

คุณสมบัตินี้ `InputMode` ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง `Input` โดยข้อมูลจะเลือกได้ 2 ประเภท คือ

`ComInputModeText` สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น “0” และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

`ComInputModeBinary` สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปแบบไบนารีรวมกันอยู่เป็นไบต์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัพเฟอร์ส่งข้อมูล
รูปแบบการใช้งานคำสั่ง

Object.Output [=value]

ค่า Value เป็นค่าของตัวอักษรที่เขียนไปยังบัพเฟอร์ส่งข้อมูล คุณสมบัติ output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variient และมีข้อมูลภายในเป็นไบต์

2.4 โพรโตคอล TCP/IP

โพรโตคอล TCP/IP เป็นโพรโตคอลที่สำคัญที่มีการใช้กันอย่างแพร่หลายตามการขยายตัวของอินเทอร์เน็ต โดยความจริงแล้ว TCP/IP เป็นกลุ่มของโพรโตคอลหลายตัวที่ประกอบกันเป็นชุดให้ใช้งาน โดยมีชื่อเรียกว่า Transmission Control Protocol / Internet Protocol ตัวอย่างของโพรโตคอลในกลุ่มของ 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 โดยโพรโตคอลที่มีบทบาทสำคัญในการทำงานในเครือข่ายอินเทอร์เน็ต คือ IP (Internet Protocol) เนื่องจากเมื่อโพรโตคอลอื่นๆ ต้องการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตนั้น จำเป็นต้องอาศัยการห่อหุ้มข้อมูล (Encapsulation) ไปกับโพรโตคอล IP ที่มีกลไกการระบุเส้นทาง (Route Service) ผ่าน Gateway เพื่อนำข้อมูลไปยังเครือข่ายหรือเครื่องปลายทางที่ต้องการ โดยกลไกการระบุเส้นทางจะทำงานที่โพรโตคอล IP เท่านั้น หรืออาจเรียกว่า IP เป็นโพรโตคอลที่มีความสามารถในการระบุเส้นทางของการส่งผ่านข้อมูลได้ (Routable)

2.4.1 สถาปัตยกรรมของ TCP/IP (TCP/IP Reference Model)

TCP/IP จะมีการแบ่งจำนวนชั้นตอนที่ใช้รับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองระบบ ออกเป็น 4 ชั้นเท่านั้น หรือเรียกว่าเป็น TCP/IP Stack โดยมีชื่อเรียกแตกต่างกัน ดังแสดงได้ดังรูปที่

2.17

Process Layer (FTP , Telnets , SNMP)
Host-to-Host Layer (TCP)
Internetwork Layer (IP)
Network Interface (IEEE 802.3 , 802.5)

รูปที่ 2.17 ส่วนประกอบของ TCP/IP Stack

1) Process Layer

จะเป็น Application Protocol ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้และให้บริการต่างๆ เช่น FTP , Telnet เป็นต้น

2) Host-to-Host Layer

จะเป็น TCP หรือ UDP ที่ทำหน้าที่คล้ายกับชั้นที่ 4 ของ OSI 7 – Layer Model คือควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล และตัดข้อมูลออกเป็นส่วนย่อยให้เหมาะกับเครือข่ายที่ใช้รับส่งข้อมูล รวมทั้งประกอบข้อมูลส่วนย่อยๆ นี้เข้าด้วยกันเมื่อถึงปลายทาง

3) Internetwork Layer

ได้แก่ส่วนของโปรโตคอล IP ซึ่งทำหน้าที่คล้ายกับชั้นที่ 3 ของ OSI 7-Layer Model คือเชื่อมต่อคอมพิวเตอร์เข้าระบบเครือข่ายที่อยู่ชั้นล่างลงไป และทำหน้าที่เลือกเส้นทางการรับส่งข้อมูลในลักษณะเครือข่ายต่างๆ จนไปถึงผู้รับข้อมูล ในชั้นนี้จะจัดการกับกลุ่มข้อมูลในลักษณะที่เรียกว่า Frame ในรูปแบบของ TCP/IP ที่เรารู้จักกันนั่นเอง

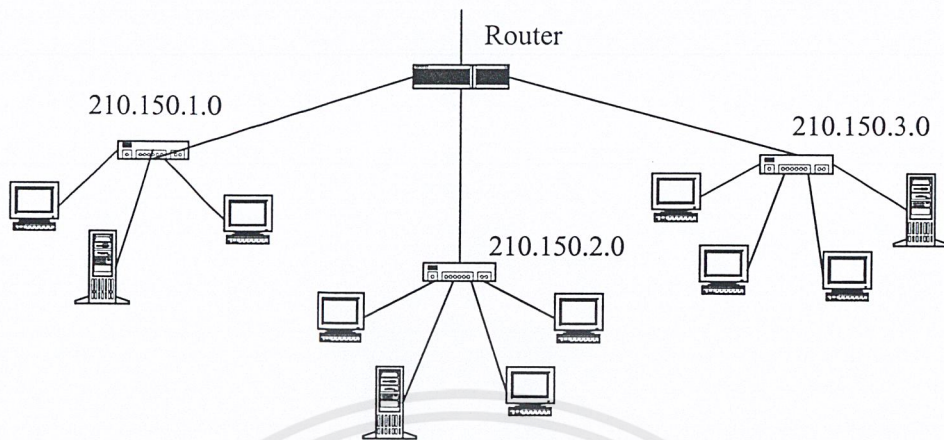
4) Network Internet

คือชั้นที่ควบคุมฮาร์ดแวร์รับส่งข้อมูลผ่านเครือข่าย ซึ่งเทียบได้กับชั้นที่ 1 และ 2 ของ OSI7-Layer Model ในชั้นนี้จะทำหน้าที่เชื่อมต่อกับฮาร์ดแวร์ และควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย ซึ่งที่ใช้กันอยู่จะเป็นตามมาตรฐานของ IEEE

2.4.2 IP ADDRESS

IP Address เป็นหมายเลขที่ถูกกำหนดขึ้นมาเป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่างๆ ที่เชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต โดยอุปกรณ์แต่ละชิ้นจะมี IP Address ไม่ซ้ำกัน แต่อย่างไรก็ตามหมายเลขนี้จะไม่ถูกผูกติดกับตัวอุปกรณ์ สามารถกำหนดหรือแก้ไขเปลี่ยนแปลงได้ ทั้งนี้เนื่องจากการกำหนดด้วยซอฟต์แวร์ แตกต่างกับหมายเลข MAC Address (Media Access Control Address) ซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์ที่ต่ออยู่ในเครือข่าย ค่า MAC Address จะถูกกำหนดมาจากบริษัทผู้ผลิตอุปกรณ์ในระดับล่างสุด (Physical Layer) ของกลไกการรับส่งข้อมูลผ่านเครือข่าย

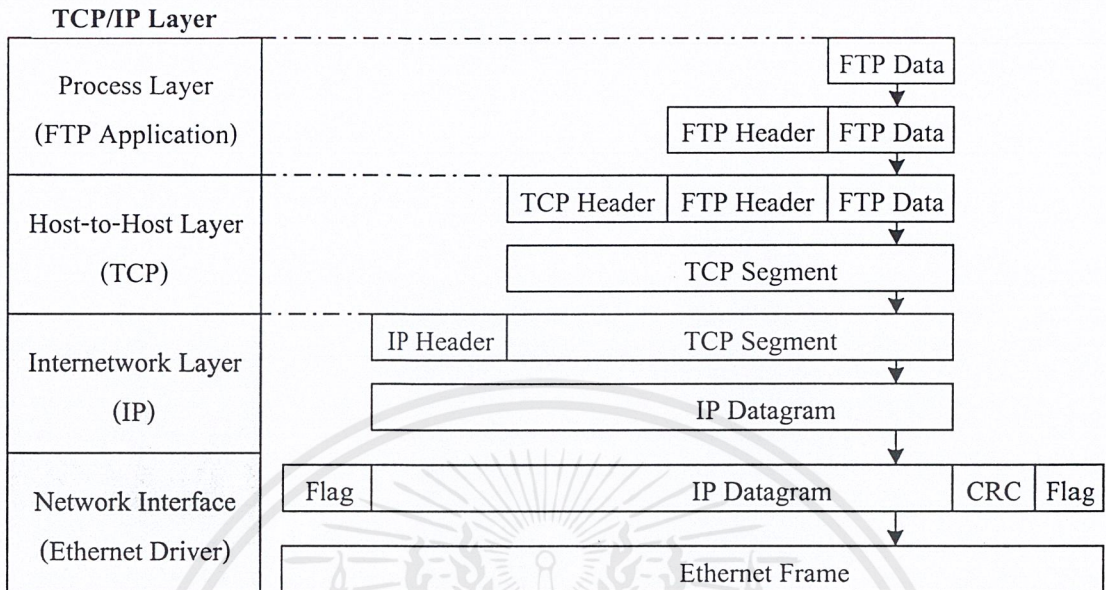
การทำงานของ Internet Protocol (IP) จำเป็นต้องอาศัย IP Address ในการระบุและอ้างถึงอุปกรณ์ต่างๆ ที่ต่ออยู่ในเครือข่าย หมายเลข IP Address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วนและคั่นด้วยเครื่องหมายจุด (.) ดังนั้นตัวเลขแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (2^8) โดยค่าของ IP Address 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 บนสายสัญญาณที่เชื่อมโยงเส้นเดียวกันแต่มีหมายเลขประจำตัวเครื่องที่แตกต่างกันคือ 2 และ 3 ตามลำดับ



รูปที่ 2.18 เครื่องข่าย 3 เครื่องข่ายที่แต่ละเครื่องข่ายมี IP Address ของตัวเอง

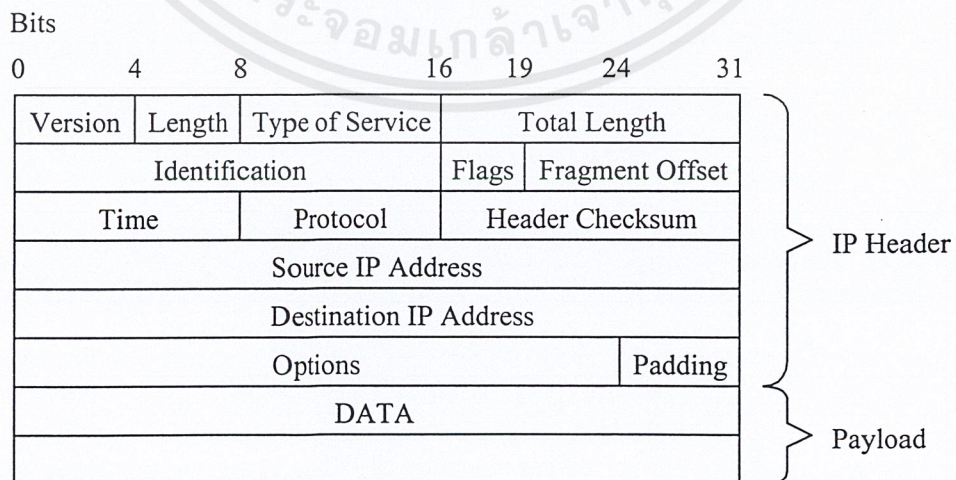
ในการรับส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตนั้น ตัวข้อมูลที่จะส่งจะถูกทำให้มีขนาดเล็กลง โดยแบ่งออกเป็นส่วนย่อยๆ เรียกว่า Data Packet หรือ Datagram ซึ่งการแบ่งข้อมูลเป็นส่วนย่อยนี้ จะช่วยให้เครือข่ายสามารถส่งข้อมูลได้อย่างราบรื่นไม่ติดขัด อีกทั้งยังสามารถทำการตรวจทาน ความถูกต้องของข้อมูลที่ปลายทางและแก้ไขข้อผิดพลาดได้อย่างมีประสิทธิภาพด้วย และที่สำคัญ ช่องสัญญาณที่ใช้ในการเชื่อมต่อในระบบเครือข่ายมีจำกัดและเป็นสิ่งที่ต้องใช้ร่วมกัน เมื่อมี อุปกรณ์ใดส่งข้อมูล อุปกรณ์อื่นๆ ก็ต้องรอให้การส่งข้อมูลนั้นเสร็จสิ้นก่อน ดังนั้นการแบ่งข้อมูล เป็นส่วนย่อยจะเป็นการแบ่งเวลาให้กับผู้ใช้งานรายอื่นๆ ด้วย

โดยตัวของข้อมูลที่ถูกแยกออกเป็น Data Packet หรือ Datagram จะมีลักษณะเป็นข้อมูล แบบต่อเนื่อง (Stream Byte) คือมีการลำดับก่อนหลังของข้อมูล เพื่อให้ประกอบข้อมูลย่อยคืนสู่ สภาพเดิมได้อย่างถูกต้อง และมีรูปแบบที่แน่นอน คือ Datagram จะประกอบด้วยส่วนของ Header และส่วนของตัวข้อมูล (Body) โดยในส่วนหัวของข้อมูลจะประกอบด้วยข้อมูลต่างๆ ที่ระบุที่อยู่ ปลายทางที่จะต้องส่งข้อมูลไป , หมายเลขต้นทางที่ส่งข้อมูลมา , ค่าความยาวของ Datagram นี้และ ข้อมูลอื่นๆ โดย Datagram นี้จะถูกผนึก (Encapsulation) เข้าไปกับโปรโตคอล IP เรียกรวมกันเป็น IP Datagram นับว่าเป็นกลไกสำคัญของการส่งข้อมูลด้วย TCP/IP โดยขบวนการที่ใช้จะมีขั้นตอน คร่าวๆ ดังรูปที่ 2.19 ซึ่งเป็นตัวอย่างของการเรียกใช้โปรแกรมแอปพลิเคชัน FTP (File Transfer Protocol) ร่วมกับ TCP/IP ในการส่งผ่านข้อมูลไปในเครือข่ายแบบ Ethernet



รูปที่ 2.19 ตัวอย่างการทำ Encapsulation ของข้อมูล FTP เทียบกับ TCP/IP Layer

หลังจากขบวนการ Encapsulation นั้น เราจะพบว่าข้อมูลในรูปของ Datagram จะถูกแปลงเป็น Ethernet Frame หรือเฟรมข้อมูลในรูปแบบอื่นๆ ตามลักษณะการเชื่อมต่อทางกายภาพ และเพื่อให้ข้อมูลสามารถส่งออกสู่เครือข่ายไปสู่อินเทอร์เน็ตได้ ตัวข้อมูลที่ถูกแปลงมาเป็น IP Datagram นี้จะประกอบด้วย 2 ส่วน IP Header ที่มีขนาด 32 ไบต์ และส่วนเนื้อข้อมูลที่เรียกว่า Payload โดยจะมีขนาดขึ้นอยู่กับข้อมูล ทำให้ IP Datagram มีขนาดไม่แน่นอน ดังรูปที่ 2.20



รูปที่ 2.20 รูปแบบของ IP Datagram ที่ประกอบด้วย Header และ Payload

ส่วนของ IP Header มีการแบ่งย่อยเพื่อระบุพารามิเตอร์ในการใช้งานต่างๆ ดังนี้

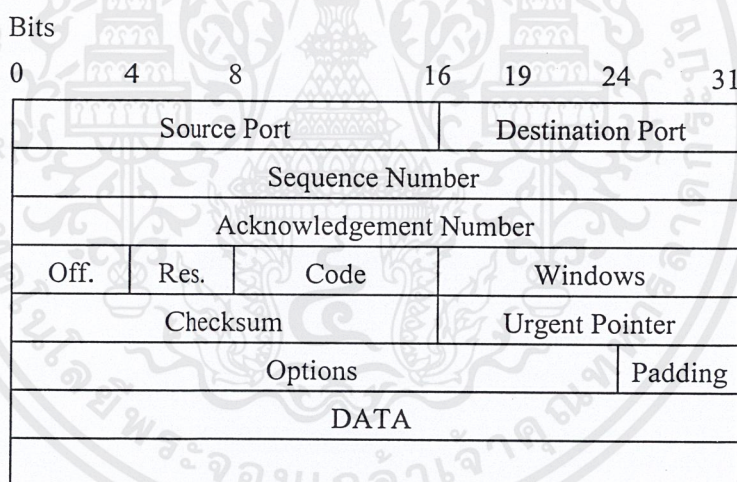
1. Version มีขนาด 4 บิต กำหนดเป็น 4 เมื่อใช้มาตรฐาน IPv4 และเป็น 6 เมื่อใช้เป็น IPv6 (มาตรฐานที่จะใช้งานในอนาคต)
2. Length มีขนาด 4 บิต กำหนดความยาวของ IP Header
3. Type of Service มีขนาด 8 บิต บอกให้ทราบว่าจะดำเนินการกับข้อมูลนี้อย่างไร เช่น Low delay , High Throughout เป็นต้น
4. Total Length มีขนาด 16 บิต กำหนดความยาวสุทธิของ IP Datagram เป็นจำนวนไบต์ ซึ่งมีค่าไม่เกิน 2^{16} หรือ 65,535 ไบต์
5. Identification เป็นข้อมูลที่บอกให้ทราบว่า IP Datagram นี้มาจากที่ใด
6. Flag และ Fragment Offset เป็นส่วนข้อมูลที่ใช้ระบุการแยกและรวมข้อมูล เพื่อให้ข้อมูลที่ถูกรวมออกเป็นข้อมูลย่อย (Fragment) สามารถกลับมารวมกันใหม่ตามลำดับได้อย่างถูกต้อง
7. Time เป็นข้อมูลแสดงจำนวนมากที่สุดของ IP Datagram นี้สามารถส่งผ่านเครือข่ายไปยังปลายทางได้ มีหน่วยเป็นวินาที ปกติเป็น 32 โดยเมื่อ IP Datagram ถูกส่งผ่าน Router ตัวหนึ่งก็จะถูกลดค่า Time ลงไปหนึ่ง ทำให้สามารถใช้ค่านี้นับจำนวนเครือข่ายที่ IP Datagram นี้ถูกส่งผ่านมาได้ เรียกว่า Hop Count
8. Protocol เป็นการระบุโปรโตคอลที่ทำงานในชั้น (Layer) ข้างบนที่ผืนiculumารใน IP Datagram
9. Header Checksum เป็นส่วนของข้อมูลที่ใช้ตรวจสอบความถูกต้องของข้อมูลในส่วนของ IP Header เท่านั้น โดยเมื่อเกิดความผิดพลาดของข้อมูล IP Datagram นั้นจะถูกยกเลิกไปไม่นำมาใช้งาน
10. Source/Destination IP Address เป็นส่วนที่เก็บหมายเลข IP Address ของต้นทางและปลายทางตามลำดับ
11. Option มีขนาดไม่แน่นอน ใช้เก็บค่าพารามิเตอร์ส่วนประกอบปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการใช้งาน
12. Padding ทำหน้าที่เป็นส่วนเติมข้อมูลให้ IP Header ครบ 32 ไบต์

2.4.3 กลไกการทำงานของโปรโตคอล TCP

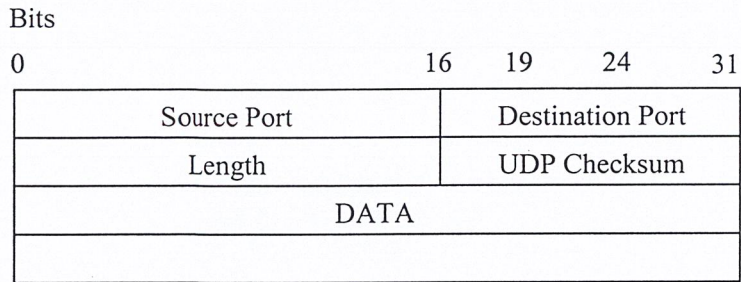
โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ Stream Oriented Protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆ ก่อน (Datagram) แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็น

ลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งหายไป 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

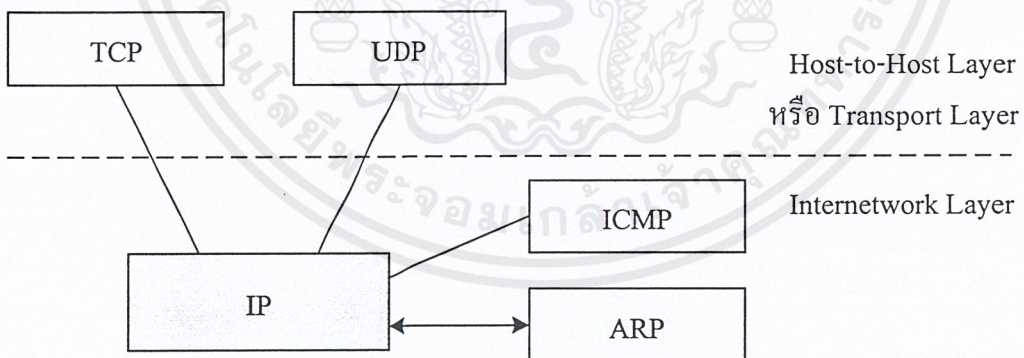


รูปที่ 2.21 รูปแบบของ TCP Packet จะเห็นว่ามี Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล

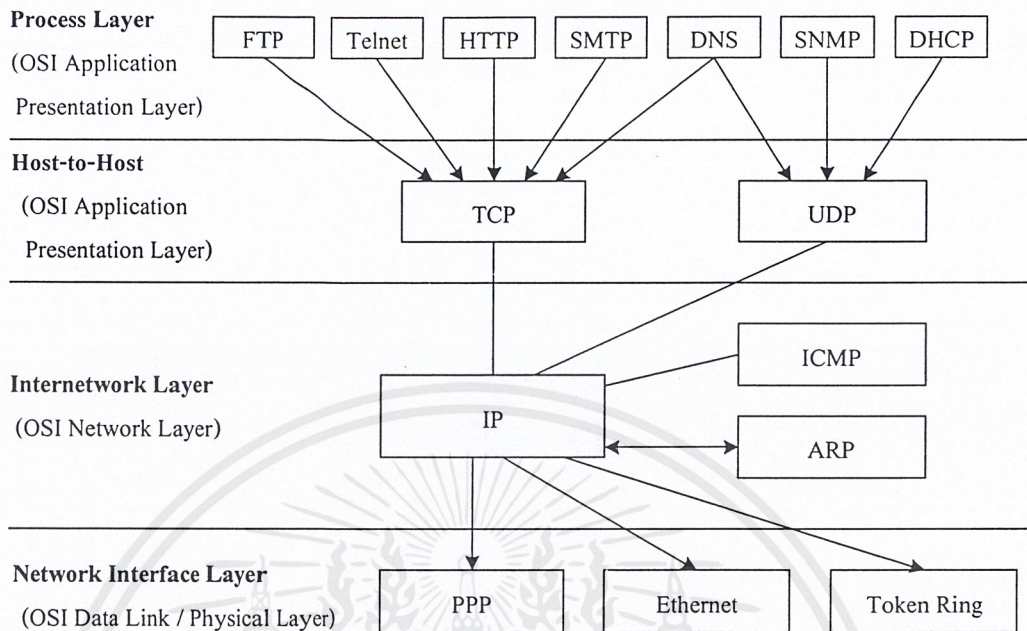


รูปที่ 2.22 รูปแบบของ UDP Packet จะเห็นว่ามีส่วน Header ที่มีขนาดเล็กมากจึงใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP

โดยนอกจากโปรโตคอล TCP แล้วยังมีโปรโตคอล UDP (User Datagram Protocol) ที่จะทำงานในลักษณะที่แตกต่างจากโปรโตคอล TCP คือจะทำการติดต่อสื่อสารแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (Connectionless-Oriented) และไม่มีการตรวจสอบความถูกต้องครบถ้วนของข้อมูลในการรับส่งด้วย จึงทำให้โปรโตคอล UDP ใช้ทรัพยากรของระบบน้อยกว่าโปรโตคอล TCP เนื่องจากการทำงานที่ต่างกันทำให้โปรโตคอลทั้งสองแบบมีแอปพลิเคชันที่เรียกใช้งานต่างกันดังในรูปที่ 2.23



รูปที่ 2.23 แอปพลิเคชันต่างๆ ใน Transport Layer ที่เรียกใช้โปรโตคอล TCP และ UDP



รูปที่ 2.24 โพรโทคอล TCP และ UDP อาศัยโปรโตคอล IP ในชั้น Internetwork Layer ในการส่งผ่านข้อมูลและแสดงความสัมพันธ์ของโปรโตคอล IP, ICMP และ ARS

2.4.4 กลไกการทำงานของโปรโตคอล IP

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งข้ามไปยังเครือข่ายใดๆ ได้อย่างถูกต้อง โดยจะทำงานแบบ Packet Switching คือมีการส่งผ่านข้อมูลผ่านอุปกรณ์ Router หรือ Gateway ในระบบเครือข่ายจนกว่าจะถึงปลายทางที่กำหนดโดย IP Address เมื่อข้อมูลไปถึงปลายทางที่กำหนดโดย IP Address แล้วจะมีกลไกการแปลงหมายเลข IP Address ให้เป็นหมายเลขประจำตัวอุปกรณ์ (MAC Address) ที่ถูกต้องอีกทีหนึ่งด้วยโปรโตคอล ARP (Address Resolution Protocol) นอกจากนี้ยังมีการใช้งานโปรโตคอล ICMP (Internet Control Message Protocol) ควบคู่ไปด้วย โดยจะมีหน้าที่ทำการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้งานทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งข้อความที่โปรโตคอล ICMP ส่งนั้นแบ่งออกได้เป็น 2 แบบ คือ ICMP Error Message หรือข้อความแจ้งความผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูล ไปยังปลายทางที่อยู่คนละเครือข่ายกัน 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 จะเรียกใช้บริการ โปรโตคอล ARS เพื่อแปลงหมายเลข IP Address ปลายทางให้เป็นค่าหมายเลขประจำตัวของอุปกรณ์

3. โปรโตคอล IP ส่งข้อมูล IP Datagram ไปยัง Router ที่กำหนดไว้ จากนั้น Router ก็จะทำการส่งผ่านข้อมูลข้ามเครือข่ายต่อไป

2.4.5 การส่งข้อมูลโดยอาศัย Router และ Routing Algorithm

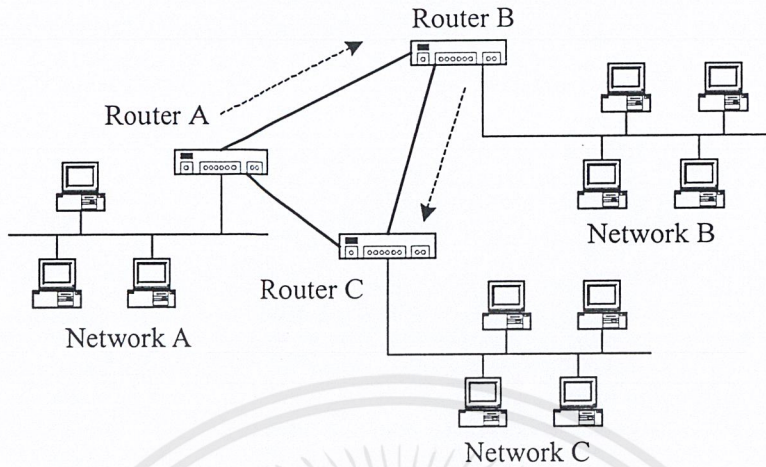
อุปกรณ์ Router ทำหน้าที่ในการเชื่อมโยงเครือข่ายที่อยู่ห่างไกลเข้าด้วยกัน ไม่ว่าจะเครือข่ายนั้นจะต่างกันหรือเหมือนกันทางด้านกายภาพก็ตาม การเชื่อมโยงนี้มีได้หลายลักษณะ โดยส่วนใหญ่แล้วจะทำการเชื่อมโยงกัน โดยผ่านบริการ Wide Area Network Services (WAN Service) เช่น ISDN (Intregrated Service Digital Network) , Frame Relay , Point-to-point Leased Circuit , X.25 เป็นต้น โดยมีบทบาทการทำงานอย่างคร่าวๆ ดังต่อไปนี้

1. เมื่ออุปกรณ์ Router ได้รับ IP Datagram มาจะทำการตรวจสอบความถูกต้องของข้อมูลก่อนด้วยขบวนการ Checksum จนกว่าจะได้ข้อมูลที่ถูกต้อง

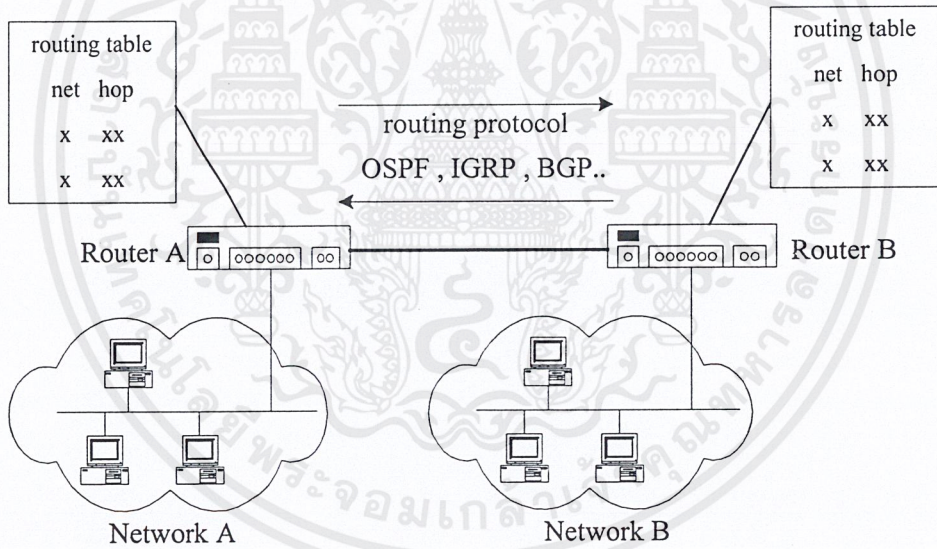
2. ต่อมา Router จะทำการพิจารณาเส้นทางในการส่งผ่านข้อมูล ด้วยขบวนการ Router Algorithm

3. เมื่อทราบเส้นทางที่จะทำการส่งข้อมูลแล้ว ก็จะทำการนำข้อมูลใส่ลงในลำดับ (Queue) เพื่อรอการส่งต่อไป

4. นอกจากนี้ Router จะทำการรับส่งข้อมูล Routing Table ระหว่าง Router ด้วยกันเอง เพื่อเป็นการแลกเปลี่ยนข้อมูลเกี่ยวกับเส้นทางในการส่งผ่านข้อมูลอยู่เป็นระยะๆ ตามเวลาที่กำหนด โดยในการแลกเปลี่ยนนั้นจะอาศัย โปรโตคอลที่เรียกว่า Routing Protocol



รูปที่ 2.25 การส่งผ่านข้อมูลผ่านเครือข่าย



รูปที่ 2.26 หน้าทีของ Router ในการพิจารณาเส้นทางและการแลกเปลี่ยนข้อมูล Routing Table ระหว่าง Router ด้วยกัน

ขบวนการพิจารณาเส้นทางของการส่งผ่านข้อมูล หรือที่เรียกว่า Routing Algorithm มีบทบาทในการปรับปรุงข้อมูลใน Routing Table ให้ใหม่อยู่เสมอ ตามสถานะที่มีการเปลี่ยนแปลงไปของเครือข่าย รวมทั้งยังมีบทบาทในการพิจารณาว่า การส่งข้อมูลนั้นจะเป็นไปอย่างมีประสิทธิภาพ ตามเงื่อนไขที่กำหนดไว้หรือไม่ ช่วยให้ไม่พบกับปัญหาการส่งข้อมูลไม่ถึงปลายทาง ดังนั้นในการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

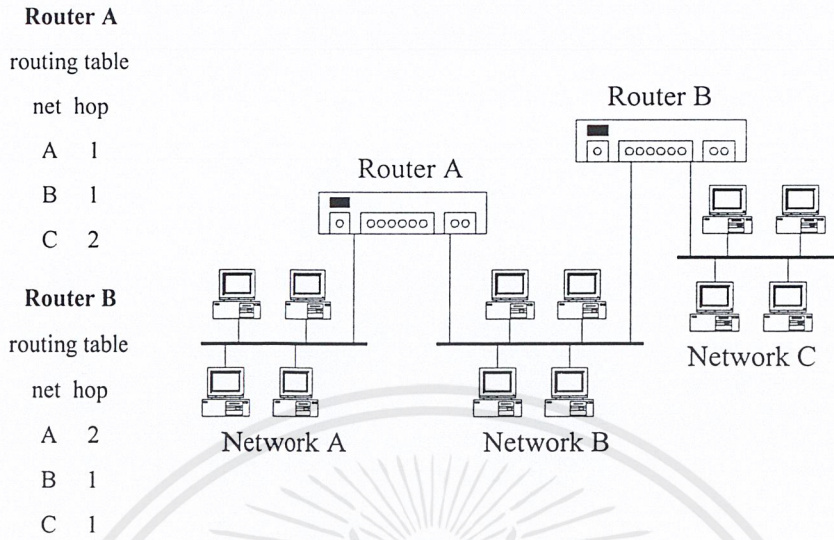
ติดต่อระหว่าง Router รวมถึงวิธีคิดพิจารณาเส้นทางการส่งผ่านข้อมูลที่สามารถทำได้หลายแบบ ซึ่ง จะกำหนดได้เป็น โพรโตคอลสำหรับการพิจารณาเส้นทางการส่งผ่านข้อมูลหรือ Routing Protocol ซึ่งจะทำงานบนโพรโตคอล IP อีกทีหนึ่ง โดยโพรโตคอล IP สามารถผนึกข้อมูลหรือโพรโตคอลที่ จะระบุเส้นทางผ่านข้อมูลเข้าไปได้

Routing Algorithm มีหลายประเภท ซึ่งแต่ละประเภทมีวิธีการคำนวณพิจารณาหาเส้นทาง และวิธีปรับปรุงข้อมูลระหว่าง Router ด้วยกันที่แตกต่างกันด้วย ดังนั้นถ้าต้องการให้สามารถเชื่อมโยงโครงข่ายระหว่างกันได้ และสามารถส่งข้อมูลข้ามเครือข่ายได้อย่างถูกต้องแล้วละก็ อุปกรณ์ Router ที่ใช้เชื่อมต่อกันนั้นจำเป็นต้องใช้งาน Routing Protocol ที่เหมือนกันด้วย โดยสามารถแบ่งประเภทของ Routing Algorithm ได้เป็น 2 ประเภทใหญ่ๆ คือ

1) Interior Routing Protocol

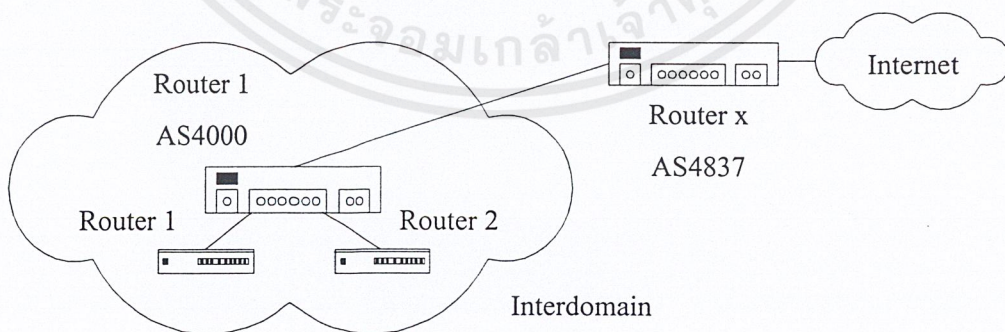
มักใช้งานกับเครือข่ายขนาดเล็กที่มีเครือข่ายย่อยเชื่อมต่อเป็นสมาชิกอยู่ โดยใช้เป็นเส้นทางการติดต่อแลกเปลี่ยนข้อมูลภายในกลุ่มสมาชิกด้วยกัน เช่น บริษัทผู้ให้บริการอินเทอร์เน็ต (ISP) จะมีการเชื่อมต่อเครือข่ายย่อยๆ คือเครือข่ายของลูกค้าแบบองค์กรที่เชื่อมต่อเครือข่ายของตนเข้ากับเครือข่ายของ ISP ทำให้เครือข่ายของลูกค้าอยู่ในเครือข่ายอินเทอร์เน็ตได้ โดยในการติดต่อส่งผ่านข้อมูลอุปกรณ์ Router จะทำการแลกเปลี่ยนข้อมูลของ Routing Table เพื่อให้ทราบว่าเส้นทางใดเป็นเส้นทางในการส่งผ่านข้อมูล ตัว Router ของบริษัทผู้ให้บริการ (ISP) จะทำหน้าที่เป็น Router หลักอยู่ในระบบที่เรียกว่า Autonomous System (AS) ที่จะทำหน้าที่เชื่อมโยงไปสู่ระบบภายนอก และออกสู่ระบบอินเทอร์เน็ตต่อไป โดย AS นี้จะทำการเชื่อมโยงระหว่าง Router หลักของผู้ให้บริการเข้าด้วยกัน Router หลักของ ISP จะมีหมายเลขประจำตัวหรือ AS Number ที่ไม่ซ้ำกัน และมีโพรโตคอลบางชนิดที่ใช้งานค่าหมายเลข AS Number ในการแลกเปลี่ยนสถานะของเครือข่ายและข้อมูล Routing Table โดยมีแบ่งย่อยอีกได้เป็นอีกหลายแบบ ตัวอย่าง เช่น

1. Distance-Vector Routing Protocol มีลักษณะที่สำคัญคือ ในแต่ละ Router จะมีข้อมูล Routing Table เอาไว้พิจารณาเส้นทางการส่งข้อมูล โดยพิจารณาจากระยะทางที่ข้อมูลจะไปถึงปลายทางเป็นหลักโดยจะนับเป็นจำนวน Hop ดังรูปที่ 2.27 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 2.27 Distance-Vector Routing Protocol

2. Link-State Routing Protocol มีกลไกการทำงานคือ อุปกรณ์ Router จะทำการ Broadcast ข้อมูลการเชื่อมต่อของเครือข่ายตนเองไปให้ตัวอุปกรณ์อื่นๆ ทราบ ข้อมูลนี้เรียกว่า Link-State ซึ่งเกิดจากการคำนวณโดยตัวอุปกรณ์ที่ทำกรคำนวณ จะทำการคำนวณค่าการเชื่อมต่อโดยการพิจารณาให้ Router ของตนเองเป็นหลักในการสร้าง Routing Table ขึ้นมา ดังนั้นข้อมูล Link-State ที่ส่งออกไปในเครือข่ายจะเป็นข้อมูลที่บอกเส้นทางที่ดีที่สุดของตนเอง โดยไม่สนใจอุปกรณ์อื่น บางครั้งเรียกว่าเป็น Interdomain Routing Protocol ดังรูปที่ 2.28



รูปที่ 2.28 Intradomain / Interdomain ในการติดต่อระหว่างอุปกรณ์ Router

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Exterior Routing Protocol

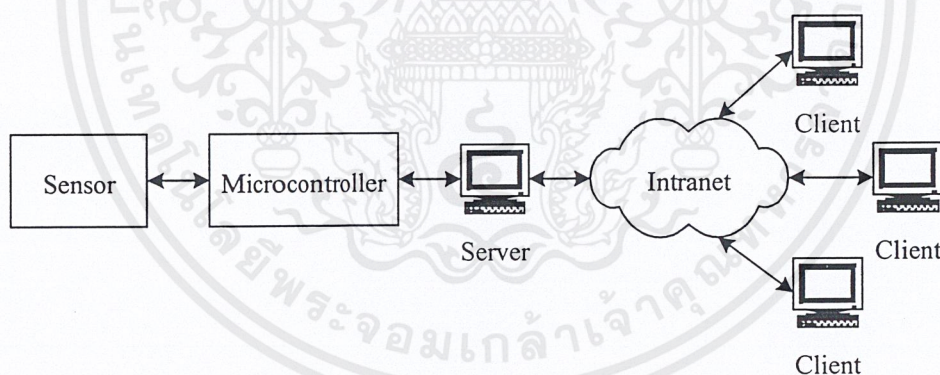
เป็นการเชื่อมต่อเครือข่ายภายใน เช่น ISP เข้ากับเครือข่ายอินเทอร์เน็ต จำเป็นต้องมีการเชื่อมต่อแบบ Exterior Routing Protocol และอาศัยหมายเลข AS Number ในการติดต่อ ถ้าอาศัยหลักการของ Interior Routing Protocol คือให้ Router ในเครือข่ายทุกตัวในอินเทอร์เน็ตส่งค่า Routing Table ออกไปในเครือข่ายนั้นไม่สามารถทำได้เพราะช่องสัญญาณที่มีอยู่ไม่เพียงพอ และส่งผลให้เกิดการติดขัดได้ บางครั้งเราเรียกโปรโตคอล Exterior Routing Protocol ว่าเป็น Interdomain Routing Protocol โดยหลักการงานจะใช้ในลักษณะของ Distance-Vector Routing Protocol และ Link-State Routing Protocol เช่นกัน แต่จะใช้การแลกเปลี่ยนข้อมูลระหว่าง Router หลักที่มีหมายเลข AS Number แทน ประกอบกับต้องมีการพิจารณาปัจจัยอื่นๆ ประกอบด้วย เช่น ขนาดของช่องสัญญาณ (Bandwidth) , จำนวน Delay ที่เกิดขึ้นในช่องสัญญาณ , ความหนาแน่นของช่องสัญญาณ และความน่าเชื่อถือของช่องสัญญาณ (Reliable) เป็นต้น ตัวอย่างโปรโตคอลที่ใช้งาน ได้แก่ RIP (Routing Information Protocol) , IGRP (Interior Gateway Routing Protocol) , OSPF (Open Shortest Path First) , EGP (Exterior Gateway Protocol) และ BGP (Border Gateway Protocol) เป็นต้น

บทที่ 3

การออกแบบ สร้างและการทำงาน

3.1 แนวคิดและองค์ประกอบของระบบ

เมื่อได้กำหนดขอบเขตความสามารถของโครงการดังที่ได้กล่าวไว้ในบทที่หนึ่งแล้ว จากนั้นทำการศึกษาเกี่ยวกับองค์ประกอบต่างๆ สามารถแสดงโครงสร้างของระบบที่จะสร้างในรูปแบบที่ 3.1 เพื่อเป็นการทดสอบการทำงานของระบบทั้งหมด จะทดสอบโดยใช้การวัดอุณหภูมิในห้องทดลอง เช่น การวัดอุณหภูมิของการเพาะเลี้ยงเนื้อเยื่อในห้องทดลอง และการชั่งน้ำหนักของสสารขนาดเล็ก เช่น การชั่งน้ำหนักเพื่อดูการเปลี่ยนแปลงน้ำหนักของเมล็ดพันธุ์พืชในห้องทดลอง เป็นต้น โดยส่วนของการเชื่อมต่อของวงจรรภาค Sensor กับวงจรรภาคไมโครคอนโทรลเลอร์และเครื่องไมโครคอมพิวเตอร์หรือเครื่อง Server และการติดต่อของข้อมูลผ่านระบบเครือข่ายภายใน ในลักษณะของ เครื่อง Server และ เครื่อง Client แบบ Multi-User



รูปที่ 3.1 ส่วนประกอบของระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน

จากรูปที่ 3.1 สามารถแบ่งส่วนประกอบของระบบเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายใน ออกได้เป็น 3 ส่วนใหญ่ๆ ได้แก่

1. วงจรรภาค Sensor ประกอบไปด้วย 2 ส่วน ได้แก่

- วงจรรภาควัดอุณหภูมิ
- วงจรเครื่องชั่งน้ำหนักขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

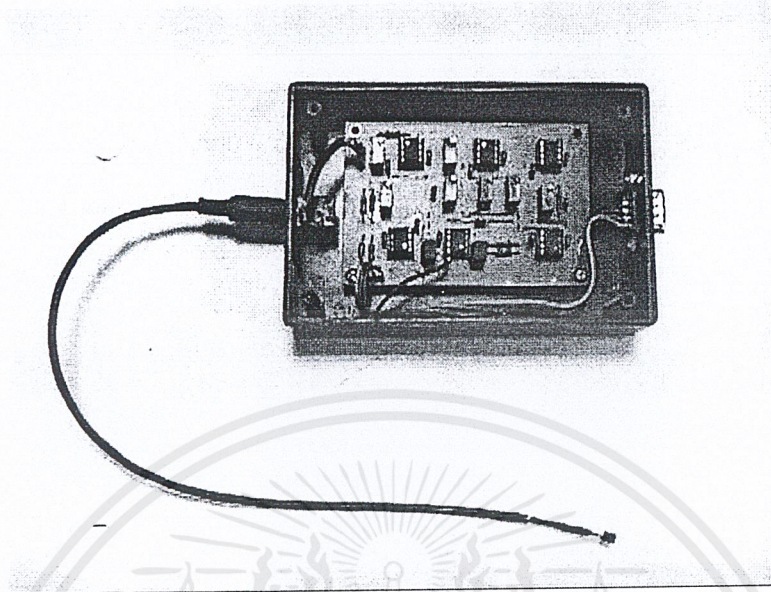
2) การทำงานของวงจรภาควัดอุณหภูมิ

การทำงานเริ่มจาก LM335 ซึ่งเป็นตัวเซ็นเซอร์อุณหภูมิจะให้ค่าแรงดันที่เปลี่ยนแปลงตามอุณหภูมิ $+10\text{mV/K}$ ($+10$ มิลลิโวลต์ ต่อ 1 เคลวิน) โดยขณะที่อุณหภูมิ 0 องศาเซลเซียสจะได้ค่าแรงดันเอาต์พุตที่ LM335 เท่ากับ 2.73 โวลต์ และขณะที่อุณหภูมิเพิ่มเป็น 100 องศาเซลเซียสจะได้ค่าแรงดันเอาต์พุตที่ LM335 เท่ากับ 3.73 โวลต์ แรงดันจะเปลี่ยนแปลงตั้งแต่ 2.73 โวลต์ ถึง 3.73 โวลต์ แล้วต่อแรงดันเอาต์พุตเข้าสู่ขา 3 ของ Op-Amp1 (OP07) ซึ่งต่อในลักษณะของวงจรบัฟเฟอร์ (Buffer) เอาต์พุตของ Op-Amp1 นี้จะต่อเข้ากับ Op-Amp3 ซึ่งเป็นวงจรดิฟเฟอเรนเชียลแอมพลิไฟเออร์ (Differential Amplifier) และขาสัญญาณขั้วบวกของ Op-Amp2 ซึ่งเป็นวงจรบัฟเฟอร์นี้จะใช้ V_{R1} เป็นตัวปรับให้ได้ค่าแรงดันคงที่เท่ากับ 2.73 โวลต์ แล้วป้อนเอาต์พุตเข้าสู่ Op-Amp3 วงจรดิฟเฟอเรนเชียลแอมพลิไฟเออร์เช่นกัน เพื่อจะทำหน้าที่หักล้างกับแรงดันที่ได้จากวงจรภาค Sensor แล้วต่อเข้า Op-Amp4 เป็นวงจรบัฟเฟอร์ จากนั้นเข้าสู่วงจรขยายสัญญาณ 5 เท่า โดยใช้ Op-Amp5 ป้อนกลับแบบลบ (Negative Feedback) เพื่อเป็นการขยายย่านการวัดให้ได้แรงดัน 0 ถึง 5 โวลต์ สุดท้ายจะผ่านเข้าสู่วงจรโลพาสฟิลเตอร์อันดับสอง (Second order lowpass-filter) เพื่อกำจัดสัญญาณรบกวนก่อนที่เข้าสู่วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลหรือวงจรภาคไมโครคอนโทรลเลอร์ต่อไป

โดยความถี่คัทออฟ (Cut-off Frequency) ของวงจรโลพาสฟิลเตอร์กำหนดได้ด้วยค่าของ R และ C ดังนี้

$$F_{OH} = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 100K \times 0.47 \mu F} = 3.386 \text{ Hz} \quad (3.1)$$

ลักษณะของวงจรภาควัดอุณหภูมิแสดงได้ดังรูปที่ 3.3 และลักษณะของเครื่องวัดอุณหภูมิแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 เครื่องวัดอุณหภูมิ

3.2.2 วงจรเครื่องชั่งน้ำหนักขนาดเล็ก

วงจรส่วนของเครื่องชั่งน้ำหนักขนาดเล็กนี้เป็นส่วนที่ใช้ในการวัดสัญญาณอินพุตเข้ามาเพื่อทำการวัดและบันทึกผลข้อมูล และส่วนของตัว Sensor ได้จัดทำขึ้นมาเอง โดยใช้ขดลวดของมิเตอร์ (Coil Meter) ใช้เข็มของมิเตอร์ทำเป็นแกนของตาชั่ง และใช้กระดาษอะลูมิเนียม (Aluminium Foil) ทำเป็นจานสำหรับชั่งสสารหรือวัดมวลมาแขวนไว้ที่ปลายของเข็มมิเตอร์

โดยใช้หลักการทางแสงหรือหลักการรับส่งของโมดูล LED อินฟราเรด และโฟโตทรานซิสเตอร์ แล้วใช้กระดาษตัดเป็นรูปวงกลมเล็กๆ มาติดไว้ที่เข็มมิเตอร์หรือแกนของตาชั่ง ป้อนกระแสให้แก่ขดลวดมิเตอร์นี้แล้วปรับกระแสให้เข็มมิเตอร์หรือแกนของตาชั่งอยู่ในระดับกึ่งกลางเพื่อไปตัดขวางเส้นทางการรับส่งของโมดูล LED อินฟราเรด (LED Infrared) และโฟโตทรานซิสเตอร์ (Photo Transistor) ซึ่งถ้าเข็มมิเตอร์หรือแกนของตาชั่งเกิดการเปลี่ยนแปลงก็จะทำให้แรงดันเอาต์พุตที่โฟโตทรานซิสเตอร์เปลี่ยนแปลงไปด้วย ก็จะนำแรงดันเอาต์พุตที่เปลี่ยนแปลงนี้ไปขยายแรงดันแล้วนำไปเข้าสู่วงจรภาคไมโครคอนโทรลเลอร์ใช้งานต่อไป

ในสภาวะปกติหรือยังไม่ได้ชั่งวัตถุเข็มมิเตอร์หรือแกนของตาชั่งจะอยู่ในระดับกึ่งกลางพอดี จะไปปิดเส้นทางการรับส่งของโมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์ ทำให้แรงดันเอาต์พุตที่โฟโตทรานซิสเตอร์ได้ประมาณ Vcc หรือประมาณ 8.88 โวลต์ ถ้ามีสสารหรือวัตถุมาชั่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้ำหนักก็จะทำให้เข็มมิเตอร์หรือแกนของตาชั่งตกลงมา ก็จะส่งผลให้ช่องทางการรับส่งทางแสงของโมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์มีมากขึ้น ทำให้แรงดันเอาต์พุตที่โฟโตทรานซิสเตอร์มีแรงดันลดลง โดยน้ำหนักของวัตถุหรือสสารที่ชั่งได้สูงสุดจะทำให้แรงดันเอาต์พุตได้ประมาณ 3.88 โวลต์ ซึ่งแรงดันเอาต์พุตนี้จะแตกต่างกัน 5 โวลต์พอดีทำให้ง่ายต่อการนำไปใช้งานในวงจรภาคไมโครคอนโทรลเลอร์ต่อไป

1) วงจรรับส่งโมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์

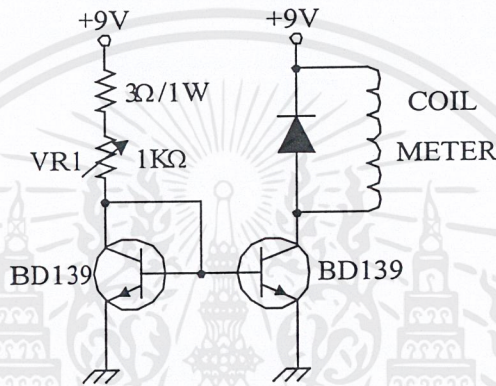
วงจรทางด้านส่ง LED อินฟราเรดให้กระแสไหลผ่าน LED ประมาณ 1.8 มิลลิแอมป์ และวงจรด้านรับโฟโตทรานซิสเตอร์จะใช้ตัวต้านทานปรับค่าได้ 10K Ω ปรับแรงดันเอาต์พุตขณะทำการชั่งน้ำหนักได้สูงสุดให้ได้แรงดันประมาณ 3.88 โวลต์ วงจรภาครับส่งแสดงได้ดังรูปที่ 3.5 โดยใช้เข็มมิเตอร์หรือแกนของตาชั่งตัดขวางเส้นทางการรับส่งทางแสงของวงจรทางด้านส่งและด้านรับ เพื่อให้แรงดันเอาต์พุตเปลี่ยนแปลง



รูปที่ 3.5 วงจรรับส่งโมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์

2) วงจรปรับกระแสให้ขดลวดมิเตอร์

วงจรปรับกระแสนี้จะใช้วงจรแบ่งกระแส (Current Miller) และตัวต้านทานปรับค่าได้ 1 K Ω เป็นตัวปรับกระแสให้แก่ขดลวดมิเตอร์ประมาณ 110 มิลลิแอมป์ ให้เข็มมิเตอร์หรือแกนของตาซึ่งอยู่ตำแหน่งกึ่งกลางพอดีหรือปรับให้ได้แรงดันเอาต์พุตที่โฟโตทรานซิสเตอร์มีค่าคงที่ประมาณ 8.88 โวลต์ เพื่อพร้อมที่จะใช้งานในการชั่งน้ำหนักของสสารหรือวัตถุต่อไป



รูปที่ 3.6 วงจรปรับกระแสให้ขดลวดมิเตอร์

3) วงจรส่วนภาคขยายของเครื่องชั่งน้ำหนักขนาดเล็ก

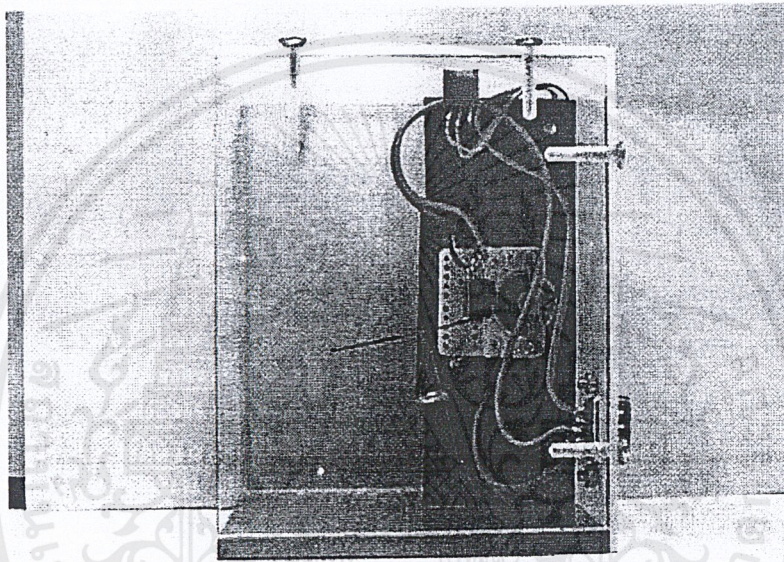
การทำงานของวงจรจะเริ่มจากเอาต์พุตที่ได้จากโฟโตทรานซิสเตอร์ โดยที่แรงดันจะเปลี่ยนแปลงตั้งแต่ 8.88 โวลต์ ถึง 3.88 โวลต์ แล้วต่อแรงดันเอาต์พุตเข้าสู่ขา 3 ของ Op-Amp1 (OP07) ซึ่งต่อในลักษณะของวงจรบัฟเฟอร์ (Buffer) เอาต์พุตของ Op-Amp1 นี้จะต่อเข้ากับ Op-Amp3 ซึ่งเป็นวงจรดิฟเฟอเรนเชียลแอมพลิไฟเออร์ (Differential Amplifier) และขาสัญญาณขั้วบวกของ Op-Amp2 ซึ่งเป็นวงจรบัฟเฟอร์นี้จะใช้ VR₁ เป็นตัวปรับให้ได้ค่าแรงดันคงที่เท่ากับ 8.88 โวลต์ แล้วป้อนเอาต์พุตเข้าสู่ Op-Amp3 วงจรดิฟเฟอเรนเชียลแอมพลิไฟเออร์เช่นกัน เพื่อจะทำหน้าที่หักล้างกับแรงดันที่ได้จากวงจรภาค Sensor หรือวงจรรับส่ง โมดูล LED อินฟราเรดและโฟโตทรานซิสเตอร์ ทำให้ได้แรงดัน 0 ถึง 5 โวลต์ สุดท้ายจะผ่านเข้าสู่วงจรโลพาสฟิลเตอร์อันดับสอง (Second order lowpass-filter) เพื่อกำจัดสัญญาณรบกวนก่อนที่จะเข้าสู่วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลหรือวงจรภาคไมโครคอนโทรลเลอร์ต่อไป

โดยความถี่คัทออฟของวงจรโลพาสฟิลเตอร์กำหนดได้ด้วยค่าของ R และ C ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

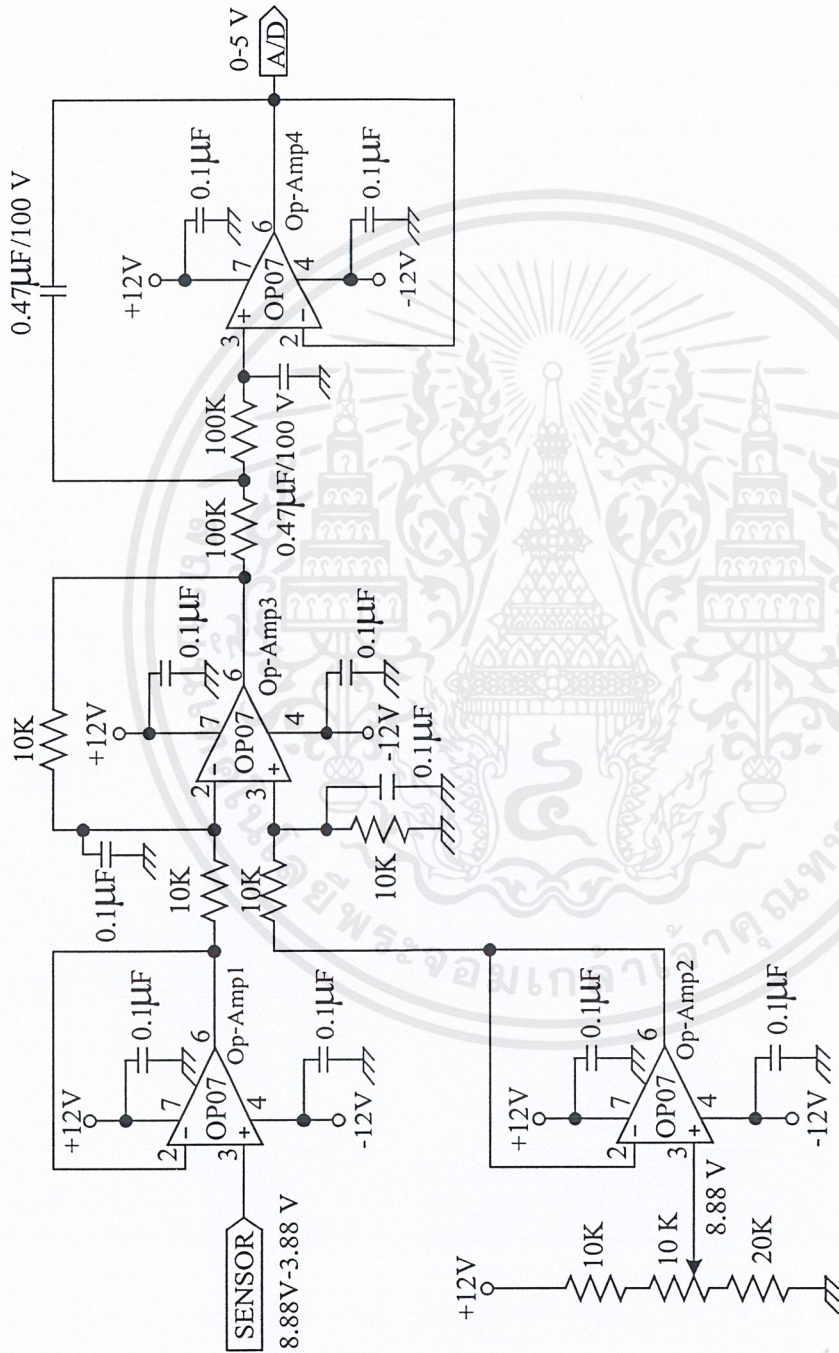
$$F_{OH} = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 100K \times 0.47\mu F} = 3.386 H_z \quad (3.2)$$

ลักษณะของเครื่องชั่งน้ำหนักขนาดเล็กแสดงได้ดังรูปที่ 3.7 และลักษณะวงจรส่วนภาคขยายของเครื่องชั่งน้ำหนักขนาดเล็กแสดงได้ดังรูปที่ 3.8



รูปที่ 3.7 เครื่องชั่งน้ำหนักขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรส่วนภาคขยายของเครื่องชั่งน้ำหนักขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรภาคไมโครคอนโทรลเลอร์

วงจรภาคไมโครคอนโทรลเลอร์เป็นส่วนที่ทำหน้าที่เชื่อมต่อวงจรภาค Sensor เข้ากับเครื่องคอมพิวเตอร์ ทำการแปลงสัญญาณอนาล็อกระดับแรงดัน 0 ถึง 5 โวลต์เป็นสัญญาณดิจิทัล 12 บิต โดยใช้ LTC1298 และทำการวัดด้วยระยะทางไกลโดยผ่านระบบเครือข่ายภายใน

3.3.1 คุณสมบัติของ LTC1298

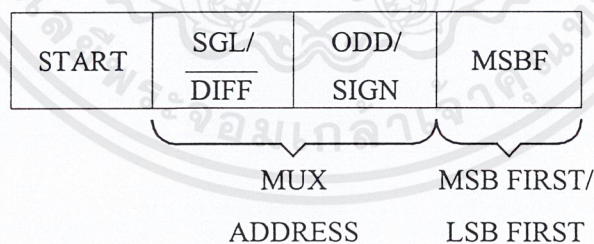
LTC1298 เป็นอุปกรณ์ที่ใช้ในการแปลงสัญญาณอนาล็อกเป็นดิจิทัล แบบ Serial ADC 2 Channel 12 บิต โดยมีรายละเอียดดังนี้

1) Data Transfer

การสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับ LTC1298 จะเป็นในลักษณะของ Synchronizes ระหว่าง Clock กับ Data

2) Input Data Word

LTC1298 จะต้องส่งอินพุท Input Data Word เข้าที่ขา D_{in} จำนวน 4 บิต ต่อจาก bit Start โดยมี Diagram ของ Input Data ดังรูปที่ 3.9



รูปที่ 3.9 Diagram ของ LTC1298

3) Start Bit

คือบิตแรกที่ Shift เข้าในขา D_{in} หลังจากขา CS มีลอจิก LOW

4) Multiplexer (MUX) Address

บิตที่ Shift ต่อจากบิต Start จะเป็นบิตที่ใช้เลือก Channel การอ่านสัญญาณอนาล็อก ซึ่งในรูปที่ 3.10 จะแสดงเงื่อนไขของ Channel Input ทั้ง 2 Channel โดยสามารถกำหนดให้ LTC1298 ทำงานในลักษณะ Single-ended หรือ Differential อีกทั้งสามารถที่จะกำหนดให้แต่ละ Channel ใดมีศักย์ไฟฟ้าเป็นบวกกับอีก Channel หนึ่งได้

LTC1298 Channel Selection

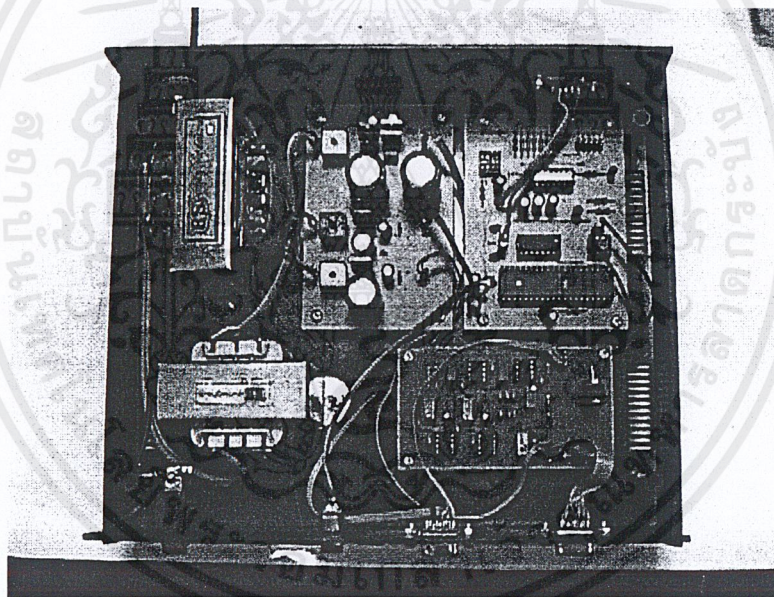
	MUX ADDRESS		CHANNEL #		GND
	SGL/DIFF	ODD/SIGN	0	1	
SINGLE-ENDED MUX MODE	1	0	+	—	—
	1	1	—	+	—
DIFFERENTIAL MUX MODE	0	0	+	—	—
	0	1	—	+	—

รูปที่ 3.10 เงื่อนไขของ Channel Input

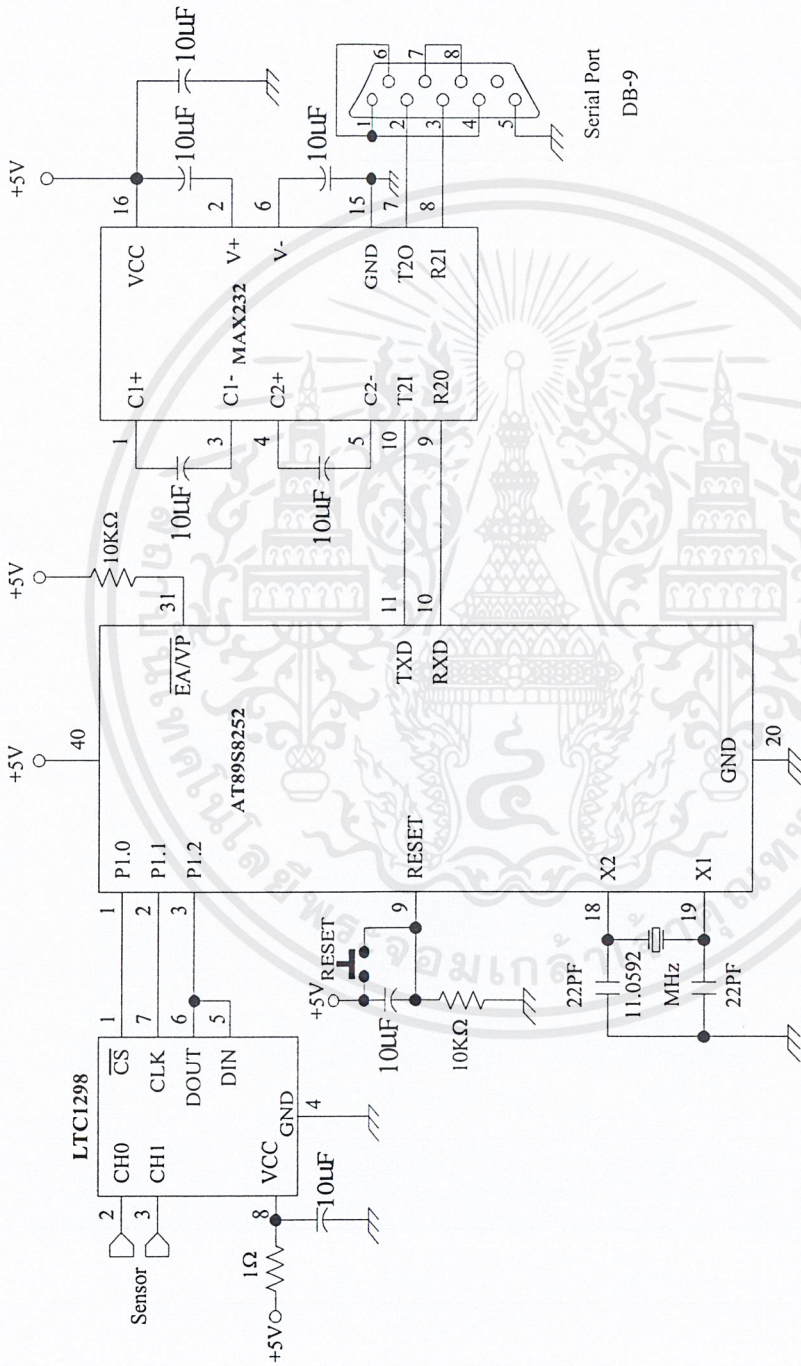
3.3.2 การทำงานของวงจรรภาคไมโครคอนโทรลเลอร์

วงจรรภาคไมโครคอนโทรลเลอร์ดังรูปที่ 3.12 จะประกอบไปด้วย ส่วนแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลโดยใช้ LTC1298 ขนาด 12 บิต 2 Channel แบบ Serial ซึ่งทำให้มีระดับของสัญญาณ 2^{12} เท่ากับ 4096 ระดับ ดังนั้นแต่ละระดับจะมีค่าแรงดันเท่ากับ 1.22 mV ต่อเข้ากับ Port 1 ของไมโครคอนโทรลเลอร์ AT89S8252 ซึ่งทำหน้าที่อ่านสัญญาณอินพุต เลือกช่องสัญญาณเข้ามาแล้วส่งต่อออกไปแสดงผลที่เครื่องคอมพิวเตอร์ โดยผ่านมาตรฐานการติดต่อแบบอนุกรม RS232C

ลักษณะของเครื่องเก็บบันทึกข้อมูลอัตโนมัติและวงจรรภาคไมโครคอนโทรลเลอร์แสดงดังรูปที่ 3.11 และลักษณะวงจรรส่วนภาคไมโครคอนโทรลเลอร์แสดงได้ดังรูปที่ 3.12



รูปที่ 3.11 เครื่องเก็บบันทึกข้อมูลอัตโนมัติและวงจรรภาคไมโครคอนโทรลเลอร์

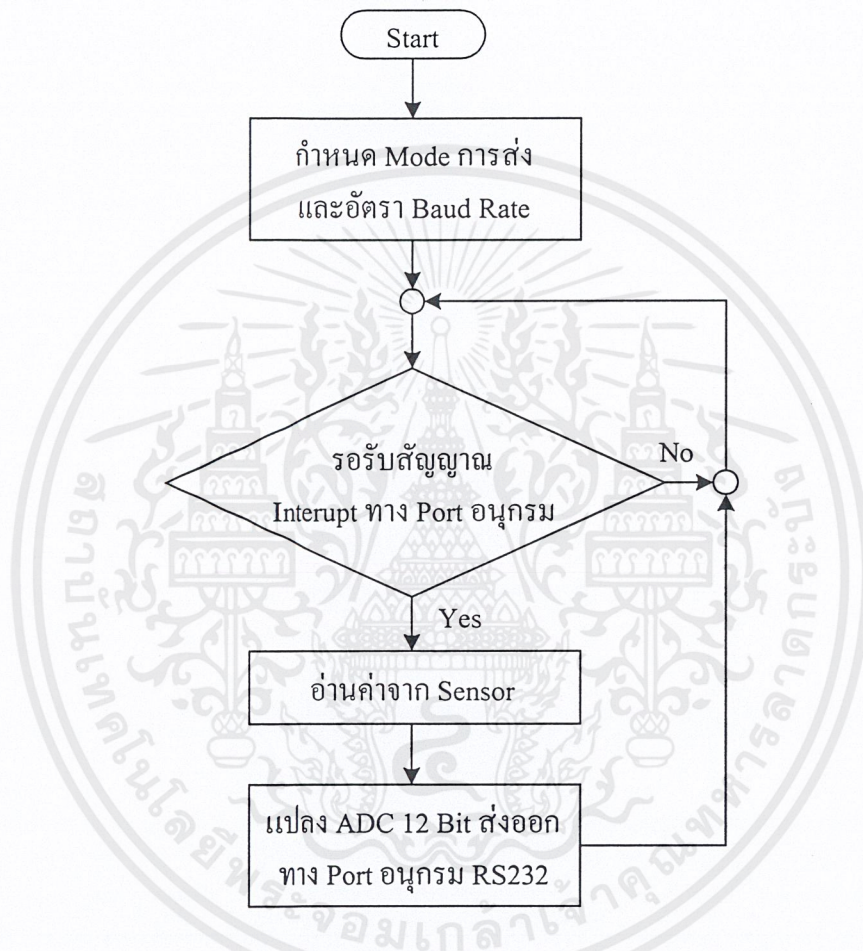


รูปที่ 3.12 วงจรส่วนภาคไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 โปรแกรมส่วนไมโครคอนโทรลเลอร์

ใช้ภาษาแอสเซมบลี (Assembly) และใช้ SXA51 เป็นตัวแปลภาษา (Compiler) แผนผังการทำงานแสดงดังรูปที่ 3.13



รูปที่ 3.13 แผนผังการทำงานของไมโครคอนโทรลเลอร์

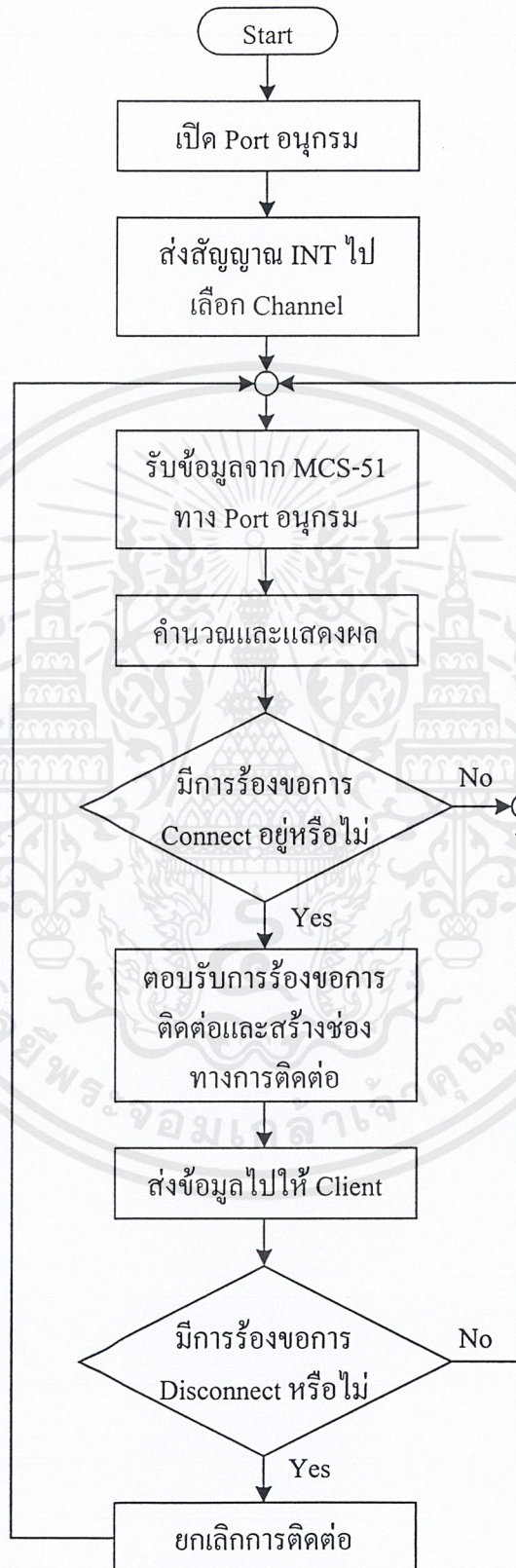
3.4 โปรแกรมส่วนติดต่อกับระบบเครือข่ายภายในโดยใช้โปรโตคอล TCP/IP

3.4.1 โปรแกรมส่วนของ Server

ส่วนของ Server เป็นส่วนที่สำคัญที่สุดของระบบการเก็บบันทึกข้อมูล โปรแกรมส่วนนี้จะประกอบไปด้วยโปรแกรมส่วนที่ทำหน้าที่เชื่อมต่อกับ Client และส่วนที่ทำหน้าที่เชื่อมต่อกับวงจรรักษาข้อมูลและเครื่องซึ่งนำหน้าขนาดเล็ก ซึ่งส่วนของ Server จะทำการเตรียมพร้อมรอรับการร้องขอทำการติดต่อกับ Client อยู่ตลอดเวลา โดยการทำงานปกติ Server จะทำการวัดค่าข้อมูลของสัญญาณอินพุตตลอดเวลา เมื่อมีการร้องขอการติดต่อกับ Client เครื่อง Server จึงจะทำการสร้างช่องทางการติดต่อและส่งสัญญาณตอบรับการติดต่อกลับไปให้ Client จากนั้นก็จะเริ่มทำการส่งข้อมูลที่วัดค่าได้จากพอร์ตอนุกรม จนกว่าจะมีการขอยกเลิกการติดต่อ Server จึงค่อยปิดช่องทางการติดต่อกับ Client และแสดงผลของข้อมูลและบันทึกผลบนไมโครคอมพิวเตอร์ โดยจะแสดงผลเป็นตัวเลขและกราฟเทียบกับแกนเวลา การทำงานของระบบทั้งหมดจะถูกควบคุมด้วยคอมพิวเตอร์ เช่น การวัดค่า , การตั้งเวลาในการวัด , การส่งผ่านระบบเครือข่ายภายใน , การเก็บบันทึกข้อมูล เป็นต้น ได้พัฒนาโดยใช้ภาษา Delphi™ แผนผังการทำงานแสดงได้ดังรูปที่ 3.14

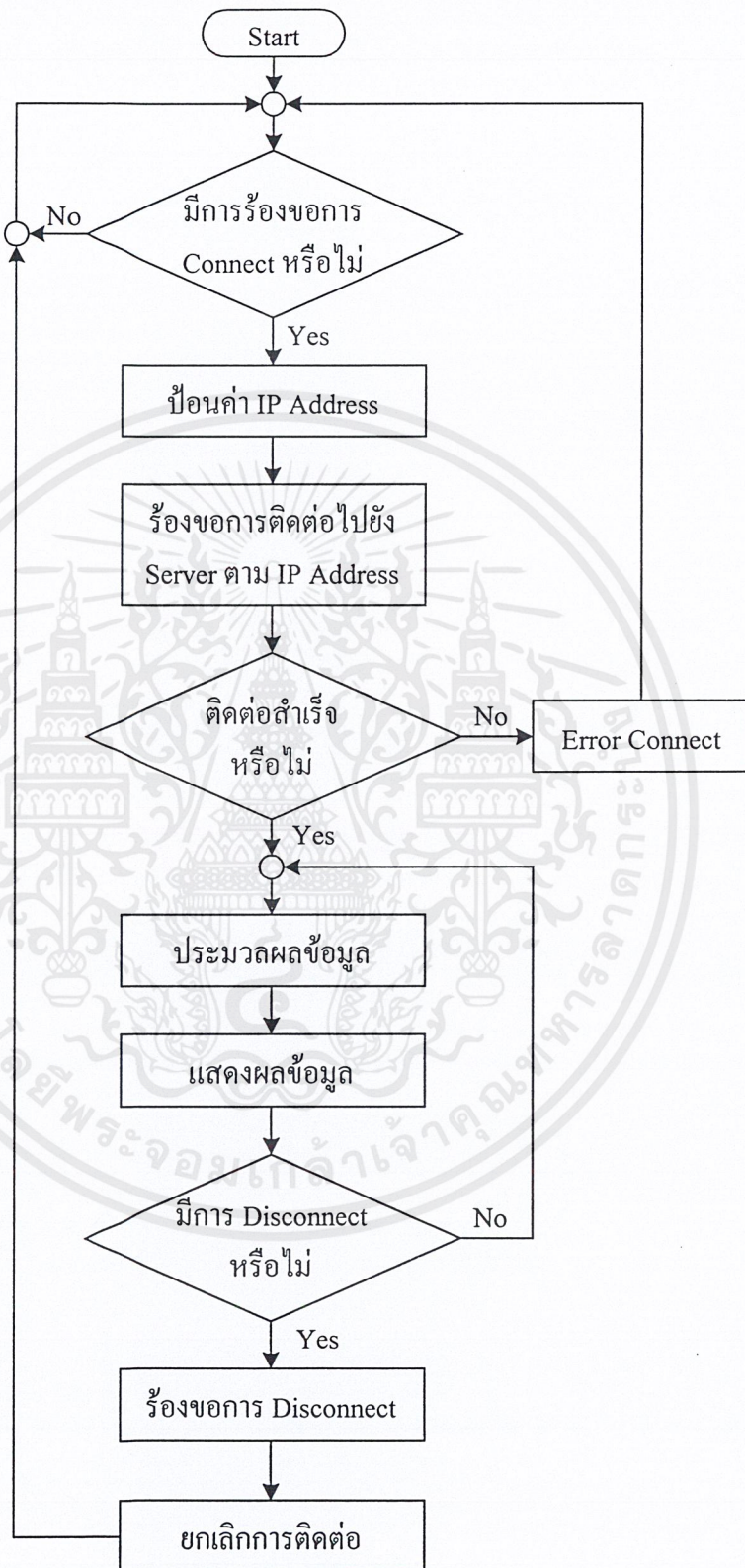
3.4.2 โปรแกรมส่วนของ Client

ส่วนของ Client การทำงานจะมีหน้าที่หลักๆ อยู่ 2 ส่วนคือส่วนที่ทำหน้าที่ในการทำการร้องขอการติดต่อกับ Server และส่งสัญญาณยกเลิกการติดต่อกับ Server กับอีกส่วนหนึ่งคือส่วนของการแสดงผล โดยจะแสดงผลเป็นตัวเลขและกราฟเปรียบเทียบกับเวลาต่างๆ แผนผังการทำงานแสดงได้ดังรูปที่ 3.15



รูปที่ 3.14 แผนผังการทำงานของโปรแกรมส่วน Server

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ห้ามการคัดลอก เหนือกว่าเงื่อนไขไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แผนผังการทำงานของโปรแกรมส่วน Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ส่วนของวงจรวัดอุณหภูมิ

ส่วนของวงจรวัดอุณหภูมิจะใช้ LM335 ในการวัดอุณหภูมิ จากนั้นทำการทดลองวัดค่าอุณหภูมิ 0 องศาเซลเซียส ถึง 50 องศาเซลเซียส แล้วเปรียบเทียบค่าที่วัดได้ระหว่างเทอร์โมมิเตอร์แบบปรอทกับค่าที่วัดได้บนหน้าจอกอมพิวเตอร์ และวัดค่าแรงดันเอาต์พุทของวงจรภาควัดอุณหภูมิ ดังแสดงในตารางที่ 4.1

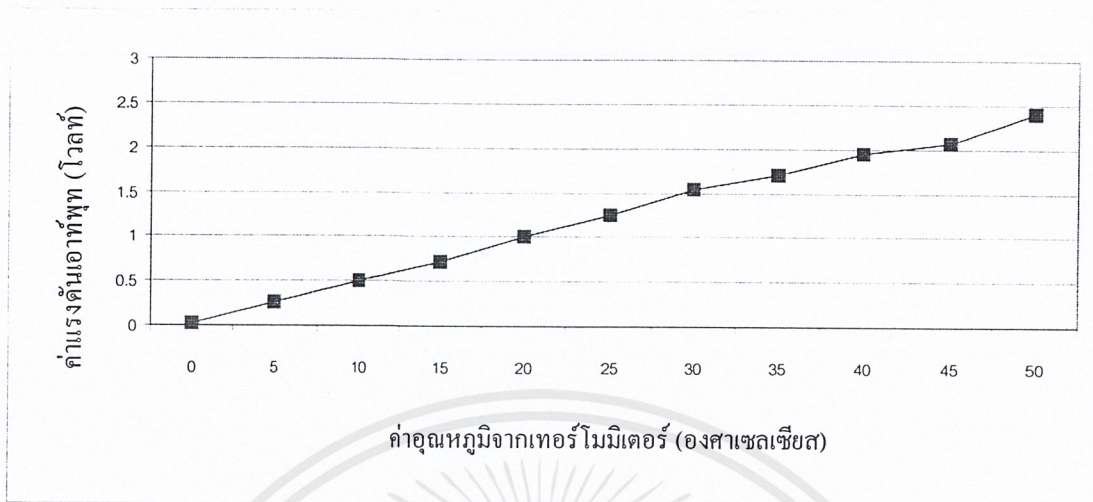
ตารางที่ 4.1 ผลของการวัดค่าแรงดันเอาต์พุทของวงจรภาควัดอุณหภูมิเปรียบเทียบกับเทอร์โมมิเตอร์แบบปรอท

ค่าที่วัดได้จากเทอร์โมมิเตอร์ (°C)	ค่าที่วัดได้จากเอาต์พุทของภาควัดอุณหภูมิ (V)	ค่าที่คำนวณได้ (°C)	ค่าความผิดพลาด
0	0.02	0.4	0.02
5	0.26	5.2	0.04
10	0.50	10	0.00
15	0.72	14.4	0.04
20	1.02	20.4	0.02
25	1.25	25	0.00
30	1.54	30.8	0.02
35	1.71	34.2	0.02
40	1.95	39	0.02
45	2.07	41.4	0.08
50	2.39	47.8	0.04

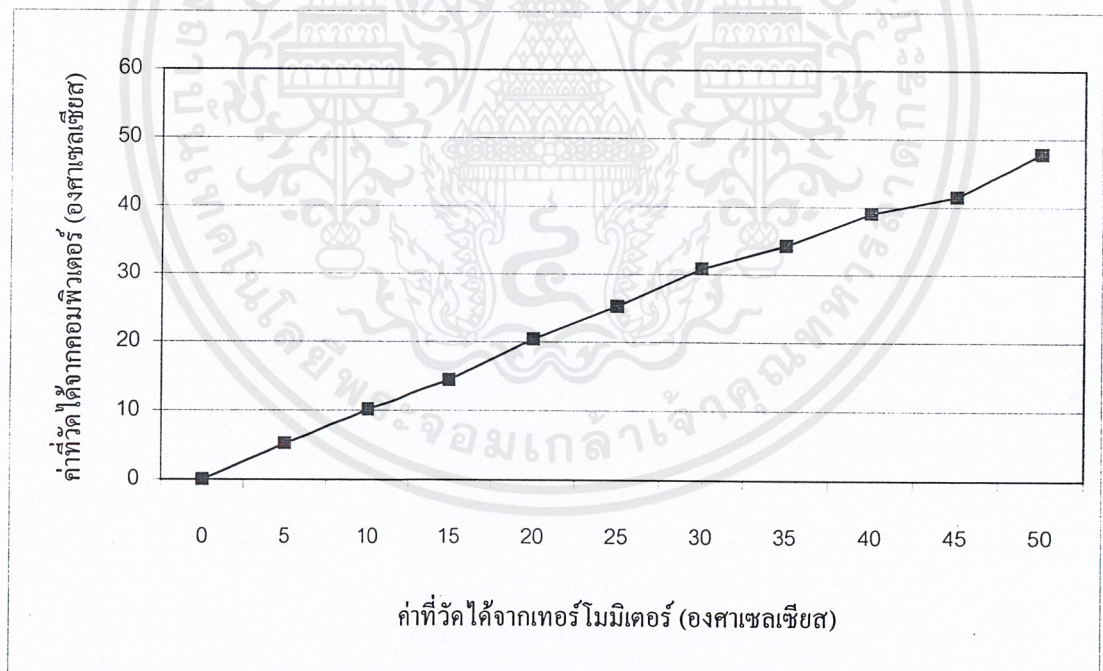
ตารางที่ 4.2 ผลของการวัดค่าอุณหภูมิจากคอมพิวเตอร์เปรียบเทียบกับเทอร์โมมิเตอร์แบบปรอท

ค่าที่วัดได้จากเทอร์โมมิเตอร์ (°C)	ค่าที่วัดได้จากคอมพิวเตอร์ (°C)	ค่าความผิดพลาด
0	0	0.00
5	5.23	0.04
10	10.15	0.01
15	14.45	0.03
20	20.42	0.02
25	25.26	0.01
30	30.84	0.02
35	34.25	0.02
40	39.06	0.02
45	41.49	0.07
50	47.88	0.04

จากตารางผลการทดลองจะเห็นได้ว่าค่าอุณหภูมิที่วัดได้จากเครื่องวัดอุณหภูมิจะมีค่าใกล้เคียงกับค่าอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์แบบปรอท แต่เมื่ออุณหภูมิมีค่าเพิ่มขึ้น ค่าที่วัดได้จะเริ่มมีความผิดพลาดสูงขึ้นตาม อาจเนื่องมาจากความผิดพลาดในขณะที่ทำการอ่านและค่าความผิดพลาดจากตัวอุปกรณ์



รูปที่ 4.1 กราฟแรงดันเอาต์พุตจากวงจรวัดอุณหภูมิเปรียบเทียบกับค่าอุณหภูมิจากเทอร์โมมิเตอร์แบบปรอท

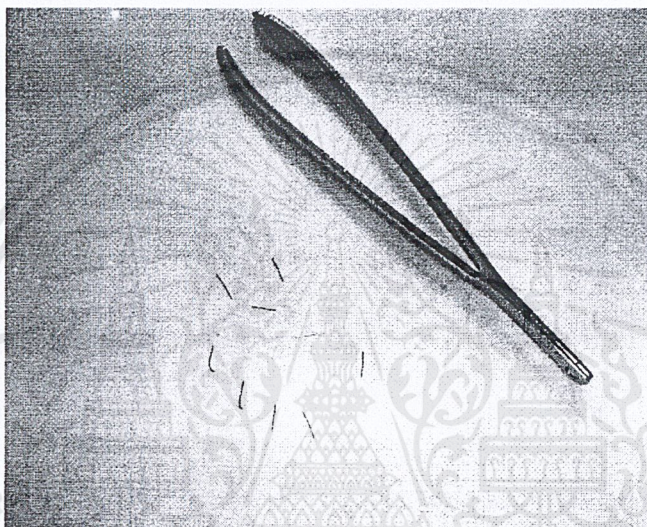


รูปที่ 4.2 กราฟเปรียบเทียบอุณหภูมิที่วัดได้จากเครื่องวัดอุณหภูมิกับอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์แบบปรอท

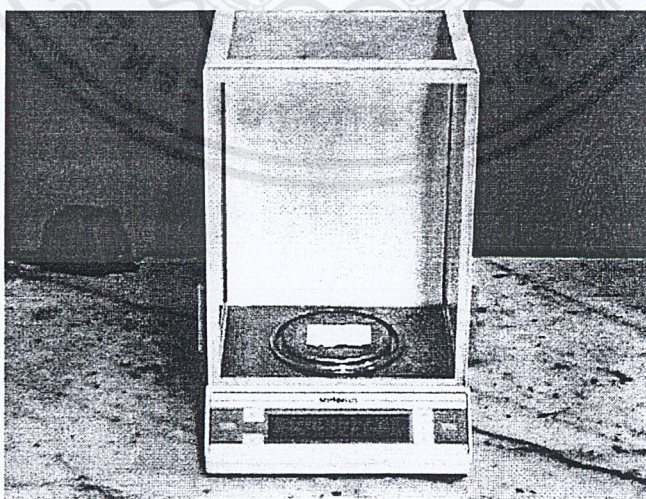
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของเครื่องชั่งน้ำหนักขนาดเล็ก

ทำการตัดเส้นลวดให้มีขนาดเล็กๆ ดังรูปที่ 4.3 แล้วนำไปชั่งน้ำหนักบนน้ำหนักมาตรฐานเป็นมิลลิกรัมกับเครื่องชั่งน้ำหนักมาตรฐานแสดงดังรูปที่ 4.4 เพื่อนำมาเปรียบเทียบกับเครื่องชั่งน้ำหนักขนาดเล็กที่ได้สร้างขึ้นมา ซึ่งเมื่อหาค่าน้ำหนักเฉลี่ยของเส้นลวดแต่ละเส้นแล้วจะได้น้ำหนักเฉลี่ยเส้นละประมาณ 0.601 มิลลิกรัม



รูปที่ 4.3 เส้นลวดขนาดเล็กที่นำไปชั่งน้ำหนักกับเครื่องชั่งน้ำหนักมาตรฐาน



รูปที่ 4.4 เครื่องชั่งน้ำหนักมาตรฐานยี่ห้อ Sartorius รุ่น BD210D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการทดลองชั่งน้ำหนักเส้นลวด แล้วเปรียบเทียบน้ำหนักเส้นลวดที่ชั่งโดยเครื่องชั่งน้ำหนักมาตรฐานกับค่าที่ชั่งได้บนหน้าจอคอมพิวเตอร์ และวัดค่าแรงดันเอาต์พุตของวงจรเครื่องชั่งน้ำหนัก ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ตารางผลของการวัดค่าแรงดันเอาต์พุตของวงจรเครื่องชั่งน้ำหนัก กับ น้ำหนักโดยเฉลี่ย

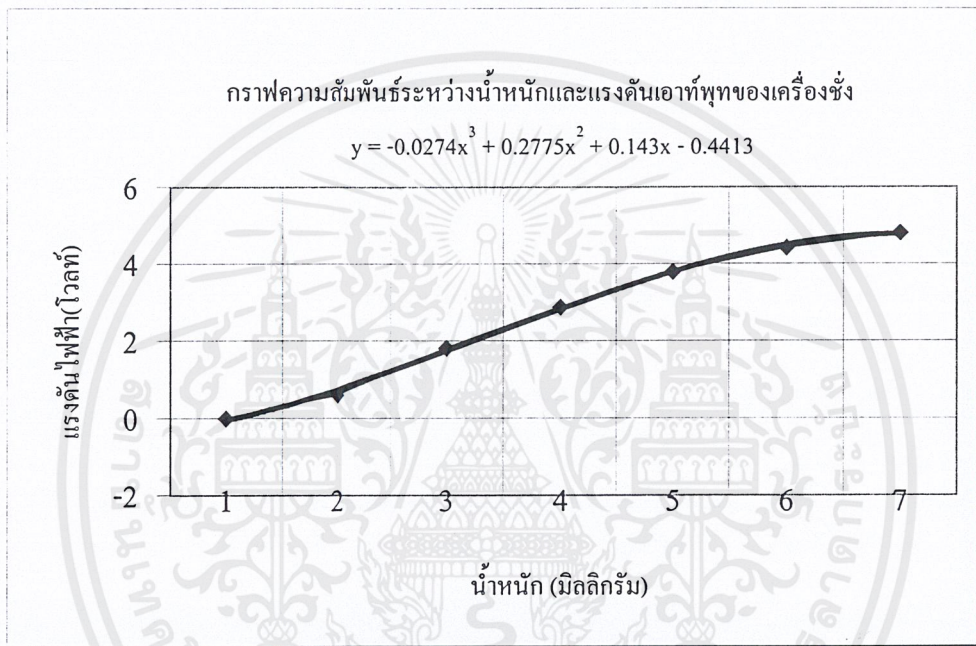
ค่าน้ำหนักเฉลี่ยที่นำมาชั่ง (มิลลิกรัม)	ค่าแรงดันที่วัดได้จากเอาต์พุต ของเครื่องชั่งน้ำหนัก (โวลต์)
2.404	70.4 mV
3.005	0.608 V
3.606	1.814 V
4.207	2.871 V
4.808	3.797 V
5.409	4.42 V
6.010	4.81 V

ตารางที่ 4.4 ผลของการชั่งน้ำหนักจากคอมพิวเตอร์เปรียบเทียบกับเครื่องชั่งน้ำหนักมาตรฐาน

ค่าน้ำหนักเฉลี่ยที่นำมาชั่ง (มิลลิกรัม)	ค่าที่วัดได้จากคอมพิวเตอร์ (มิลลิกรัม)	ค่าความผิดพลาด
2.404	0.08	0.96
3.005	0.94	0.68
3.606	2.82	0.21
4.207	5.01	0.19
4.808	5.88	0.22
5.409	6.85	0.26
6.010	7.48	0.24

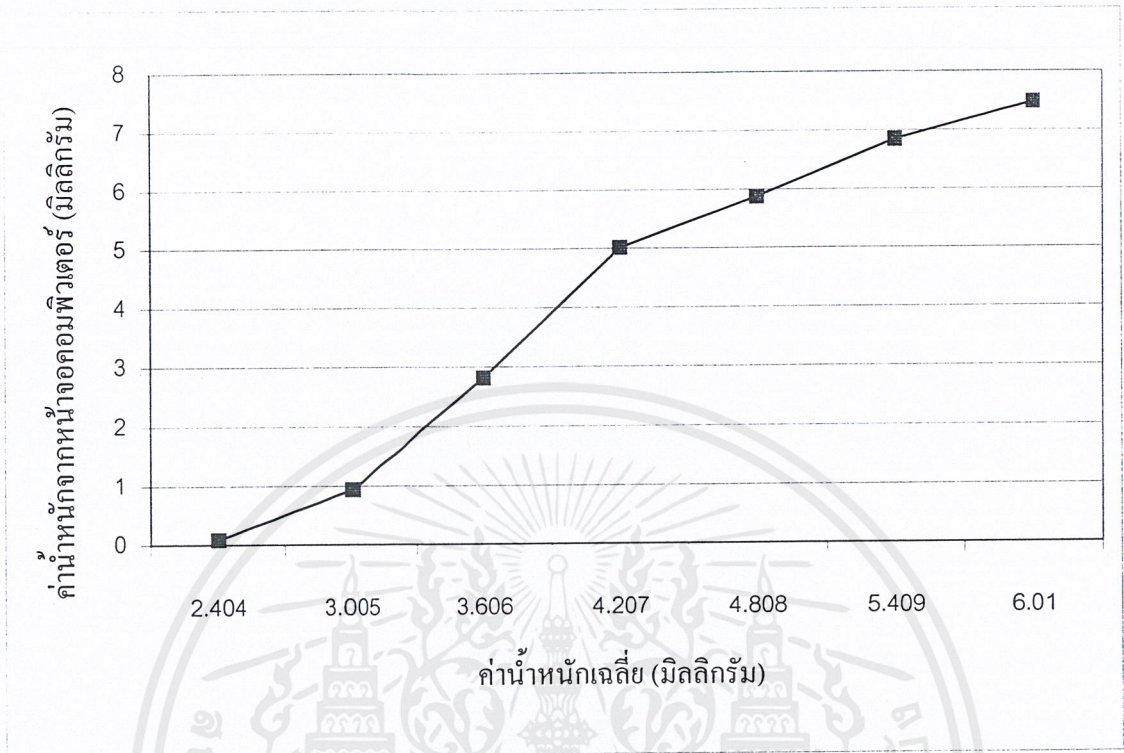
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางผลการทดลองที่ 4.4 จะเห็นได้ว่าค่าแรงดันเอาต์พุตที่ได้จากเครื่องชั่งน้ำหนักนั้น ยังไม่เป็นเชิงเส้น และค่าน้ำหนักที่วัดได้จากหน้าจอบทคอมพิวเตอร์กับค่าน้ำหนักเปรียบเทียบกับเครื่องชั่งมาตรฐานยังมีค่าความผิดพลาดอยู่ โดยเฉพาะค่าน้ำหนักที่อยู่ในช่วงประมาณ 0 ถึง 3.6 มิลลิกรัมจะมีค่าความผิดพลาดสูงมาก เนื่องมาจากโครงสร้างทาง Hardware ของเครื่องชั่งน้ำหนัก ขนาดเล็กนั่นเอง



รูปที่ 4.5 กราฟความสัมพันธ์ระหว่างน้ำหนักและแรงดันเอาต์พุตของเครื่องชั่ง

กราฟรูปที่ 4.5 เป็นการหาความสัมพันธ์ระหว่างน้ำหนักและแรงดันเอาต์พุตของเครื่องชั่ง น้ำหนักขนาดเล็กที่วัดได้จากผลการทดลองตารางที่ 4.3 ซึ่งจะได้ฟังก์ชันสมการคณิตศาสตร์ดังนี้ คือ $y = -0.0274x^3 + 0.2775x^2 + 0.143x - 0.4413$ แล้วนำสมการที่ได้นี้ไปคำนวณในส่วนของโปรแกรมที่เครื่อง Server เพื่อคำนวณหาค่าน้ำหนักที่ชั่งได้ออกมาแสดงผล



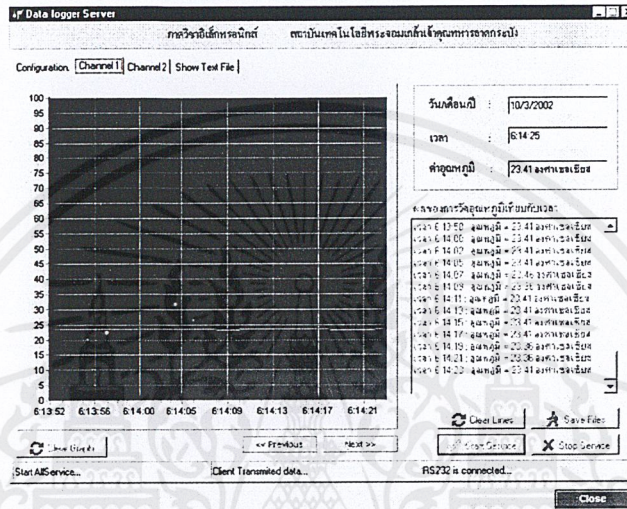
รูปที่ 4.6 กราฟเปรียบเทียบค่าน้ำหนักที่วัดได้จากเครื่องชั่งน้ำหนักขนาดเล็ก
เปรียบเทียบกับน้ำหนักเฉลี่ยที่นำมาชั่ง

จากกราฟที่ 4.6 จะเห็นได้ว่าค่าน้ำหนักที่นำมาชั่งกับเครื่องชั่งน้ำหนักขนาดเล็กได้นี้จะอยู่ในช่วงประมาณ 2.4 ถึง 6 มิลลิกรัม เนื่องจากโครงสร้างทาง Hardware ของเครื่องชั่งน้ำหนักขนาดเล็กนั่นเอง

4.3 ส่วนของโปรแกรม Server

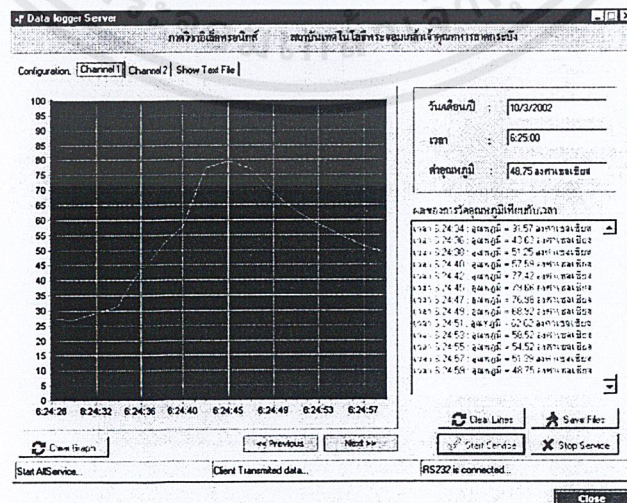
4.3.1 ส่วนของ Channel 1 ที่ใช้วัดอุณหภูมิ

1. ทำการวัดอุณหภูมิที่เครื่อง Server โดยทำการวัดค่าของอุณหภูมิห้องปกติ จะปรากฏหน้าจอที่แสดงผลของการวัดอุณหภูมิดังรูปที่ 4.7



รูปที่ 4.7 ผลของการวัดค่าอุณหภูมิห้องปกติ

2. ทำการเปลี่ยนแปลงอุณหภูมิโดยให้ความร้อนแก่ LM335 แล้วให้ LM335 เกิดการเย็นตัว จะปรากฏหน้าจอแสดงผลดังรูปที่ 4.8

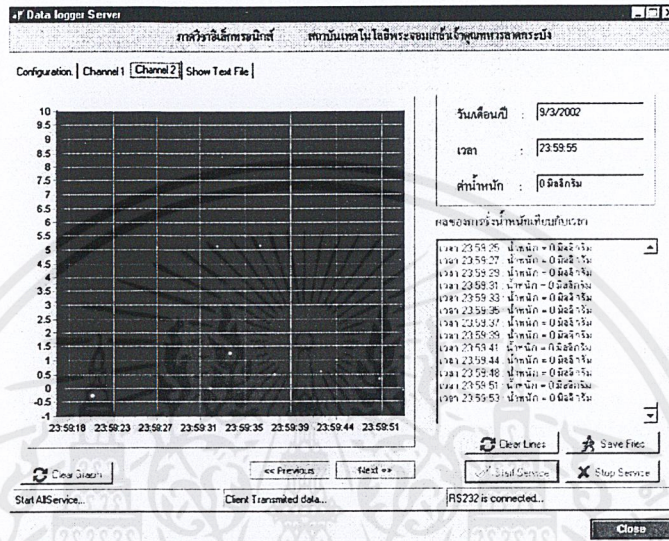


รูปที่ 4.8 ผลของการวัดค่าอุณหภูมิโดยให้ความร้อนแก่ LM335 และให้ LM335 เกิดการเย็นตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

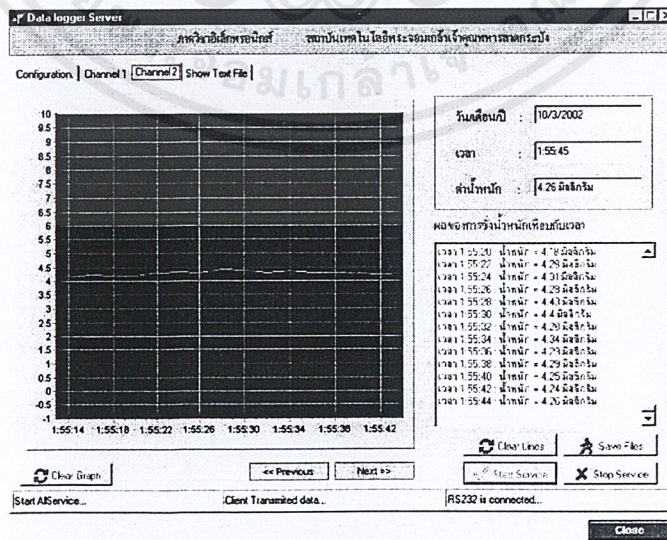
4.3.2 ส่วนของ Channel 2 ที่ใช้ข้งนำหน้า

1. ทำการทดสอบการข้งนำหน้า ในกรณีที่ข้งนำไม่ได้ นำเอาวัตถุใดๆ มาข้งที่เครื่องข้ง จะปรากฏหน้าจอแสดงผลดังรูปที่ 4.9



รูปที่ 4.9 ขณที่ข้งนำไม่ได้ ทำการข้งวัตถุใดๆ

2. ทำการทดสอบ โดยการนำข้งเส้นลวดเล็กๆ มาข้ง จะปรากฏหน้าจอแสดงผลของการข้งวัตถุดังรูปที่ 4.10

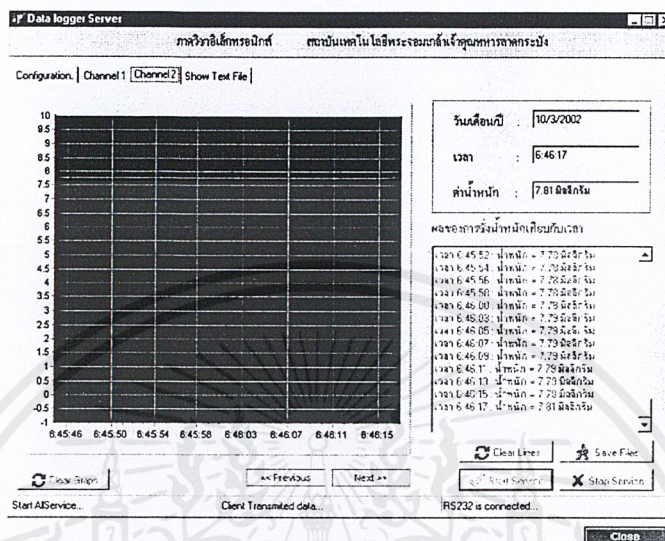


รูปที่ 4.10 ผลของนำการข้งวัตถุมาข้งนำหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อวัตถุที่นำมาชั่งมีน้ำหนักมากเกินกว่า 8 มิลลิกรัม จะปรากฏหน้าจอแสดงผลดังรูปที่

4.11

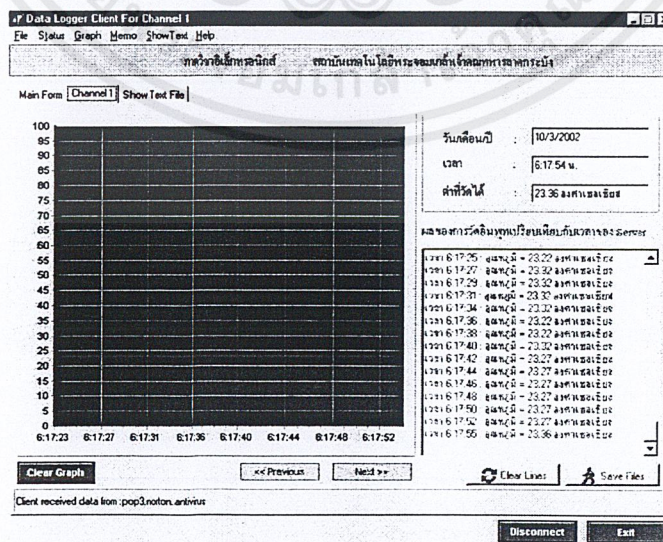


รูปที่ 4.11 เมื่อวัตถุมีน้ำหนักมากเกินกว่า 8 มิลลิกรัม

4.4 ส่วนของโปรแกรม Client

4.4.1 โปรแกรม Client สำหรับการวัดอุณหภูมิ

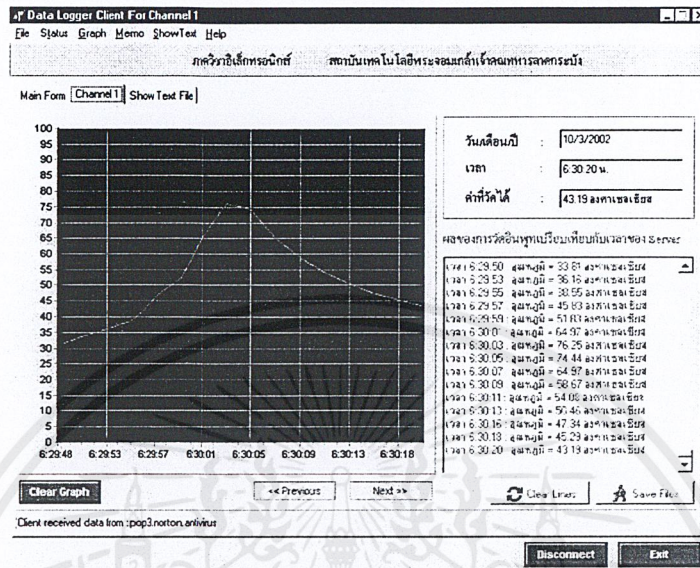
1. แสดงผลของการวัดค่าอุณหภูมิห้องปกติจากเครื่อง Server ดังรูปที่ 4.12



รูปที่ 4.12 แสดงผลของการวัดค่าอุณหภูมิห้องปกติจากเครื่อง Server

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับโรงเรียนที่ออกให้เท่านั้น มิใช่ให้ผู้เห็นแจ้งประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

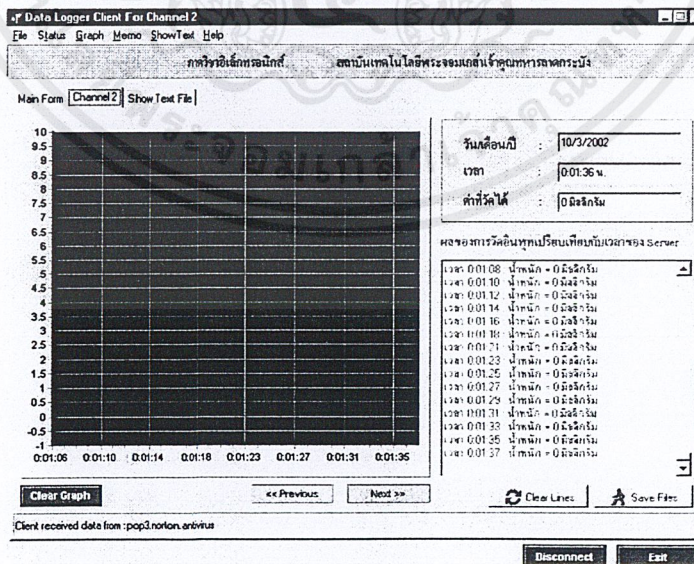
2. แสดงผลของการเปลี่ยนแปลงค่าอุณหภูมิจากเครื่อง Server ดังรูปที่ 4.13



รูปที่ 4.13 แสดงผลของการเปลี่ยนแปลงค่าอุณหภูมิจากเครื่อง Server

4.4.2 โปรแกรม Client สำหรับการซิงค์น้ำหนัก

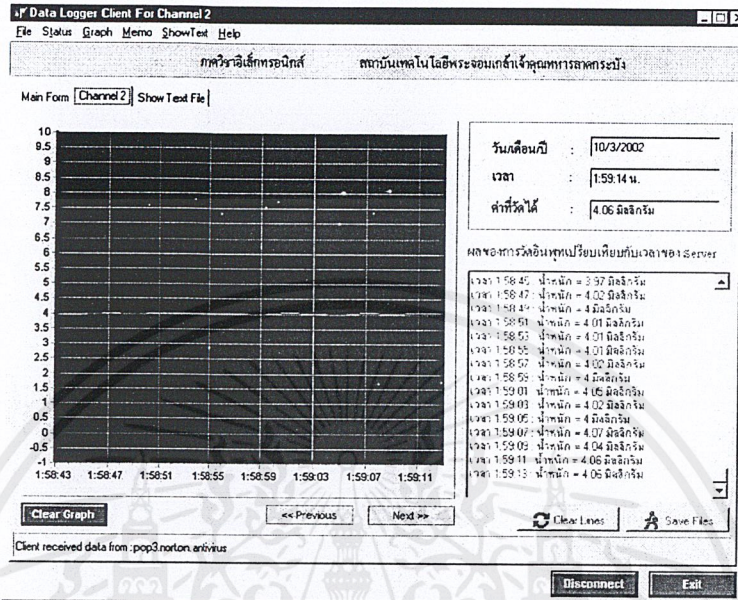
1. แสดงผลในขณะที่ยังไม่ได้นำเข้าวัตถุใดๆ มาซิงค์ที่เครื่องซิงค์ ดังรูปที่ 4.14



รูปที่ 4.14 แสดงผลในขณะที่ยังไม่ได้นำเข้าวัตถุใดๆ มาซิงค์ที่เครื่องซิงค์จากเครื่อง Server

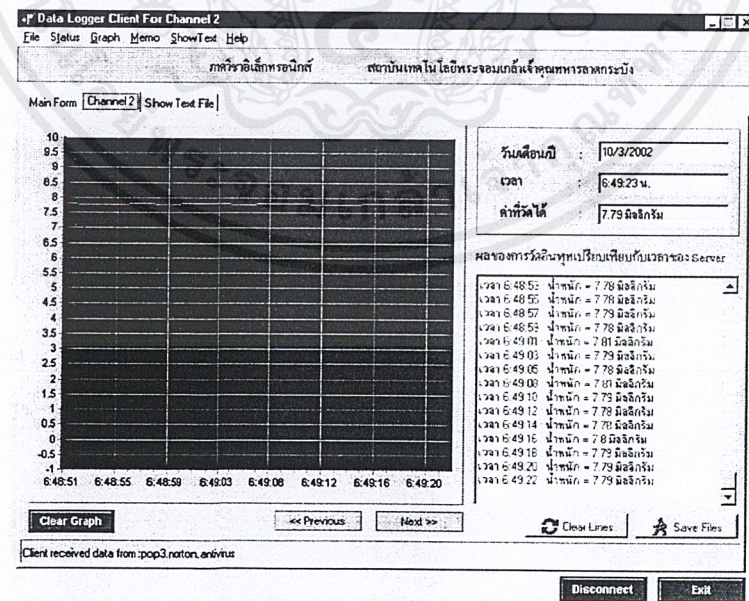
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แสดงผลของการชั่งเส้นลวดเล็กๆ ดังรูปที่ 4.15



รูปที่ 4.15 แสดงผลของการนำวัตถุมาชั่งน้ำหนักจากเครื่อง Server

3. แสดงผลเมื่อชั่งวัตถุที่มีน้ำหนักมากกว่า 8 มิลลิกรัม ดังรูปที่ 4.16



รูปที่ 4.16 แสดงผลเมื่อวัตถุมีน้ำหนักมากกว่า 8 มิลลิกรัมจากเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าค่าอุณหภูมิและค่าน้ำหนักที่วัดได้จากเครื่อง Client ทั้ง 2 ส่วนนี้ สามารถรับข้อมูลได้ตรงกับค่าที่วัดได้จากเครื่อง Server แต่ความเร็วในการติดต่อสื่อสารข้อมูลจะขึ้นอยู่กับเครื่องคอมพิวเตอร์ด้วย

จากการทดลองการติดต่อสื่อสารข้อมูลทั้งหมดนี้ใช้เครื่องคอมพิวเตอร์ทดสอบเพียงเครื่องเดียว โดยทำการร้องขอการติดต่อไปที่ localhost



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

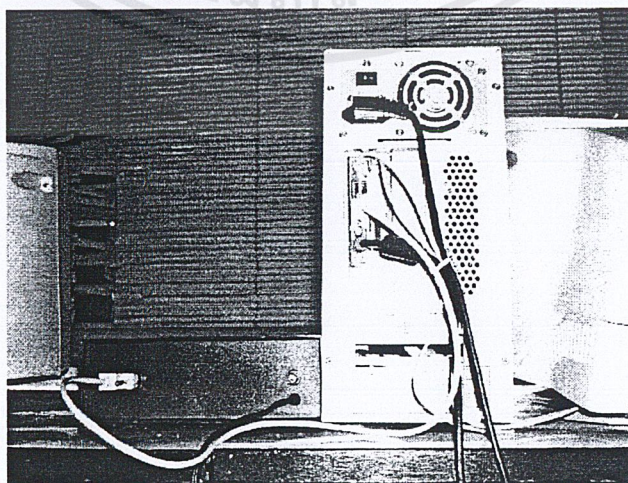
การนำไปใช้งาน

การนำไปใช้งานของเครื่องเก็บบันทึกข้อมูลอัตโนมัติส่งผ่านระบบเครือข่ายภายในนั้น จะเน้นการใช้งานภายในสถานที่เป็นหลักแบ่งออกเป็น 2 ช่องสัญญาณ (Channel) ได้แก่ ช่องสัญญาณที่ 1 (Channel 1) ใช้เก็บบันทึกข้อมูลอุณหภูมิของอากาศโดยมีย่านการวัดอุณหภูมิอยู่ในช่วงระหว่าง 0 ถึง 100 องศาเซลเซียสและช่องสัญญาณที่ 2 (Channel 2) ใช้เก็บบันทึกข้อมูลน้ำหนักของสสารหรือวัตถุนานเล็กโดยมีย่านการชั่งน้ำหนักอยู่ในช่วงระหว่าง 0 ถึง 7.8 มิลลิกรัม

ทำการติดตั้งโปรแกรม Data logger Server ยังเครื่องคอมพิวเตอร์ที่ใช้เป็นเครื่อง Server ที่จะบันทึกข้อมูลและที่โฟลเดอร์ My Documents ให้สร้างโฟลเดอร์ใหม่ คือ Server File และที่โฟลเดอร์นี้สร้างโฟลเดอร์ คือ Channel 1 และ Channel 2 ทำการติดตั้งโปรแกรม Data logger Client ทั้ง 2 ช่องสัญญาณยังเครื่องคอมพิวเตอร์ที่ใช้เป็นเครื่อง Client เพื่อใช้ดูข้อมูลจากเครื่อง Server โดยที่โฟลเดอร์ My Documents ให้สร้างโฟลเดอร์ใหม่ คือ Client File ที่โฟลเดอร์นี้สร้างโฟลเดอร์ คือ Channel 1 และ Channel 2 และปรับความละเอียดของหน้าจอที่ 1024 x 768 pixels

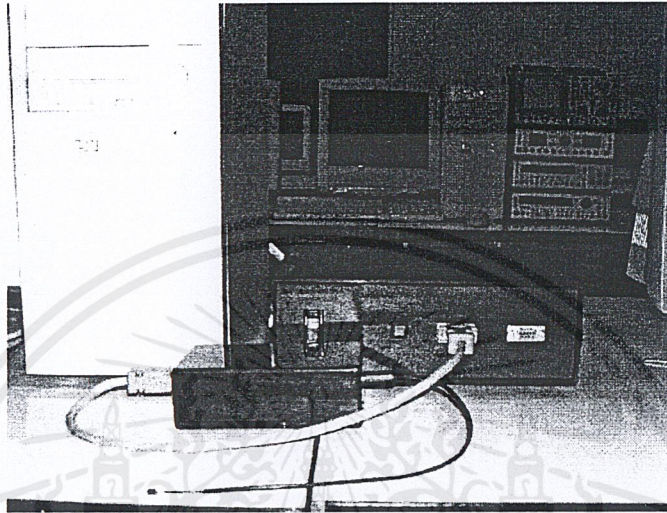
5.1 การติดตั้งอุปกรณ์ที่เครื่องคอมพิวเตอร์ Server ที่ใช้บันทึกข้อมูล

1. นำสาย RS-232 มาติดตั้งเข้ากับ Serial Port ของเครื่องคอมพิวเตอร์และที่ Serial Port ของเครื่อง Data logger ดังรูปที่ 5.1



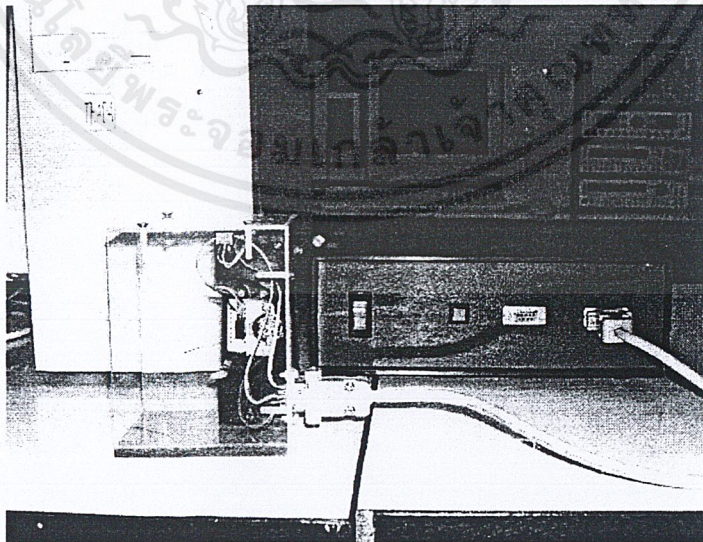
เอกสารนี้เป็นเอกสารลิขสิทธิ์ การติดตั้งสาย RS-232 กับเครื่อง Data Logger และเครื่องคอมพิวเตอร์ รับผิดชอบด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. นำสาย Port มาติดตั้งเข้ากับกล่องวัดอุณหภูมิและเครื่อง Data Logger ที่ช่องสัญญาณที่ 1 ดังรูปที่ 5.2



รูปที่ 5.2 การติดตั้งสาย Port กับกล่องวัดอุณหภูมิและช่องสัญญาณที่ 1

3. นำสาย Port มาติดตั้งที่เครื่องชั่งน้ำหนักขนาดเล็กและเครื่อง Data Logger ที่ช่องสัญญาณที่ 2 ดังรูปที่ 5.3

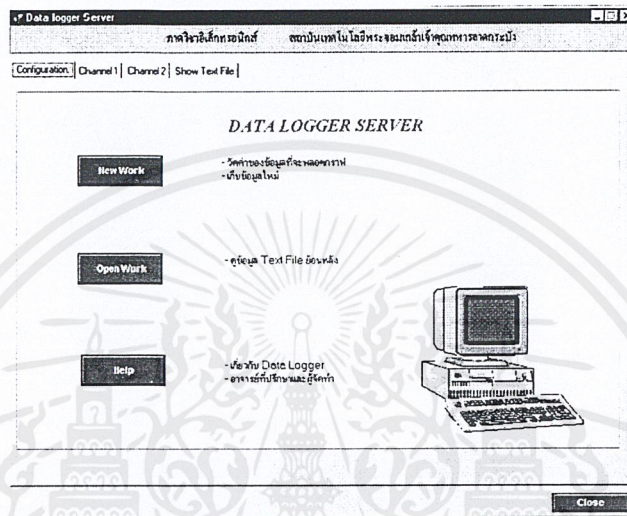


รูปที่ 5.3 การติดตั้งสาย Port กับเครื่องชั่งน้ำหนักขนาดเล็กและช่องสัญญาณที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

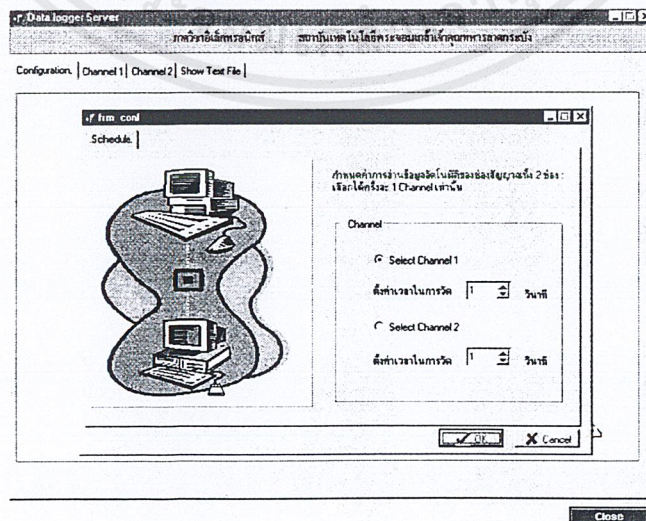
5.1.1 การใช้งานโปรแกรมส่วนการวัดอุณหภูมิ

เมื่อติดตั้งอุปกรณ์ทั้งหมดเสร็จเรียบร้อยแล้วเปิดสวิทซ์เครื่อง Data logger และเปิดโปรแกรม Data logger Server จะปรากฏหน้าจอแสดงในรูปที่ 5.4



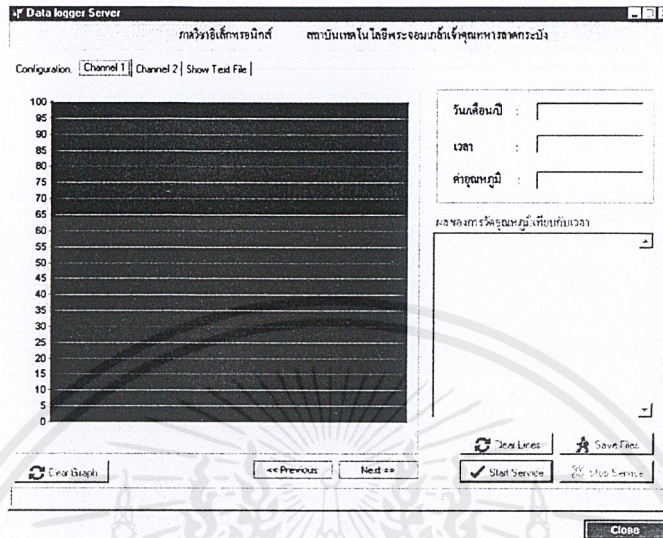
รูปที่ 5.4 โปรแกรมส่วน Data Logger Server

1. ทำการเลือก New Work เพื่อทำการวัดค่าของข้อมูลที่จะพลอตกราฟและเก็บบันทึกข้อมูลใหม่ จะปรากฏหน้าจอแสดงในรูปที่ 5.5
2. เลือก Select Channel 1 โดยสามารถตั้งค่าเวลาในการวัดได้ตั้งแต่ 1 ถึง 60 วินาที



เอกสารนี้เป็นเอกสารรูปที่ 5.5 หน้าจอในส่วนของการเลือกช่องสัญญาณอินพุตและตั้งค่าเวลาในการวัดในด้านการค้าไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

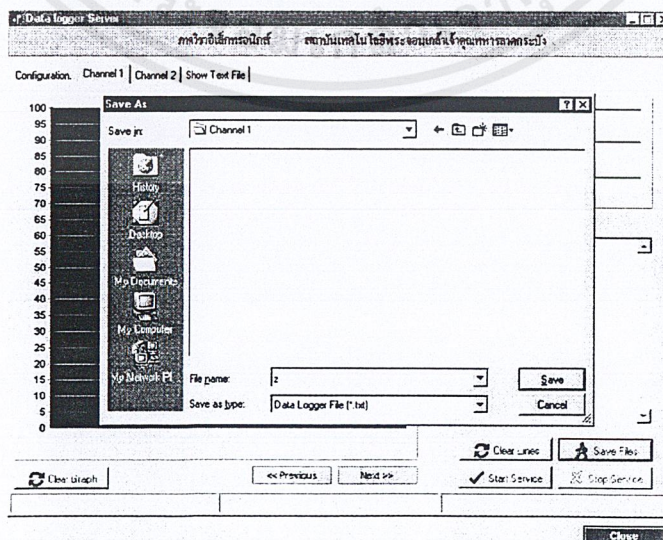
3. เมื่อตั้งค่าเรียบร้อยแล้วเลือกปุ่ม OK จะปรากฏหน้าจอดังรูปที่ 5.6



รูปที่ 5.6 หน้าจอในส่วนของ Channel 1

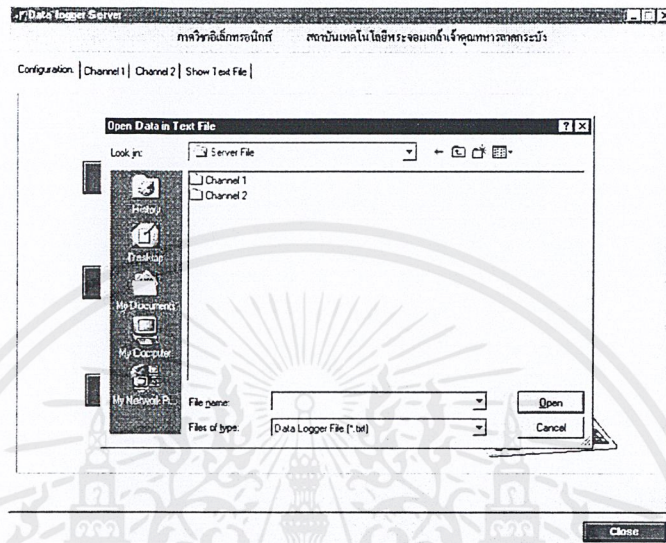
4. จากนั้นเลือก Start Service โปรแกรมจะทำการวัดอุณหภูมิแสดงเป็นตัวเลขและกราฟออกมาเมื่อต้องการหยุดวัดอุณหภูมิให้เลือก Stop Service

5. เมื่อต้องการลบกราฟให้เลือก Clear Graph เมื่อต้องการลบตัวเลขแสดงผลให้เลือก Clear Lines และเมื่อต้องการเก็บข้อมูลให้เลือก Stop Service จากนั้นเลือก Save Files จะปรากฏหน้าจอ ดังรูปที่ 5.7 ใส่ชื่อไฟล์ข้อมูลนั้นแล้วเลือก Save ข้อมูลจะถูกบันทึกลงไมโครคอมพิวเตอร์



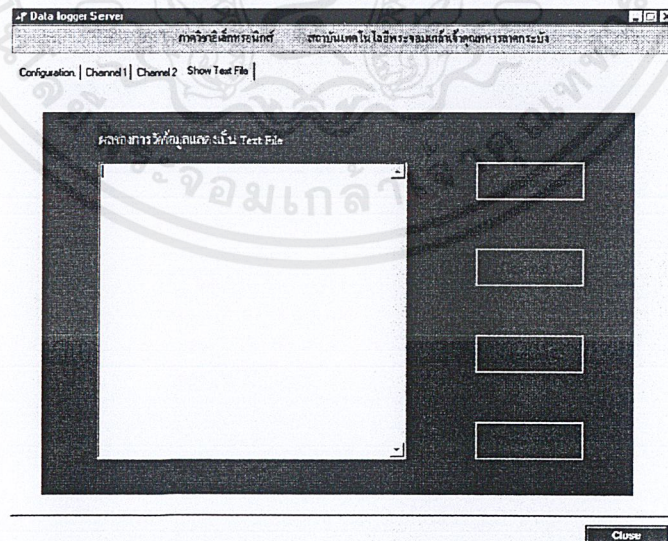
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาควิชาวิศวกรรมเครื่องกลเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.7 หน้าจอใช้ในการเก็บข้อมูล
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อต้องการดูข้อมูลที่เก็บไว้ให้กลับมาที่หน้าจอหลักดังรูปที่ 5.4 เลือก Open Work และเลือกข้อมูลที่จะเปิด เลือก Open จะปรากฏข้อมูลบนหน้าจอ ดังรูปที่ 5.8



รูปที่ 5.8 หน้าจอใช้ในการดูข้อมูลที่เก็บไว้

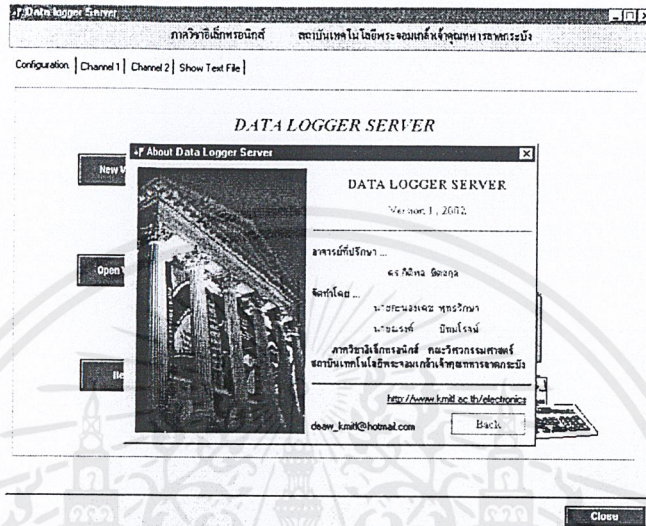
7. จะปรากฏหน้าจอ ดังรูปที่ 5.9 เมื่อเลือก Clear Files จะลบข้อมูลที่นำมาแสดง



รูปที่ 5.9 หน้าจอแสดงข้อมูลที่บันทึกไว้เป็น Text File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อต้องการดูเกี่ยวกับ Data Logger Server และผู้จัดทำ จากหน้าจอหลักดังรูปที่ 5.4 เลือก Help จะปรากฏหน้าจอ ดังรูปที่ 5.10 เมื่อเลือก BACK จะกลับไปยังหน้าจอหลักเหมือนเดิม



รูปที่ 5.10 เกี่ยวกับ Data Logger Server

9. เมื่อคลิกเลือกที่ <http://www.kmitl.ac.th/electronics> จะเข้าสู่เว็บไซต์ของภาควิชาอิเล็กทรอนิกส์ ดังรูปที่ 5.11



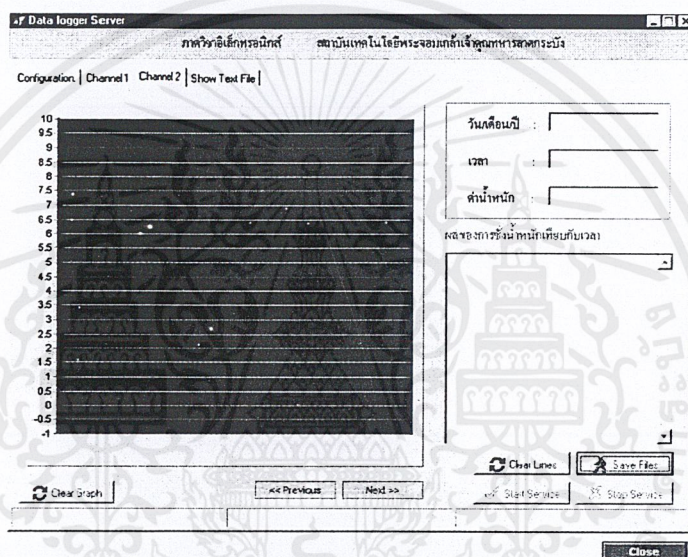
รูปที่ 5.11 เว็บไซต์ภาควิชาอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 การใช้งานโปรแกรมส่วนการชั่งน้ำหนัก

ถ้าต้องการวัดการชั่งน้ำหนัก จากหน้าจอแสดงในรูปที่ 5.4

1. ทำการเลือก New Work เพื่อทำการวัดค่าของข้อมูลที่จะพลอตกราฟและเก็บบันทึกข้อมูลใหม่ จะปรากฏหน้าจอจดังรูปที่ 5.5
2. เลือก Select Channel 2 โดยสามารถตั้งค่าเวลาในการวัดได้ตั้งแต่ 1 ถึง 60 วินาที
3. เมื่อตั้งค่าเรียบร้อยแล้วเลือกปุ่ม OK จะปรากฏหน้าจอจดังรูปที่ 5.12



รูปที่ 5.12 หน้าจอในส่วนของ Channel 2

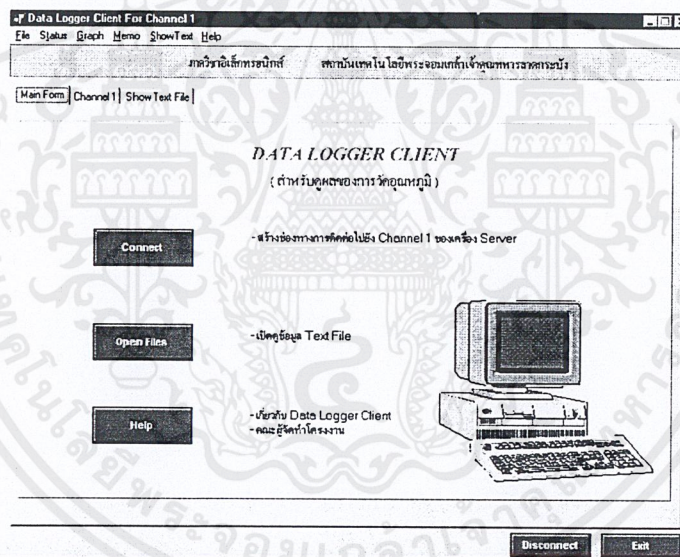
4. จากนั้นเลือก Start Service โปรแกรมจะทำการชั่งน้ำหนักแสดงเป็นตัวเลขและกราฟออกมาเมื่อต้องการหยุดชั่งน้ำหนักให้เลือก Stop Service
5. เมื่อต้องการลบกราฟให้เลือก Clear Graph เมื่อต้องการลบตัวเลขแสดงผลให้เลือก Clear Lines และเมื่อต้องการหยุดการวัดข้อมูลให้เลือก Stop Service สามารถเลือก Previous เพื่อเลื่อนดูกราฟถอยหลังและเลือก Next เพื่อเลื่อนดูกราฟไปข้างหน้าได้ จากนั้นเลือก Save Files จะปรากฏหน้าจอจดังรูปที่ 5.7 ใส่ชื่อไฟล์ข้อมูลนั้นแล้วเลือก Save ข้อมูลจะถูกบันทึก
6. เมื่อต้องการดูข้อมูลที่เก็บไว้ให้กลับมาที่หน้าจอหลักจดังรูปที่ 5.4 เลือก Open Work จะปรากฏข้อมูลบนหน้าจอจดังรูปที่ 5.8 เลือกข้อมูลที่จะเปิด และเลือก Open จะปรากฏหน้าจอแสดงผลข้อมูลเป็น Text File ดังรูปที่ 5.9

5.2 การใช้เครื่องคอมพิวเตอร์ Client ที่ใช้ในการดูข้อมูลจากเครื่องคอมพิวเตอร์ Server

ติดตั้งโปรแกรม Data logger Client ยังเครื่องคอมพิวเตอร์ที่ต้องการให้เป็น Client โดยโปรแกรม Data logger Client นี้จะประกอบด้วย 2 ส่วนคือ โปรแกรม Data logger Client ส่วนดูข้อมูลการวัดอุณหภูมิและโปรแกรม Data logger Client ส่วนดูข้อมูลการชั่งน้ำหนัก เครื่องคอมพิวเตอร์นี้ต้องมีการติดตั้งระบบเครือข่ายภายใน โดยใช้โปรโตคอล TCP/IP ไว้เรียบร้อยแล้ว

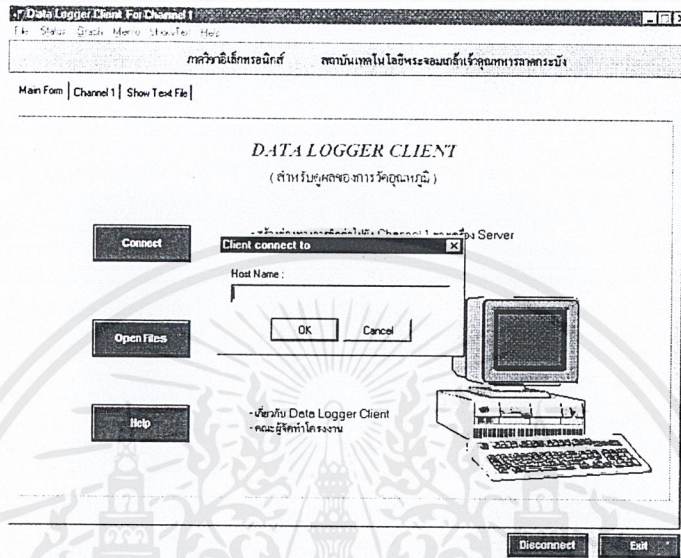
5.2.1 การใช้โปรแกรม Client เพื่อดูข้อมูลการวัดอุณหภูมิ

1. เปิดโปรแกรม Data logger Client สำหรับดูข้อมูลการวัดอุณหภูมิจะปรากฏหน้าจอ ดังรูปที่ 5.13



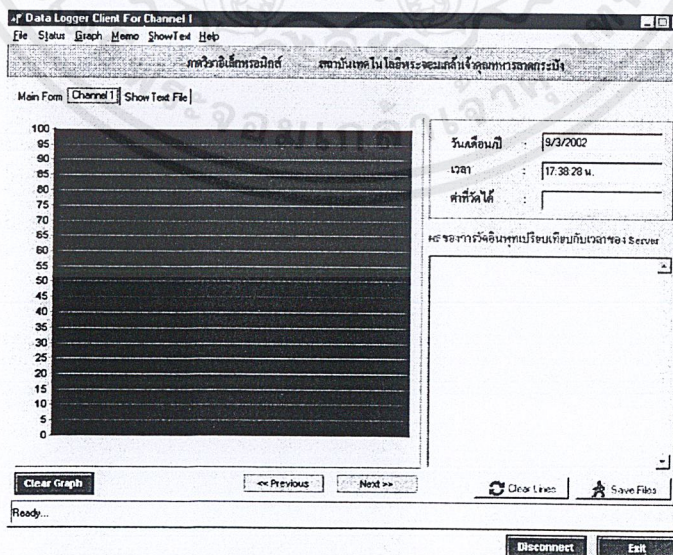
รูปที่ 5.13 หน้าจอหลักของ Data logger Client สำหรับดูข้อมูลการวัดอุณหภูมิ

2. เลือก Connect จะปรากฏหน้าจอ ดังรูปที่ 5.14 และให้ใส่หมายเลข IP Address ของเครื่อง Server



รูปที่ 5.14 การสร้างช่องทางการติดต่อกับเครื่อง Server

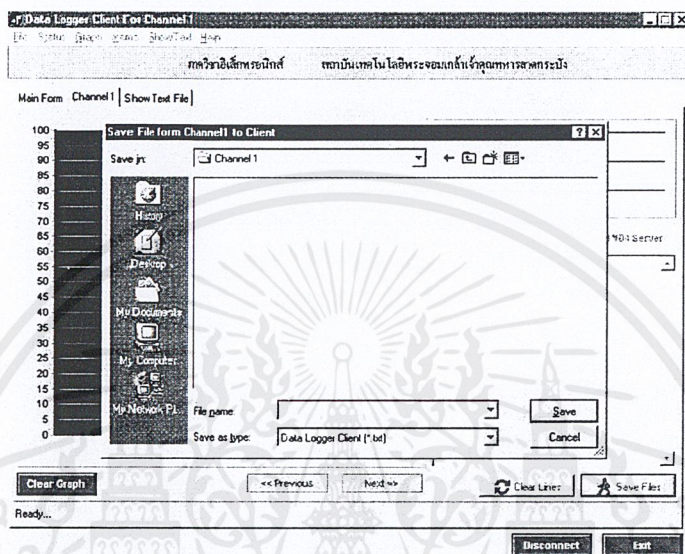
3. เมื่อเลือก OK จะปรากฏหน้าจอที่ใช้ในการแสดงข้อมูลการวัดอุณหภูมิจากเครื่อง Server ดังรูปที่ 5.15



รูปที่ 5.15 หน้าจอที่ใช้ในการแสดงข้อมูลการวัดอุณหภูมิจากเครื่อง Server

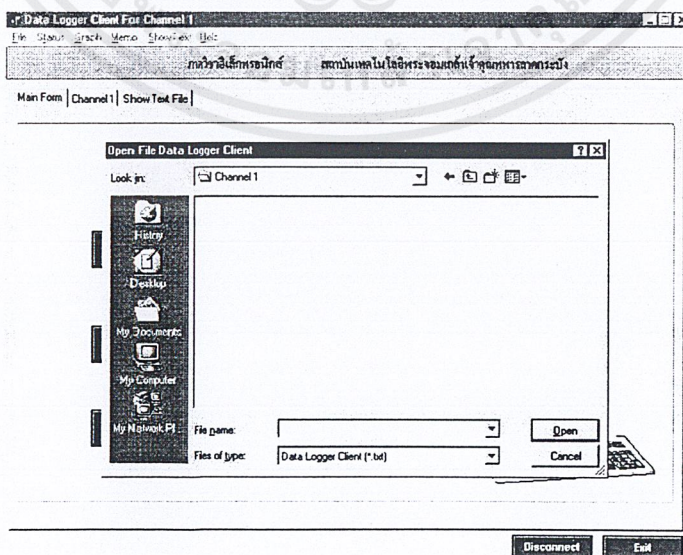
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อต้องการลบกราฟให้เลือก Clear Graph เมื่อต้องการลบตัวเลขแสดงผลให้เลือก Clear Lines เมื่อต้องการเลื่อนดูกราฟให้เลือก Previous หรือ Next และเมื่อต้องการเก็บข้อมูลให้เลือก Save Files จะปรากฏหน้าจอ ดังรูปที่ 5.16 ใส่ชื่อไฟล์ข้อมูลนั้นแล้วเลือก Save ข้อมูลจะถูกบันทึก



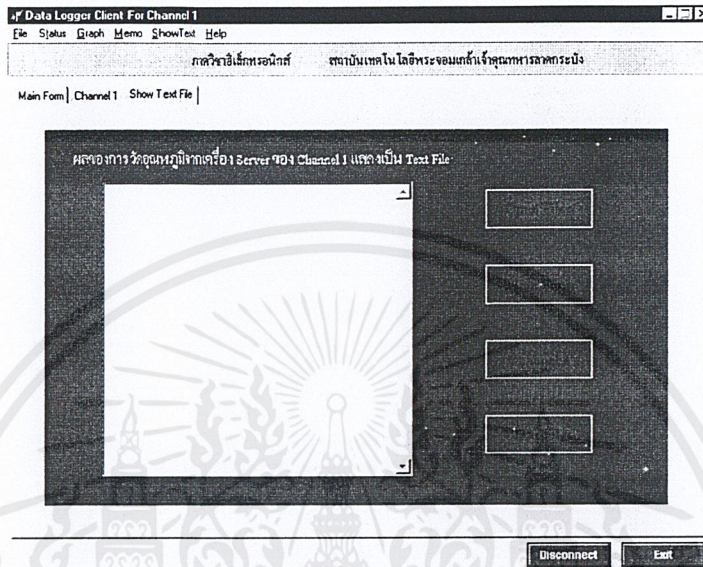
รูปที่ 5.16 หน้าจอใช้ในการเก็บข้อมูลจากเครื่อง Server

5. เมื่อต้องการดูข้อมูลการวัดอุณหภูมิจากเครื่อง Server ให้กลับไปหน้าจอหลักดังรูปที่ 5.13 เลือก Open Files จะปรากฏหน้าจอ ดังรูปที่ 5.17



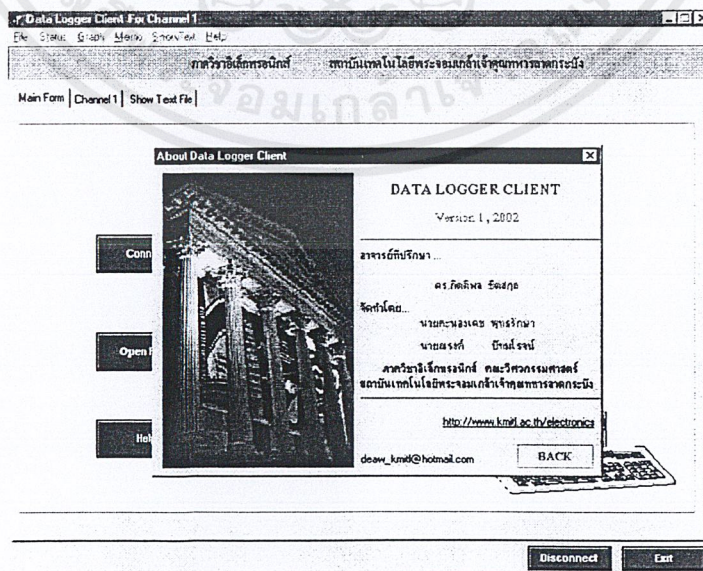
เอกสารนี้เป็นเอกสารที่รูปที่ 5.17 หน้าจอที่ใช้ในการดูข้อมูลจากเครื่อง Server ที่เก็บไว้ ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เลือกชื่อไฟล์ข้อมูลที่ต้องการดูข้อมูลการวัดอุณหภูมิจากเครื่อง Server และเลือก Open จะปรากฏหน้าจอ ดังรูปที่ 5.18



รูปที่ 5.18 หน้าจอแสดงข้อมูลที่บันทึกแสดงผลเป็น Text File

7. เมื่อต้องการดูเกี่ยวกับ Data Logger Client และผู้จัดทำ จากหน้าจอหลักดังรูปที่ 5.13 เลือก Help จะปรากฏหน้าจอ ดังรูปที่ 5.19 เมื่อเลือก BACK จะกลับไปยังหน้าจอหลักเหมือนเดิม



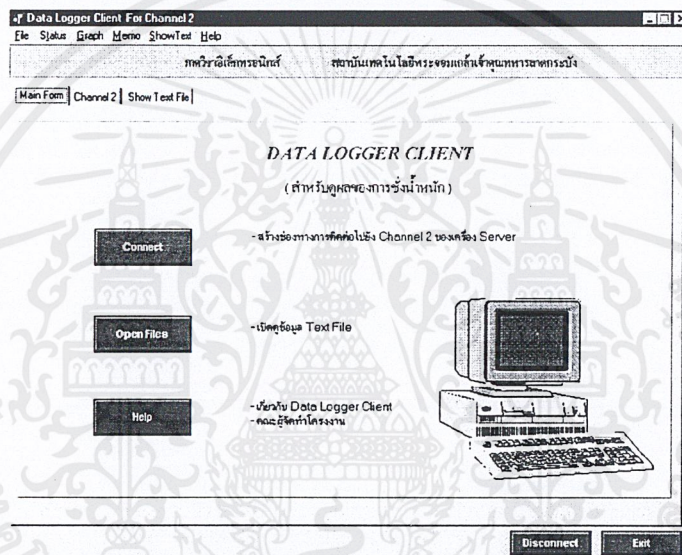
รูปที่ 5.19 เกี่ยวกับ Data Logger Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ประโยชน์ในการศึกษาเท่านั้น มิได้อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อกดคลิกเลือกที่ <http://www.kmitl.ac.th/electronics> จะเข้าสู่เว็บไซต์ของภาควิชาอิเล็กทรอนิกส์ ดังรูปที่ 5.11

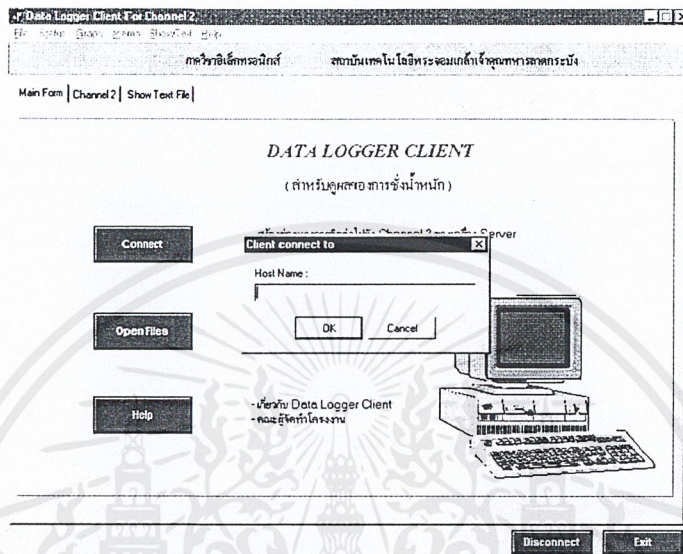
5.2.2 การใช้โปรแกรม Client เพื่อดูข้อมูลการชั่งน้ำหนัก

1. เปิดโปรแกรม Data logger Client สำหรับดูข้อมูลการชั่งน้ำหนักจะปรากฏหน้าจอ ดังรูปที่ 5.20



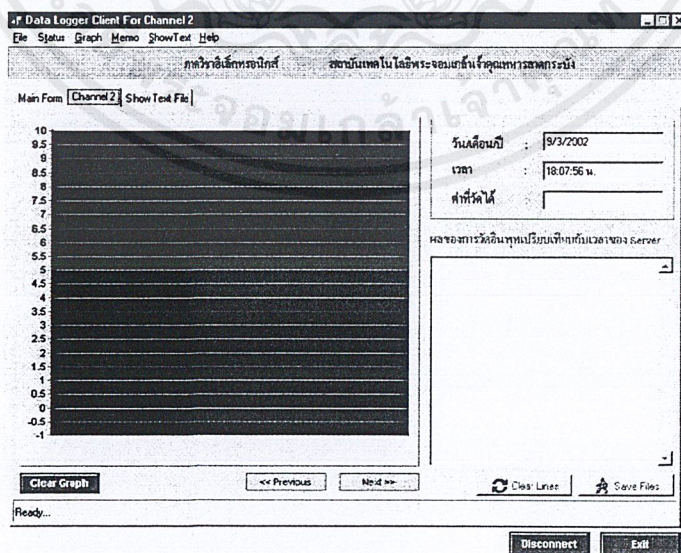
รูปที่ 5.20 หน้าจอหลักของ Data logger Client ส่วนดูข้อมูลการชั่งน้ำหนัก

2. เลือก Connect จะปรากฏหน้าจอดังรูปที่ 5.21 และให้ใส่หมายเลข IP Address ของเครื่อง Server



รูปที่ 5.21 การติดต่อกับเครื่อง Server

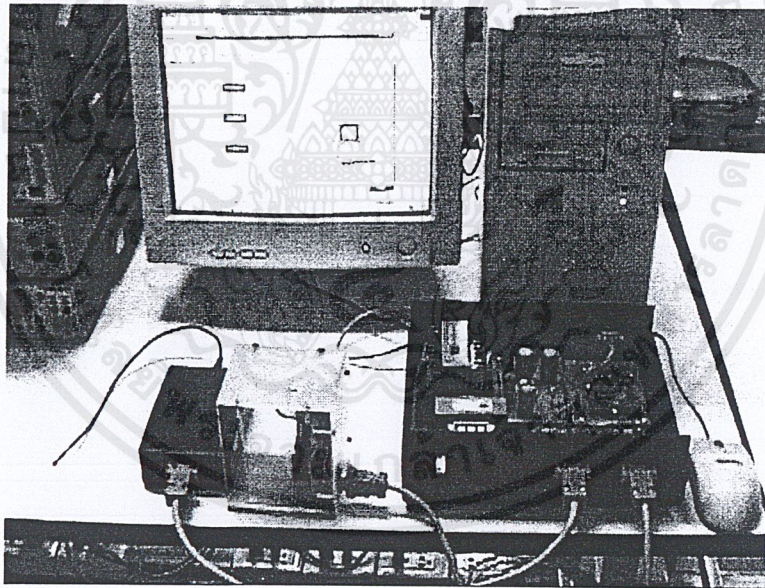
3. เมื่อเลือก OK จะปรากฏหน้าจอที่ใช้ในการแสดงข้อมูลการชั่งน้ำหนักจากเครื่อง Server ดังรูปที่ 5.22



รูปที่ 5.22 หน้าจอที่ใช้ในการแสดงข้อมูลการชั่งน้ำหนักจากเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อต้องการลบกราฟให้เลือก Clear Graph เมื่อต้องการลบตัวเลขแสดงผลให้เลือก Clear Lines และเมื่อต้องการบันทึกข้อมูลให้เลือก Save Files จะปรากฏหน้าจอ ดังรูปที่ 5.16 ใส่ชื่อไฟล์ ข้อมูลนั้นแล้วเลือก Save ข้อมูลจะถูกบันทึก
5. เมื่อต้องการดูข้อมูลการชั่งน้ำหนักจากเครื่อง Server ให้กลับไปหน้าจอหลักดังรูปที่ 5.20 เลือก Open Files จะปรากฏหน้าจอ ดังรูปที่ 5.17
6. เลือกชื่อไฟล์ข้อมูลที่ต้องการดูข้อมูลการชั่งน้ำหนักจากเครื่อง Server และเลือก Open จะปรากฏหน้าจอ ดังรูปที่ 5.18
7. เมื่อต้องการดูเกี่ยวกับ Data Logger Client และผู้จัดทำ จากหน้าจอหลักดังรูปที่ 5.20 เลือก Help จะปรากฏหน้าจอ ดังรูปที่ 5.19 เมื่อเลือก BACK จะกลับไปยังหน้าจอหลักเหมือนเดิม
8. เมื่อคลิกเลือกที่ <http://www.kmitl.ac.th/electronics> จะเข้าสู่เว็บไซต์ของภาควิชา อิเล็กทรอนิกส์ ดังรูปที่ 5.11



รูปที่ 5.23 ระบบเก็บบันทึกข้อมูลอัตโนมัติ

บทที่ 6

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

6.1 บทสรุป

ได้กล่าวถึงการสร้างระบบเก็บข้อมูลอัตโนมัติในรูปแบบดิจิทัล ระบบประกอบด้วยส่วนฮาร์ดแวร์ที่ทำหน้าที่ปรับสัญญาณอินพุตขนาดออกพร้อมกับเปลี่ยนให้เป็นสัญญาณดิจิทัล การสุ่มจะถูกควบคุมด้วยไมโครคอมพิวเตอร์ในอัตรา 1 sample/sec ถึง 1 sample/minute และข้อมูลจะถูกส่งผ่านไปเก็บที่หน่วยความจำ ข้อมูลที่เก็บได้สามารถเก็บในฐานข้อมูลในคอมพิวเตอร์ควบคุมที่ทำหน้าที่เป็น server ของเครือข่าย ทำให้คอมพิวเตอร์ client ในเครือข่ายสามารถร้องขอข้อมูลที่บันทึกนี้ไว้ได้ ในการทดสอบการทำงานของระบบได้ทดลองสร้างสัญญาณทดสอบเป็นแรงดันที่ได้จากเทอร์โมมิเตอร์อิเล็กทรอนิกส์ที่สามารถวัดอุณหภูมิในช่วง 0 ถึง 100 องศาเซลเซียส และเป็นเชิงเส้น นอกจากนี้ได้สร้างเครื่องชั่งน้ำหนักขนาดเล็กทำจาก moving coil ของมิเตอร์สามารถ ชั่งน้ำหนักได้ในช่วง 0 ถึง 7.8 มิลลิกรัม ปัญหาในการสร้างระบบสรุปได้ต่อไปนี้

6.1.1 เครื่องวัดอุณหภูมิ

จากผลของการทดลองสามารถวัดค่าอุณหภูมิได้จริงทั้งเครื่อง Server และเครื่อง Client โดยเครื่อง Client สามารถรับค่าอุณหภูมิได้ตรงตามเครื่อง Server แต่ค่าที่วัดได้ยังมีค่าความผิดพลาดอยู่ โดยเฉพาะที่ค่าอุณหภูมิสูงๆ ส่วนค่าที่วัดได้อยู่ในช่วง 0 ถึง 100 องศาเซลเซียส การแสดงผลสามารถแสดงผลได้ทั้งกราฟเปรียบเทียบกับเวลาและตัวเลข การทำงานของโครงการนี้จะต้องใช้คอมพิวเตอร์ควบคุมการทำงานทั้งหมด ในส่วนของการเก็บบันทึกข้อมูลมีการบันทึกข้อมูลเป็น Text Files และการแสดงผลที่หน้าจอคอมพิวเตอร์จะแสดงได้เพียงครั้งละ 1 ช่องสัญญาณ

6.1.2 เครื่องชั่งน้ำหนักขนาดเล็ก

จากผลการทดลองสามารถชั่งน้ำหนักได้จริงทั้งเครื่อง Server และเครื่อง Client โดยเครื่อง Client สามารถรับค่าน้ำหนักได้ตรงตามเครื่อง Server ส่วนน้ำหนักที่ชั่งได้นั้นอยู่ในช่วงประมาณ 0 ถึง 7.8 มิลลิกรัม แต่ค่าที่ชั่งได้ยังมีค่าผิดพลาดในช่วงน้ำหนักที่เบาๆ อยู่มาก และส่วนการทำงานอื่นๆ จะเหมือนกับเครื่องวัดอุณหภูมิ

6.2 ปัญหา

6.2.1 ส่วนของโปรแกรม

1. ปัญหาจากการเขียนโปรแกรมในการอ่านค่าอินพุตจากตัววัดอุณหภูมิเข้ามาคำนวณ แล้วแสดงผลเป็นกราฟและตัวเลขซึ่งใช้วงจรแปลงอนาลอกเป็นดิจิตอลขนาด 12 บิตแต่การรับส่งระหว่างไมโครคอนโทรลเลอร์และไมโครคอมพิวเตอร์ส่งได้ครั้งละ 8 บิต ทำให้ใช้เวลาในการศึกษาโปรแกรมนาน
2. ความเร็วในการติดต่อสื่อสารข้อมูลระหว่างเครื่อง Server และเครื่อง Client จะขึ้นอยู่กับเครือข่ายของระบบ Network และเครื่องคอมพิวเตอร์ที่ใช้
3. การติดต่อสื่อสารข้อมูลผ่านระบบเครือข่ายภายใน จะช้ากว่าการที่ติดต่อสื่อสารข้อมูลกันโดยใช้เครื่องคอมพิวเตอร์เครื่องเดียว
4. ในขณะที่ทำการเปิดเครื่อง Server ใวนั้น ถ้าหากใช้โปรแกรมในส่วนอื่นของเครื่อง Server หรือให้เครื่องทำงานในส่วนอื่น จะทำให้ข้อมูลที่อ่านได้ไม่ใช่ค่าที่ถูกต้องเพราะเครื่อง Server จะเสียเวลาในการอ่านค่าเข้ามาทำให้ข้อมูลผิดพลาด
5. สามารถวัดได้ครั้งละ 1 ช่องสัญญาณเท่านั้น
6. ไม่สามารถตั้งเวลาในการหยุดวัดสัญญาณอินพุตได้

6.2.2 เครื่องวัดอุณหภูมิ

1. ค่าอุณหภูมิที่วัดได้ยังมีค่าความผิดพลาดอยู่ โดยเฉพาะที่อุณหภูมิค่าสูงๆ เป็นเพราะความไม่เป็นเชิงเส้นของตัววัดอุณหภูมิ LM335
2. วงจรภาควัดอุณหภูมิให้แรงดันเอาต์พุตไม่คงที่ ต้องทำให้ใช้เวลาในการหาสาเหตุและแก้ไขนาน

6.2.3 เครื่องชั่งน้ำหนักขนาดเล็ก

1. ค่าน้ำหนักที่ชั่งได้ยังมีค่าความผิดพลาดอยู่สูงมาก โดยเฉพาะในช่วงน้ำหนักที่เบา ๆ
2. วงจรที่ใช้ในการปรับกระแสที่จ่ายให้กับขดลวดซึ่งใช้ทรานซิสเตอร์มาต่อวงจรแบบ Current Miller กระแสไหลผ่านทรานซิสเตอร์ 2 ตัวไม่เท่ากันจริง จึงทำให้การจ่ายกระแสคงที่ให้กับขดลวดนั้นควบคุมได้ยาก
3. ในการชั่งน้ำหนักเมื่อมีลมพัดผ่านมายังเครื่องชั่งน้ำหนักจะส่งผล ให้ค่าที่ชั่งได้ผิดพลาด

เพราะจะทำให้เข็มมิเตอร์เกิดการแกว่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขดลวดที่ใช้ทำเครื่องชั่งน้ำหนักต้องการกระแสมากในการควบคุมเพราะไม่ใช่ขดลวดที่ทำหน้าที่นี้โดยเฉพาะ

5. ค่าแรงดันเอาต์พุตที่ได้ยังไม่คงที่พอ ส่งผลให้ค่าเอาต์พุตที่วัดมาได้ไม่คงที่ตามไปด้วย

6.3 แนวทางแก้ไขและพัฒนา

6.3.1 ส่วนของโปรแกรม

1. ใช้ระบบเครือข่ายภายใน และเครื่องคอมพิวเตอร์ที่มีความเร็วสูง
2. ไม่ควรใช้เครื่องคอมพิวเตอร์ที่ใช้เป็นเครื่อง Server และเครื่อง Client ทำกิจกรรมอื่น
3. ต้องพัฒนาโปรแกรมให้ทำงานให้เร็วขึ้นและสามารถใช้งานได้ 2 ช่องสัญญาณพร้อมกัน
4. ต้องพัฒนาโปรแกรมให้สามารถตั้งค่าการหยุดวัดสัญญาณอินพุตได้

6.3.2 เครื่องวัดอุณหภูมิ

1. ทำค่าแรงดันเอาต์พุตให้คงที่โดย ใช้วงจร Buffer ใส่ก่อนอินพุตและเอาต์พุต
2. ใช้วงจร Low Pass Filter ช่วยในการกรองสัญญาณรบกวนออกไป

6.3.3 เครื่องชั่งน้ำหนักขนาดเล็ก

1. ทรานซิสเตอร์ 2 ตัวที่ใช้ต่อเป็นวงจร Current Mirror ควรจะอยู่ใน Package เดียวกันเพื่อให้ทำงานได้เท่ากัน

2. ถอดใส่เครื่องชั่งน้ำหนักควรปิดสนิทเพื่อกันลมเข้ามาทำให้การชั่งผิดพลาด
3. ในการชั่งน้ำหนักควรชั่งในบริเวณที่อากาศนิ่งๆ ไม่มีลมพัดผ่าน
4. ขดลวดที่ใช้ทำเครื่องชั่งน้ำหนักควรเป็นขดลวดที่กินกระแสไม่มากนักเพื่อที่จะได้ควบคุมง่ายขึ้น

5. ใช้ LED อินฟราเรดกับโฟโตทรานซิสเตอร์ที่มีพื้นที่หน้าตัดในการรับส่งมากกว่านี้
6. ทำค่าแรงดันเอาต์พุตให้คงที่โดย ใช้วงจร Buffer ใส่ก่อนอินพุตและเอาต์พุต
7. ใช้วงจร Low Pass Filter ช่วยในการกรองสัญญาณรบกวนออกไป



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LTC1286/LTC1298

Micropower Sampling 12-Bit A/D Converters In SO-8 Packages

FEATURES

- 12-Bit Resolution
- 8-Pin SOIC Plastic Package
- Low Cost
- Low Supply Current: 250 μ A Typ.
- Auto Shutdown to 1nA Typ.
- Guaranteed $\pm 3/4$ LSB Max DNL
- Single Supply 5V to 9V Operation
- On-Chip Sample-and-Hold
- 60 μ s Conversion Time
- Sampling Rates:
 - 12.5 ksps (LTC1286)
 - 11.1 ksps (LTC1298)
- I/O Compatible with SPI, Microwire, etc.
- Differential Inputs (LTC1286)
- 2-Channel MUX (LTC1298)
- 3V Versions Available: LTC1285/LTC1288

APPLICATIONS

- Battery-Operated Systems
- Remote Data Acquisition
- Battery Monitoring
- Handheld Terminal Interface
- Temperature Measurement
- Isolated Data Acquisition

DESCRIPTION

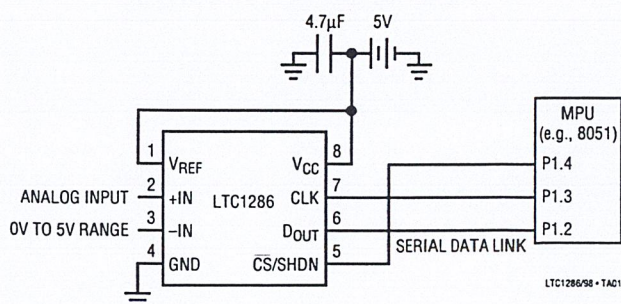
The LTC1286/LTC1298 are micropower, 12-bit, successive approximation sampling A/D converters. They typically draw only 250 μ A of supply current when converting and automatically power down to a typical supply current of 1nA whenever they are not performing conversions. They are packaged in 8-pin SO packages and operate on 5V to 9V supplies. These 12-bit, switched-capacitor, successive approximation ADCs include sample-and-holds. The LTC1286 has a single differential analog input. The LTC1298 offers a software selectable 2-channel MUX.

On-chip serial ports allow efficient data transfer to a wide range of microprocessors and microcontrollers over three wires. This, coupled with micropower consumption, makes remote location possible and facilitates transmitting data through isolation barriers.

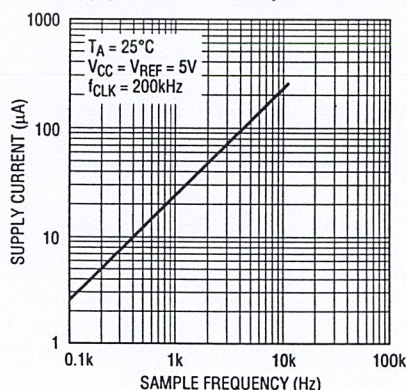
These circuits can be used in ratiometric applications or with an external reference. The high impedance analog inputs and the ability to operate with reduced spans (to 1.5V full scale) allow direct connection to sensors and transducers in many applications, eliminating the need for gain stages.

TYPICAL APPLICATIONS

25 μ W, SO-8 Package, 12-Bit ADC
Samples at 200Hz and Runs Off a 5V Supply



Supply Current vs Sample Rate

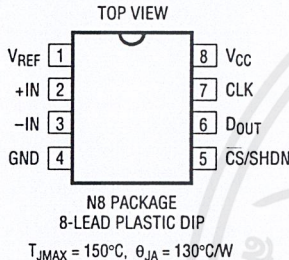
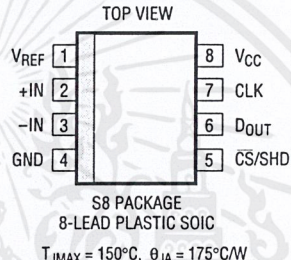
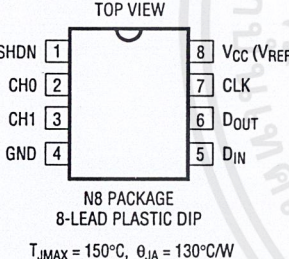
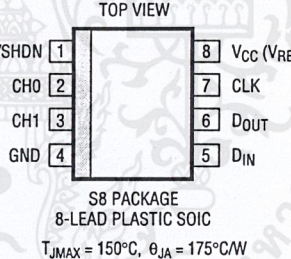


LTC1286/LTC1298

ABSOLUTE MAXIMUM RATINGS (Notes 1 and 2)

Supply Voltage (V_{CC}) to GND	12V	Power Dissipation	500mW
Voltage		Operating Temperature Range	
Analog and Reference	-0.3V to $V_{CC} + 0.3V$	LTC1286C/LTC1298C	0°C to 70°C
Digital Inputs	-0.3V to 12V	LTC1286I/LTC1298I	-40°C to 85°C
Digital Output	-0.3V to $V_{CC} + 0.3V$	Storage Temperature Range	-65°C to 150°C
		Lead Temperature (Soldering, 10 sec.)	300°C

PACKAGE/ORDER INFORMATION

 <p>N8 PACKAGE 8-LEAD PLASTIC DIP $T_{JMAX} = 150^{\circ}C, \theta_{JA} = 130^{\circ}C/W$</p>	ORDER PART NUMBER	 <p>S8 PACKAGE 8-LEAD PLASTIC SOIC $T_{JMAX} = 150^{\circ}C, \theta_{JA} = 175^{\circ}C/W$</p>	ORDER PART NUMBER
	LTC1286CN8 LTC1286IN8		LTC1286CS8 LTC1286IS8
			PART MARKING
			1286C 1286I
 <p>N8 PACKAGE 8-LEAD PLASTIC DIP $T_{JMAX} = 150^{\circ}C, \theta_{JA} = 130^{\circ}C/W$</p>	ORDER PART NUMBER	 <p>S8 PACKAGE 8-LEAD PLASTIC SOIC $T_{JMAX} = 150^{\circ}C, \theta_{JA} = 175^{\circ}C/W$</p>	ORDER PART NUMBER
	LTC1298CN8 LTC1298IN8		LTC1298CS8 LTC1298IS8
			PART MARKING
			1298C 1298I

Consult factory for military grade parts.

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
V_{CC}	Supply Voltage (Note 3)	LTC1286 LTC1298	4.5	9.0	5.5	V
f_{CLK}	Clock Frequency	$V_{CC} = 5V$	(Note 4)	200		kHz
t_{CYC}	Total Cycle Time	LTC1286, $f_{CLK} = 200kHz$ LTC1298, $f_{CLK} = 200kHz$	80			μs
t_{hDI}	Hold Time, D_{IN} After $CLK\uparrow$	$V_{CC} = 5V$	150			ns
t_{suCS}	Setup Time $\overline{CS}\downarrow$ Before First $CLK\uparrow$ (See Operating Sequence)	LTC1286, $V_{CC} = 5V$ LTC1298, $V_{CC} = 5V$	2			μs
t_{suDI}	Setup Time, D_{IN} Stable Before $CLK\uparrow$	$V_{CC} = 5V$	400			ns
t_{WHCLK}	CLK High Time	$V_{CC} = 5V$	2			μs
t_{WLCLK}	CLK Low Time	$V_{CC} = 5V$	2			μs
t_{WHCS}	\overline{CS} High Time Between Data Transfer Cycles	$V_{CC} = 5V$	2			μs
t_{WLCS}	\overline{CS} Low Time During Data Transfer	LTC1286, $f_{CLK} = 200kHz$ LTC1298, $f_{CLK} = 200kHz$	75			μs
			85			μs

2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CONVERTER AND MULTIPLEXER CHARACTERISTICS (Note 5)

PARAMETER	CONDITIONS	LTC1286			LTC1298			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Resolution (No Missing Codes)		●	12		12		Bits	
Integral Linearity Error	(Note 6)	●	±3/4	±2	±3/4	±2	LSB	
Differential Linearity Error		●	±1/4	±3/4	±1/4	±3/4	LSB	
Offset Error		●	3/4	±3	3/4	±3	LSB	
Gain Error		●	±2	±8	±2	±8	LSB	
Analog Input Range	(Note 7 and 8)	●	-0.05V to $V_{CC} + 0.05V$				V	
REF Input Range (LTC1286)	$4.5 \leq V_{CC} \leq 5.5V$		1.5V to $V_{CC} + 0.05V$				V	
(Notes 7, 8, and 9)	$5.5V < V_{CC} \leq 9V$		1.5V to 5.55V				V	
Analog Input Leakage Current (Note 10)		●	±1		±1		μA	

DIGITAL AND DC ELECTRICAL CHARACTERISTICS (Note 5)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
V_{IH}	High Level Input Voltage	$V_{CC} = 5.25V$	●	2		V
V_{IL}	Low Level Input Voltage	$V_{CC} = 4.75V$	●		0.8	V
I_{IH}	High Level Input Current	$V_{IN} = V_{CC}$	●		2.5	μA
I_{IL}	Low Level Input Current	$V_{IN} = 0V$	●		-2.5	μA
V_{OH}	High Level Output Voltage	$V_{CC} = 4.75V, I_O = 10\mu A$	●	4.0	4.64	V
		$V_{CC} = 4.75V, I_O = 360\mu A$	●	2.4	4.62	V
V_{OL}	Low Level Output Voltage	$V_{CC} = 4.75V, I_O = 1.6mA$	●		0.4	V
I_{OZ}	Hi-Z Output Leakage	$\overline{CS} = High$	●		±3	μA
I_{SOURCE}	Output Source Current	$V_{OUT} = 0V$		-25		mA
I_{SINK}	Output Sink Current	$V_{OUT} = V_{CC}$		45		mA
R_{REF}	Reference Input Resistance (LTC1286)	$\overline{CS} = V_{CC}$ $\overline{CS} = GND$		5000 55		MΩ kΩ
I_{REF}	Reference Current (LTC1286)	$\overline{CS} = V_{CC}$ $t_{CYC} \geq 640\mu s, f_{CLK} \leq 25kHz$ $t_{CYC} = 80\mu s, f_{CLK} = 200kHz$	●	0.001 90 90	2.5 140 140	μA μA μA
I_{CC}	Supply Current	$\overline{CS} = V_{CC}$	●	0.001	±3.0	μA
		LTC1286, $t_{CYC} \geq 640\mu s, f_{CLK} \leq 25kHz$	●	200	400	μA
		LTC1286, $t_{CYC} = 80\mu s, f_{CLK} = 200kHz$	●	250	500	μA
		LTC1298, $t_{CYC} \geq 720\mu s, f_{CLK} \leq 25kHz$	●	290	490	μA
		LTC1298, $t_{CYC} = 90\mu s, f_{CLK} = 200kHz$	●	340	640	μA

DYNAMIC ACCURACY $f_{SMPL} = 12.5kHz$ (LTC1286), $f_{SMPL} = 11.1kHz$ (LTC1298) (Note 5)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
S/(N + D)	Signal-to-Noise Plus Distortion Ratio	1kHz/7kHz Input Signal		71/68		dB
THD	Total Harmonic Distortion (Up to 5th Harmonic)	1kHz/7kHz Input Signal		-84/-80		dB
SFDR	Spurious-Free Dynamic Range	1kHz/7kHz Input Signal		90/86		dB
	Peak Harmonic or Spurious Noise	1kHz/7kHz Input Signal		-90/-86		dB

LTC1286/LTC1298

AC CHARACTERISTICS (Note 5)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t_{SMPL}	Analog Input Sample Time	See Operating Sequence		1.5		CLK Cycles
$f_{SMPL (MAX)}$	Maximum Sampling Frequency	LTC1286 LTC1298	● ●	12.5 11.1		kHz kHz
t_{CONV}	Conversion Time	See Operating Sequence		12		CLK Cycles
t_{dDO}	Delay Time, CLK↓ to D _{OUT} Data Valid	See Test Circuits	●	250	600	ns
t_{dis}	Delay Time, \overline{CS} ↑ to D _{OUT} Hi-Z	See Test Circuits	●	135	300	ns
t_{en}	Delay Time, CLK↓ to D _{OUT} Enable	See Test Circuits	●	75	200	ns
t_{hDO}	Time Output Data Remains Valid After CLK↓	$C_{LOAD} = 100pF$		230		ns
t_f	D _{OUT} Fall Time	See Test Circuits	●	20	75	ns
t_r	D _{OUT} Rise Time	See Test Circuits	●	20	75	ns
C_{IN}	Input Capacitance	Analog Inputs, On Channel Analog Inputs, Off Channel Digital Input		20 5 5		pF pF pF

The ● denotes specifications which apply over the full operating temperature range.

Note 1: Absolute maximum ratings are those values beyond which the life of a device may be impaired.

Note 2: All voltage values are with respect to GND.

Note 3: These devices are specified at 5V. For 3V specified devices, see LTC1285 and LTC1288.

Note 4: Increased leakage currents at elevated temperatures cause the S/H to droop, therefore it is recommended that $f_{CLK} \geq 120kHz$ at 85°C, $f_{CLK} \geq 75kHz$ at 70° and $f_{CLK} \geq 1kHz$ at 25°C.

Note 5: $V_{CC} = 5V$, $V_{REF} = 5V$ and $CLK = 200kHz$ unless otherwise specified.

Note 6: Linearity error is specified between the actual end points of the A/D transfer curve.

Note 7: Two on-chip diodes are tied to each reference and analog input which will conduct for reference or analog input voltages one diode drop below GND or one diode drop above V_{CC} . This spec allows 50mV forward bias of either diode for $4.5V \leq V_{CC} \leq 5.5V$. This means that as long as the reference or analog input does not exceed the supply voltage by more than 50mV the output code will be correct. To achieve an absolute 0V to 5V input voltage range will therefore require a minimum supply voltage of 4.950V over initial tolerance, temperature variations and loading. For $5.5V < V_{CC} \leq 9V$, reference and analog input range cannot exceed 5.55V. If reference and analog input range are greater than 5.55V, the output code will not be guaranteed to be correct.

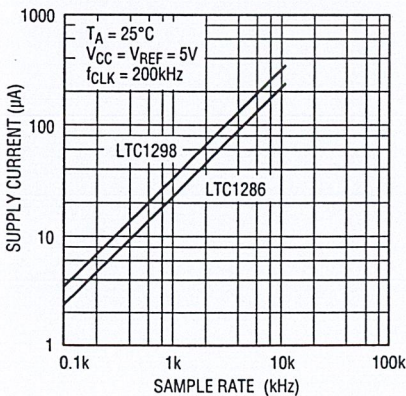
Note 8: The supply voltage range for the LTC1286 is from 4.5V to 9V, but the supply voltage range for the LTC1298 is only from 4.5V to 5.5V.

Note 9: Recommended operating conditions

Note 10: Channel leakage current is measured after the channel selection.

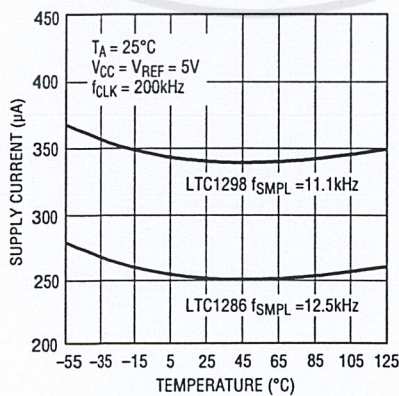
TYPICAL PERFORMANCE CHARACTERISTICS

Supply Current vs Sample Rate



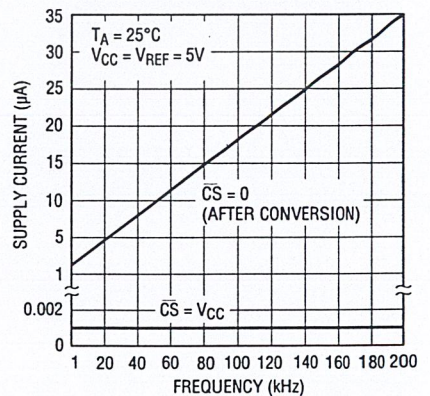
LT1286/98 G03

Supply Current vs Temperature



LT1286/98 G04

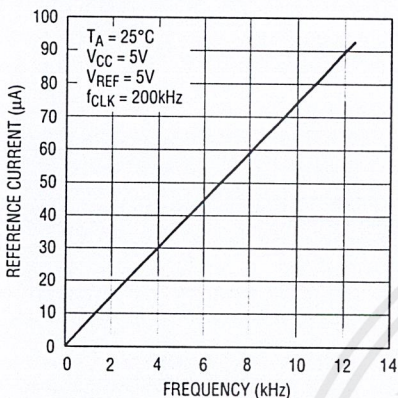
Shutdown Supply Current vs Clock Rate with CS High and CS Low



LT1286/98 G01

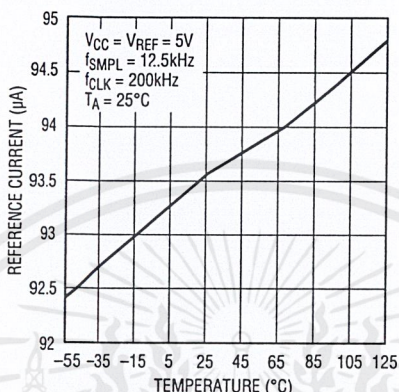
TYPICAL PERFORMANCE CHARACTERISTICS

Reference Current vs Sample Rate (LTC1286)



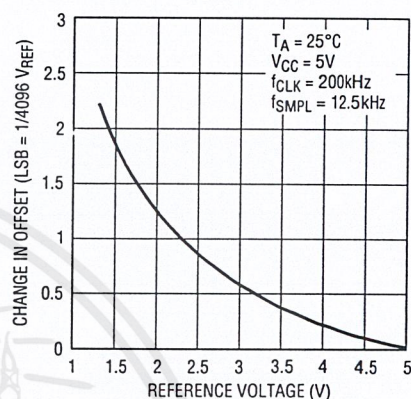
LT1286/98 G06

Reference Current vs Temperature



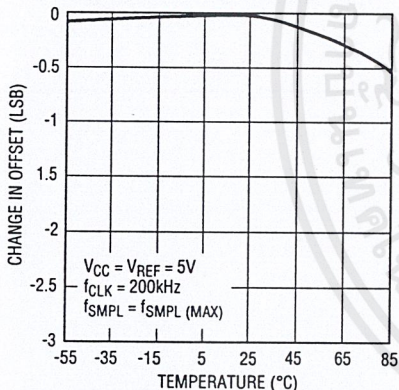
LT1286/98 G07

Change in Offset vs Reference Voltage



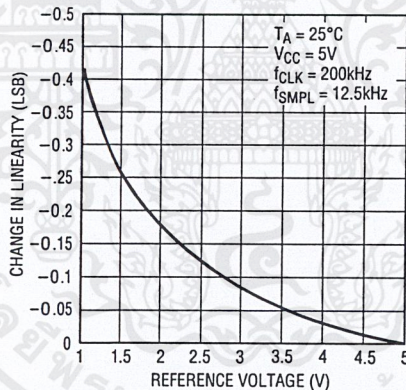
LT1286/98 G08

Change in Offset vs Temperature



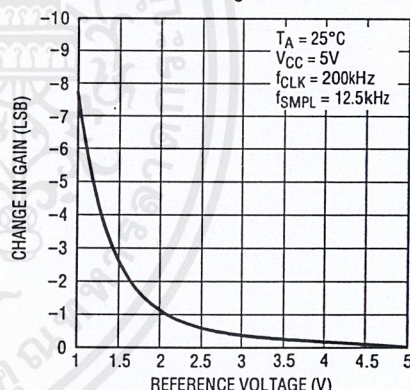
LT1286/98 G09

Change in Linearity vs Reference Voltage



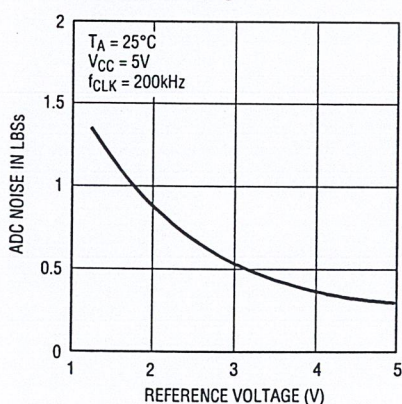
LT1286/98 G10

Change in Gain vs Reference Voltage



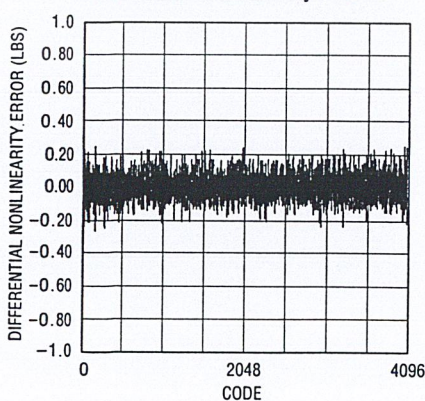
LT1286/98 G11

Peak-to-Peak ADC Noise vs Reference Voltage

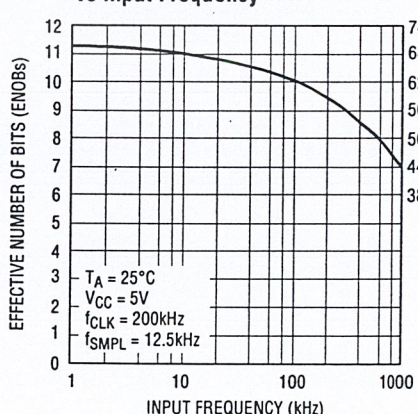


LT1286/98 G15

Differential Nonlinearity vs Code



Effective Bits and S/(N + D) vs Input Frequency

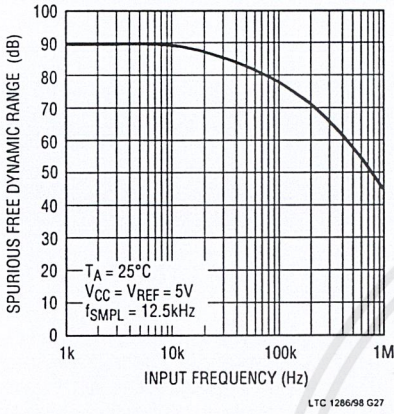


LTC 1286/98 G20

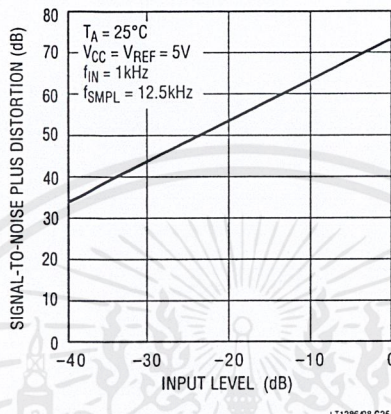
LTC1286/LTC1298

TYPICAL PERFORMANCE CHARACTERISTICS

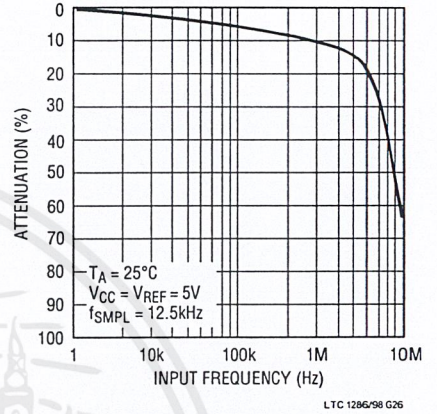
Spurious Free Dynamic Range vs Frequency



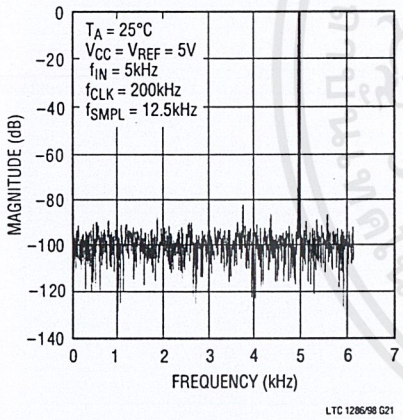
S/(N+D) vs Input Level



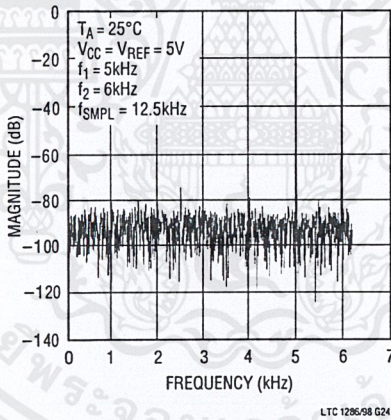
Attenuation vs Input Frequency



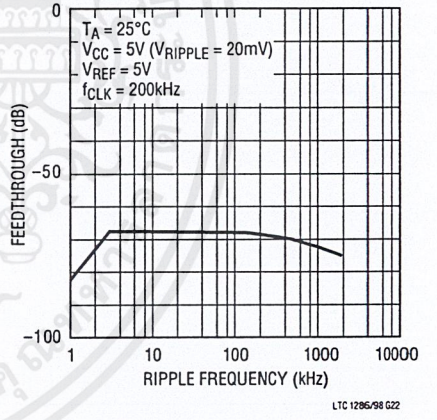
4096 Point FFT Plot



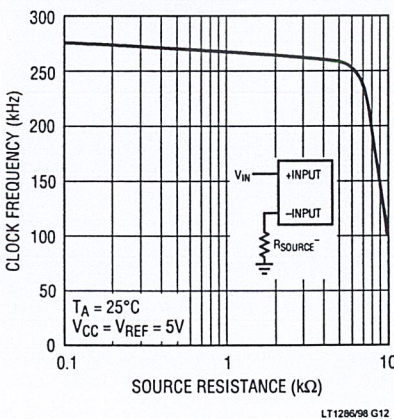
Intermodulation Distortion



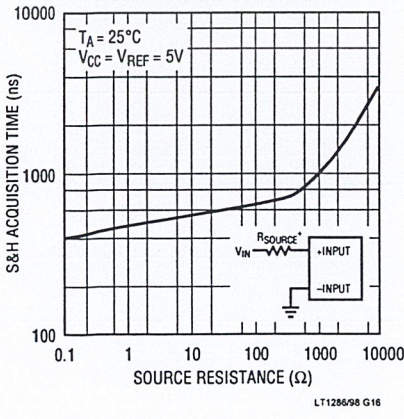
Power Supply Feedthrough vs Ripple Frequency



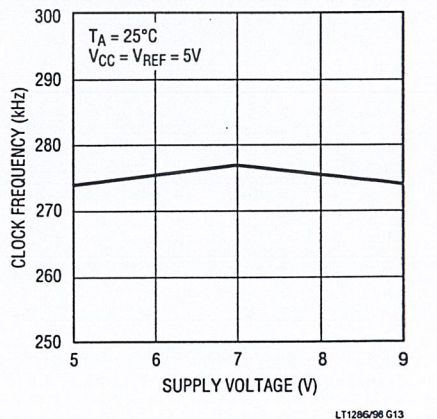
Maximum Clock Frequency vs Source Resistance



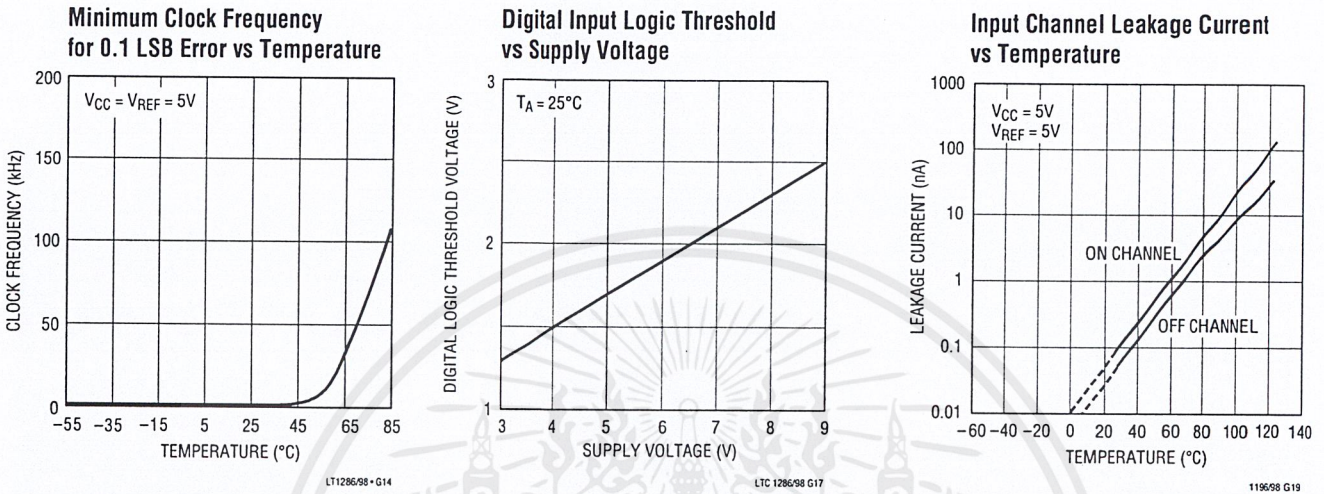
Sample and Hold Acquisition Time vs Source Resistance



Maximum Clock Frequency vs Supply Voltage



TYPICAL PERFORMANCE CHARACTERISTICS



PIN FUNCTIONS

LTC1286

V_{REF} (Pin 1): Reference Input. The reference input defines the span of the A/D converter.

IN^+ (Pin 2): Positive Analog Input.

IN^- (Pin 3): Negative Analog Input.

GND (Pin 4): Analog Ground. GND should be tied directly to an analog ground plane.

$\overline{CS}/SHDN$ (Pin 5): Chip Select Input. A logic low on this input enables the LTC1286. A logic high on this input disables and powers down the LTC1286.

D_{OUT} (Pin 6): Digital Data Output. The A/D conversion result is shifted out of this output.

CLK (Pin 7): Shift Clock. This clock synchronizes the serial data transfer and determines conversion speed.

V_{CC} (Pin 8): Power Supply Voltage. This pin provides power to the A/D converter. It must be kept free of noise and ripple by bypassing directly to the analog ground plane.

LTC1298

$\overline{CS}/SHDN$ (Pin 1): Chip Select Input. A logic low on this input enables the LTC1298. A logic high on this input disables and powers down the LTC1298.

CHO (Pin 2): Analog Input.

CH1 (Pin 3): Analog Input.

GND (Pin 4): Analog Ground. GND should be tied directly to an analog ground plane.

D_{IN} (Pin 5): Digital Data Input. The multiplexer address is shifted into this input.

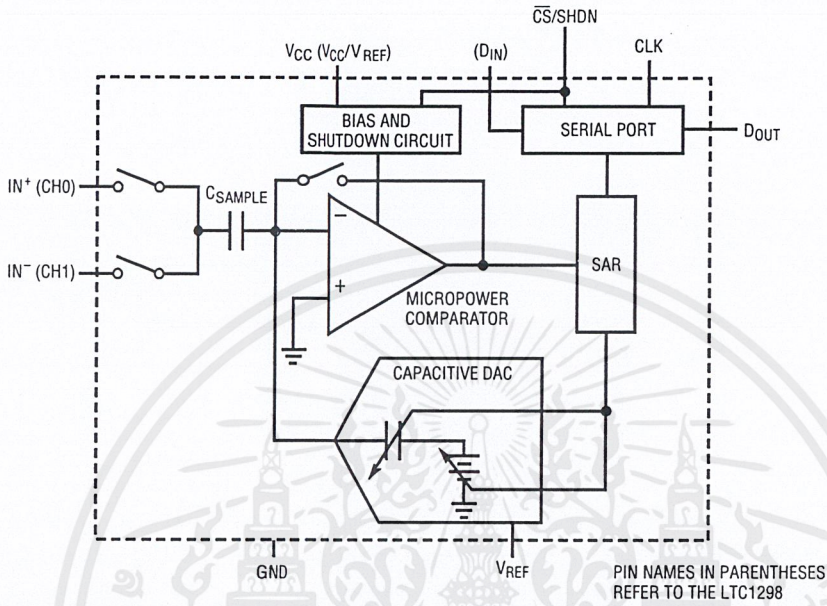
D_{OUT} (Pin 6): Digital Data Output. The A/D conversion result is shifted out of this output.

CLK (Pin 7): Shift Clock. This clock synchronizes the serial data transfer and determines conversion speed.

V_{CC}/V_{REF} (Pin 8): Power Supply and Reference Voltage. This pin provides power and defines the span of the A/D converter. It must be kept free of noise and ripple by bypassing directly to the analog ground plane.

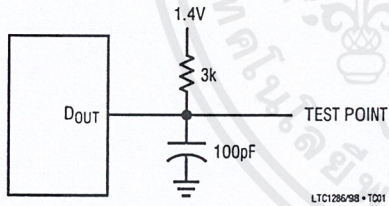
LTC1286/LTC1298

BLOCK DIAGRAM

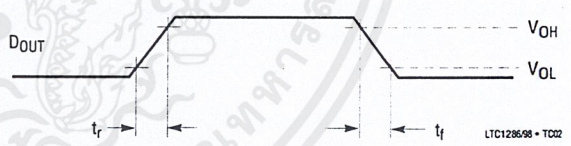


TEST CIRCUITS

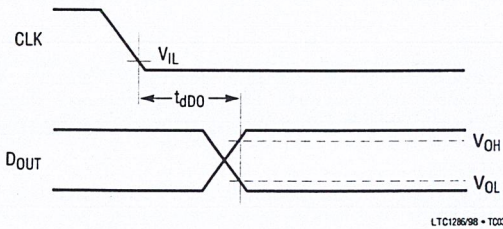
Load Circuit for t_{dDO} , t_r and t_f



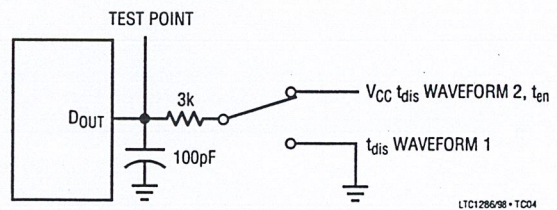
Voltage Waveforms for D_{OUT} Rise and Fall Times, t_r , t_f



Voltage Waveforms for D_{OUT} Delay Times, t_{dDO}

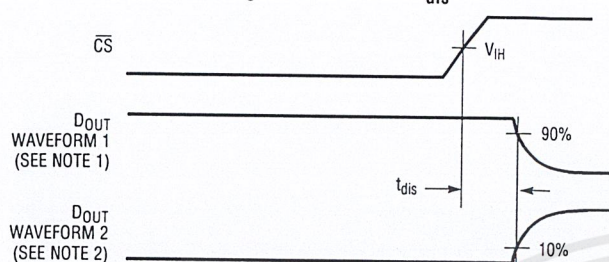


Load Circuit for t_{dis} and t_{en}



TEST CIRCUITS

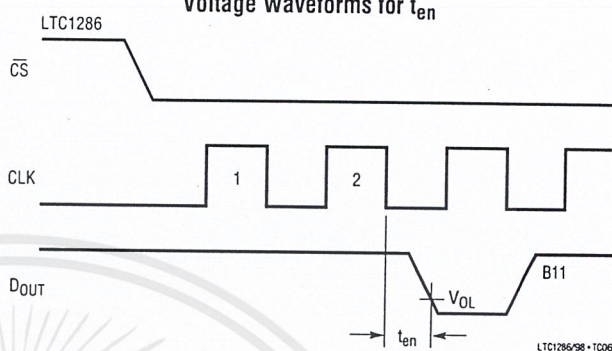
Voltage Waveforms for t_{dis}



NOTE 1: WAVEFORM 1 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS HIGH UNLESS DISABLED BY THE OUTPUT CONTROL.
 NOTE 2: WAVEFORM 2 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS LOW UNLESS DISABLED BY THE OUTPUT CONTROL.

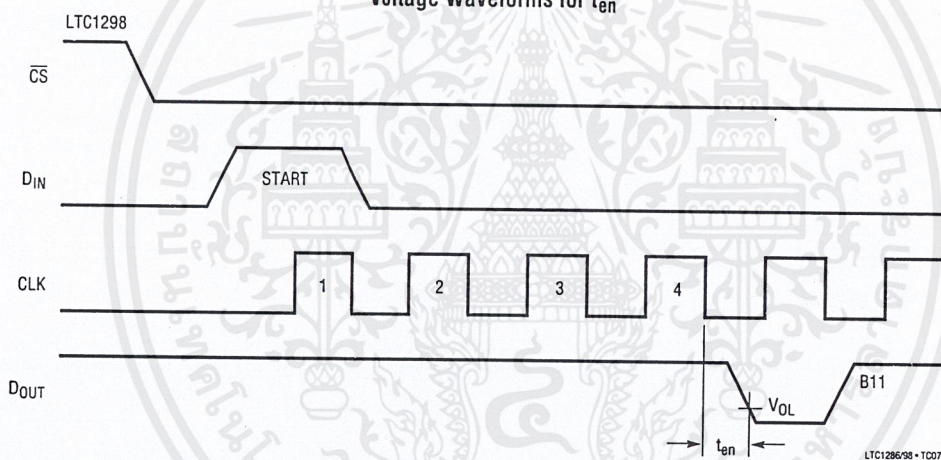
LTC1286-98 • TC05

Voltage Waveforms for t_{en}



LTC1286-98 • TC06

Voltage Waveforms for t_{en}



LTC1286-98 • TC07

APPLICATION INFORMATION

OVERVIEW

The LTC1286 and LTC1298 are micropower, 12-bit, successive approximation sampling A/D converters. The LTC1286 typically draws 250 μ A of supply current when sampling at 12.5kHz while the LTC1298 nominally consumes 350 μ A of supply current when sampling at 11.1 kHz. The extra 100 μ A of supply current on the LTC1298 comes from the reference input which is intentionally tied to the supply. Supply current drops linearly as the sample rate is reduced (see Supply Current vs Sample Rate). The ADCs automatically power down when not performing conversions, drawing only leakage current. They are packaged in 8-pin SO and DIP packages. The LTC1286 operates on a single supply from 4.5V to 9V,

while the LTC1298 operates from a 4.5V to 5.5V supply.

Both the LTC1286 and the LTC1298 contain a 12-bit, switched-capacitor ADC, a sample-and-hold, and a serial port (see Block Diagram). Although they share the same basic design, the LTC1286 and LTC1298 differ in some respects. The LTC1286 has a differential input and has an external reference input pin. It can measure signals floating on a DC common-mode voltage and can operate with reduced spans to 1V. Reducing the spans allows it to achieve 244 μ V resolution. The LTC1298 has a two-channel input multiplexer and can convert either channel with respect to ground or the difference between the two. The reference input is tied to the supply pin.

LTC1286/LTC1298

APPLICATION INFORMATION

SERIAL INTERFACE

The 2-channel LTC1298 communicates with microprocessors and other external circuitry via a synchronous, half duplex, 4-wire serial interface. The single channel LTC1286 uses a 3-wire interface (see Operating Sequence in Figures 1 and 2).

Data Transfer

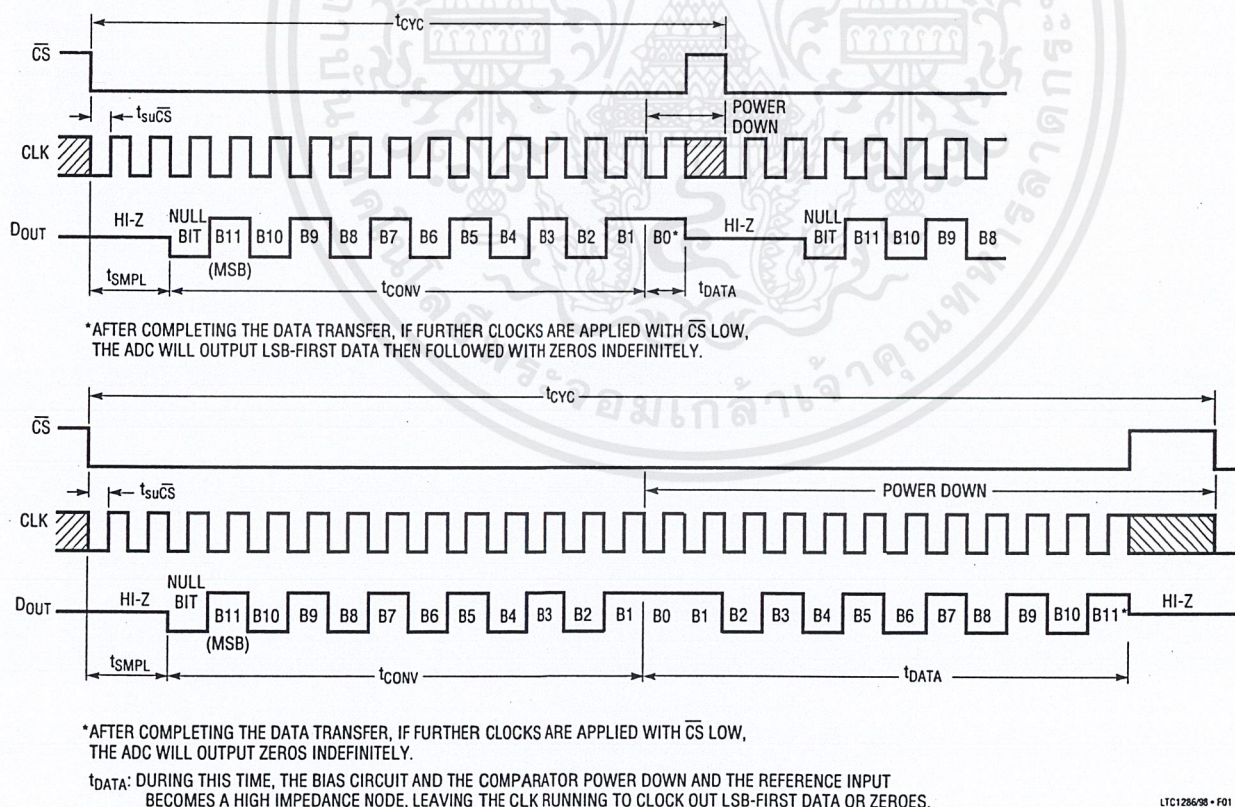
The CLK synchronizes the data transfer with each bit being transmitted on the falling CLK edge and captured on the rising CLK edge in both transmitting and receiving systems.

The LTC1286 does not require a configuration input word and has no D_{IN} pin. A falling \overline{CS} initiates data transfer as shown in the LTC1286 operating sequence. After \overline{CS} falls the second CLK pulse enables D_{OUT} . After one null bit the

A/D conversion result is output on the D_{OUT} line. Bringing \overline{CS} high resets the LTC1286 for the next data exchange.

The LTC1298 first receives input data and then transmits back the A/D conversion result (half duplex). Because of the half duplex operation, D_{IN} and D_{OUT} may be tied together allowing transmission over just 3 wires: \overline{CS} , CLK and DATA (D_{IN}/D_{OUT}).

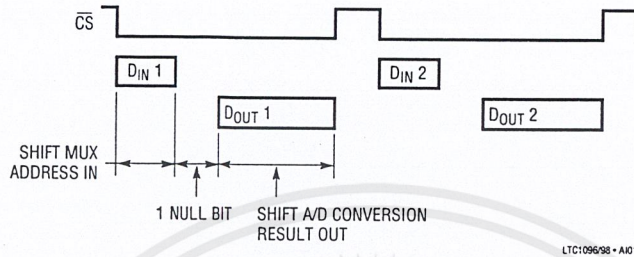
Data transfer is initiated by a falling chip select (\overline{CS}) signal. After \overline{CS} falls the LTC1298 looks for a start bit. After the start bit is received, the 3-bit input word is shifted into the D_{IN} input which configures the LTC1298 and starts the conversion. After one null bit, the result of the conversion is output on the D_{OUT} line. At the end of the data exchange \overline{CS} should be brought high. This resets the LTC1298 in preparation for the next data exchange.



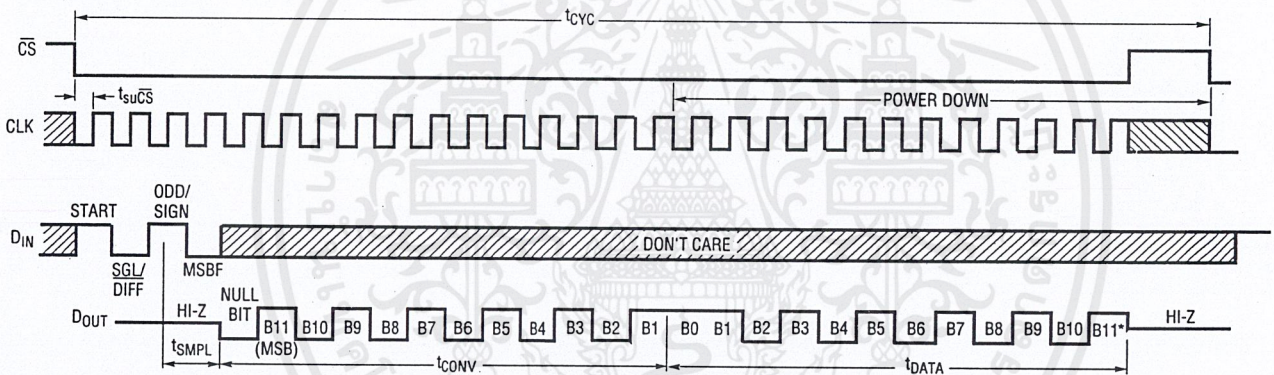
LTC1286-08 • F01

Figure 1. LTC1286 Operating Sequence

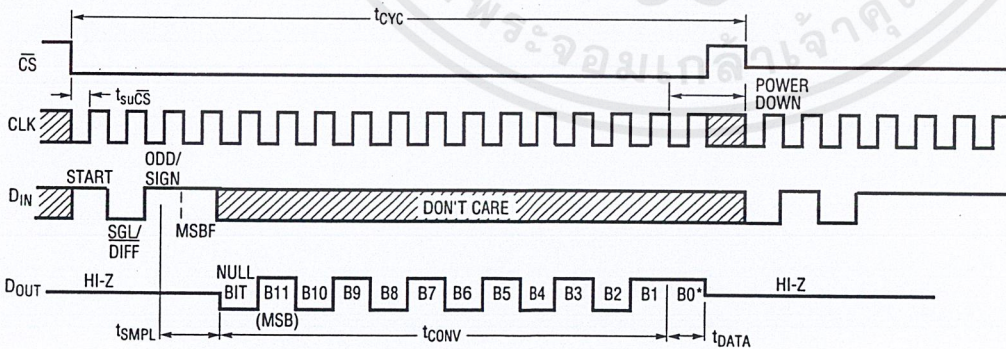
APPLICATION INFORMATION



MSB-First Data (MSBF = 0)



MSB-First Data (MSBF = 1)



*AFTER COMPLETING THE DATA TRANSFER, IF FURTHER CLOCKS ARE APPLIED WITH \overline{CS} LOW, THE ADC WILL OUTPUT ZEROS INDEFINITELY.
 t_{DATA} : DURING THIS TIME, THE BIAS CIRCUIT AND THE COMPARATOR POWER DOWN AND THE REFERENCE INPUT BECOMES A HIGH IMPEDANCE NODE, LEAVING THE CLK RUNNING TO CLOCK OUT LSB-FIRST DATA OR ZEROS.

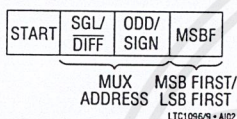
Figure 2. LTC1298 Operating Sequence Example: Differential Inputs (CH⁺, CH⁻)

LTC1286/LTC1298

APPLICATION INFORMATION

Input Data Word

The LTC1286 requires no D_{IN} word. It is permanently configured to have a single differential input. The conversion result appears on the D_{OUT} line. The data format is MSB first followed by the LSB sequence. This provides easy interface to MSB or LSB first serial ports. For MSB first data the \overline{CS} signal can be taken high after B0 (see Figure 1). The LTC1298 clocks data into the D_{IN} input on the rising edge of the clock. The input data words are defined as follows:



Start Bit

The first "logical one" clocked into the D_{IN} input after \overline{CS} goes low is the start bit. The start bit initiates the data transfer. The LTC1298 will ignore all leading zeros which precede this logical one. After the start bit is received, the remaining bits of the input word will be clocked in. Further inputs on the D_{IN} pin are then ignored until the next \overline{CS} cycle.

Multiplexer (MUX) Address

The bits of the input word following the START bit assign the MUX configuration for the requested conversion. For a given channel selection, the converter will measure the voltage between the two channels indicated by the + and - signs in the selected row of the following tables. In single-ended mode, all input channels are measured with respect to GND.

LTC1298 Channel Selection

	MUX ADDRESS		CHANNEL #		GND
	SGL/DIFF	ODD/SIGN	0	1	
SINGLE-ENDED MUX MODE	1	0	+	-	
	1	1		+	-
DIFFERENTIAL MUX MODE	0	0	+	-	
	0	1	-	+	

LTC1096/9-A103

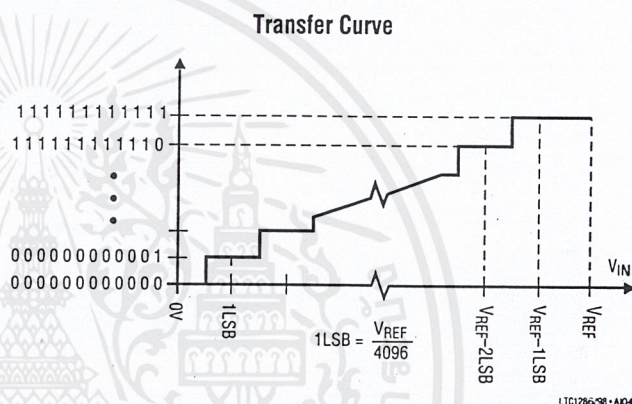
MSB First/LSB First (MSBF)

The output data of the LTC1298 is programmed for MSB first or LSB first sequence using the MSBF bit. When the MSBF bit is a logical one, data will appear on the D_{OUT} line in MSB first format. Logical zeros will be filled in indefinitely following the last data bit. When the

MSBF bit is a logical zero, LSB first data will follow the normal MSB first data on the D_{OUT} line. (see Operating Sequence)

Transfer Curve

The LTC1286/LTC1298 are permanently configured for unipolar only. The input span and code assignment for this conversion type are shown in the following figures.



Output Code

OUTPUT CODE	INPUT VOLTAGE	INPUT VOLTAGE ($V_{REF} = 5.000V$)
111111111111	$V_{REF} - 1LSB$	4.99878V
111111111110	$V_{REF} - 2LSB$	4.99756V
⋮	⋮	⋮
000000000001	1LSB	0.00122V
000000000000	0V	0V

LTC1286/98-A105

Operation with D_{IN} and D_{OUT} Tied Together

The LTC1298 can be operated with D_{IN} and D_{OUT} tied together. This eliminates one of the lines required to communicate to the microprocessor (MPU). Data is transmitted in both directions on a single wire. The processor pin connected to this data line should be configurable as either an input or an output. The LTC1298 will take control of the data line and drive it low on the 4th falling CLK edge after the start bit is received (see Figure 3). Therefore the processor port line must be switched to an input before this happens to avoid a conflict.

In the Typical Applications section, there is an example of interfacing the LTC1298 with D_{IN} and D_{OUT} tied together to the Intel 8051 MPU.

APPLICATION INFORMATION

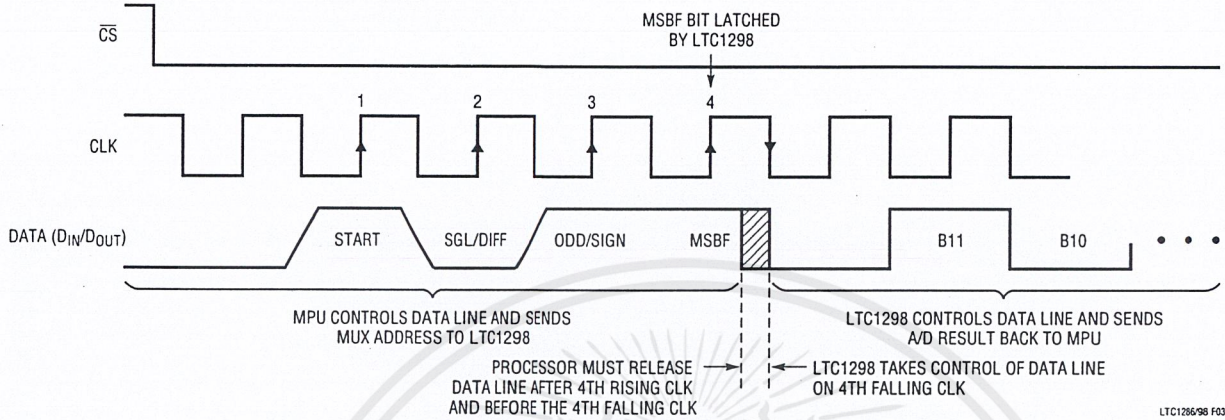


Figure 3. LTC1298 Operation with D_{IN} and D_{OUT} Tied Together

ACHIEVING MICROWATT PERFORMANCE

With typical operating currents of $250\mu A$ and automatic shutdown between conversions, the LTC1286/LTC1298 achieves extremely low power consumption over a wide range of sample rates (see Figure 4). The auto-shutdown allows the supply curve to drop with reduced sample rate. Several things must be taken into account to achieve such a low power consumption.

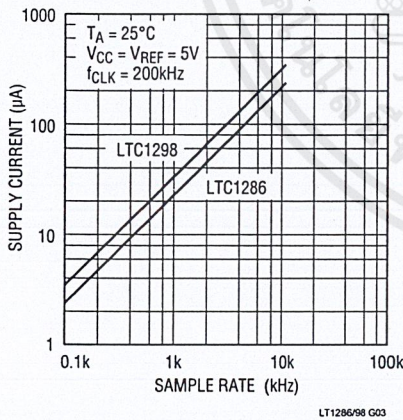


Figure 4. Automatic Power Shutdown Between Conversions Allows Power Consumption to Drop with Sample Rate.

Shutdown

The LTC1286/LTC1298 are equipped with automatic shutdown features. They draw power when the \overline{CS} pin is low and shut down completely when that pin is high. The bias circuit and comparator powers down and the reference

input becomes high impedance at the end of each conversion leaving the CLK running to clock out the LSB first data or zeroes (see Figures 1 and 2). If the \overline{CS} is not running rail-to-rail, the input logic buffer will draw current. This current may be large compared to the typical supply current. To obtain the lowest supply current, bring the \overline{CS} pin to ground when it is low and to supply voltage when it is high.

When the \overline{CS} pin is high (= supply voltage), the converter is in shutdown mode and draws only leakage current. The status of the D_{IN} and CLK input have no effect on supply current during this time. There is no need to stop D_{IN} and CLK with $\overline{CS} = \text{high}$; they can continue to run without drawing current.

Minimize \overline{CS} Low Time

In systems that have significant time between conversions, lowest power drain will occur with the minimum \overline{CS} low time. Bringing \overline{CS} low, transferring data as quickly as possible, and then bringing it back high will result in the lowest current drain. This minimizes the amount of time the device draws power. After a conversion the ADC automatically shuts down even if \overline{CS} is held low (see Figures 1 and 2). If the clock is left running to clock out LSB-data or zero, the logic will draw a small current. Figure 5 shows that the typical supply current with $\overline{CS} = \text{ground}$ varies from $1\mu A$ at 1kHz to $35\mu A$ at 200kHz. When $\overline{CS} = V_{CC}$, the logic is gated off and no supply current is drawn regardless of the clock frequency.

LTC1286/LTC1298

APPLICATION INFORMATION

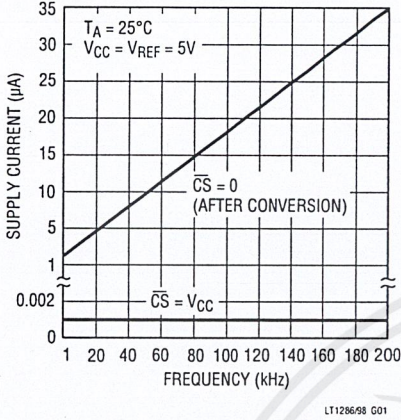


Figure 5. Shutdown current with \overline{CS} high is 1nA typically, regardless of the clock. Shutdown current with \overline{CS} = ground varies from 1µA at 1kHz to 35µA at 200kHz.

D_{OUT} Loading

Capacitive loading on the digital output can increase power consumption. A 100pF capacitor on the D_{OUT} pin can add more than 50µA to the supply current at a 200kHz clock frequency. An extra 50µA or so of current goes into charging and discharging the load capacitor. The same goes for digital lines driven at a high frequency by any logic. The $C \times V \times f$ currents must be evaluated and the troublesome ones minimized.

OPERATING ON OTHER THAN 5V SUPPLIES (LTC1286)

The LTC1286 operates from 4.5V to 9V supplies and the LTC1298 operates from a 5V supply. To operate the LTC1286 on other than 5V supplies a few things must be kept in mind.

Input Logic Levels

The input logic levels of \overline{CS} , CLK and D_{IN} are made to meet TTL on a 5V supply. When the supply voltage varies, the input logic levels also change. For the LTC1286 to sample and convert correctly, the digital inputs have to be in the proper logical low and high levels relative to the operating supply voltage (see typical curve of Digital Input Logic Threshold vs Supply Voltage). If achieving micropower consumption is desirable, the digital inputs must go rail-to-rail between supply voltage and ground (see ACHIEVING MICROPOWER PERFORMANCE section).

Clock Frequency

The maximum recommended clock frequency is 200kHz for the LTC1286/LTC1298 running off a 5V supply. With the supply voltage changing, the maximum clock frequency for the devices also changes (see the typical curve of Maximum Clock Rate vs Supply Voltage). If the maximum clock frequency is used, care must be taken to ensure that the device converts correctly.

Mixed Supplies

It is possible to have a microprocessor running off a 5V supply and communicate with the LTC1286 operating on a 9V supply. The requirement to achieve this is that the outputs of \overline{CS} and CLK from the MPU have to be able to trip the equivalent inputs of the LTC1286 and the output of D_{OUT} from the LTC1286 must be able to toggle the equivalent input of the MPU (see typical curve of Digital Input Logic Threshold vs Supply Voltage). With the LTC1286 operating on a 9V supply, the output of D_{OUT} may go between 0V and 9V. The 9V output may damage the MPU running off a 5V supply. The way to get around this possibility is to have a resistor divider on D_{OUT} (Figure 6) and connect the center point to the MPU input. It should be noted that to get full shutdown, the \overline{CS} input of the LTC1286 must be driven to the V_{CC} voltage to keep the \overline{CS} input buffer from drawing current. An alternative is to leave \overline{CS} low after a conversion, clock data until D_{OUT} outputs zeros, and then stop the clock low.

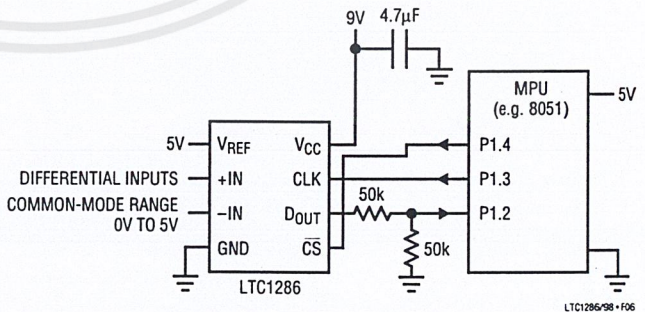


Figure 6. Interfacing a 9V Powered LTC1286 to a 5V System



APPLICATION INFORMATION

BOARD LAYOUT CONSIDERATIONS

Grounding and Bypassing

The LTC1286/LTC1298 are easy to use if some care is taken. They should be used with an analog ground plane and single point grounding techniques. The GND pin should be tied directly to the ground plane.

The V_{CC} pin should be bypassed to the ground plane with a $10\mu\text{F}$ tantalum capacitor with leads as short as possible. If the power supply is clean, the LTC1286/LTC1298 can also operate with smaller $1\mu\text{F}$ or less surface mount or ceramic bypass capacitors. All analog inputs should be referenced directly to the single point ground. Digital inputs and outputs should be shielded from and/or routed away from the reference and analog circuitry.

SAMPLE-AND-HOLD

Both the LTC1286 and the LTC1298 provide a built-in sample-and-hold (S&H) function to acquire signals. The S&H of the LTC1286 acquires input signals from “+” input relative to “-” input during the t_{SMPL} time (see Figure 1). However, the S&H of the LTC1298 can sample input signals in the single-ended mode or in the differential inputs during the t_{SMPL} time (see Figure 7).

Single-Ended Inputs

The sample-and-hold of the LTC1298 allows conversion of rapidly varying signals. The input voltage is sampled during the t_{SMPL} time as shown in Figure 7. The sampling interval begins as the bit preceding the MSBF bit is shifted in and continues until the falling CLK edge after the MSBF bit is received. On this falling edge, the S&H goes into hold mode and the conversion begins.

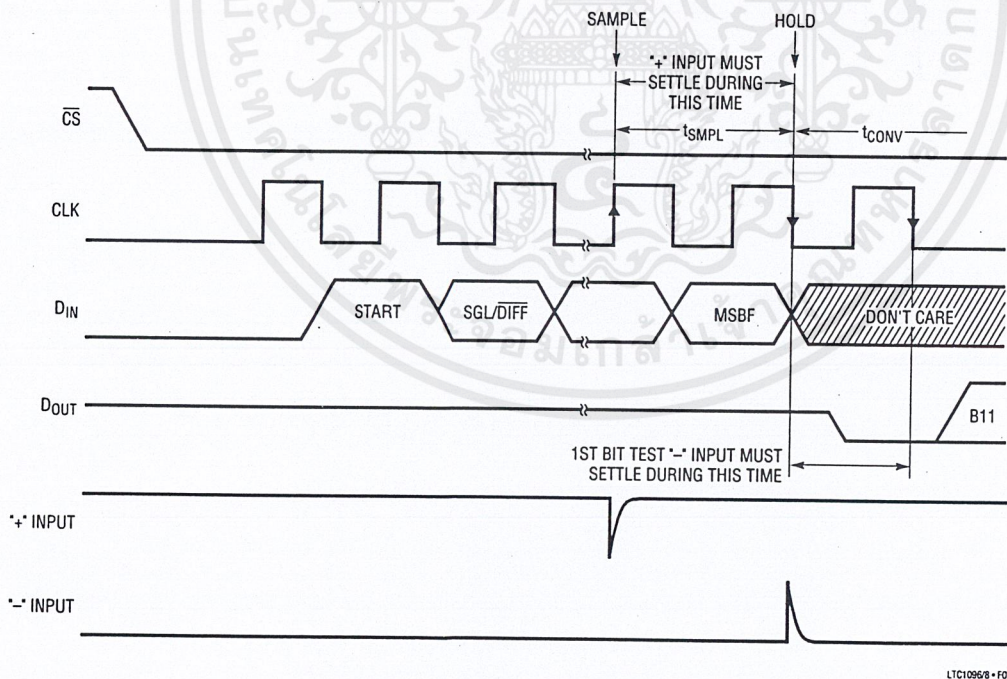


Figure 7. LTC1298 “+” and “-” Input Settling Windows

LTC1096B • F07

LTC1286/LTC1298

APPLICATION INFORMATION

Differential Inputs

With differential inputs, the ADC no longer converts just a single voltage but rather the difference between two voltages. In this case, the voltage on the selected “+” input is still sampled and held and therefore may be rapidly time varying just as in single-ended mode. However, the voltage on the selected “-” input must remain constant and be free of noise and ripple throughout the conversion time. Otherwise, the differencing operation may not be performed accurately. The conversion time is 12 CLK cycles. Therefore, a change in the “-” input voltage during this interval can cause conversion errors. For a sinusoidal voltage on the “-” input this error would be:

$$V_{\text{ERROR (MAX)}} = V_{\text{PEAK}} \times 2 \times \pi \times f(\text{“-”}) \times 12/f_{\text{CLK}}$$

Where $f(\text{“-”})$ is the frequency of the “-” input voltage, V_{PEAK} is its peak amplitude and f_{CLK} is the frequency of the CLK. In most cases V_{ERROR} will not be significant. For a 60Hz signal on the “-” input to generate a 1/4LSB error (305 μ V) with the converter running at CLK = 200kHz, its peak value would have to be 13.48mV.

ANALOG INPUTS

Because of the capacitive redistribution A/D conversion techniques used, the analog inputs of the LTC1286/LTC1298 have capacitive switching input current spikes. These current spikes settle quickly and do not cause a problem. However, if large source resistances are used or if slow settling op amps drive the inputs, care must be taken to insure that the transients caused by the current spikes settle completely before the conversion begins.

“+” Input Settling

The input capacitor of the LTC1286 is switched onto “+” input during the t_{SMPL} time (see Figure 1) and samples the input signal within that time. However, the input capacitor of the LTC1298 is switched onto “+” input during the sample phase (t_{SMPL} , see Figure 7). The sample phase is 1 1/2 CLK cycles before conversion starts. The voltage on the “+” input must settle completely within t_{SMPL} for the LTC1286 and the LTC1298 respectively. Minimizing R_{SOURCE}^+ and C1 will improve the input settling time. If a large “+” input source resistance must be used, the

sample time can be increased by using a slower CLK frequency.

“-” Input Settling

At the end of the t_{SMPL} , the input capacitor switches to the “-” input and conversion starts (see Figures 1 and 7). During the conversion, the “+” input voltage is effectively “held” by the sample-and-hold and will not affect the conversion result. However, it is critical that the “-” input voltage settles completely during the first CLK cycle of the conversion time and be free of noise. Minimizing R_{SOURCE}^- and C2 will improve settling time. If a large “-” input source resistance must be used, the time allowed for settling can be extended by using a slower CLK frequency.

Input Op Amps

When driving the analog inputs with an op amp it is important that the op amp settle within the allowed time (see Figure 7). Again, the “+” and “-” input sampling times can be extended as described above to accommodate slower op amps. Most op amps, including the LT1006 and LT1413 single supply op amps, can be made to settle well even with the minimum settling windows of 6 μ s (“+” input) which occur at the maximum clock rate of 200kHz.

Source Resistance

The analog inputs of the LTC1286/LTC1298 look like a 20pF capacitor (C_{IN}) in series with a 500 Ω resistor (R_{ON}) as shown in Figure 8. C_{IN} gets switched between the selected “+” and “-” inputs once during each conversion cycle. Large external source resistors and capacitances

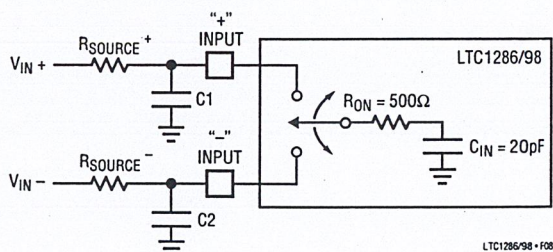


Figure 8. Analog Input Equivalent Circuit

APPLICATION INFORMATION

will slow the settling of the inputs. It is important that the overall RC time constants be short enough to allow the analog inputs to completely settle within the allowed time.

RC Input Filtering

It is possible to filter the inputs with an RC network as shown in Figure 9. For large values of C_F (e.g., $1\mu\text{F}$), the capacitive input switching currents are averaged into a net DC current. Therefore, a filter should be chosen with a small resistor and large capacitor to prevent DC drops across the resistor. The magnitude of the DC current is approximately $I_{DC} = 20\text{pF} \times V_{IN}/t_{CYC}$ and is roughly proportional to V_{IN} . When running at the minimum cycle time of $64\mu\text{s}$, the input current equals $1.56\mu\text{A}$ at $V_{IN} = 5\text{V}$. In this case, a filter resistor of 75Ω will cause 0.1LSB of full-scale error. If a larger filter resistor must be used, errors can be eliminated by increasing the cycle time.

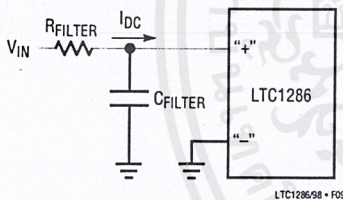


Figure 9. RC Input Filtering

Input Leakage Current

Input leakage currents can also create errors if the source resistance gets too large. For instance, the maximum input leakage specification of $1\mu\text{A}$ (at 125°C) flowing through a source resistance of 240Ω will cause a voltage drop of $240\mu\text{V}$ or 0.2LSB . This error will be much reduced at lower temperatures because leakage drops rapidly (see typical curve of Input Channel Leakage Current vs Temperature).

REFERENCE INPUTS

The reference input of the LTC1286 is effectively a $50\text{k}\Omega$ resistor from the time $\overline{\text{CS}}$ goes low to the end of the conversion. The reference input becomes a high impedance node at any other time (see Figure 10). Since the voltage on the reference input defines the voltage span of the A/D

converter, the reference input should be driven by a reference with low R_{OUT} (ex. LT1004, LT1019 and LT1021) or a voltage source with low R_{OUT} .

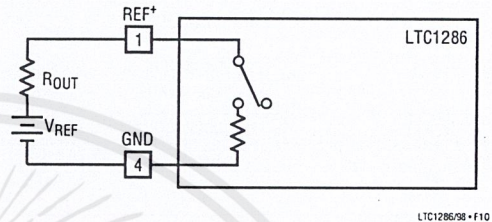


Figure 10. Reference Input Equivalent Circuit

Reduced Reference Operation

The minimum reference voltage of the LTC1298 is limited to 4.5V because the V_{CC} supply and reference are internally tied together. However, the LTC1286 can operate with reference voltages below 1V .

The effective resolution of the LTC1286 can be increased by reducing the input span of the converter. The LTC1286 exhibits good linearity and gain over a wide range of reference voltages (see typical curves of Change in Linearity vs Reference Voltage and Change in Gain vs Reference Voltage). However, care must be taken when operating at low values of V_{REF} because of the reduced LSB step size and the resulting higher accuracy requirement placed on the converter. The following factors must be considered when operating at low V_{REF} values:

1. Offset
2. Noise
3. Conversion speed (CLK frequency)

Offset with Reduced V_{REF}

The offset of the LTC1286 has a larger effect on the output code. When the ADC is operated with reduced reference voltage. The offset (which is typically a fixed voltage) becomes a larger fraction of an LSB as the size of the LSB is reduced. The typical curve of Change in Offset vs Reference Voltage shows how offset in LSBs is related to reference voltage for a typical value of V_{OS} . For example, a V_{OS} of $122\mu\text{V}$ which is 0.1LSB with a 5V reference becomes 0.5LSB with a 1V reference and 2.5LSBs with a

LTC1286/LTC1298

APPLICATION INFORMATION

0.2V reference. If this offset is unacceptable, it can be corrected digitally by the receiving system or by offsetting the “-” input of the LTC1286.

Noise with Reduced V_{REF}

The total input referred noise of the LTC1286 can be reduced to approximately 400 μ V peak-to-peak using a ground plane, good bypassing, good layout techniques and minimizing noise on the reference inputs. This noise is insignificant with a 5V reference but will become a larger fraction of an LSB as the size of the LSB is reduced.

For operation with a 5V reference, the 400 μ V noise is only 0.33LSB peak-to-peak. In this case, the LTC1286 noise will contribute virtually no uncertainty to the output code. However, for reduced references the noise may become a significant fraction of an LSB and cause undesirable jitter in the output code. For example, with a 2.5V reference this same 400 μ V noise is 0.66LSB peak-to-peak. This will reduce the range of input voltages over which a stable output code can be achieved by 1LSB. If the reference is further reduced to 1V, the 400 μ V noise becomes equal to 1.65LSBs and a stable code may be difficult to achieve. In this case averaging multiple readings may be necessary.

This noise data was taken in a very clean setup. Any setup induced noise (noise or ripple on V_{CC} , V_{REF} or V_{IN}) will add to the internal noise. The lower the reference voltage to be used the more critical it becomes to have a clean, noise free setup.

Conversion Speed with Reduced V_{REF}

With reduced reference voltages, the LSB step size is reduced and the LTC1286 internal comparator overdrive is reduced. Therefore, it may be necessary to reduce the maximum CLK frequency when low values of V_{REF} are used.

DYNAMIC PERFORMANCE

The LTC1286/LTC1298 have exceptional sampling capability. Fast Fourier Transform (FFT) test techniques are used to characterize the ADC's frequency response, dis-

tortion and noise at the rated throughput. By applying a low distortion sine wave and analyzing the digital output using an FFT algorithm, the ADC's spectral content can be examined for frequencies outside the fundamental. Figure 11 shows a typical LTC1286 plot.

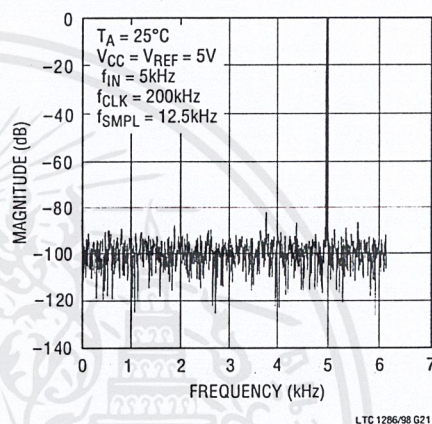


Figure 11. LTC1286 Non-Averaged, 4096 Point FFT Plot

Signal-to-Noise Ratio

The Signal-to-Noise plus Distortion Ratio (S/N + D) is the ratio between the RMS amplitude of the fundamental input frequency to the RMS amplitude of all other frequency components at the ADC's output. The output is band limited to frequencies above DC and below one half the sampling frequency. Figure 12 shows a typical spectral content with a 12.5kHz sampling rate.

Effective Number of Bits

The Effective Number of Bits (ENOBs) is a measurement of the resolution of an ADC and is directly related to S/(N+D) by the equation:

$$\text{ENOB} = [S/(N + D) - 1.76]/6.02$$

where S/(N + D) is expressed in dB. At the maximum sampling rate of 12.5kHz with a 5V supply, the LTC1286 maintains above 11 ENOBs at 10kHz input frequency. Above 10kHz the ENOBs gradually decline, as shown in Figure 12, due to increasing second harmonic distortion. The noise floor remains low.

APPLICATION INFORMATION

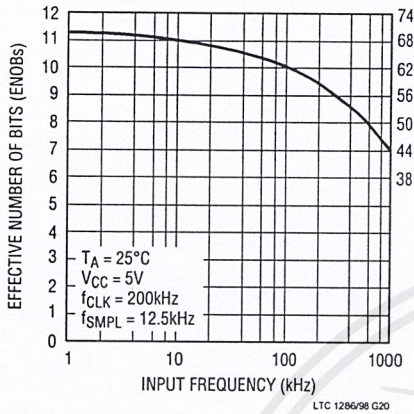


Figure 12. Effective Bits and S/(N + D) vs Input Frequency

Total Harmonic Distortion

Total Harmonic Distortion (THD) is the ratio of the RMS sum of all harmonics of the input signal to the fundamental itself. The out-of-band harmonics alias into the frequency band between DC and half of the sampling frequency. THD is defined as:

$$\text{THD} = 20\log \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \dots + V_N^2}}{V_1}$$

where V_1 is the RMS amplitude of the fundamental frequency and V_2 through V_N are the amplitudes of the second through the N^{th} harmonics. The typical THD specification in the Dynamic Accuracy table includes the 2nd through 5th harmonics. With a 7kHz input signal, the LTC1286/LTC1298 have typical THD of 80dB with $V_{CC} = 5V$.

Intermodulation Distortion

If the ADC input signal consists of more than one spectral component, the ADC transfer function nonlinearity can produce intermodulation distortion (IMD) in addition to THD. IMD is the change in one sinusoidal input caused by the presence of another sinusoidal input at a different frequency.

If two pure sine waves of frequencies f_a and f_b are applied to the ADC input, nonlinearities in the ADC transfer function can create distortion products at sum and difference frequencies of $mf_a \pm nf_b$, where m and $n = 0, 1, 2, 3$, etc. For example, the 2nd order IMD terms include $(f_a + f_b)$ and $(f_a - f_b)$ while 3rd order IMD terms include $(2f_a + f_b)$, $(2f_a - f_b)$, $(f_a + 2f_b)$, and $(f_a - 2f_b)$. If the two input sine waves are equal in magnitudes, the value (in dB) of the 2nd order IMD products can be expressed by the following formula:

$$\text{IMD}(f_a \pm f_b) = 20\log \left[\frac{\text{amplitude}(f_a \pm f_b)}{\text{amplitude at } f_a} \right]$$

For input frequencies of 5kHz and 6kHz, the IMD of the LTC1286/LTC1298 is 73dB with a 5V supply.

Peak Harmonic or Spurious Noise

The peak harmonic or spurious noise is the largest spectral component excluding the input signal and DC. This value is expressed in dBs relative to the RMS value of a full-scale input signal.

Full-Power and Full-Linear Bandwidth

The full-power bandwidth is that input frequency at which the amplitude of the reconstructed fundamental is reduced by 3dB for a full-scale input.

The full-linear bandwidth is the input frequency at which the effective bits rating of the ADC falls to 11 bits. Beyond this frequency, distortion of the sampled input signal increases. The LTC1286/LTC1298 have been designed to optimize input bandwidth, allowing the ADCs to undersample input signals with frequencies above the converters' Nyquist Frequency.

LTC1286/LTC1298

TYPICAL APPLICATIONS

MICROPROCESSOR INTERFACES

The LTC1286/LTC1298 can interface directly without external hardware to most popular microprocessor (MPU) synchronous serial formats (see Table 1). If an MPU without a dedicated serial port is used, then 3 or 4 of the MPU's parallel port lines can be programmed to form the serial link to the LTC1286/LTC1298. Included here is one serial interface example and one example showing a parallel port programmed to form the serial interface.

Motorola SPI (MC68HC11)

The MC68HC11 has been chosen as an example of an MPU with a dedicated serial port. This MPU transfers data MSB-first and in 8-bit increments. The D_{IN} word sent to the data register starts with the SPI process. With three 8-bit transfers, the A/D result is read into the MPU. The second 8-bit transfer clocks B11 through B8 of the A/D conversion result into the processor. The third 8-bit transfer clocks the remaining bits, B7 through B0, into the MPU. The data is right justified into two memory locations. ANDing the second byte with $0F_{HEX}$ clears the four most significant bits. This operation was not included in the code. It can be inserted in the data gathering loop or outside the loop when the data is processed.

MC68HC11 Code

In this example the D_{IN} word configures the input MUX for a single-ended input to be applied to CHO. The conversion result is output MSB-first.

Table 1. Microprocessor with Hardware Serial Interfaces Compatible with the LTC1286/LTC1298

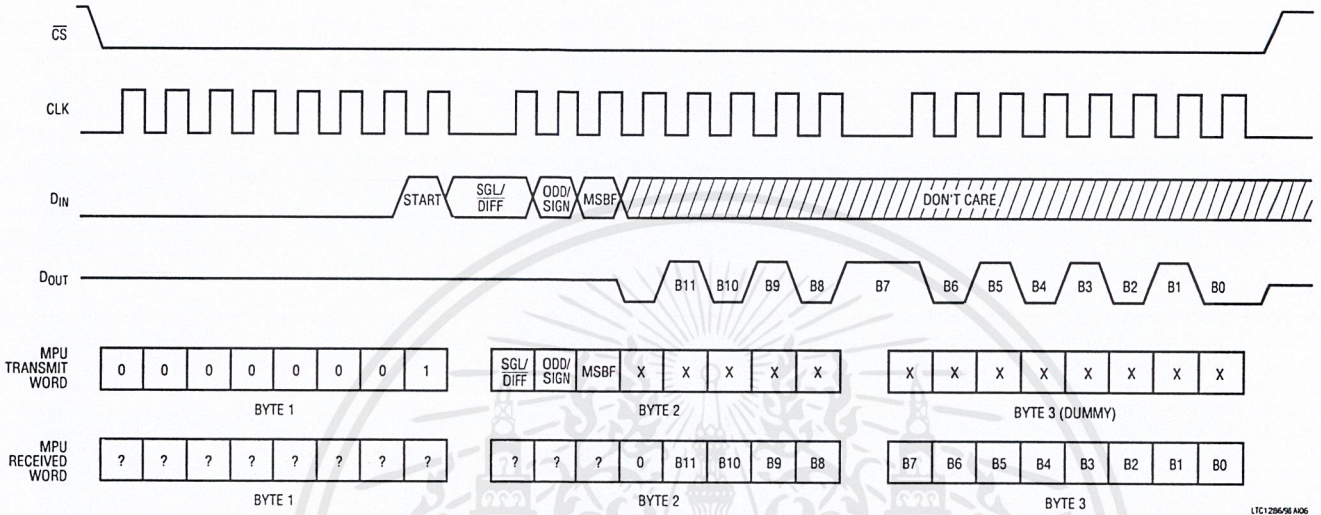
PART NUMBER	TYPE OF INTERFACE
Motorola	
MC6805S2,S3	SPI
MC68HC11	SPI
MC68HC05	SPI
RCA	
CDP68HC05	SPI
Hitachi	
HD6305	SCI Synchronous
HD63705	SCI Synchronous
HD6301	SCI Synchronous
HD63701	SCI Synchronous
HD6303	SCI Synchronous
HD64180	CSI/O
National Semiconductor	
COP400 Family	MICROWIRE [†]
COP800 Family	MICROWIRE/PLUS [†]
NS8050U	MICROWIRE/PLUS [†]
HPC16000 Family	MICROWIRE/PLUS [†]
Texas Instruments	
TMS7002	Serial Port
TMS7042	Serial Port
TMS70C02	Serial Port
TMS70C42	Serial Port
TMS32011*	Serial Port
TMS32020	Serial Port
Intel	
8051	Bit Manipulation on Parallel Port

* Requires external hardware

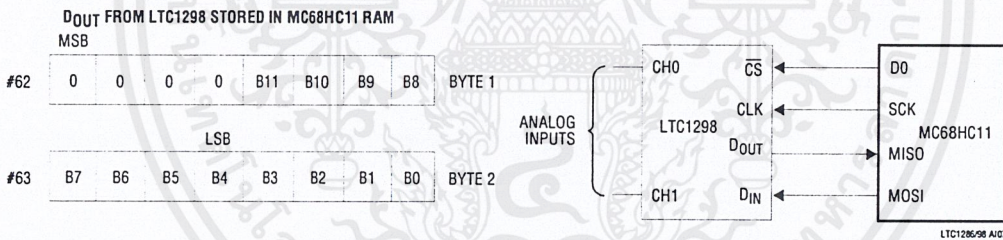
[†] MICROWIRE and MICROWIRE/PLUS are trademarks of National Semiconductor Corp.

TYPICAL APPLICATIONS

Timing Diagram for Interface to the MC68HC11



Hardware and Software Interface to the MC68HC11



LABEL	MNEMONIC	OPERAND	COMMENTS	LABEL	MNEMONIC	OPERAND	COMMENTS
	LDA	#\$50	CONFIGURATION DATA FOR SPCR	WAIT1	BPL	WAIT1	CHECK IF TRANSFER IS DONE
	STAA	\$1028	LOAD DATA INTO SPCR (\$1028)	LDA	\$51		LOAD DIN INTO ACC A FROM \$51
	LDA	#\$1B	CONFIG. DATA FOR PORT D DDR	STAA	\$102A		LOAD DIN INTO SPI, START SCK
	STAA	\$1009	LOAD DATA INTO PORT D DDR	WAIT2	LDA	\$1029	CHECK SPI STATUS REG
	LDA	#\$01	LOAD DIN WORD INTO ACC A	BPL	WAIT2		CHECK IF TRANSFER IS DONE
	STAA	\$50	LOAD DIN DATA INTO \$50	LDA	\$102A		LOAD LTC1291 MSBs INTO ACC A
	LDA	#\$A0	LOAD DIN WORD INTO ACC A	STAA	\$62		STORE MSBs IN \$62
	STAA	\$51	LOAD DIN DATA INTO \$51	LDA	\$52		LOAD DUMMY INTO ACC A FROM \$52
	LDA	#\$00	LOAD DUMMY DIN WORD INTO ACC A	STAA	\$102A		LOAD DUMMY DIN INTO SPI, START SCK
	STAA	\$52	LOAD DUMMY DIN DATA INTO \$52	WAIT3	LDA	\$1029	CHECK SPI STATUS REG
	LDX	#\$1000	LOAD INDEX REGISTER X WITH \$1000	BPL	WAIT3		CHECK IF TRANSFER IS DONE
LOOP	BCLR	\$08,X,#\$01	DO GOES LOW (CS GOES LOW)	BSET	\$08,X,#\$01		DO GOES HIGH (CS GOES HIGH)
	LDA	\$50	LOAD DIN INTO ACC A FROM \$50	LDA	\$102A		LOAD LTC1291 LSBs IN ACC
	STAA	\$102A	LOAD DIN INTO SPI, START SCK	STAA	\$63		STORE LSBs IN \$63
	LDA	\$1029	CHECK SPI STATUS REG	JMP	LOOP		START NEXT CONVERSION

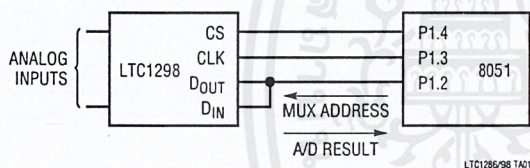
LTC1286/LTC1298

TYPICAL APPLICATIONS

Interfacing to the Parallel Port of the INTEL 8051 Family

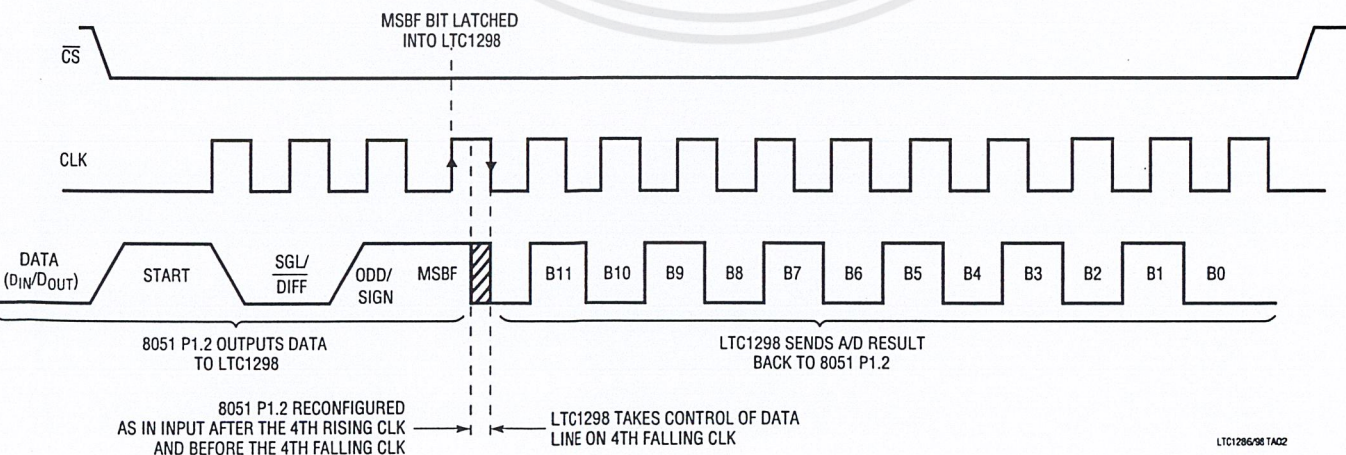
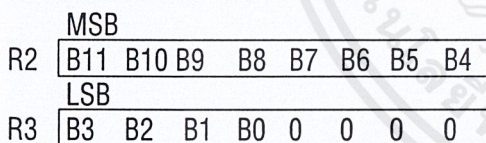
The Intel 8051 has been chosen to demonstrate the interface between the LTC1298 and parallel port micro-processors. Normally the \overline{CS} , CLK and D_{IN} signals would be generated on 3 port lines and the D_{OUT} signal read on a 4th port line. This works very well. However, we will demonstrate here an interface with the D_{IN} and D_{OUT} of the LTC1298 tied together as described in the SERIAL INTERFACE section. This saves one wire.

The 8051 first sends the start bit and MUX address to the LTC1298 over the data line connected to P1.2. Then P1.2 is reconfigured as an input (by writing to it a one) and the 8051 reads back the 12-bit A/D result over the same data line.



LABEL	MNEMONIC	OPERAND	COMMENTS
LOOP 1	MOV	A, #FFH	D_{IN} word for LTC1298
	SETB	P1.4	Make sure \overline{CS} is high
	CLR	P1.4	\overline{CS} goes low
	MOV	R4, #04	Load counter
LOOP 2	RLC	A	Rotate D_{IN} bit into Carry
	CLR	P1.3	SCLK goes low
	MOV	P1.2, C	Output D_{IN} bit to LTC1298
	SETB	P1.3	SCLK goes high
LOOP 3	DJNZ	R4, LOOP 1	Next bit
	MOV	P1, #04	Bit 2 becomes an input
	CLR	P1.3	SCLK goes low
	MOV	R4, #09	Load counter
LOOP 4	MOV	C, P1.2	Read data bit into Carry
	RLC	A	Rotate data bit into Acc.
	SETB	P1.3	SCLK goes high
	CLR	P1.3	SCLK goes low
LOOP 5	DJNZ	R4, LOOP 2	Next bit
	MOV	R2, A	Store MSBs in R2
	CLR	A	Clear Acc.
	MOV	R4, #04	Load counter
LOOP 6	MOV	C, P1.2	Read data bit into Carry
	RLC	A	Rotate data bit into Acc.
	SETB	P1.3	SCLK goes high
	CLR	P1.3	SCLK goes low
LOOP 7	DJNZ	R4, LOOP 3	Next bit
	MOV	R4, #04	Load counter
	RRC	A	Rotate right into Acc.
	DJNZ	R4, LOOP 4	Next Rotate
LOOP 8	MOV	R3, A	Store LSBs in R3
	SETB	P1.4	\overline{CS} goes high

D_{OUT} FROM 1298 STORED IN 8501 RAM



TYPICAL APPLICATIONS

A “Quick Look” Circuit for the LTC1286

Users can get a quick look at the function and timing of the LTC1286 by using the following simple circuit (Figure 13). V_{REF} is tied to V_{CC} . V_{IN} is applied to the +IN input and the -IN input is tied to the ground. \overline{CS} is driven at 1/16 the clock rate by the 74C161 and D_{OUT} outputs the data. The output data from the D_{OUT} pin can be viewed on an oscilloscope that is set up to trigger on the falling edge of \overline{CS} (Figure 14). Note the LSB data is partially clocked out before \overline{CS} goes high.

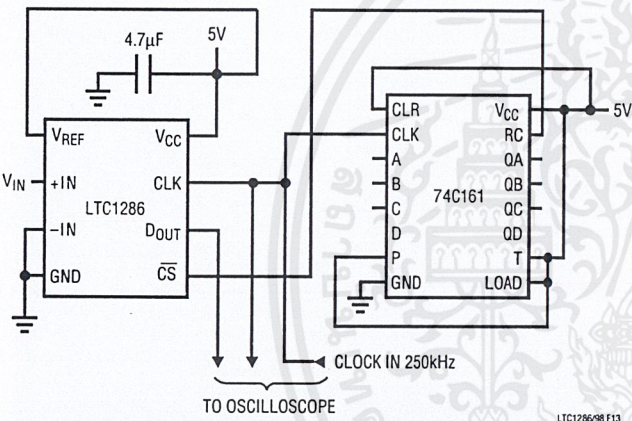


Figure 13. “Quick Look” Circuit for the LTC1286

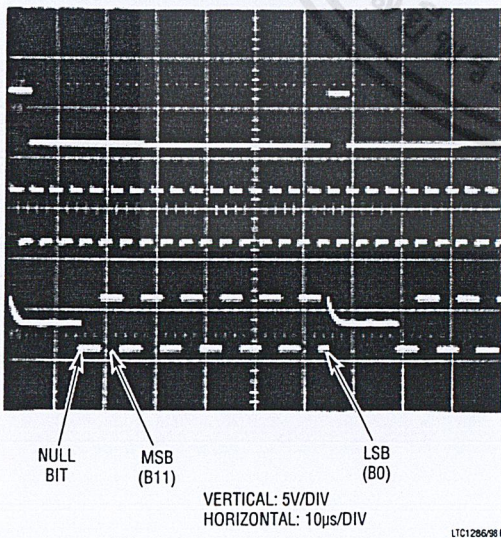


Figure 14. Scope Trace the LTC1286 “Quick Look” Circuit Showing A/D Output 1010101010 (AAHex)

Micropower Battery Voltage Monitor

A common problem in battery systems is battery voltage monitoring. This circuit monitors the 10 cell stack of NiCad or NiMH batteries found in laptop computers. It draws only $67\mu A$ from the 5V supply at $f_{SAMPL} = 0.1kHz$ and $25\mu A$ to $55\mu A$ from the battery. The 12-bits of resolution of the LTC1286 are positioned over the desired range of 8V to 16V. This is easily accomplished by using the ADC's differential inputs. Tying the -input to the reference gives an ADC input span of V_{REF} to $2V_{REF}$ (2.5V to 5V). The resistor divider then scales the input voltage for 8V to 16V.

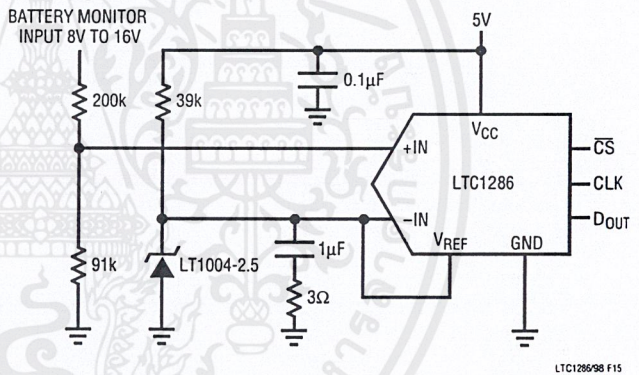
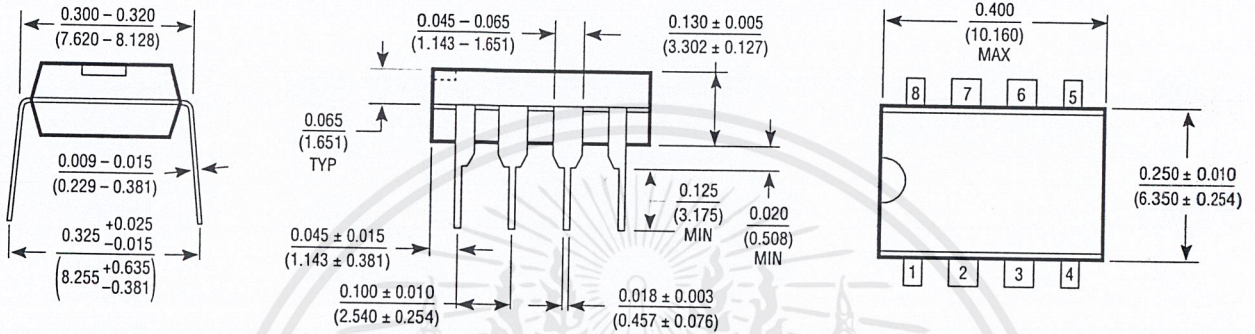


Figure 15. Micropower Battery Voltage Monitor

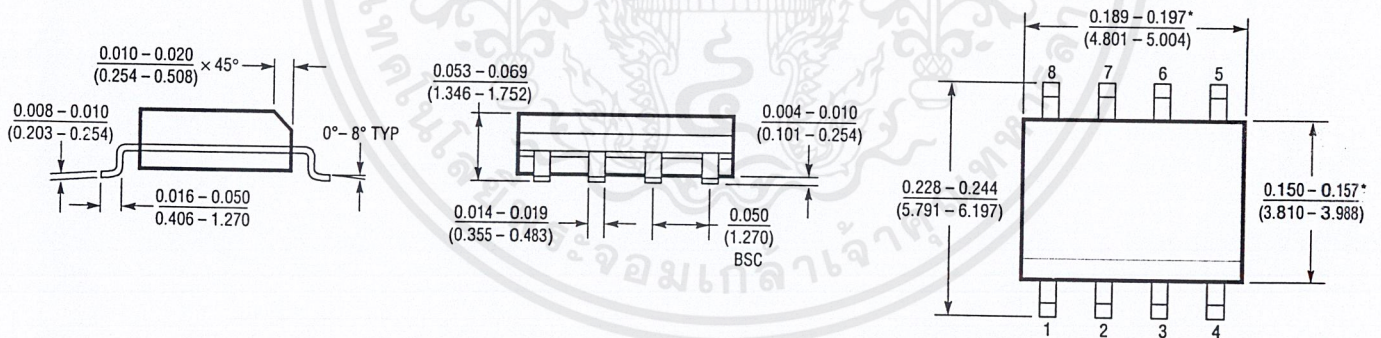
LTC1286/LTC1298

PACKAGE DESCRIPTION Dimensions in inches (millimeters) unless otherwise noted.

**N8 Package
8-Lead Plastic DIP**



**S8 Package
8-Lead Plastic SOIC**



*THESE DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.006 INCH (0.15mm).

SD8 029



บรรณานุกรม

1. กฤษดา ใจเย็น , ชัยวัฒน์ ลิ้มพรจิตวิไล และ อรรถพล บุญยะโกคา. **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม.** บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด : กรุงเทพฯ , 2542.
2. จักรพงษ์ สุขประเสริฐ , สัจจะ จรัสรุ่งรวีร. **เริ่มต้นอย่างมืออาชีพด้วย Delphi 5.** สำนักพิมพ์ อินโฟเพรส : กรุงเทพฯ , 2543.
3. ชัยวัฒน์ ลิ้มพรจิตวิไล , วรพจน์ กรแก้ววัฒนกุล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51.** บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด : กรุงเทพฯ , 2542.
4. สมยศ จุณณะปิยะ. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51** พิมพ์ครั้งที่ 3. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง : กรุงเทพฯ , 2543.
5. สุวัฒน์ ปุณณชัยยะ. **เปิดโลกของ TCP/IP และโปรโตคอลของอินเทอร์เน็ต.** บริษัทด้านสุทธาการพิมพ์ จำกัด : กรุงเทพฯ , 2543.
6. Shawn Carlson. **The Amateur Scientist.** Scientific American : USA , 2000.