



ระบบส่งสัญญาณแจ้ง เหตุเพลิงไหม้
(FIRE ALARM SYSTEM)

โดย

นางสาว กัลยรัตน์ พ่วงพุ่ม

นาย สรร เสรีชัย ยิงชนะ เกียรติ

นางสาว สุกัญญา วัฒนโพธิ์



ปริญญานิพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบัน เทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้
(FIRE ALARM SYSTEM)

โดย

นางสาว กัลยรัตน์ พ่วงพุ่ม 34101013
นาย สรร เสริญ ยิงชนะ เกียรติ 34107406
นางสาว สุกัญญา โรจนโพธิ์ 34108443

อาจารย์ที่ปรึกษา

ผศ. สมยศ จุณะปิยะ

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2537

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้

โดย

1. นางสาว กัลยรัตน์ พ่วงพุ่ม 34101013
2. นาย สรรเสริญ บึงชนะเกียรติ 34107406
3. นางสาว สุกัญญา โรจนโพธิ์ 34108443



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้
(Fire Alarm System)

โดย	นางสาว กัลยรัตน์ พ่วงพุ่ม	34101013
	นาย สรรเสริญ ยิ่งชนะเกียรติ	34107406
	นางสาว สุภกัญญา วิจารณ์โพธิ์	34108443

อาจารย์ที่ปรึกษา ผศ. สมยศ จุณณะปิยะ

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ แสดงการออกแบบ ระบบการส่งสัญญาณแจ้งเหตุเพลิงไหม้ (Fire Alarm System) โดยใช้เทคนิคการสื่อสารข้อมูลแบบอนุกรม ส่งข้อมูลอะซิงโครนัส (Asynchronous) มีการเชื่อมต่อกับพอร์ตคอมพิวเตอร์ตามมาตรฐานของ RS 232 C ผ่านสายเพียงคู่เดียว โดยออกแบบให้สามารถส่งข้อมูลด้วยความเร็ว 4800บิตต่อวินาทีและมีการนำชิพที่ทำหน้าที่เป็น Addressable Asynchronous Receiver / Transmitter ทำการออกแบบเป็นวงจร Addressable Fire Detector (AFD) และมีอุปกรณ์ตรวจจับ (Sensor) สามารถตรวจสอบสถานะ ที่ตำแหน่งที่ติดตั้งอุปกรณ์ AFDแล้วส่งข้อมูลเพื่อแจ้งตำแหน่งเกิดเหตุเพลิงไหม้ หรือสภาวะอื่นๆออกทางหน้าจอคอมพิวเตอร์

สำหรับโครงงานนี้ แบ่งส่วนการทำงานเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของฮาร์ดแวร์ (Hard Ware) และส่วนของโปรแกรมซอฟต์แวร์ เพื่อใช้ในการติดต่อระหว่าง CPU กับ AFD

การติดต่อระหว่าง AFD กับ CPU จะทำงานเป็นไปตามการควบคุมของซอฟต์แวร์และการแสดงผลออกหน้าจอมีการใช้โหมดกราฟฟิก ซึ่งระบบนี้สามารถชี้ตำแหน่งที่เกิดเพลิงไหม้ได้อย่างถูกต้องชัดเจน และแม่นยำส่งผลให้สามารถแก้ไขสถานการณ์ได้ทันทั่วทั้ง

Fire Alarm System

By

Kanyarat Phongpum 34101013

Sansern Yingchanakiat 34107406

Suphakanya Rodjanapho 34108443

Somyot Junnapiya

Advisor

Academic Year 1994

Abstract

This thesis will present to design the Fire Alarm System which uses the serial data communication technique and send asynchronous. This system has the interfacing for computer port by RS 232 C and uses only two lines. It has the data transmission speed 4800 Bit/sec and uses the addressable asynchronous receiver /transmitter to design addressable fire detector (AFD) and this system has sensors which can send the status and the correct point to display the alarm on the monitor.

The communication between the CPU and AFD operated by software. The system can show any fault alarm to the right location this will cause us monitoring all the statust and help at once.

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
บทที่ 3 การออกแบบระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้	20
บทที่ 4 การทดลองและผลการทดลอง	40
บทที่ 5 สรุปผลการทดลอง	50
ภาคผนวก	
หนังสืออ้างอิง	
กิตติกรรมประกาศ	

บทที่ 1

บทนำ

ปัจจุบันนี้มีการสร้างอาคารที่มีความสูงหลายๆชั้น ขนาดใหญ่มากมายซึ่งต้องใช้ทุนก่อสร้างสูงมาก จึงจำเป็นต้องมีการออกแบบระบบรักษาความปลอดภัยโดยเฉพาะระบบการป้องกันอัคคีภัย โดยส่วนใหญ่มักจะติดตั้งระบบแจ้งเหตุเพลิงไหม้ซึ่งที่มีใช้ทั่วไปเป็นแบบ Hardwire ต้องใช้สายจำนวนมากถ้าต้องการติดตั้งตัว ตรวจจับ (Detector) จำนวนหลายตัวแล้วก็จะต้องใช้สายจำนวนมากขึ้นเช่นกัน ทำให้เพิ่มค่าใช้จ่ายมากขึ้น จึงมีแนวคิดที่จะพัฒนาให้ระบบมีการใช้สายในการรับส่งข้อมูลเพียงคู่เดียวโดยนำไอซีที่สามารถจัดตำแหน่งที่อยู่ (Address) ของตัวเองได้ มาต่อเข้ากับวงจรที่ทำหน้าที่เป็นตัวตรวจจับ (Sensor) เรียกว่า แอด-แตรสเซเบิล ไฟร์ ดีเทคเตอร์ (Addressable Fire Detector หรือ AFD) แล้วนำ AFD แต่ละตัวต่อขนานกันภายใต้สายเพียงคู่เดียว

การออกแบบระบบแจ้งเหตุเพลิงไหม้โดยใช้ AFD นี้จะมีการติดต่อรับส่งข้อมูลระหว่าง CPU กับ AFD ใช้การส่งข้อมูลแบบอะซิงโครนัส มีการเชื่อมต่อพอร์ตอนุกรมตามมาตรฐาน RS 232 C โดยส่วนฮาร์ดแวร์นั้นมี 3 ส่วน คือส่วนควบคุมในการติดต่อสื่อสาร (เพื่อแปลงระดับสัญญาณให้เหมาะสม) ส่วนของวงจร AFD และส่วนตรวจจับจะกล่าวถึงรายละเอียดในบทที่ 3 ต่อไป

ทางด้านซอฟต์แวร์ใช้การเขียนโปรแกรมด้วยภาษาซี ออกแบบรับส่งข้อมูลบนพอร์ตอนุกรมแล้วแสดงผลออกทางหน้าจอคอมพิวเตอร์ โดยสามารถเลือกได้ว่า ผู้ใช้ (User) สอบถามสถานะที่ติดตั้งอุปกรณ์ AFD หรือให้ระบบตรวจสอบสถานะเอง สำหรับระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้ นี้ จะสามารถแสดงจุดที่เกิดเหตุเพลิงไหม้ได้บนหน้าจอคอมพิวเตอร์ ได้อย่างชัดเจนและถูกต้อง เพื่อจะช่วยให้แก้ไขเหตุการณ์ได้ทันที่ ทำให้ลดความเสียหายที่อาจเกิดขึ้นจากอัคคีภัยได้

บทที่ 2

ทฤษฎีและหลักการ

ทฤษฎีและหลักการต่างๆที่เกี่ยวข้องที่ใช้ในการทดลองในโครงงานนี้มีดังนี้

2.1 การทำงานของ Addressable Aynchronous receiver/transmitter(MC 14469)

ไอซี MC 14469 เป็นไอซีที่ทำหน้าที่รับข้อมูลแบบขนานมาแปลงเป็นอนุกรม และรับข้อมูลแบบอนุกรม มาแปลงเป็นแบบขนาน ซึ่งสามารถกำหนดแอดเดรสบนตัวมันเองได้ จึงเหมาะสมที่จะใช้ทำหน้าที่เป็นตัวกำหนดจุดหมายปลายทางที่ต้องการจะติดต่อและรับสัญญาณควบคุมที่ถูกส่งมาเป็นข้อมูลแบบอนุกรม

การทำงานของ ไอซี MC 14469

1. เมื่อมีสัญญาณแบบอนุกรมเข้ามาที่ขา RI(Receive Input) สัญญาณนาฬิกา (Clock) ภายใน MC 14469 ซึ่งจะทำงานที่ความถี่ 1/64 เท่าของความถี่ที่ขา OSC1 จะทำให้ตัวมันเองตรวจสอบข้อมูลที่ส่งเข้ามาทีละบิต เริ่มด้วย บิตเริ่มต้น(Start bit) ตามด้วยแอดเดรสจำนวน 8 บิต ซึ่ง 7 บิตแรกจะนำมาเทียบสถานะที่ขาแอดเดรส (A0-A6) ว่าตรงกันหรือไม่ บิตที่ 8 เป็นตัวบอกว่าเวิร์ด (Word) นั้นเป็นแอดเดรส เวิร์ด (Address Word) ถ้ามันเป็น 1 ค่อยไปเป็นพาริตี บิต (Parity bit) ซึ่งจะตรวจสอบดูความถูกต้องโดยวงจรลอจิก (Logic) ภายในตัวไอซี กำหนดให้เป็นพาริตี คู่ (Even parity) สุดท้ายจะเป็นบิตจบ (Stop bit) คือบิตที่บอกการสิ้นสุดของเวิร์ด

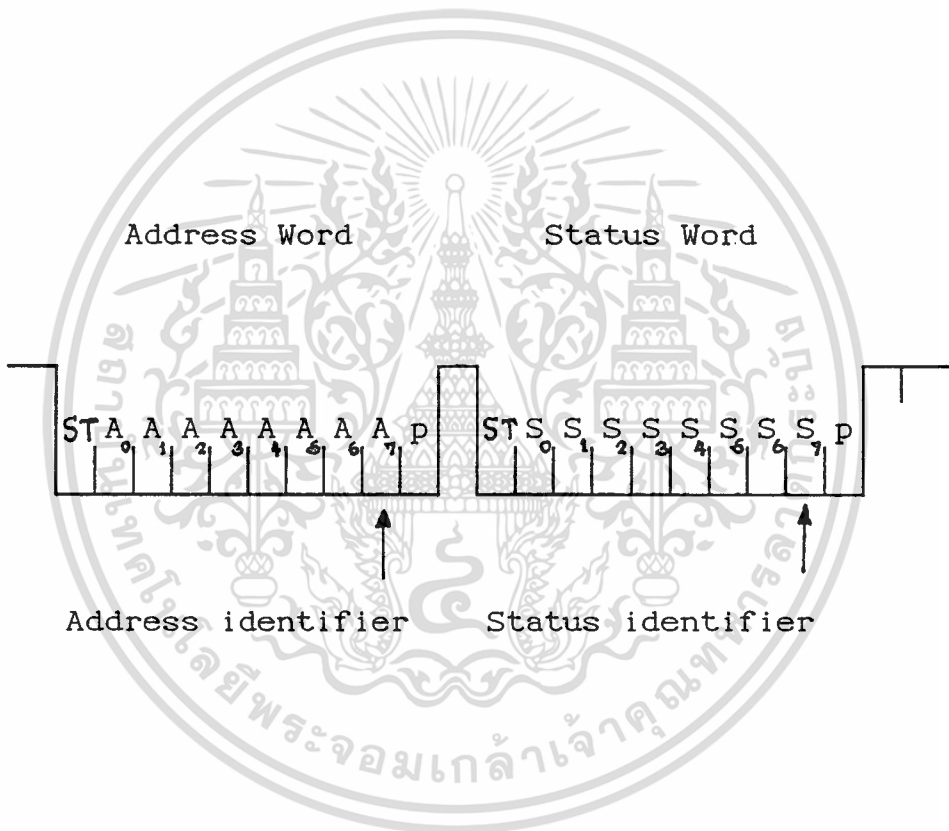
2. ถ้าแอดเดรส เวิร์ด ที่รับมาถูกต้องตามที่กำหนดไว้ที่ขา A0-A6 สัญญาณที่ขา VAP จะเกิดขึ้นและตามมาด้วย คอมมานด์ เวิร์ด ซึ่งบิตที่ 8 ของคอมมานด์ เวิร์ด เป็น 0 เพื่อบอกว่าเป็นคอมมานด์ เวิร์ด จากนั้นจะมีสัญญาณออกมาที่ขา C0-C6 ตามค่าที่ส่งมา

3. ในการส่งสัญญาณอนุกรม เริ่มต้นการส่งเมื่อขา SEND เป็น 1 มันจะรับข้อมูลเข้าที่ขา ID0-ID7 แล้วส่งออกไปก่อนตามด้วยข้อมูลที่ขา S0-S7 จะถูกส่งตามออกไป โดยการส่งจะส่งแบบอนุกรมออกไปที่ขา TRO

สำหรับไอซี MC 14469 นี้เราสามารถกำหนดแอดเดรสบนตัวมันได้ถึง 27 = 128 ตำแหน่ง เพราะมีขาแอดเดรส 7 ขา จึงทำให้เราสามารถควบคุมได้ถึง 128

ตัวบอกรหัสว่า เวอร์ด นี้เป็น แอดแตรส เวอร์ด ($A7 = 1$) ส่วน $C0-C6$ เป็น คอมมานด์ เวอร์ด $C7$ เป็นตัวบอกรหัสว่า เวอร์ด นี้เป็น คอมมานด์ เวอร์ด ($C7=0$)

ในทำนองเดียวกัน รูปแบบข้อมูลที่ AFD ส่งหรือตอบรับไปยัง CPU แสดง ในรูปที่ โดยที่ $A0 -A6$ คือ แอดแตรส ของ AFD และ $S0-S6$ เป็นสแตตัส เวอร์ด โดยมี $S0-S6$ เป็นตัวบอกรหัสของ AFD ว่าอยู่ในสภาวะใด $S7$ เป็น ตัวบอกรหัสว่าเป็น สแตตัส เวอร์ด



รูปที่ 2.2 แสดงAFD ส่งข้อมูลตอบรับ CPU

2.3 มาตรฐาน RS-232

DCE: เป็นอุปกรณ์ที่มีฟังก์ชันการทำงานต่างๆ ที่ทำให้เกิดการเชื่อมต่ออย่าง ดำรงต่อไป และยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณ และ สร้างรหัสสัญญาณต่างๆ ที่จำเป็นต้องใช้ในการสื่อสารข้อมูลระหว่าง DTE และ Data circuit โดย DCE อาจเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์ หรือไม่ก็ได้

DTE: 1. เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (data source) หรือ ตัวรับข้อมูล (data sink) หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้

2. เป็นอุปกรณ์ที่ประกอบด้วย ฟังก์ชัน ยูนิค ต่อไปนี้ คือ คอนโทรล โลจิก(control logic), บัฟเฟอร์ สตอร์(buffer store)และอุปกรณ์อินพุท หรือเอาต์พุทจำนวนหนึ่งตัวหรือมากกว่าก็ได้ หรือรวมเครื่องคอมพิวเตอร์เข้าไป ด้วยก็ได้ DTE อาจจะรวมส่วน เออเรอร์ คอนโทรล ซิงโครไนเซชัน (error control synchronization)และความสามารถในการบ่งหรือระบุว่าต้องการ เกี่ยวกับอุปกรณ์ตัวใด(station identification capability)เข้าไป ด้วยก็ได้ถ้าเทอร์มินัลและคอมพิวเตอร์ของเราเป็น DCE และ DTE ทั้งคู่เราจะ ทำการสื่อสารข้อมูลได้อย่างไรปัญหาที่เกิดขึ้นเราแก้ไขโดยใช้สายเคเบิล(cable) ที่เรียกว่า "นูล โมเด็ม" (null modem)

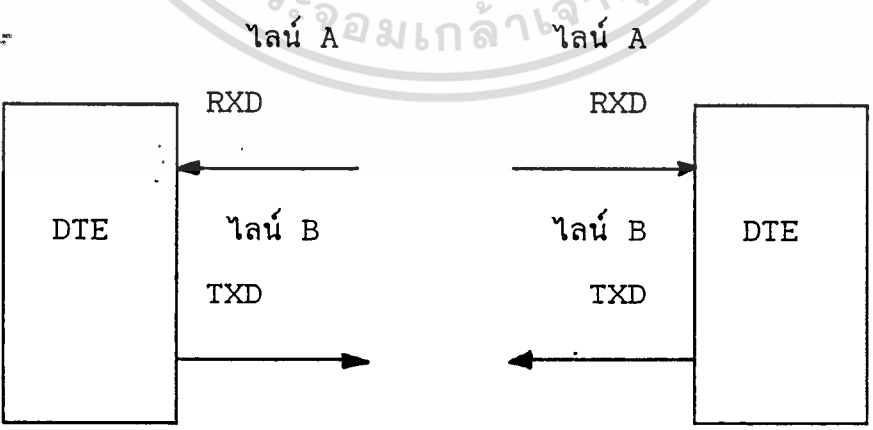
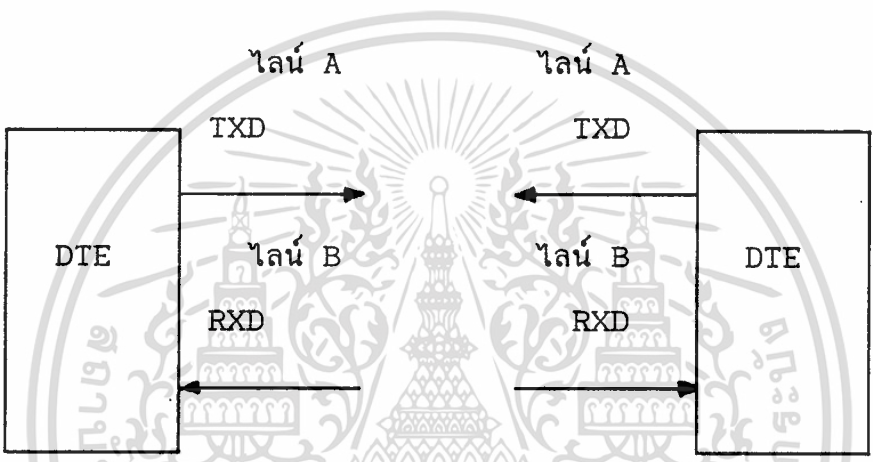
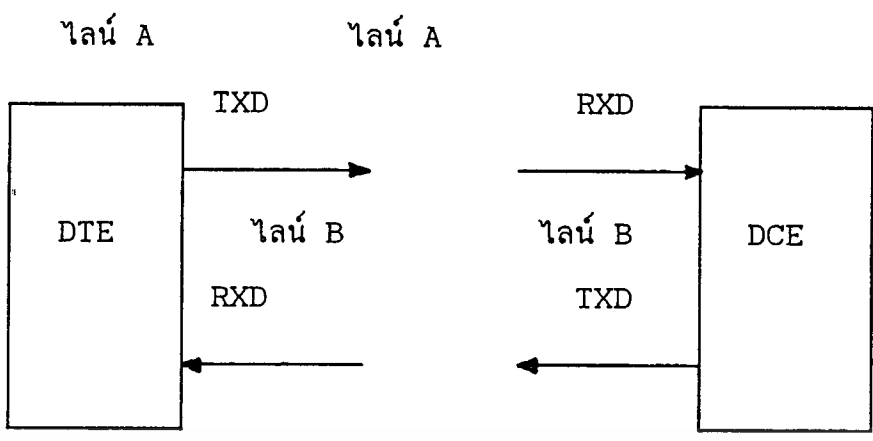
1. null หมายความว่า อุปกรณ์ตัวนี้ไม่สามารถทำอะไรได้ เรา ใช้อุปกรณ์ตัวนี้ เมื่อต้องการเปลี่ยนทิศทางการเคลื่อนที่ของข้อมูลเท่านั้น

2. modem แสดงว่าอุปกรณ์ตัวนี้เป็น DCE เหตุนี้เราจึงใช้ null modem แทรกเข้าไประหว่าง DTE ทั้งสองตัว เพื่อให้เราสามารถ ทำการ สื่อสารข้อมูลโดยผ่าน RS-232C ได้

สำหรับลักษณะของสาย (line) ที่ใช้ในการรับและส่งข้อมูลของ DTE และ DCE เป็นดังนี้

1. สายที่ใช้ในการส่งและรับข้อมูลของ DTE และ DCE มีอยู่สอง เส้น แต่ละเส้นจะมีทิศทางการเคลื่อนที่ของข้อมูลกำหนดได้ต่างทิศทางกัน

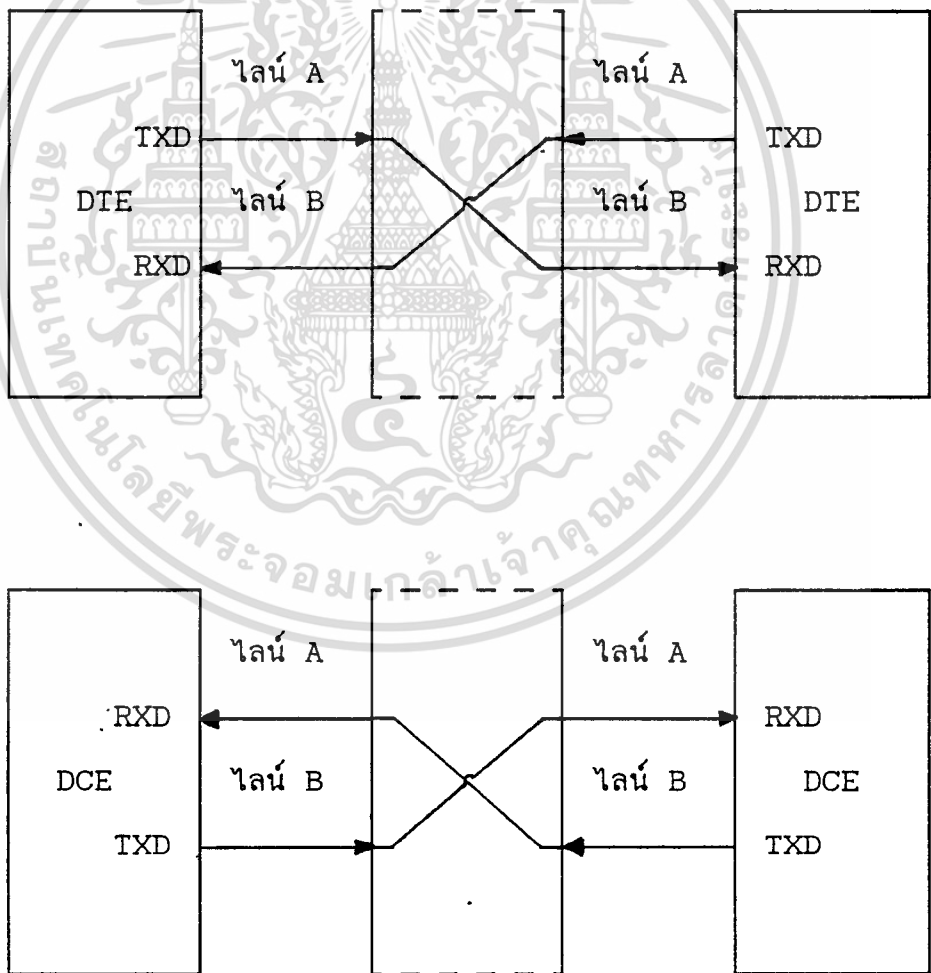
2. DTE จะส่งข้อมูลทาง ไลน์ A และ DCE จะรับข้อมูลทางไลน์ A เช่นเดียวกัน DCE จะส่งข้อมูลทาง ไลน์ B และ DTE รับข้อมูลทางไลน์ B ดังแสดงในรูปที่



รูปที่ 2.3 ลักษณะการส่งข้อมูลและรับข้อมูลของ DTE และ DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ถ้า DTE 2 ตัว ทำการแลกเปลี่ยนข้อมูลกัน ข้อมูลจะถูกส่งออกสองทาง ไลน์ A และรับทาง ไลน์ B ทั้งคู่ ดังนั้นการสื่อสารข้อมูลจะไม่สามารถเกิดขึ้นได้ ในกรณีที่ DCE สองตัวทำการแลกเปลี่ยนข้อมูลกัน เมื่อเราต่อ ไลน์ A เข้าด้วยกันและต่อ ไลน์ B เข้าด้วยกัน ข้อมูลใน ไลน์ B จะต้านกันเอง ส่วนในไลน์ A ไม่มีข้อมูลที่จะรับ ปัญหานี้แก้ได้โดยการใช้ นูล โมเด็ม (null modem) เข้ามาช่วย นูล โมเด็ม เคเบิล (null modem cable) จะทำการไขว้ ไลน์ A เข้ากับ ไลน์ B ดังรูปที่ 2.4



รูปที่ 2.4 ลักษณะการทำงานของ นูล โมเด็ม (null modem)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรที่ใช้การอินเทอร์เฟซ DTE หรือ DCE สองตัว
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราใช้มาตรฐาน RS-232C ในการสื่อสารข้อมูลแบบอนุกรมระหว่าง DCE กับ DTE โดยการส่งข้อมูลจะถูกกำหนดให้อยู่ระหว่าง 0 ถึง 2000 บิตต่อวินาที ในการประยุกต์ใช้งาน RS-232C ความเร็วสูงสุดที่ควรใช้จะมีค่าไม่เกิน 19.2 กิโลบิตต่อวินาที

พอร์ตแบบอนุกรมส่วนใหญ่ จะมีรูปร่างขึ้นอยู่กับมาตรฐานของ RS-232C คือมีขา 25 ขาที่คอนเนคเตอร์แต่ละปลายสายส่ง (แต่ IBM AT จะมีเพียง 9 ขา) แต่ถึงแม้ว่าจะมีจำนวนขาเท่ากัน แต่พอร์ตส่วนใหญ่จะมีสัญญาณที่ไม่เหมือนสัญญาณของ RS-232 ทั้งหมด เพราะว่าบางสัญญาณไม่จำเป็นต้องใช้สำหรับสัญญาณพื้นฐานของ RS-232 แสดงดังตารางที่ 1

สัญญาณ	ชื่อย่อ	หมายเลขขา
Request to send	RTS	4
Clean to send	CTS	5
Data set ready	DSR	6
Data terminal ready	DTR	20
Transmit data	TxD	2
Receive data	RxD	3
Ground	GRD	7

ตารางที่ 1 แสดงสัญญาณพื้นฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สัญญาณทั้งหมดมีมากกว่านี้ เพราะเริ่มแรกนั้น พอร์ตอนุกรมถูกออกแบบมาเพื่อใช้ร่วมกับโมเด็ม ดังนั้นเมื่อนำไปใช้กับอุปกรณ์อื่น บางสัญญาณจึงไม่จำเป็น

2.4 คุณสมบัติของสัญญาณไฟฟ้า

สัญญาณที่ขาทุกขาที่คอนเนคเตอร์ของRS-232C จะเป็นสภาวะใดสภาวะหนึ่งในแต่ละคู่ต่อไปนี้

MARK/SPACE

ON/OFF

LOGIC 0/LOGIC 1

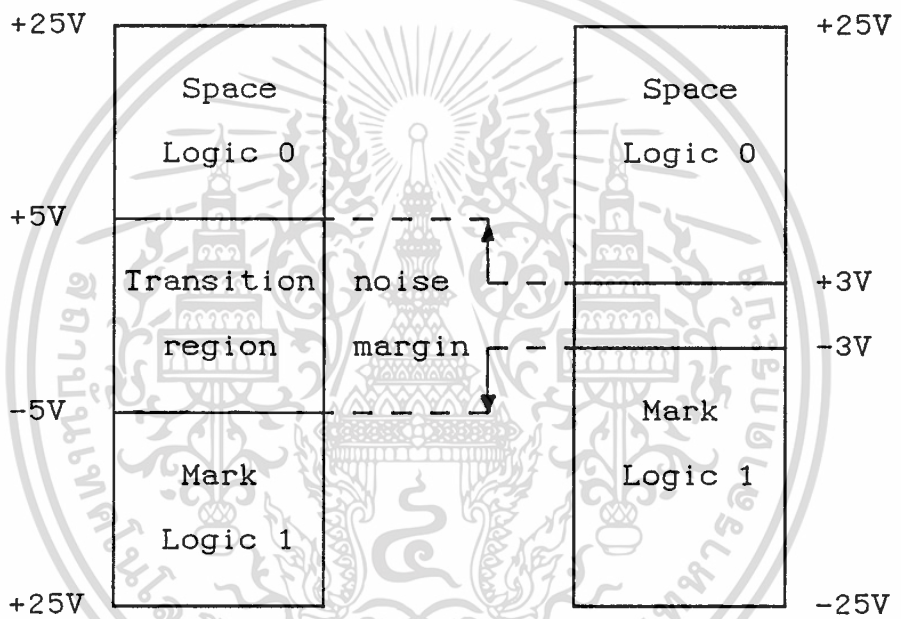
ความสัมพันธ์ระหว่างสถานะของสัญญาณคู่ต่างๆ กับระดับแรงดันได้แสดงไว้ในตารางที่ 2

Status	Signal Voltage	
	$-25V < V_1 < -3V$	$3V < V_1 < 25V$
Binary Logic	1	0
Status		
Signal Condition	Mark	Space
Function	OFF	ON

ตารางที่ 2 แสดงความสัมพันธ์ระหว่างสถานะของสัญญาณกับแรงดัน

ช่วงของระดับแรงดันระหว่าง -3 ถึง +3 โวลต์ จะเป็นช่วงของการเปลี่ยนลอจิก ดังนั้นจึงไม่มีการระบุสถานะของสัญญาณในช่วงนี้ ในการแทนลอจิก 1 หรือสถานะ mark ตัวขับสัญญาณ(driver) ต้องจ่ายแรงดันระหว่าง-5 ถึง -25 โวลต์ ส่วนในการแทนลอจิก 0 หรือ space ตัวขับสัญญาณต้องจ่ายแรงดันระหว่าง +5 ถึง +25 โวลต์

RS-232C ยอมให้มี นอยส์ มาร์จิ้น(noise margin)ได้ไม่เกิน 2 โวลต์ สำหรับความสัมพันธ์ระหว่างระดับแรงดันและสถานะได้แสดงไว้ในรูปที่



รูปที่ 2.5 คุณสมบัติทางไฟฟ้าของการอินเตอร์เฟส แบบ RS-232C

จากรูปเห็นได้ว่าถ้า ไลน์ ไดรฟ์เวอร์ (line driver) หรือตัวกำเนิดสัญญาณต้องการส่งลอจิก 0 ไลน์ ไดรฟ์เวอร์ จะต้องจ่ายแรงดันระหว่าง+5 ถึง +25 โวลต์ ส่วน ไลน์ รีซีฟเวอร์(line receive) หรือตัวรับสัญญาณปลายทาง จะถือว่าแรงดันที่อยู่ในช่วง+3 ถึง +25 แทนลอจิก 0 จากการเปรียบเทียบระดับสัญญาณของตัวส่งและตัวรับจะเห็นว่า RS-232C ยอมให้มีการครอป (drop) ของสัญญาณในช่วง 2 โวลต์เกิดขึ้นได้ สำหรับในด้านการส่งลอจิก1ก็เป็นเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า จึงกล่าวได้ว่า สัญญาณทางไฟฟ้า หมายถึง การรวมข้อมูลเกี่ยวกับการจ่ายไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันและกระแสของแต่ละขาสัญญาณด้วย ซึ่งถ้าเกิดการลัดวงจรกันระหว่างขาขึ้นไปแล้วจะต้องไม่ทำให้อุปกรณ์เกิดการเสียหาย (ไม่ได้หมายความว่าอุปกรณ์นั้นจะต้องไม่พัง) สำหรับส่วนของหน้าที่ซึ่งอาจเป็นส่วนที่สำคัญที่สุดก็ได้ นั้นเป็นส่วนกำหนดลำดับของสัญญาณและการตอบสนองการทำงานของ DTE และ DCE

2.5 การติดต่อสื่อสารภายใต้การควบคุมของซอฟต์แวร์

ไมโครโปรเซสเซอร์ ตัวหนึ่งที่เรียกว่า ยูนิเวอร์ซอล อะซิงโครนัส รีซีฟเวอร์ ทรานสมิตชัน (UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER หรือ UART) ซึ่งอยู่บนอะซิงโครนัส คอมมูนิเคชัน อะแดปเตอร์ (ASYNCHRONOUS COMMUNICATION ADAPTER) สามารถส่งข้อมูลผ่านเข้า-ออกผ่าน RS-232C ได้ ภายใต้การควบคุมของซอฟต์แวร์ 8250 อะซิงโครนัส ไมโครโปรเซสเซอร์ (ASYNCHRONOUS MICROPROCESSOR) เป็น UART ที่ติดตั้งใน IBM PC COMPATIBLE การใช้งานไมโครโปรเซสเซอร์ตัวนี้ ทำได้โดยการใช้คำสั่งในภาษาใดก็ได้ที่เกี่ยวกับ อินพุท/เอาต์พุท พอร์ต (I/O PORT) ในโครงงานนี้จะใช้ภาษาซี สำหรับการส่งข้อมูลที่มีความหมายต่อ 8250-USRT เมื่อรับข้อมูลที่ส่งมาได้ ก็จะตีความออกเป็นคำสั่ง แล้วไปดำเนินการตามกระบวนการที่ได้กำหนดไว้แล้ว เมื่อต้องการส่งข้อมูล บิตแรก จะต้องทำการตรวจสอบสถานะของ 8250-UART โดยการดูว่าใน ทรานสมิตเตอร์ โฮลดิ้ง รีจิสเตอร์ (TRANSMITTER HOLDING REGISTER) ว่างหรือไม่ถ้าว่างก็สามารถส่งข้อมูลออกไปได้ เมื่อข้อมูลอยู่ใน ทรานสมิตเตอร์ โฮลดิ้ง รีจิสเตอร์ เรียบร้อยแล้ว 8250-UART จะย้ายข้อมูลไปไว้ที่ ทรานสมิตเตอร์ ชิฟ รีจิสเตอร์ หลังจากข้อมูลก่อนหน้านี้ออกไปแล้วซึ่งขณะนี้ ทรานสมิตเตอร์ โฮลดิ้ง รีจิสเตอร์ จะว่างพร้อมจะส่งข้อมูลต่อไปจากคอมพิวเตอร์ในกรณีที่มีขนาดข้อมูลมีน้อยกว่า 8 บิต จะทำการส่งเฉพาะบิตค่าเท่านั้น

เมื่อมีการส่งข้อมูลออกไป 8250-UART จะเพิ่มบิตเริ่มต้น, บิตสูงสุด และพาริตีบิตเข้าไปด้วย เมื่อ 8250-UART รับข้อมูลครบทุกบิตแล้ว 8250-UART จะนำข้อมูลนั้นไปเก็บไว้ที่ รีซีฟเวอร์ ชิฟ รีจิสเตอร์ (RECEIVER SHIFT REGISTER) หลังจากตรวจสอบบิตและตัดบิตเริ่มต้นและบิตสูงสุดแล้ว ข้อมูลจริงจะถูกเก็บไว้ที่ รีซีฟเวอร์ ดาต้า รีจิสเตอร์ (RECEIVER DATA REGISTER) ขณะนี้ 8250-UART

จะอยู่ในสถานะรับข้อมูลเรียบร้อยแล้ว เมื่อข้อมูลถูกคอมพิวเตอร์อ่านไป สถานะจะเปลี่ยนเป็นตรงกันข้ามซึ่งแสดงว่าพร้อมรับข้อมูลต่อไป

2.6 โครงสร้างของ 8250

ไอซี 8250 เป็นไอซีที่ออกแบบมาเพื่องานสื่อสารข้อมูลโดยเฉพาะและใช้ในการสื่อสารข้อมูลแบบ อะซิงโครนัส สามารถโปรแกรมให้รับส่งข้อมูล ในอัตราบอดเรทต่างๆได้อย่างง่ายและสะดวก สามารถโปรแกรมจำนวน สตอปบิท, พาริตีบิท, ความยาวของคาตาบิท เป็น 5, 6, 7, 8 บิทได้ นอกจากนี้ยังสามารถตรวจสอบความผิดพลาดของข้อมูลที่ได้จากการรับหรือส่งข้อมูลได้

8250 มีรีจิสเตอร์ภายในที่สำคัญอยู่ 8 รีจิสเตอร์ที่ใช้ในการควบคุมการทำงานทั้งหมดของการสื่อสารข้อมูล

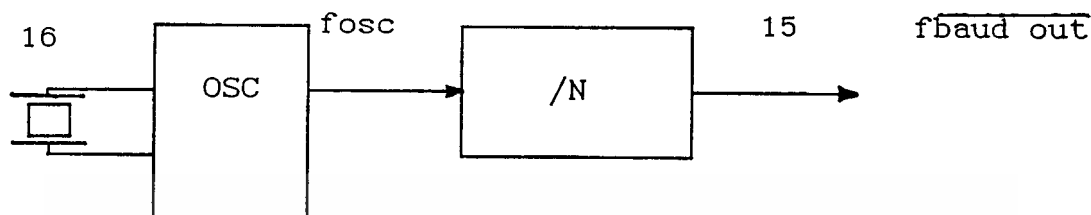
- R0 - RBR, THR, DLL
- R1 - Interrupt Enable Register, DLM
- R2 - Interrupt Identification Register
- R3 - Line Control Register
- R4 - Modem Control Register
- R5 - Line Status Register
- R6 - Modem Control Register
- R7 - Scratch Register

RBR = Receiver Buffer Register DLL = Divisor Latch(LSB)

THR = Transmitter Holding Register DLM = Divisor Latch(MSB)

2.7 การโปรแกรม บอร์ดเรท

8250 สามารถโปรแกรม บอร์ดเรท ให้ค่าต่างๆที่ต้องการได้ โดยมีลำดับการโปรแกรมดังนี้



1. ความถี่บอร์ดเอาท์ ที่ขา 15 ของ 8250 จะมีค่าเป็น 16 เท่าของอัตราบอร์ดเรทที่ต้องการ

$$f_{baud\ out} = 16 \text{ บอร์ดเรท}$$

2. ความถี่บอร์ดเอาท์ที่ขา 15 ได้จากความถี่ออสซิลเลเตอร์ที่ขา 16, 17 ผ่านเข้าไปในวงจรหารความถี่ด้วยค่า N (ค่า N สามารถโปรแกรมให้มีค่าใดๆก็ได้ ขนาด 16 บิต) ความถี่ที่ได้จากวงจรหารความถี่นี้เช่นความถี่บอร์ดเอาท์

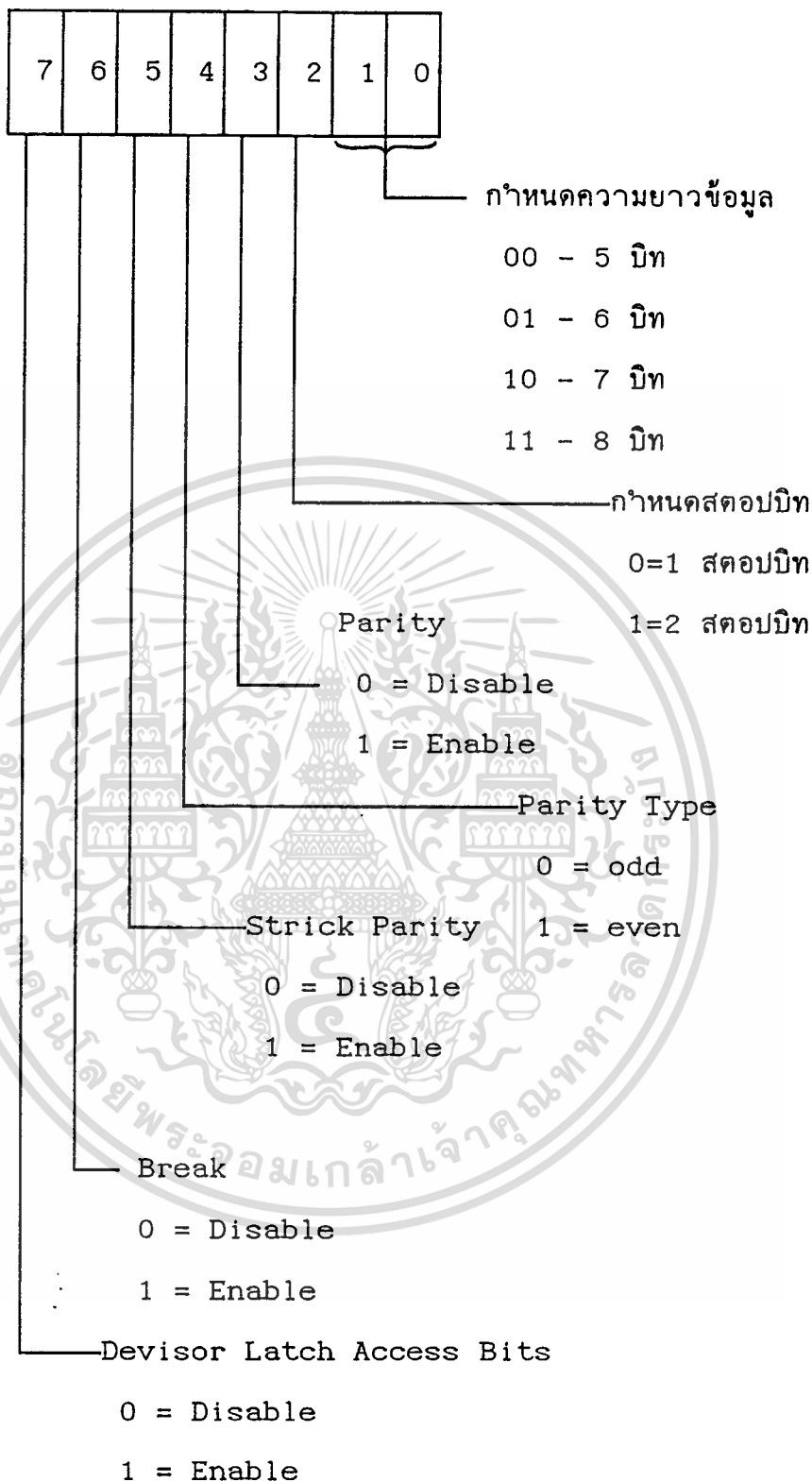
$$f_{osc}/N = f_{baud\ out}$$

$$\text{อัตราบอร์ดเรท} = f_{osc}/16 \times N$$

จะเห็นว่าอัตราบอร์ดเรท สามารถเปลี่ยนแปลงได้ตามค่า f_{osc} และ N

2.8 ไลน์คอนโทรล รีจิสเตอร์ (R3)

สำหรับไลน์คอนโทรลรีจิสเตอร์ (R3) นี้ ใช้เป็นตัวกำหนดจำนวนสตอปบิต, พาริตีบิต, จำนวนความยาวของคาต้าบิต โดยมีโครงสร้างดังนี้



รูปที่ 2.6 แสดงโครงสร้างของไลน์ คอนโทรล รีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 7 ใช้สำหรับควบคุม ดีไวเซอร์แลทซ์รีจิสเตอร์ เพื่อการโปรแกรมตัว
 ทหาร N ถ้าบิตนี้เป็นลอจิก 1 รีจิสเตอร์ $R0$ และ $R1$ จะเป็นดีไวเซอร์แลทซ์
 รีจิสเตอร์ ซึ่งสามารถเขียนค่า N ที่ต้องการไปที่ $R0$ และ $R1$ ได้ ถ้าบิตที่ 7
 เป็นลอจิก 0 รีจิสเตอร์ $R0$ และ $R1$ เป็นสภาพปกติเดิมของไอซี คือ

$R0$ เป็นรีซีฟเวอร์ บัฟเฟอร์ รีจิสเตอร์ ในจังหวะอ่าน หรือเป็น
 ทรานสมิตเตอร์ ในจังหวะเขียน

$R1$ เป็นอินเทอร์รัป ไอเดนทิฟิเคชัน รีจิสเตอร์

บิตที่ 0, 1 ใช้เพื่อกำหนดขนาดความยาวของข้อมูลเป็น 5, 6, 7 หรือ 8 บิต

บิตที่ 2 กำหนดจำนวนสตอปบิต (Stop bit)

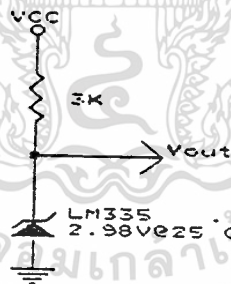
บิตที่ 3 เป็นการตรวจสอบอีนาเบิล หรือ คีสเอเบิล พาริตี

บิตที่ 4 เป็นการกำหนดบิตคู่หรือบิตคี่ (even, odd)

บิตที่ 5 เป็นการกำหนดสตรีก พาริตี (Strick parity)

บิตที่ 6 เป็นการกำหนดเบรค อีเวนท์ (Break event)

โดยพื้นฐานแล้ว LM 335 ก็ทำงานเช่นเดียวกับ ซีเนอร์ ไดโอด (Zener diode) ดังรูปที่ แรงดันพังทลาย ซึ่งหมายถึงแรงดัน เอาท์พุท จากวงจรนี้จะแปรโดยตรงกับอุณหภูมิสมบูรณ์โดยมีค่าเท่ากับ 10 มิลลิโวลต์ ต่ออุณหภูมิที่เพิ่มขึ้น 1 องศาเซลเซียส ในย่านอุณหภูมิที่ออกแบบมาใช้งานค่าของตัวต้านทานในรูปที่จะเป็นตัวกำหนดค่าของกระแสที่ไหลผ่านอุปกรณ์ตัวนี้ แต่เนื่องจาก ค่าไดนามิก อิมพีแดนซ์ที่กระแส 1 มิลลิโวลต์ จะมีค่าโดยปกติ 0.6 โอห์ม อุปกรณ์ตัวนี้จึงสามารถทำงานได้ในย่านกระแสตั้งแต่ 400 ไมโครแอมป์ จนถึง 5 มิลลิแอมป์ โดยเสมือนว่าไม่มีการเปลี่ยนแปลงคุณสมบัติของตัวมันเลยและค่ากระแส รีเวิร์ส (reverse current) หรือ กระแส ฟรอเวิร์ด (Forward current) สูงสุด โดยไม่ทำให้อุปกรณ์เสียหายควรจะเป็น 10 มิลลิโวลต์ ถ้ากระแสสูงกว่านี้จะทำให้เกิดความเสียหายต่ออุปกรณ์ตัวนี้ได้ ที่อุณหภูมิ 25 องศาเซลเซียส และที่ รีเวิร์ส (reverse) = 1 มิลลิโวลต์ แรงดัน เอาท์พุท จากวงจรรูปที่ 2.8 จะมีค่าตามที่ออกแบบไว้เท่ากับ 2.98 โวลต์



รูปที่ 2.8 แสดงลักษณะวงจรของตัววัดอุณหภูมิ

ค่าเปรียบเทียบจุดอ้างอิงของมาตรฐาน

จุดศูนย์สมบูรณ์	=	-273.15 °C	มีค่า	=	0 K
จุดเยือกแข็งของน้ำ	=	0 °C	มีค่า	=	273.15 K

สูตรการเปลี่ยน °C เป็น K

$$\dots\dots\dots K = \dots\dots\dots\text{°C} + 273.15$$

จากคุณสมบัติการเปลี่ยนแปลง 10 มิลลิโวลต์ / °C

ถ้า 0 °C มีค่า = 273.15 K

เทียบค่าเป็นแรงดัน = 273 x 10 (10 : 10 มิลลิโวลต์ / °C)

= 2730 มิลลิโวลต์

= 2.73 โวลต์

เทียบค่า 25 °C เป็นแรงดัน = 25 x 10 (10 : 10 มิลลิโวลต์ / °C)

= 250 มิลลิโวลต์

= 0.25 โวลต์

ทำการเปลี่ยนค่า K เป็น °C

จะได้ 0 °C = 273 K

ถ้า 25 °C = 2.73 + 0.25

ฉะนั้น 25 °C = 2.98 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเป็นเชิงเส้น

จุดเด่นของตัววัดที่เป็น เซมิคอนดักเตอร์ เช่นเบอร์ LM 335 จะให้แรงดัน เอาท์พุท ที่เป็นเชิงเส้นเมื่อเทียบกับอุณหภูมิที่เปลี่ยนแปลงในย่านที่กำหนด ซึ่งไม่เหมือนกับ เอาท์พุท ที่ได้จากตัวตรวจจับอุณหภูมิส่วนใหญ่ ซึ่ง เอาท์พุทจะไม่เป็นเชิงเส้น ซึ่งถ้าเรานำเอาค่าแรงดัน เอาท์พุท มาเขียนเป็นกราฟระหว่างค่าแรงดันกับอุณหภูมิตลอดย่านการทำงานของLM 335 เมื่อลากเส้นไปตัดแกนอุณหภูมิที่ 0 องศาเซลเซียส ค่าแรงดัน เอาท์พุท ที่อ่านได้จากกราฟจะเป็น 0 โวลต์

จากความเป็นเชิงเส้นที่กล่าว จะต้องขึ้นอยู่กับ การปรับความ ถูกต้องของ อุณหภูมิเพียงครั้งเดียว จะให้ความถูกต้องตลอดย่านอุณหภูมิที่ใช้งานนั้น ก็เนื่องมาจาก เอาท์พุท จะแปรผันโดยตรงอย่างเที่ยงตรงกับอุณหภูมิสัมบูรณ์โดยเอาท์พุท จะลดลงเป็น 0 โวลต์ ที่ 0 อุณหภูมิองศาเซลเซียส ดังนั้น การปรับความลาดชันที่ อุณหภูมิค่าหนึ่งให้ถูกต้อง จะทำให้เกิดความถูกต้องตลอดย่านอุณหภูมิ

ความร้อนที่เกิดขึ้นในตัว

เช่นเดียวกับตัวตรวจจับอุณหภูมิใดๆก็ตาม ความร้อนใดๆ ที่เกิดขึ้นจาก กระแสที่ไหลผ่านอุปกรณ์ที่ตัวตรวจจับจะมีผลต่อค่าอุณหภูมิของตัวมันตลอดจนค่าแรงดัน เอาท์พุท ที่เกิดขึ้นสำหรับ LM 335 นั้น ควรจะให้ทำงานที่กระแสต่ำสุด ซึ่งเพียงพอที่จะจับให้วงจรภายในไอซีทำงานได้ โดยกระแสประมาณ 400 มิลลิแอมป์ จะเป็นกระแสต่ำสุดที่ ไอซี จะทำงานได้ตามปกติ (ค่ากระแสนี้จาก space)

ตัวแปลงสัญญาณ

จากตัววัดอุณหภูมิซึ่งใช้ ไอซี LM 335 ซึ่งให้ เอาท์พุทเป็นแรงดันที่เปลี่ยนแปลงค่าตามอุณหภูมิที่เปลี่ยนไป และแรงดันที่ได้นี้เป็นสัญญาณแบบ อนาลอก ฉะนั้น จำเป็นอย่างยิ่งที่จะต้องทำการแปลงค่า แรงดันซึ่งเป็น อนาลอก ให้เป็นแบบ ดิจิตอล เพื่อให้ค่าที่เป็นไบนารี เป็นค่าข้อมูล(data)ส่งค่าทางสาย ข้อมูล(data bus) ให้กับไอซี เบอร์ MC 14469 เพื่อให้ ไอซี ดังกล่าวส่งค่าอุณหภูมิที่วัดได้ให้กับ CPU แสดงผลออกหน้าจอในการแปลงสัญญาณ อนาลอกเป็นสัญญาณ ดิจิตอล นี้ เราใช้ ไอซี เบอร์ ADC 0804 เป็นตัวแปลงสัญญาณซึ่ง ADC 0804 จะมี 1 เอาท์พุท ที่รับสัญญาณ อนาลอก และ เอาท์พุท เป็น ดิจิตอล (เป็นค่า ไบนารี) ขนาด 8 บิต แต่เราต้องการใช้เพียง 5 บิต ดังนั้นช่วงอุณหภูมิที่วัดได้คือ 32 ระดับ ตั้งแต่ 1 องศาเซลเซียส ถึง 64 องศาเซลเซียส

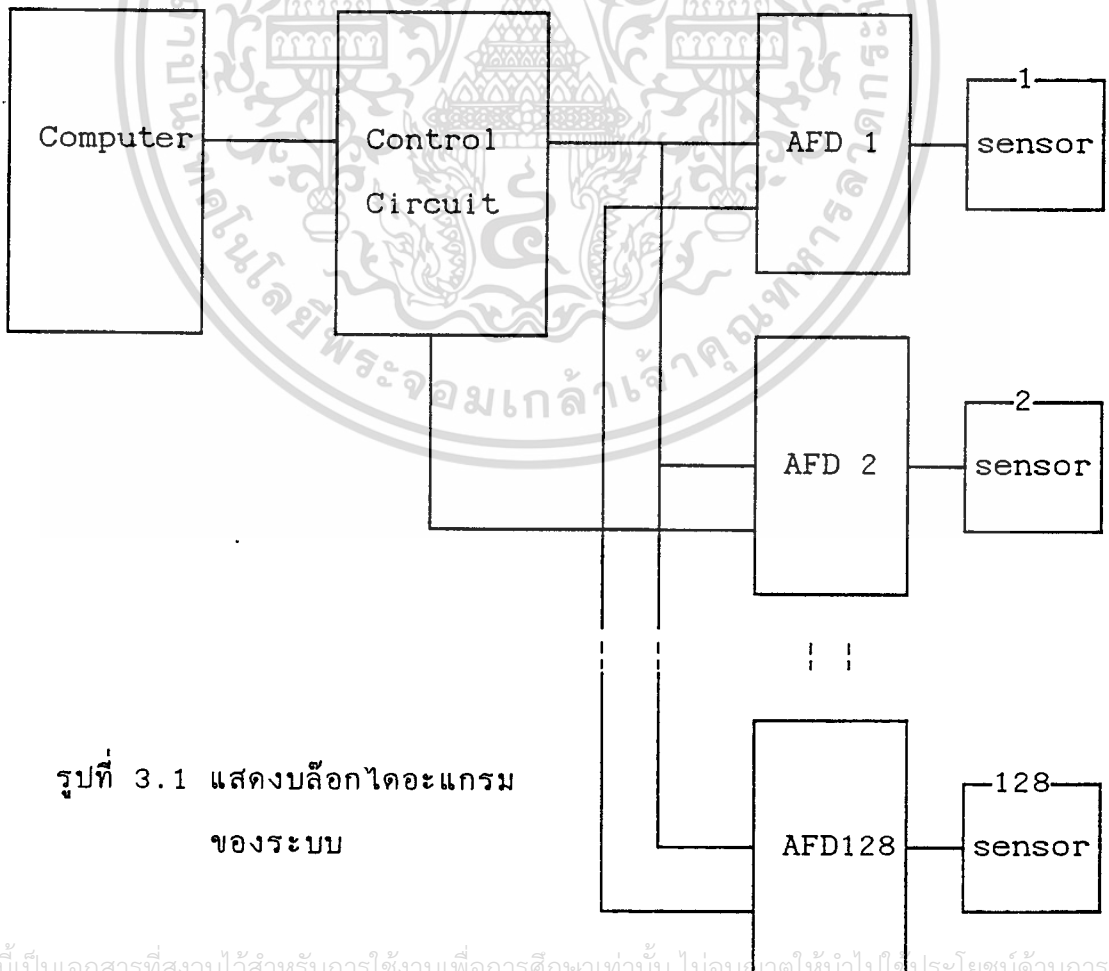
บทที่ 3

การออกแบบระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้

3.1 บล็อกโคอะแกรมของระบบ

ระบบส่งสัญญาณแจ้งเหตุเพลิงไหม้ ในโครงการนี้ประกอบด้วยส่วนหลัก คือ ส่วนของโปรแกรมซอฟต์แวร์ จากเครื่องคอมพิวเตอร์ ใช้ในการควบคุมการรับส่งข้อมูลและแสดงผลทางหน้าจอคอมพิวเตอร์ ส่วนที่สองคือ วงจรส่วนควบคุมใช้ในการแปลงระดับสัญญาณ เนื่องจากในการติดต่อกับ IBM-PC ซึ่งใช้การเชื่อมต่อตามมาตรฐาน RS 232 C ซึ่งมีระดับสัญญาณเฉพาะตัว จึงต้องใช้วงจรแปลงระดับสัญญาณให้เหมาะสมสำหรับติดต่อกับอุปกรณ์ปลายทางได้

ส่วนของวงจร Addressable Fire Detector (AFD) เป็นส่วนที่สำคัญ เป็นส่วนที่ทำการตรวจจับและส่งข้อมูลแจ้งไปยัง CPU ซึ่งจุดเด่นของ AFD คือมีตำแหน่งที่อยู่ (Address) ประจำของตนเอง และส่วนของตัวตรวจจับ (หรือส่วน SENSOR) โดยแสดงบล็อกโคอะแกรมของระบบ ดังรูปที่ 3.1



รูปที่ 3.1 แสดงบล็อกโคอะแกรม
ของระบบ

3.2 การออกแบบส่วนของฮาร์ดแวร์(hardware)

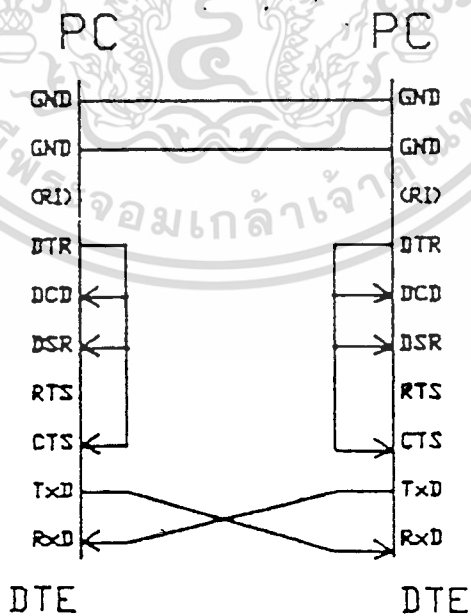
3.2.1 ส่วนการแปลงระดับสัญญาณโวลท์ เคจ(voltage)

และสวิชชิง(switching)

เริ่มแรก ต่อพอร์ตอนุกรม(serial port) ที่ com 1 ซึ่งเป็นพอร์ต(port) ที่มี 25ขา(pin)โดย pin ที่ใช้มีดังนี้

- pin 1 : GND (system ground)
- pin 2 : TxD (transmit)
- pin 3 : RxD (receive)
- pin 5 : CTS (clear to send)
- pin 6 : DSR (data set ready)
- pin 7 : GND (signal ground)
- pin 8 : DCD (data carrier detected)
- pin 20 : DTR (data terminal ready)

โดยต่อแบบนัลโมเด็ม(null modem)ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ 3.2 แสดงการต่อนัลโมเด็ม(null modem) ระเบียบข้อดำเนินการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อเพื่อให้ DTE เข้าใจว่ามีโมเด็ม (modem) ต่อที่อีกด้านหนึ่ง และ DTE จะได้ส่งอย่างสบายใจ หลังจากข้อมูลถูกส่งจาก ขา(pin) TxD (pin 2) จะถูกนำไปแปลงระดับโวลต์เตท (voltage) เพราะลอจิก (logic) " 1 " ในพอร์ตอนุกรม (serial port) ส่งมานั้นไม่เหมือนกับลอจิก(logic)" 1 " ในระดับ TTL จึงต้องมีการแปลงโดย ไอซี (IC)เบอร์ 1489 หลังจากแปลงแล้วผ่านวงจร สวิตซ์ซึ่ง (switching) โดยใช้หลักการที่ว่าหากเป็นลอจิก (logic) " 1 " จะทำให้โวลต์เตจ (voltage) ที่ขา BASEของ TIP 107 มีค่ามากกว่า 0.6โวลต์(volt)ซึ่งทำงานได้ ยังผลให้ส่งโวลต์เตจ(voltage) มีค่าเท่ากับ 12 โวลต์(v.) ผ่านสายส่งเหมือนกับสภาวะ idle ก็ส่ง 12 v.ไป เช่นกัน แต่หากระดับลอจิก (logic)" 0 " จากพอร์ตอนุกรม(serial port) ก็จะทำให้ขึ้นตอนเหมือนกันทุกประการ ทำให้ส่งโวลต์เตจ (voltage) ที่มีค่าเท่ากับ 0โวลต์(v.) หากมองด้านรับข้อมูลจาก ซึ่ง AFD จะส่งระดับ โวลต์เตจ (voltage) มา 2 ระดับ คือ 5โวลต์(v.) กับ 0 โวลต์ (v.) ผ่านสายส่ง ที่มีทั้งสิ้น 3 เส้น คือ TxD , RxD และ GND พบว่า ต้องแปลง ระดับโวลต์เตจ voltให้เป็นลอจิก(logic) " 1 " ซึ่งพอร์ตอนุกรม(serial port) เข้าใจว่าเป็นลอจิก(logic) " 1 " ดังนี้

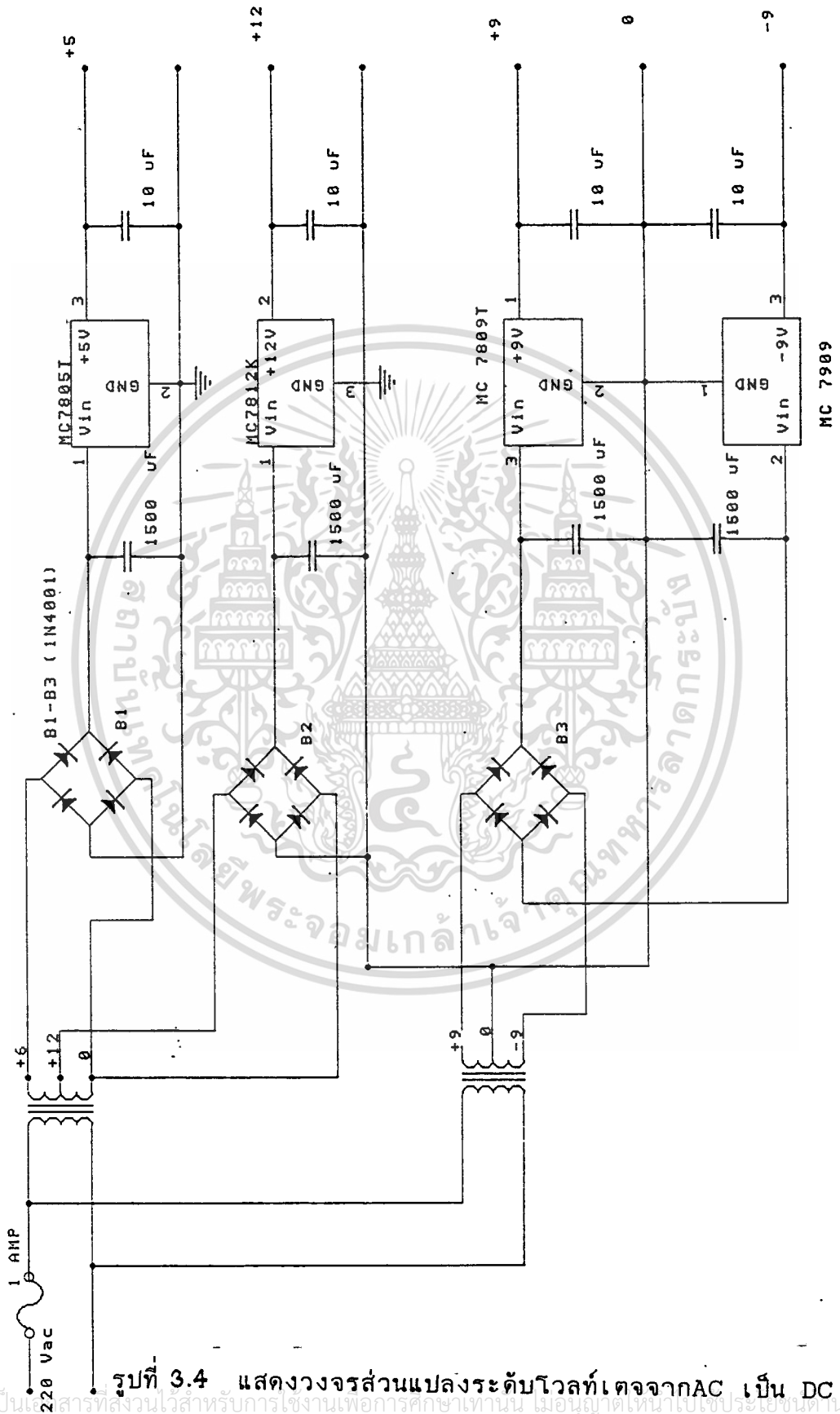
ลอจิก(logic) " 1 " คือ 5โวลต์(v.) แปลงเป็น -3 ถึง-25 v.

ลอจิก(logic) " 0 " คือ 0โวลต์(v.) แปลงเป็น +3 ถึง+15 v.

โดยผ่าน ไอซี (IC) เบอร์ 1488 ซึ่ง outputของ (IC) ไอซีจะถูกต่อเข้ากับ พอร์ตอนุกรม (serial port)

-หมายเหตุ ไฟเลี้ยง ของ ไอซี เบอร์ 1488 คือ +9 v. กับ -9 v.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไฟเลี้ยง ของ ไอซี เบอร์ 1489 คือ +5 v.
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงวงจรส่วนแปลงระดับโวลต์เตจจาก AC เป็น DC

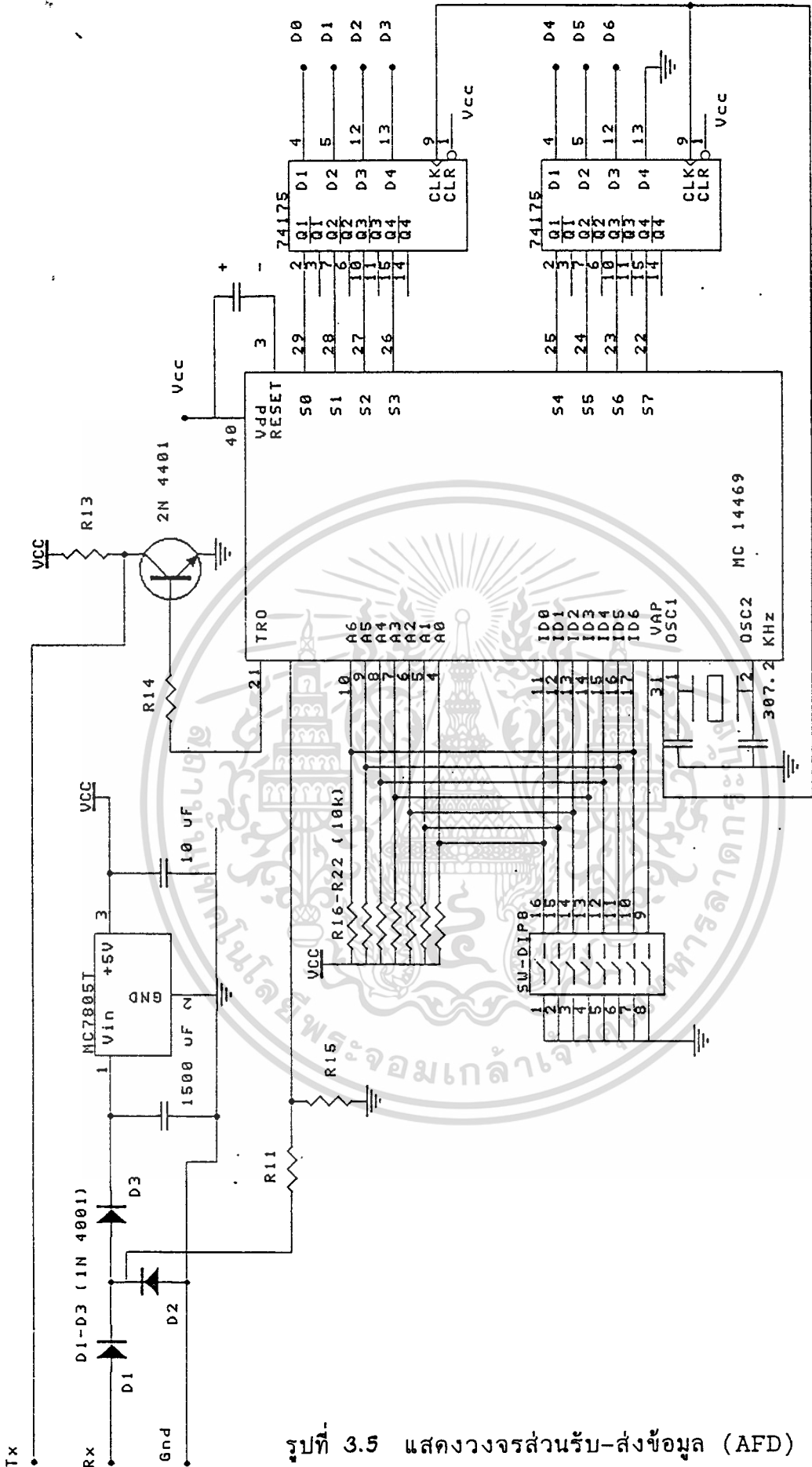
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนแอดแตรสเชเบิล ไฟร์ ดีเทคเตอร์ (AFD) (Addressable Fire Detector)

เป็นตัวรับ-ส่งข้อมูล ซึ่งตัว AFD สามารถตั้งแอดแตรสได้ มีส่วนประกอบดังนี้

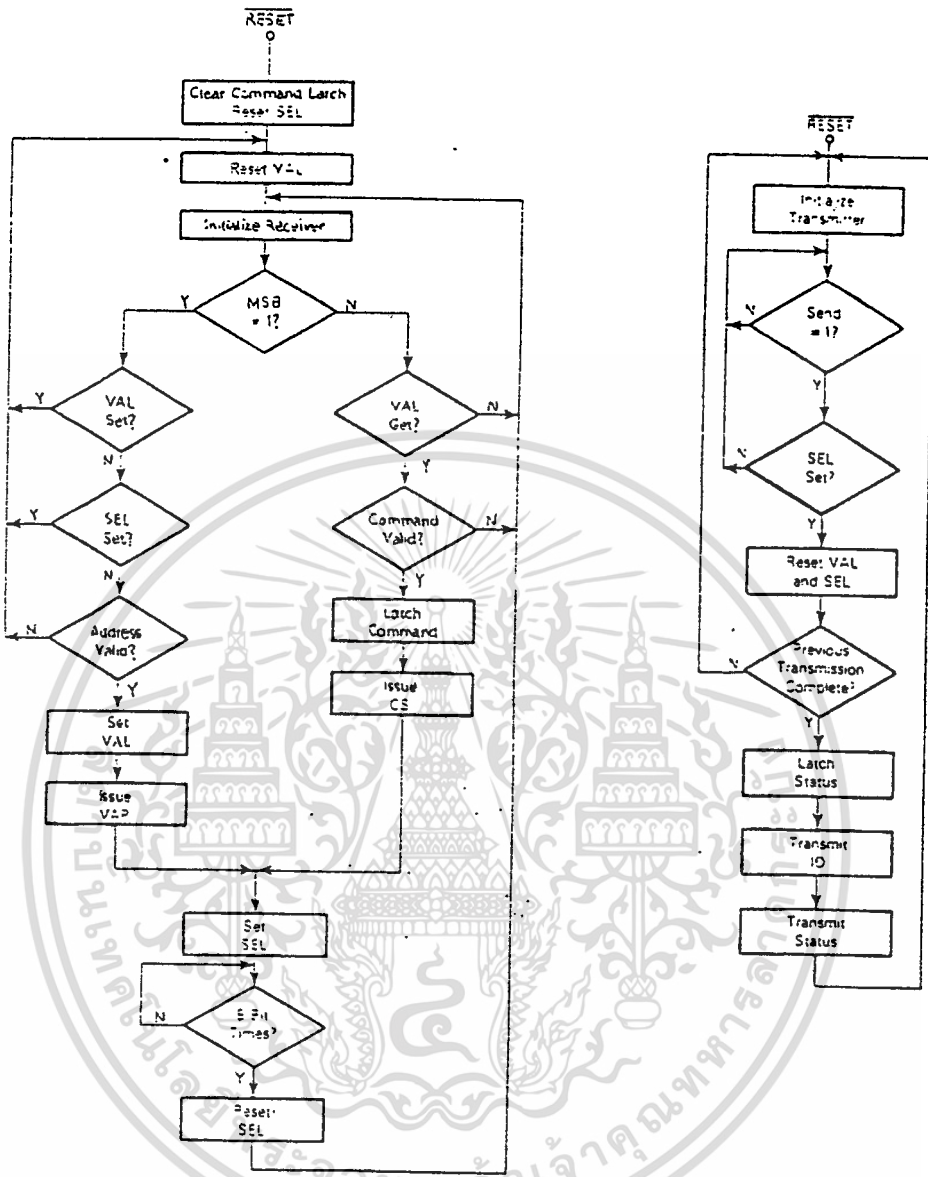
1. ไอซี(IC) เบอร์ 14469
2. ตัวแปลงระดับโวลท์เตจ(voltage) ให้คงที่ (regulator)
3. ตัวพักข้อมูล (latch)

จะขออธิบายตรงที่ ไอซี(IC)เบอร์ 14469 ก่อนเป็นไอซี(IC)ประเภท CMOS ซึ่งเป็นไอซี(IC)ที่กระแสมากที่สุดประมาณ 75 ไมโครแอมป์ ที่ 25 องศาเซลเซียส นับว่ากินกระแส น้อย แล้วยังมีคุณสมบัติที่น่าสนใจคือสามารถทำเป็นฟูลดูเพล็กซ์ (FULL DUPLEX) ซึ่งการทดลองครั้งนี้ก็ทำเป็นแบบฟูลดูเพล็กซ์(FULL DUPLEX) ด้วยส่วนขั้นตอนการทำงานต้องดูที่โฟลชาร์ท(FLOW CHART)และไซเคิล(CYCLE) การทำงาน ของไอซี MC 14469



รูปที่ 3.5 แสดงวงจรส่วนรับ-ส่งข้อมูล (AFD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

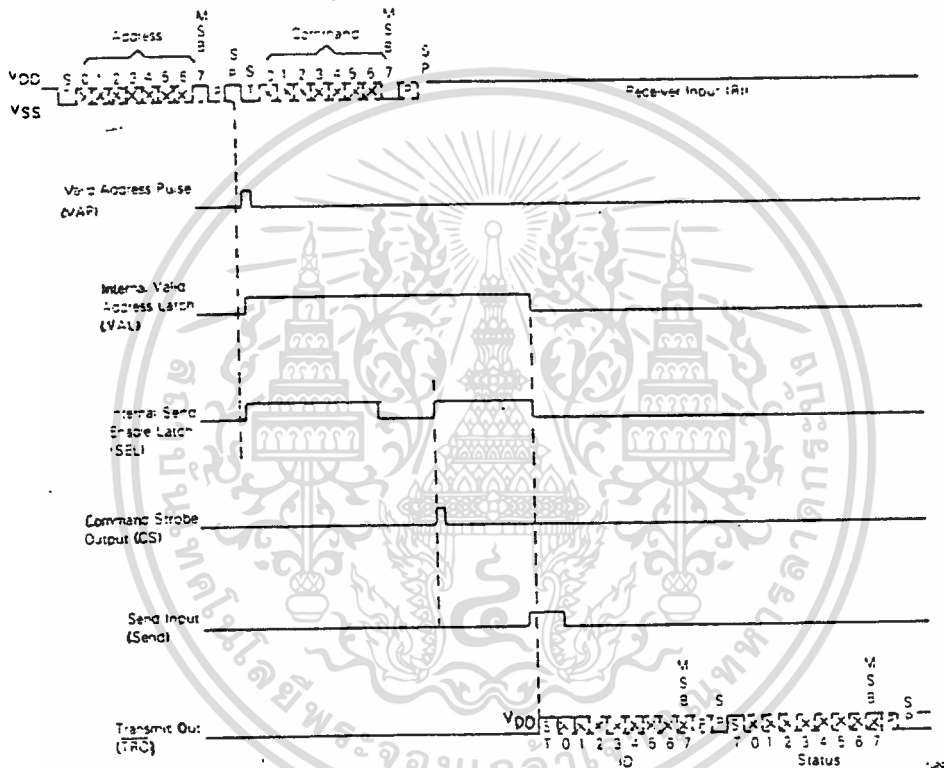


รูปที่ 3.6 แสดงโพลซาร์ทการทำงานของไอซี MC 14469

พบว่าหากตัว AFD รับข้อมูลจาก คอมพิวเตอร์มันจะตรวจว่าข้อมูลซึ่งเป็น 1 เวิร์ด มี 2 ไบท์ โดยมันจะเช็ค ไบท์ แรก แล้วนำไปเช็คกับแอดแตรสที่ต้งไว้หากตรง จะมีการทริก(trig)ขา VAP ดูรูปขั้นตอน การทำงาน ขา VAP จะไปรีเซ็ต ที่ D ฟลิปฟลอปทำให้เกิดการถ่ายข้อมูลไปยังที่ไอซี 14469 ที่ ขา S2- S7 ทั้งนี้ ขั้นตอนต่อไปคือตัวมันส่งค่า ID (identification) เป็น byte แรก จะตาม ด้วย ค่า S0- S7 เป็น ไบท์ ที่สองดังรูป การส่งข้อมูลนั้นผ่าน ขาที่ 21 ผ่าน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยได้

เราพบว่าการทำงานของตัว AFD จะสัมพันธ์ได้นั้นจะต้องอาศัยส่วนแลทช์ (latch) เป็น D ฟลิปฟลอปและอีกตัวคือ เรกคูลเรเตอร์ (regulator) โดยอาศัย อินพุต จาก สายส่งที่เป็น 12 โวลต์ แล้วนำมาผ่าน 780 ทำให้ระดับ โวลต์เตจ คงที่ ที่ 5 โวลต์ป้อนไปสู่ขาไฟเลี้ยงของไอซี 14469



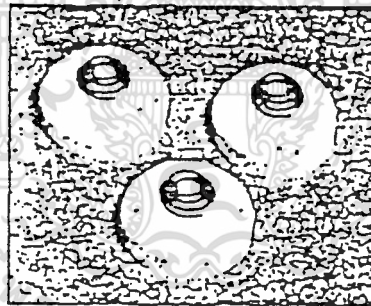
รูปที่ 3.7 แสดงไขเคลการทำงานของไอซี MC 14469

3.2.3 ส่วนตรวจจับ (Sensor)

ส่วน SENSOR ซึ่งประกอบด้วย

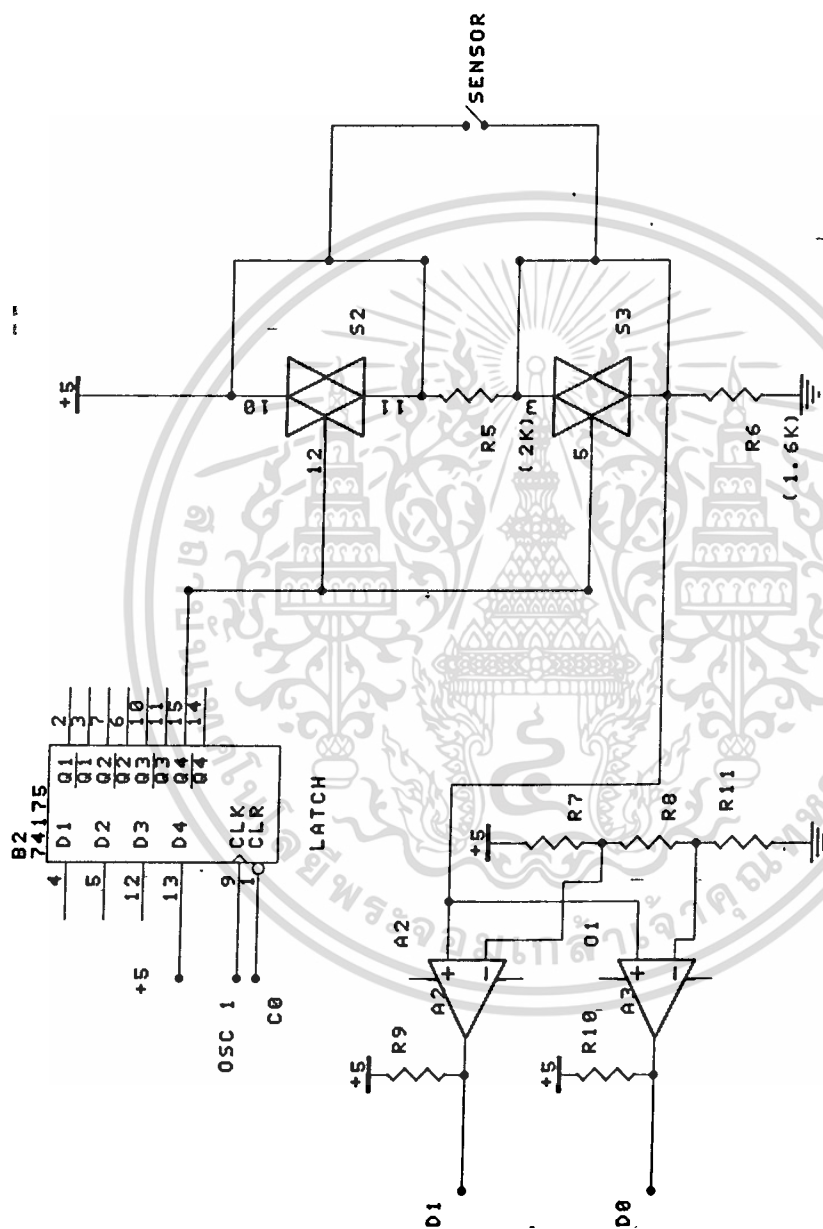
1. SENSOR ตรวจจับความร้อน

ปัจจุบันตัวตรวจจับความร้อน ที่มีขายในท้องตลาดมีอยู่ 2 หลักการด้วยกัน คือ แบบแรกจะเป็นตัวตรวจจับความร้อนแบบอุณหภูมิคงที่ (Fixed temperature - heat sensor) ตั้งในรูป ข้างล่าง ตัวตรวจจับความร้อนแบบนี้ในสภาวะปกติ สวิตช์ที่อาศัยผลของความร้อนสัมผัส เข้าหากันทำให้เกิดการลัดวงจรขึ้นตัวอย่างใน รูปสวิตช์จะลัดวงจรเมื่ออุณหภูมิสูงขึ้น 60 องศาเซนติเกรด โดยมีความผิดพลาด บวกลบไม่เกิน 5 องศาเซนติเกรด การติดตั้งควรจะติดตั้งไว้ตามฝ้าเพดานใน ตำแหน่งที่คิดว่าจะตรวจจับการเกิดไฟไหม้ได้ดีที่สุด พื้นที่ในการตรวจจับต่อตัวโดย ทั่วไป ประมาณไม่เกิน 900 ตารางฟุต (ตามที่ระบุใน UL list no. 746H.)



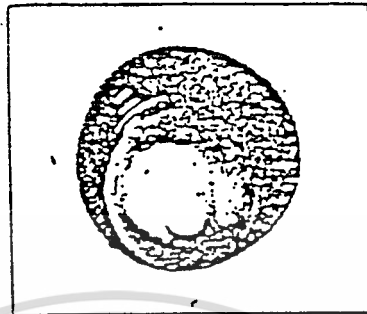
รูปที่ 3.8 แสดงตัวตรวจจับความร้อนแบบอุณหภูมิคงที่

แบบที่สองของตัวตรวจจับความร้อนนี้ เป็นแบบตรวจจับทั้งอุณหภูมิคงที่เหมือน แบบแรกรวมกับการตรวจจับอัตราการเพิ่มของอุณหภูมิ กล่าวคือ ถ้าอัตราการเพิ่ม ของอุณหภูมิรวดเร็วถึงขนาด 5 องศาฟาเรนไฮต์ต่อ 20 วินาที แล้วตัวตรวจจับจะ ทำงานทันที การติดตั้งก็เช่นเดียวกับแบบแรกคือ ติดใต้ฝ้าเพดานโดยหันหัวจุกลง ตัวตรวจจับมาตรฐานแบบนี้จะครอบคลุมพื้นที่ประมาณ 2,500 ตารางฟุตได้ (ตาม ที่ระบุใน (UL listed no. 759G.) ดูดังรูป



รูปที่ 3.9 วงจรส่วนตรวจสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงลักษณะของตัวตรวจจับอุณหภูมิ

2. SENSOR ตรวจจับอุณหภูมิ

โดยในการวัดค่าอุณหภูมิ จะใช้ LM 335 z ซึ่งมีคุณสมบัติ ทนความร้อนได้ถึง 70 องศาเซนติเกรด หากเกินกว่านี้จะไม่ทำงาน เราจะใช้หลักการที่ โวลต์เตจ (voltage) มี การเปลี่ยนแปลงตาม อุณหภูมิที่เปลี่ยนไป ในวงจรใช้ค่าเท่ากับ

$$V = 1 \text{ mV} / 1 \text{ kalvin}$$

กล่าวคือที่อุณหภูมิ 25 องศาเซลเซียส

$$V = 2.98 \text{ v.}$$

และทำงานที่ อุณหภูมิ 70 °c เป็นค่าสุดท้าย

หลังจากได้ v.ก็จะนำไปผ่าน ADC 0801 ซึ่งเป็นตัวแปลงอนาลอก (analog) ไปเป็นดิจิตอล(digital)โดยเอาท์พุท(output)ออกมาเป็นดิจิตอล(digital) 8 บิต(bit)แต่เราจะนำไปใช้ 5บิต(bit) (เป็น 5 MSB)

- หมายเหตุ หากโวลต์เตจ (voltage) เกินกว่าที่กำหนดไว้ คือต้อง ไม่ต่ำกว่า2.73โวลต์(v)ที่ 0 องศาเซลเซียสและต้องไม่เกินกว่า3.37โวลต์(v) ที่ 64 องศาเซลเซียส หากอยู่นอกขอบเขตนี้จะทำให้ ADC 0801ไม่ทำงานเพื่อ

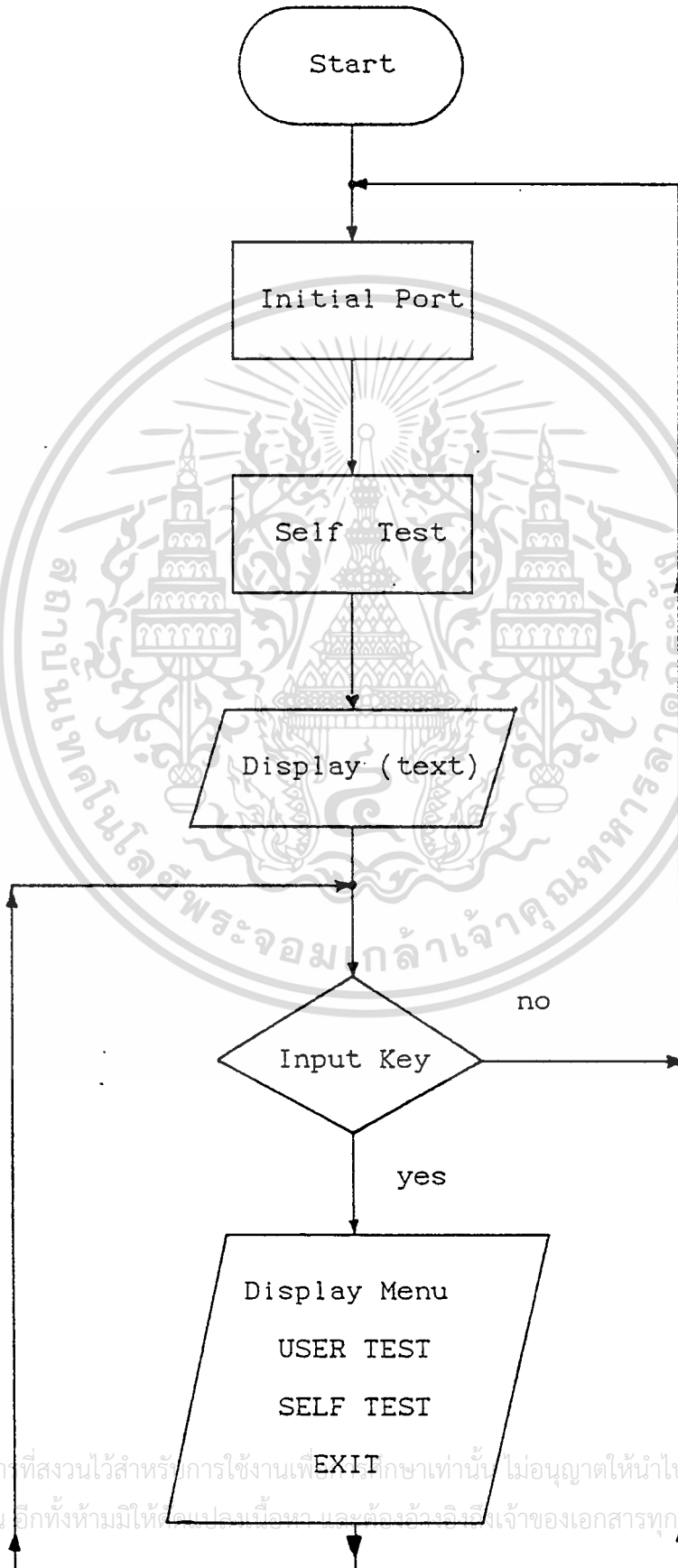
เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับใช้งานเพิ่มความรู้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า เป็นการป้องกัน ไม่ให้ทำงานขณะที่ร้อนด้วย ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

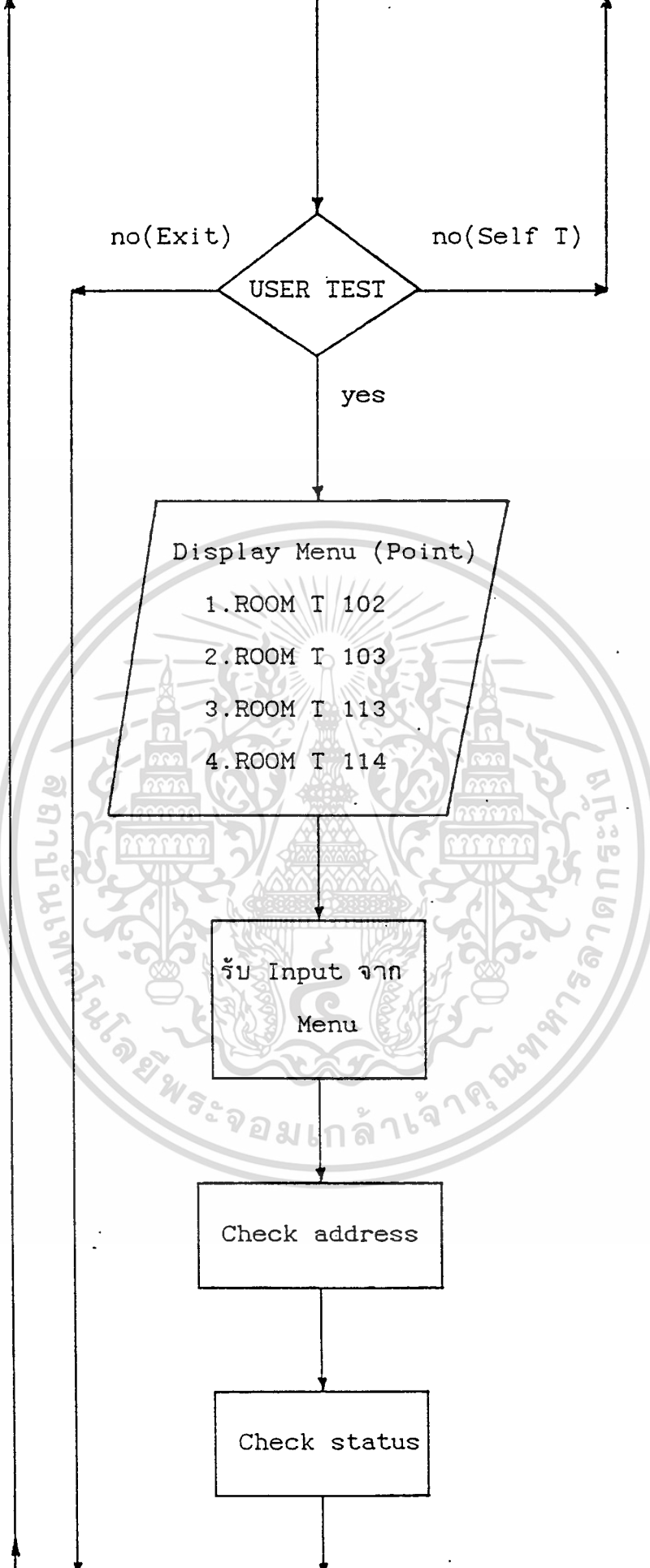
3.3 การออกแบบส่วนของซอฟต์แวร์ (Software)

ในการออกแบบซอฟต์แวร์นี้จะใช้ควบคุมการติดต่อสื่อสารแบบอนุกรม ส่งข้อมูลแบบ Asynchronous ซึ่งจะต้องมีการอินิเชียล(Initial)พอร์ตสื่อสารก่อน โดยโปรแกรมค่าต่างๆลงในรีจิสเตอร์ควบคุมสายสื่อสาร (Line Control Register) โดยกำหนดให้ มีการส่งข้อมูลขนาด 8 บิต , ไม่มีพาริตีบิต สามารถส่งข้อมูลด้วยความเร็ว 4800 บิตต่อวินาที สำหรับภาษาที่ใช้จะใช้ ภาษาซี

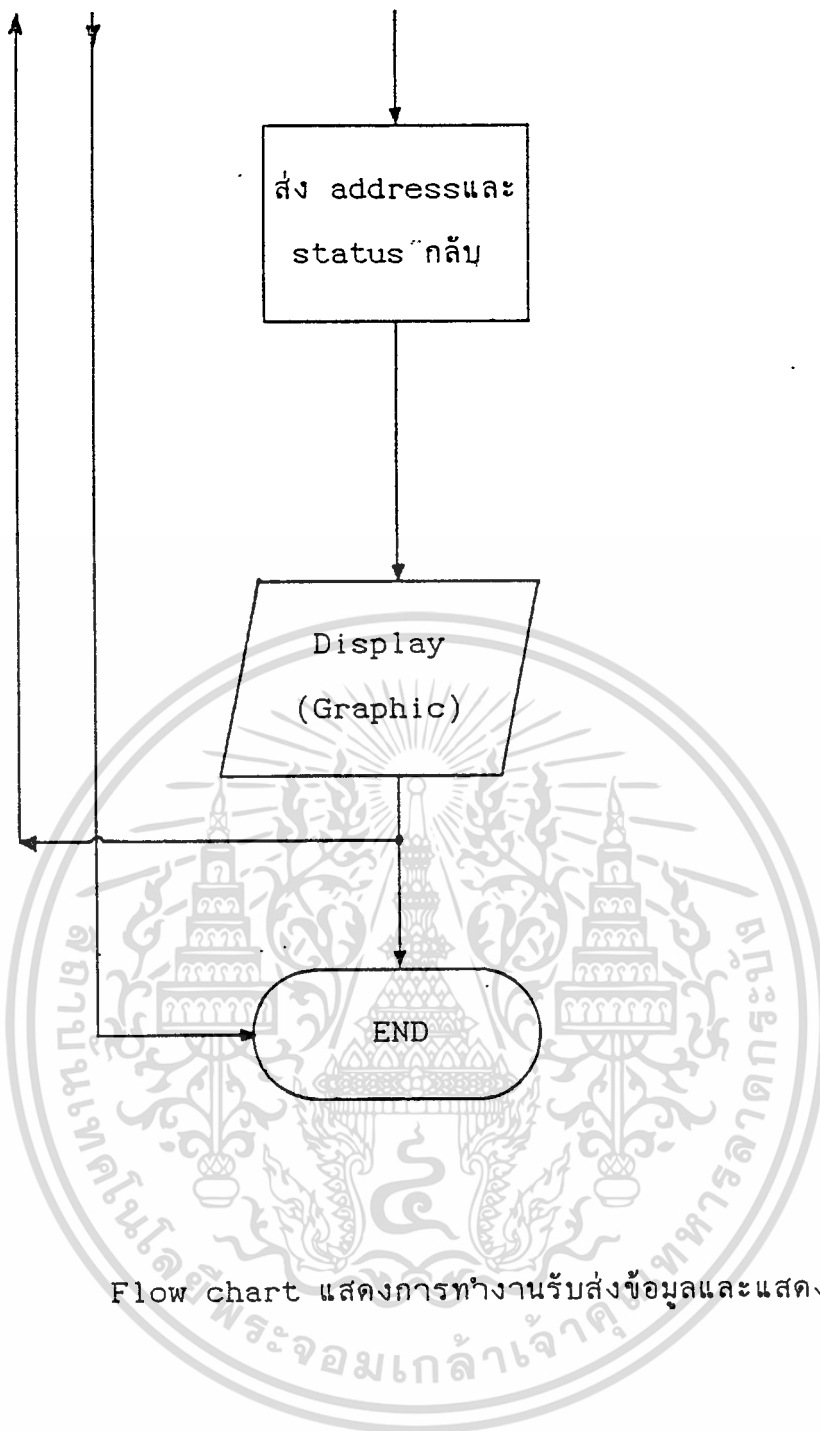
สำหรับการแสดงผล ทางหน้าจอคอมพิวเตอร์จะมีแสดงทั้ง โหมดตัวอักษร (Text Mode) และ กราฟฟิกโหมด (Graphic Mode) โดยในการทำงานรับส่งข้อมูลและแจ้งผลนี้ สามารถเรียกใช้ได้ แบบ โพลลิ่ง (Polling) คือมีการส่งค่าแอดเดรส (address) ซึ่งมาจากการสั่งงานจากซอฟต์แวร์ ว่าจะให้เป็นแบบตรวจสอบตัวเอง(Self Test)หรือ ตรวจสอบตามคำสั่งของผู้ใช้(User Test) เมื่อแอดเดรสส่งผ่านพอร์ตอนุกรมตามมาตรฐาน RS-232 แล้ว อุปกรณ์ AFD ก็จะตรวจสอบว่าแอดเดรสที่ส่งมานั้นตรงกับ ตัวมันเองหรือไม่ ถ้าถูกต้อง AFD ตัวนั้น จะทำการติดต่อกับส่วนเซนเซอร์ ให้ตรวจจับสถานะและอุณหภูมิของตัวมัน แล้วส่งข้อมูลกลับมาเพื่อทำการแสดงผลออกทางหน้าจอคอมพิวเตอร์ โดยถ้าเป็นกรณีการตรวจสอบตัวเอง จะแสดงเป็นโหมดตัวอักษร เพราะมันจะทำการวนตรวจสอบอุปกรณ์ทุกตัว ซึ่งสามารถมีได้ถึง 128 ตัว แต่ในโครงงานนี้ เรามีอุปกรณ์เพียงตัวเดียว สำหรับกรณีการตรวจสอบจากผู้ใช้ แสดงผลโดยใช้กราฟฟิกโหมด เพื่อที่จะได้แสดงตำแหน่งจุดที่เกิดเหตุ และแสดงสถานะที่ตรวจสอบได้อย่างชัดเจนโดยจะแสดงเป็นภาพแปลนของอาคารที่ติดตั้งอุปกรณ์นี้ แบ่งเป็นห้องๆชัดเจน สังเกตเห็นง่าย เมื่อส่งข้อมูลของสถานะและตำแหน่งแอดเดรสกลับมา จะมาแสดงผลบนหน้าจอที่มีภาพแปลนอาคารนั้นอย่างถูกต้อง

3.3.1 โฟลว์ชาร์ท (Flow Chart) ของการทำงานรับส่งข้อมูลและแสดงผล สำหรับการออกแบบโปรแกรมที่ใช้ติดต่อรับส่งข้อมูลระหว่าง CPU กับอุปกรณ์ปลายทาง (AFD) และการแสดงผล นั้นมีลักษณะขั้นตอนตาม Flowchart





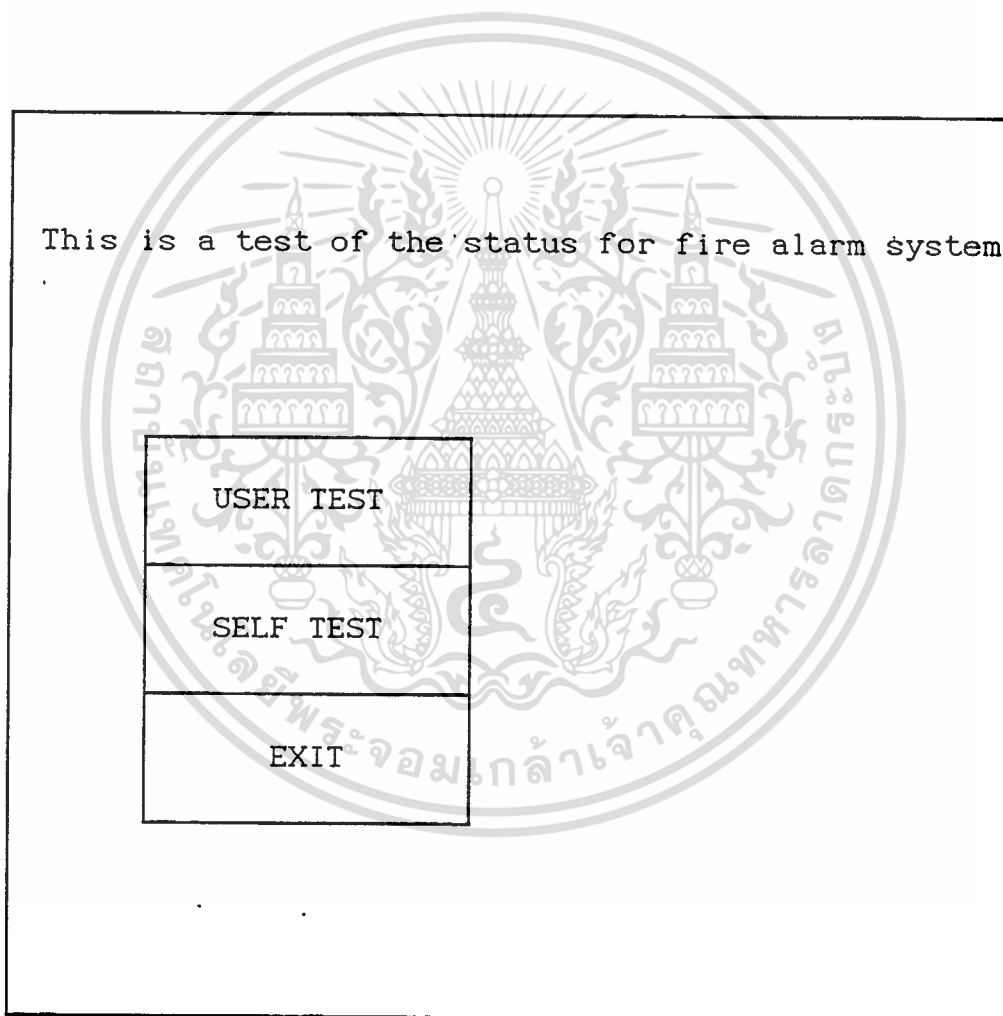
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Flow chart แสดงการทำงานรับส่งข้อมูลและแสดงผล

3.3.3 การแสดงผลทางหน้าจอ

จากที่กล่าวมาแล้วว่า การทำงานของระบบสามารถเลือกได้ว่า ต้องการแบบ ตรวจสอบตัวเอง (Self Test) หรือ ตรวจสอบโดยผู้ใช้ (User Test) ถ้าเป็นแบบตรวจสอบตัวเอง จะแสดงที่หน้าจอเป็นโหมดตัวอักษร (Text Mode) คือ เป็นตัวอักษรแสดงว่า เป็นการตรวจสอบอุปกรณ์ AFD ตัวใดและแสดงสถานะที่ตรวจสอบได้ หลังจากนั้นจึงจะไปตรวจสอบอุปกรณ์ AFD ตัวต่อไปจนกว่าจะมีการกดแป้น (Key) ใดๆ หน้าจอก็จะขึ้น Menu ให้เลือกดังนี้



รูปที่ 3.12 แสดงหน้าจอที่แสดงเมนูเลือกแบบการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีเมนูขึ้นดังรูปแล้ว ผู้ใช้สามารถเลือกว่าต้องการตรวจสอบแบบใดถ้าเลือก แถบของตรวจสอบโดยผู้ใช้เมื่อกดแป้นENTERก็จะปรากฏเมนูสำหรับเลือกว่าต้องการตรวจสอบที่ตำแหน่งใด (ห้องใด) โดยจะสมมติให้มีตัวเลือก 8 ห้อง มีลักษณะหน้าจอดังนี้

This is a test of the status for fire alarm system

USER TEST	
SELF TEST	1.ROOM T 102
EXIT	2.ROOM T 103
	3.ROOM T 106
	4.ROOM T 109
	5.ROOM T 110
	6.ROOM T 111
	7.ROOM T 113
	8.ROOM T 114

เอกสารนี้เป็นเอกสารที่ 3.13 แสดงหน้าจอที่แสดงเมนูสำหรับเลือกจุดที่ต้องการตรวจสอบด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเลือกตำแหน่งที่ต้องการตรวจสอบแล้วเมื่อมีการส่งข้อมูลแอดเดรสของตัว AFD ที่ตำแหน่งที่เลือกนั้น และทำการตรวจสอบสถานะเรียบร้อยแล้วส่งข้อมูลกลับ ที่หน้าจอจะแสดงภาพแปลนของอาคารโดยในโครงงานนี้ ใช้แปลนของตึกโทรคมนาคม โดยใช้โหมดกราฟฟิค สถานะที่ตรวจจับได้จะแสดงตรงตำแหน่ง(ห้อง)ที่ได้เลือกไว้ อย่างชัดเจน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ส่วนแปลงระดับสัญญาณ

จากการทดสอบวงจรส่วนควบคุมระดับสัญญาณ เมื่อทำการป้อนสัญญาณรูปสี่เหลี่ยม (Square wave) ด้วยระดับของ RS-232 C ขนาด -10 ถึง $+10$ โวลต์ เข้าที่ ด้านอินพุทของตัวรีซีพเวอร์ (MC 1489) เมื่อวัดสัญญาณเอาต์พุทที่ ตัวรีซีพเวอร์ นี้จะได้สัญญาณที่วัดได้จากออสซิลโลสโคป เป็นรูปสี่เหลี่ยมขนาด $0-5$ โวลต์ ซึ่งเป็นระดับของ TTL

เมื่อทดสอบป้อนไฟเลี้ยง $+24$ โวลต์ ตามรูปวงจร แล้วจับสัญญาณเอาต์พุท ด้วยออสซิลโลสโคปจะได้สัญญาณรูปสี่เหลี่ยมขนาด $0-23$ โวลต์ ที่มีรูปสวยงาม เป็นไปตามทฤษฎี

โดยรูปแสดงผลการทดสอบ ที่แสดงบนจอของออสซิลโลสโคปมีลักษณะดังรูป



รูปที่ 4.1 แสดงผลการทดลองเป็นสัญญาณเอาต์พุทรูปสี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองของส่วนตรวจจับอุณหภูมิ
ตารางแสดงผลการทดลองของส่วนตรวจจับอุณหภูมิ

อุณหภูมิ(เซลเซียส)	ค่าโวลต์เตจที่เข้า A/Dทดลองครั้งที่1	ค่าโวลต์เตจที่เข้า A/Dทดลองครั้งที่2	ค่าโวลต์เตจที่เข้า A/Dตามทฤษฎี
64	3.21	3.27	3.37
63	3.20	3.26	3.36
62	3.20	3.26	3.35
61	3.20	3.26	3.34
60	3.19	3.25	3.33
59	3.19	3.24	3.32
58	3.19	3.23	3.31
57	3.18	3.23	3.30
56	3.17	3.22	3.29
55	3.17	3.21	3.28
54	3.16	3.21	3.27
53	3.16	3.20	3.26
52	3.15	3.19	3.25
51	3.14	3.18	3.24
50	3.14	3.18	3.23
49	3.13	3.17	3.22
48	3.12	3.16	3.21
47	3.12	3.15	3.20
46	3.11	3.14	3.19
45	3.11	3.14	3.18
44	3.10	3.13	3.17

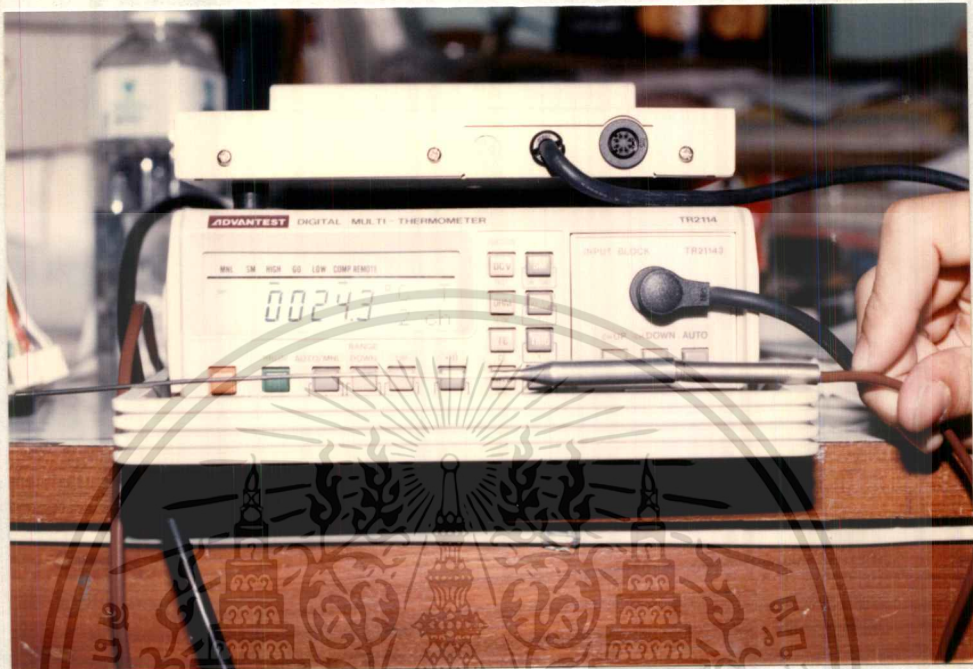
43	3.10	3.12	3.16
42	3.09	3.12	3.15
41	3.08	3.11	3.14
40	3.08	3.10	3.13
39	3.07	3.09	3.12
38	3.06	3.08	3.11
37	3.06	3.07	3.10
36	3.05	3.07	3.09
35	3.05	3.06	3.08
34	3.04	3.05	3.07
33	3.03	3.04	3.06
32	3.02	3.04	3.05
31	3.01	3.03	3.04
30	3.01	3.02	3.03
29	3.00	3.01	3.02
28	2.99	3.00	3.01
27	2.99	3.00	3.00
26	2.98	2.99	2.99
25	2.98	2.99	2.98
24	2.97	2.98	2.97
23	2.96	2.97	2.96
22	2.95	2.96	2.95
21	2.94	2.95	2.94
20	2.93	2.95	2.93
19	2.92	2.94	2.92
18	2.91	2.92	2.91
17	2.90	2.90	2.90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านอื่นๆ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16	2.89	2.89	2.89
15	2.88	2.89	2.88
14	2.87	2.88	2.87
13	2.86	2.87	2.86
12	2.85	2.86	2.85
11	2.85	2.86	2.84
10	2.84	2.85	2.83
9	2.84	2.84	2.82
8	2.83	2.83	2.81
7	2.81	2.81	2.80
6	2.80	2.80	2.79
5	2.79	2.79	2.78
4	2.79	2.79	2.76
3	2.78	2.78	2.75
2	2.78	2.78	2.74
1	2.78	2.78	2.73
0	2.78	2.78	2.72

จากตาราง เป็นการวัดอุณหภูมิ โดยการใช้เทอร์โมมิเตอร์ซึ่ง อุณหภูมิดังกล่าว เป็นอุณหภูมิที่ถูกต้อง ในส่วนของโวลต์เตจ (voltage) หรือแรงดันเป็นค่าของ เอาท์พุท (output) จากตัววัดอุณหภูมิซึ่งใช้ IC LM 335 ไว้ เอาท์พุท เป็น แรงดันที่เปลี่ยนแปลงค่า ตามอุณหภูมิที่เปลี่ยนไป แรงดันที่ได้นี้ เป็นสัญญาณแบบ อนาลอก (analog) เราจำเป็นต้องแปลงค่าจากอนาลอก(analog)เป็นดิจิตอล โดยนำแรงดันที่ได้ผ่านวงจรส่วน A/D เพื่อนำค่า ดิจิตอล(ตัวเลขแบบไบนารี) ส่งค่าให้กับคอมพิวเตอร์ แสดงผลออกมา เป็นอุณหภูมิ ช่วงอุณหภูมิที่แสดงคือ 1 - 64 องศาเซลเซียส แต่จากการทดลองค่าโวลต์เตจที่วัดได้มีความคลาด

เอกสารนี้เคลื่อนจากที่ค่าที่คำนวณได้ดังตารางด้านบนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงผลการวัดอุณหภูมิโดยใช้ดิจิตอลเทอร์โมมิเตอร์

ส่วนตรวจจับความร้อน(Heat Detector)

แสดงสถานะได้ 3 กรณี คือ

- 00 สถานะ สายขาด (open circuit)
- 01 สถานะ ปกติ (normal)
- 11 สถานะเพลิงไหม้ (fire)

สายขาด หมายถึง สายที่ต่อกับส่วนของตัวตรวจจับความร้อนขาด

การทำงานของซอฟต์แวร์ (software) และฮาร์ดแวร์ (hardware)

ซอฟต์แวร์ ในส่วนของโปรแกรมแบ่งการทำงานออกได้เป็น 2 วิธี คือ

1. การตรวจสอบตัวเอง (selftest) เป็นการติดต่อระหว่างโปรแกรมกับอุปกรณ์แจ้งเหตุเพลิงไหม้ (AFD: Addressable Fire Detector) โดยจะมีการตรวจสอบ AFD ทั้งหมด 128 ตัว (จำนวนสูงสุดที่ติดตั้ง AFD ได้) หรือ 128 ห้อง ที่มีการติดตั้งอุปกรณ์ในส่วนของอุปกรณ์ AFD สำหรับในการทดลองนี้มีการติดตั้งเพียง AFD 1 ตัวหรือ 1 ห้อง เนื่องจากมีไอซี MC 14469 เพียงตัวเดียวดังนั้นเมื่อมีการตรวจสอบจะมีเพียง AFD ตัวเดียวเท่านั้นที่แสดงสถานะออกมา โดยสามารถแสดงได้ดังนี้

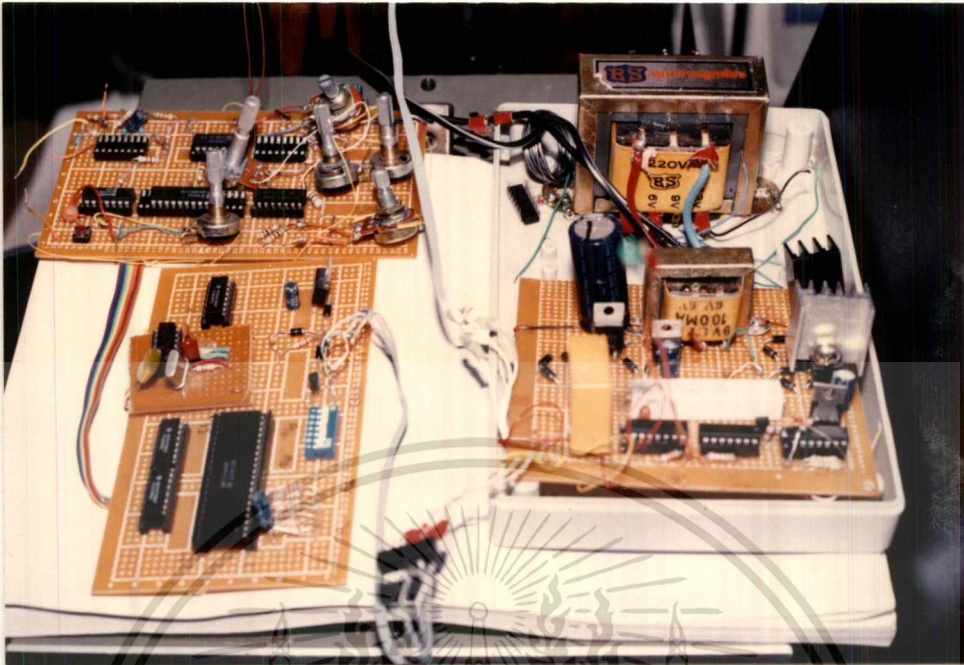
00 สายขาด (open circuit)
แสดงอุณหภูมิ 1-64 องศาเซลเซียส

01 ปกติ (normal)
แสดงอุณหภูมิ 1-64 องศาเซลเซียส

11 เพลิงไหม้ (fire)
แสดงอุณหภูมิมากกว่า 64 องศาเซลเซียส

ในห้องอื่นที่ไม่มีการติดตั้งอุปกรณ์แสดงเป็น unconnect (ไม่มีการติดตั้ง AFD) ในส่วนของ selftest แสดงเป็นเท็กซ์โหมด (text mode) แสดงเป็นตัวอักษร

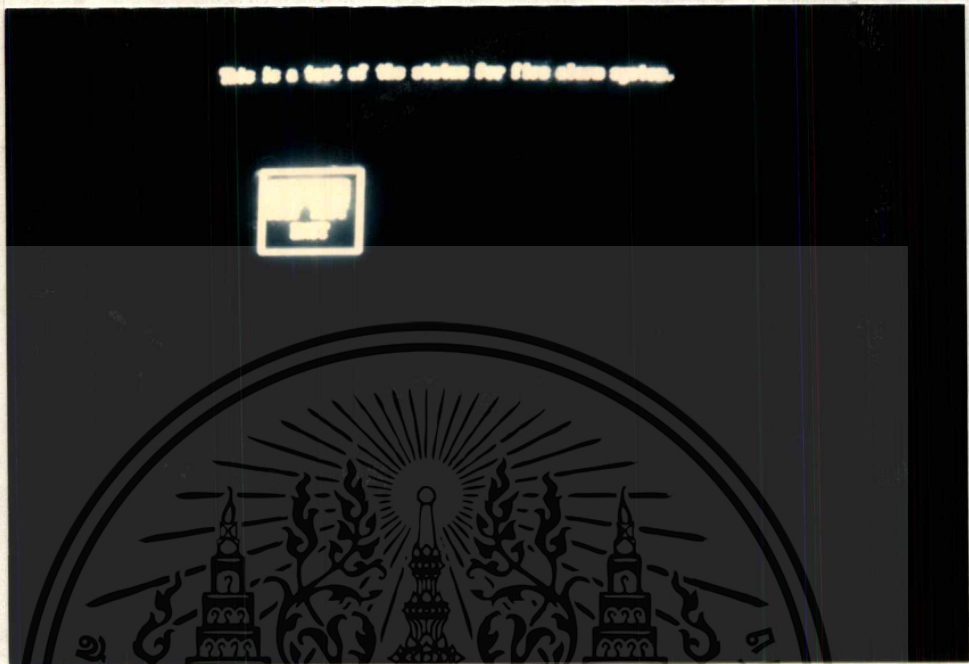
2. การตรวจสอบจากผู้ใช้ (usertest) ผู้ใช้เป็นผู้เลือกห้องที่ต้องการตรวจสอบหลังจากนั้นคอมพิวเตอร์แสดงภาพแปลนตึกโทรคมนาคมชั้นล่างและแสดงเหตุการณ์ในบริเวณห้องที่ต้องการทราบในส่วนของ usertest แสดงเป็นกราฟฟิก (graphics mode) แสดงเป็นรูปภาพ



รูปที่ 4.3 แสดงชิ้นงานของฮาร์ดแวร์ (hardware)

ประกอบด้วย

1. ส่วนแปลงระดับสัญญาณ
2. ส่วนของอุปกรณ์แจ้งเหตุเพลิงไหม้ (AFD)
 - ตรวจจับความร้อน
 - ตรวจจับอุณหภูมิ



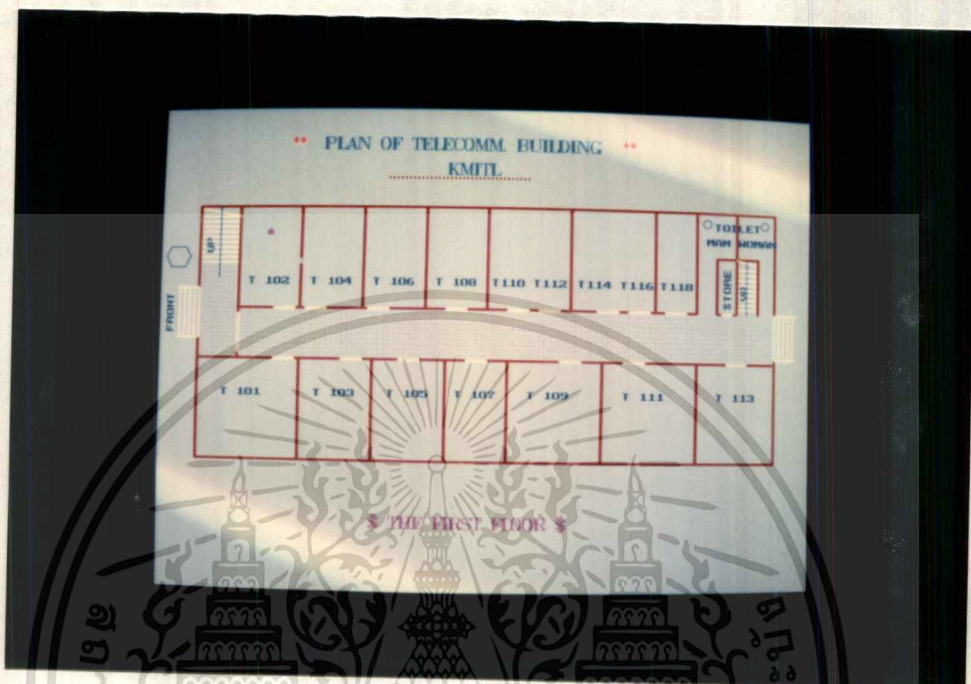
รูปที่ 4.4 แสดงการเลือกตรวจสอบจากหน้าจอคอมพิวเตอร์

1.selftest

2.usertest



รูปที่ 4.5 แสดงการเลือกห้องที่ต้องการตรวจสอบหลังจากเลือก usertest



รูปที่ 4.6 แสดงแปลนชั้นล่างของตึกโทรคมนาคม และแสดงผลเหตุการณ์ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและแนวทางการพัฒนา

จากการทดลอง ที่ผ่านมาโดยทำการติดตั้งส่วนฮาร์ดแวร์ และได้ถูกควบคุมโดยซอฟต์แวร์ที่ใช้ในเครื่องคอมพิวเตอร์ผลการทดลองที่ได้จะเกิดความผิดพลาดเนื่องจากส่วนฮาร์ดแวร์ ที่ต้องแก้ไข ดังนี้

1. ความเร็วของการส่ง (baud rate)
2. AFD (MC14469)

แต่การทดลองก่อนการทดลองทั้งระบบ พบว่า ผลการวัด อุณหภูมิที่จะตรงกับความจริงโดยขณะทำการทดลองใช้ดิจิตอล เทอร์โมมิเตอร์ (digital thermometer) เป็นตัววัดควบคุมกันไป ส่วนการจับสถานะนั้นก็ตรงกับทฤษฎีคือขณะเกิดเหตุเพลิงไหม้ นั้นจะแสดงผลเป็นโวลต์ที่แตกต่างจาก สภาวะปกติการทดลองทั้งหมดพบว่า การรับข้อมูล รับได้เพียงบางครั้งเท่านั้น โดยโอกาสที่จะรับข้อมูลนั้นน้อยมากซึ่งข้อมูลก็ ยังผิดพลาดด้วย เช่นข้อมูลคือ 0110101 ซึ่งหมายถึง เหตุการณ์ปกติ ที่ 27-28 องศาเซลเซียส ข้อมูลที่รับได้คือ 1110001 ซึ่งหมายถึง เหตุการณ์ปกติที่ 57-58 องศาเซลเซียส

ความผิดพลาดครั้งนี้ อาจเกิดจากตัวไอซี MC 14469 ซึ่งเป็นอุปกรณ์รับส่งข้อมูล การพัฒนาที่สามารถทำได้คือ

1. ควรมี AFD มากกว่านี้ คืออย่างน้อย 8 ตัวจึงจะพบเห็นประโยชน์ของระบบนี้
2. ควรใช้ประโยชน์ ของข้อมูลที่มีอยู่ 8 บิตให้ได้ประโยชน์สูงสุด
3. ควรมีการทดลองกับอุปกรณ์ ตรวจสอบจับเพลิงไหม้ จริงๆ
4. ควรระบบแสดงผลเป็น แพนผัง ณ ทุกจุด เพื่อเข้าใจได้ง่ายยิ่งขึ้น
5. ควรมีการส่งต่อไปสถานีตำรวจ เมื่อเกิดเหตุเพลิงไหม้

ภาคผนวก

โปรแกรมควบคุมการทำงานและแสดงผลของระบบ

```
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <ctype.h>
#include <bios.h>
#include <string.h>
#include <graphics.h>
#include <conio.h>
#define BORDER 1
#define ESC 27
#define REV_VID 0x70
#define NORM_VID 7
typedef struct _room {
    char szName[10];
    int PtX;
    int PtY;
} RoomStr;
RoomStr AddressToRoom[128];
#define DLH (1)
#define DLL (0)
#define LCR (3)
#define LSR (5)
#define B1200 (0)
#define B2400 (1)
#define B4800 (2)
#define B9600 (3)
#define B19200 (4)
#define COM1 (0)
#define COM2 (1)
#define D8 (0)
#define D7 (1)
#define S1 (0)
#define S2 (1)
#define PNONE (0)
#define PODD (1)
#define PEVEN (2)
#define BI 0x10
#define command (0)
#define NORMAL 100
#define FIRE 101
#define OPEN 102
#define ERROR 103
#define NOTCONNECT 104
typedef struct _result {
    char Status;
    int Tmp;
} RESULT;
void InitSerialComm(int, int, int, int, int);
int check_status(char);
int status_port(void);
int ctoi(char);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char connect(int address);
void fselftest(void);
RESULT fusertest(int);
void initializeroom(void);
int popup(char *menu[], char *keys, int count,int x, int y,
          int border);
void save_video(int startx, int endx, int starty, int endy,
               unsigned char *buf_ptr);
void restore_video(int startx, int endx, int starty, int
                  endy,
                  unsigned char *buf_ptr);
void goto_xy(int x, int y), cls(void);
void write_video(int x, int y, char *p, int attrib);
void display_menu(char *menu[], int x, int y, int count);
void draw_border(int startx, int starty, int endx, int endy)
void write_string(int x, int y, char *p, int attrib);
void write_char(int x, int y, char ch, int attrib);
void build(int,RESULT);
int get_resp(int x, int y, int count, char *menu[], char *keys);
int video_mode(void);
int readkey(void);
unsigned baud[5] = { 96,48,24,12,6 };/*for1200,2400,4800,9600&19200bps*/
char Data[2] = { 0x03,0x02 };/* feature bit of 8-bit , 7-bit */
char stop[2] = { 0x00,0x04 };/* feature bit of 1-bit , 2-bit */
char pari[3] = { 0x00,0x08,0x18 };/* feature bit of none,odd,even */
extern char str[100] ="";
extern char buffer[1024];
int ctoi(char data)
{
    return( data-'0');
}

void initializeroom(void) {
    int i,j,k,l;
    char Built[3] = {"ABC"};
    strcpy(AddressToRoom[0].szName,"T 102");
    AddressToRoom[0].PtX = 110;
    AddressToRoom[0].PtY = 125;
    strcpy(AddressToRoom[1].szName,"T 103");
    AddressToRoom[1].PtX = 175;
    AddressToRoom[1].PtY = 310;
    strcpy(AddressToRoom[2].szName,"T 106");
    AddressToRoom[2].PtX = 230;
    AddressToRoom[2].PtY = 125;
    strcpy(AddressToRoom[3].szName,"T 109");
    AddressToRoom[3].PtX = 385;
    AddressToRoom[3].PtY = 310;
    strcpy(AddressToRoom[4].szName,"T 110");
    AddressToRoom[4].PtX = 360;
    AddressToRoom[4].PtY = 125;
    strcpy(AddressToRoom[5].szName,"T 111");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AddressToRoom[4].PtY = 125;
strcpy(AddressToRoom[5].szName, "T 111");
AddressToRoom[5].PtX = 475;
AddressToRoom[5].PtY = 310;
strcpy(AddressToRoom[6].szName, "T 113");
AddressToRoom[6].PtX = 560;
AddressToRoom[6].PtY = 310;
strcpy(AddressToRoom[7].szName, "T 114");
AddressToRoom[7].PtX = 440;
AddressToRoom[7].PtY = 125;
for (i = 0; i < 3; i++)
for (k = 1; k <= 10; k++) {
sprintf(AddressToRoom[l].szName, "%c %d", Built[i], j*100+k);
AddressToRoom[l].PtX = 0;
AddressToRoom[l].PtY = 0;
}
}

char connect(int address)
{
unsigned char s_data[20]="";
int index=0;
char buffer;
outportb(0x3F8, address);
outportb(0x3F8, command)
while(index< 2)
{
while( inportb(0x3FD)&BI ); if BI set ,wait */
s_data[index] = inportb(0x3F8); /* if BI unactive ,recieve */
index++;
}
if( atoi(s_data[0])==address ) /*if address is correct*/
{
buffer =(s_data[1]);
}
else
{
buffer = '\x0';
}
return buffer;
}

int check_status(char data)
{
int status;
int temper= 0;
if( data!='\x0'){
switch(data & 0x03) /*mark recieved data with 00000011*/
{
case 0 :
strcpy(str, "OPEN");
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
        case 1 :
        strcpy(str, "NORMAL");
        break;
        case 3 :
        strcpy(str, "FIRE");
        break;
        default:
        strcpy(str, "ERROR");
        break;
    }
    if(( data&0x03==0) || (data&0x03==1))
    {
        data &= 0x7C; /* mark 5 bit upper and bit7 not used=0*/
        temper=ctoi(data);
        temper>>=2;
        temper=((2*temper)+1);
    }
    return (temper);
}
}

void InitSerialComm(int comno,int baudrate,
int databit,int stopbit,int parity)
{
    char low_divisor,high_divisor,comm_spec;
    int CommAddr;
    switch( comno ){
        case COM1 : CommAddr = 0x3F8; break;
        case COM2 : CommAddr = 0x2F8; break;
    }
    low_divisor = baud[baudrate];
    high_divisor = 0;
    comm_spec = Data[databit] | stop[stopbit] | pari[parity];
    outportb( CommAddr + LCR , comm_spec | 0x80 );
    outportb( CommAddr + DLL , low_divisor );
    outportb( CommAddr + DLH , high_divisor );
    outportb( CommAddr + LCR , comm_spec & 0x7F );
}

void fselftest(void)
{
    int COM_PORT=1; /**select com_port**/
    int address = 128;
    int i;
    char pt_buf;
    char temp;
    int RmNormal[128],RmNormalPtr = 0;
    int RmOpen[128],RmOpenPtr = 0;
    int RmFire[128],RmFirePtr = 0;
    int RmUncon[128],RmUnconPtr = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(COM_PORT==1)  InitSerialComm(COM1,2,D8,S1,PEVEN);
else if(COM_PORT==2)  InitSerialComm(COM2,2,D8,S1,PEVEN);
printf("\n");

for (i = 0; i < 128; i++) {
temp = connect(address + i);
if (temp != '\x0') { /* has data from address */
strcpy (str,"");
pt_buf = check_status(temp);
if (!strcpy(str,"NORMAL")) {
printf("Address %d Room %s\n",address+i,AddressToRoom[i].szName);
printf("Temperature case normal = %d-%d\n",pt_buf-1,pt_buf+1);
RmNormal[RmNormalPtr++] = i;
} else if (!strcpy(str,"OPEN")) {
printf("Address %d Room %s\n",address+i,AddressToRoom[i].szName);
printf("Temperature case open circuit = %d-%d\n",pt_buf-1,pt_buf+1);
RmOpen[RmOpenPtr++] = i;
} else if (!strcpy(str,"FIRE")) {
printf("Address %d Room %s\n",address+i,AddressToRoom[i].szName);
printf("Fire! temperature over 64 celcius\n");
RmFire[RmFirePtr++] = i;
} else {
printf("Address %d Room %s\n",address+i,AddressToRoom[i].szName);
printf("Error!");
}
} else { /* not connect address */
RmUncon[RmUnconPtr++] = I;
printf("Address %d Room %s Unconnect!\n",i+address,AddressToRoom[i].szName);
}
delay(1000);
if(kbhit()) {
getch();
i=129;
}
}

clrscr();
printf("TOTAL THE STATUS ");
if (RmNormalPtr > 0)
printf("Normal Room\n");
for ( i = 0; i < RmNormalPtr; i++) {
printf("%5s , ",AddressToRoom[RmNormal[i]].szName);
}
if (RmNormalPtr > 0)
if (RmNormalPtr % 10 == 0) {
gotoxy(80,wherey()-1);
printf("\b \b\b");
} else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\b\b\b  \b\b\b");
    if (RmOpenPtr > 0)
        printf("\n\nOpen Circuit Room\n");
    for ( i = 0; i < RmOpenPtr; i++) {
        printf("%5s , ",AddressToRoom[RmOpen[i]].szName);
    }
    if (RmOpenPtr > 0)
    if (RmOpenPtr % 10 == 0) {
gotoxy(80,wherey()-1);
printf("\b \b\b");
} else
printf("\b\b\b  \b\b\b");
    if (RmFirePtr > 0)
        printf("\n\nFire Room\n");
    for ( i = 0; i < RmFirePtr; i++) {
        printf("%5s , ",AddressToRoom[RmNormal[i]].szName);
    }
    if (RmFirePtr > 0)
    if (RmFirePtr % 10 == 0) {
gotoxy(80,wherey()-1);
printf("\b \b\b");
} else
printf("\b\b\b  \b\b\b");
    if (RmUnconPtr > 0)
        printf("\n\nNot Connect Room\n");
    for ( i = 0; i < RmUnconPtr; i++) {
        printf("%5s , ",AddressToRoom[RmUncon[i]].szName);
    }
    if (RmUnconPtr > 0)
    if (RmUnconPtr % 10 == 0) {
gotoxy(80,wherey()-1);
printf("\b \b\b");
} else
printf("\b\b\b  \b\b\b");
        getch();
        cls();
    }
}

RESULT fuserstest(int address)
{
    int COM_PORT=1; /**select com_port**/
    RESULT pt_buf;
    char temp;

    if(COM_PORT==1) InitSerialComm(COM1,2,D8,S1,PEVEN);
    else if(COM_PORT==2) InitSerialComm(COM2,2,D8,S1,PEVEN);

    temp = connect(address);
    if (temp != '\x0') { /* has data from address */
strcpy (str,"");
pt_buf.Tmp = check_status(temp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (!strcmp(str,"NORMAL")) {
        pt_buf.Status = NORMAL;
    } else if (!strcmp(str,"OPEN")) {
        pt_buf.Status = OPEN;
    } else if (!strcmp(str,"FIRE")) {
        pt_buf.Status = FIRE;
    } else {
        pt_buf.Status = ERROR;
    }
    } else { /* not connect address */
pt_buf.Status = NOTCONNECT;
    }
    return (pt_buf);
}

int respselect, resppoint;
char far *vid_mem;
char *point[] = {
    " 1. ROOM T 102 ",
    " 2. ROOM T 103 ",
    " 3. ROOM T 106 ",
    " 4. ROOM T 109 ",
    " 5. ROOM T 110 ",
    " 6. ROOM T 111 ",
    " 7. ROOM T 113 ",
    " 8. ROOM T 114 ",
};
char *select[] = {
    "USER TEST",
    "SELF TEST",
    "EXIT "
};

main(void)
{
    char *pt;
    int i;
    RESULT Result;
    char eff;
    initializeroom();
    cls();
    fselftest();
    goto_xy(0, 0);
    printf("This is a test of the status for fire alarm system.\n");
    while(kbhit() != ESC) {
        respselect = popup(select, "USE", 3, 5, 5, BORDER);
        switch(respselect) {
        case 0: resppoint = popup(point, "12345678", 8, 15, 7, BORDER);
            Result = fusertest(resppoint+128)
            build(resppoint, Result);
            eff = getch();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (eff == ESC)
            exit(0);
        closegraph();
        break;
    case 1: fselftest();
        break;
    case 2: exit(0);
        break;
    }
}
return 0;
}

/* Display a pop-up menu and return selection.*/

int popup(
    char *menu[], /* menu text */
    char *keys, /* hot keys */
    int count, /* number of menu items */
    int x, int y, /* X,Y coordinates of left hand corner */
    int border /* no border if 0 */
)
{
    register int i, len;
    int endx, endy, choice, vmode;
    unsigned char *p;

    if ((y > 24) || (y < 0) || (x > 79) || (x < 0)) {
        printf("range error");
        return -2;
    }

    vmode = video_mode();
    if ((vmode != 2) && (vmode != 3) && (vmode != 7)) {
        printf("video must be in 80 column text mode");
        exit(1);
    }

    /* set proper address of video RAM */
    if (vmode == 7) vid_mem = (char far *) 0xB0000000;
    else vid_mem = (char far *) 0xB8000000;

    /* make sure that menu will fit */
    len = 0;
    for (i = 0; i < count; i++)
        if (strlen(menu[i]) > len) len = strlen(menu[i]);
    endx = len + 2 + x;
    endy = count + 1 + y;
    if ((endy + 1 > 24) || (endx + 1 > 79)) {
        printf("menu won't fit");
        return -2;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* allocate enough memory to hold current contents of the screen */
p = (unsigned char *) malloc(2 * (endx-x+1) * (endy-y+1));
if(!p) exit(1); /* install your own error handler here */

/* save the current screen data */
save_video(x, endx+1, y, endy+1, p);

if(border) draw_border(x, y, endx, endy);

/* display the menu */
display_menu(menu, x+1, y+1, count);

/* get the user's response */
choice = get_resp(x, y, count, menu, keys);

/* restore the original screen */
restore_video(x, endx+1, y, endy+1, p);
free(p);
return choice;
}

/* Display the menu in its proper location. */
void display_menu(char *menu[], int x, int y, int count)
{
    register int i;
    for(i=0; i<count; i++, y++)
        write_string(x, y, menu[i], NORM_VID);
}

/* Draw a border around the window. */
void draw_border(int startx, int starty, int endx, int endy)
{
    register int i;
    char far *v, far *t;
    v = vid_mem;
    t = v;

    /* draw vertical lines */
    for(i=starty+1; i<endy; i++) {
        v += (i*160) + startx*2;
        *v++ = 179;
        *v = NORM_VID;
        v = t;
        v += (i*160) + endx*2;
        *v++ = 179;
        *v = NORM_VID;
        v = t;
    }

    /* draw horizontal lines */
    for(i=startx+1; i<endx; i++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        v += (starty*160) + i*2;
        *v++ = 196;
    *v = NORM_VID;
        v = t;
        v += (endy*160) + i*2;
        *v++ = 196;
    *v = NORM_VID;
        v = t;
    }

    /* draw corners */
    write_char(startx, starty, (char) 218, NORM_VID);
    write_char(startx, endy, (char) 192, NORM_VID);
    write_char(endx, starty, (char) 191, NORM_VID);
    write_char(endx, endy, (char) 217, NORM_VID);
}

/* Input user's selection. */
get_resp(int x, int y, int count, char *menu[], char *keys)
{
    union inkey {
        char ch[2];
        int i;
    } c;
    int arrow_choice=0;
    char *key_choice;
    x++;
    y++;

    /* highlight the first selection */
    goto_xy(x, y);
    write_string(x, y, menu[0], REV_VID); /* reverse video */

    for(;;) {
        c.i = readkey(); /* read the key */

        /* reset the selection to normal video */
        goto_xy(x, y+arrow_choice);
        write_string(x, y+arrow_choice,
            menu[arrow_choice], NORM_VID); /* redisplay */

        if(c.ch[0]) { /* is normal key */
            /* see if it is a hot key */
            key_choice = strchr(keys, tolower(c.ch[0]));
            if(key_choice) return key_choice-keys;

            /* check for ENTER or space bar */
            switch(c.ch[0]) {
                case '\r': return arrow_choice;

```

```

    case ' ' : arrow_choice++;
                break;
    case ESC : return -1; /* cancel */
                }
    }
    else { /* is special key */
        switch(c.ch[1]) {
    case 72: arrow_choice--; /* up arrow */
                break;
    case 80: arrow_choice++; /* down arrow */
                break;
        }
    }
    if (arrow_choice==count) arrow_choice=0;
    if (arrow_choice<0) arrow_choice = count-1;

    /* highlight the next selection */
    goto_xy(x, y+arrow_choice);
    write_string(x, y+arrow_choice, menu[arrow_choice], REV_VID);
}
}

/* Display a string with specified attribute. */
void write_string(int x, int y, char *p, int attrib)
{
    register int i;
    char far *v;
    v = vid_mem;
    v += (y*160) + x*2; /* compute the address */
    for(i=y; *p; i++){
        *v++ = *p++; /* write the character */
        *v++ = attrib; /* write the attribute */
    }
}

/* Write character with specified attribute. */
void write_char(int x, int y, char ch, int attrib)
{
    char far *v;
    v = vid_mem;
    v += (y*160) + x*2; /* compute the address */
    *v++ = ch; /* write the character */
    *v = attrib; /* write the attribute */
}

/* Save a portion of the screen using direct video RAM accessing.*/
void save_video(int startx, int endx, int starty, int endy,
    unsigned char *buf_ptr)
{
    register int i,j;
    char far *v, far *t;
    v = vid_mem;

```

```

        for(i=startx; i<endx; i++)
        for(j=starty; j<endy; j++) {
            t = v + (j*160) + i*2; /* compute the address */
            *buf_ptr++ = *t++; /* read the character */
            *buf_ptr++ = *t; /* read the attribute */
            *(t-1) = ' '; /* clear the window */
        }
    }

```

```

/* Restore a portion of the screen using direct video accessing.*/
void restore_video(int startx, int endx, int starty, int endy,
    unsigned char *buf_ptr)

```

```

{
    register int i,j;
    char far *v, far *t;
    v = vid_mem;
    t = v;
    for(i=startx; i<endx; i++)
    for(j=starty; j<endy; j++) {
        v = t;
        v += (j*160) + i*2; /* compute the address */
        *v++ = *buf_ptr++; /* write the character */
        *v = *buf_ptr++; /* write the attribute */
    }
}

```

```

/* Clear the screen. */

```

```

void cls(void)

```

```

{
    union REGS r;

    r.h.ah = 6; /* screen scroll code */
    r.h.al = 0; /* clear screen code */
    r.h.ch = 0; /* start row */
    r.h.cl = 0; /* start column */
    r.h.dh = 24; /* end row */
    r.h.dl = 79; /* end column */
    r.h.bh = 7; /* blank line is black */
    int86(0x10, &r, &r);
}

```

```

/* Send the cursor to x,y. */

```

```

void goto_xy(int x, int y)

```

```

{
    union REGS r;

    r.h.ah = 2; /* cursor addressing function */
    r.h.dl = x; /* column coordinate */
    r.h.dh = y; /* row coordinate */
    r.h.bh = 0; /* video page */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        int86(0x10, &r, &r);
    }

    /* Returns the current video mode. */
    video_mode(void)
    {
        union REGS r;

        r.h.ah = 15; /* get video mode */
        return int86(0x10, &r, &r) & 255;
    }

    /* Return the 16-bit scan code from the keyboard. */
    readkey(void)
    {
        union REGS r;

        r.h.ah = 0;
        return int86(0x16, &r, &r);
    }

    /* Create the plan of Telecomm Building for display the result*/
    void build(int Room,RESULT Result)
    {
        int i,x;
        int y;
        int radius;
        int polypoints[]={10,140,25,140,30,150,25,160,10,160,5,150,10,140};
        char String[256];
        int pox1,poy1;
        int driver, mode ;
        driver = VGA;
        mode = VGAHI;
        initgraph(&driver,&mode,"");
        pox1 = AddressToRoom[Room].PtX;
        poy1 = AddressToRoom[Room].PtY;
        setbkcolor(7);
        setcolor(9);
        settextstyle(1,0,1);
        outtextxy(150,25," PLAN OF TELECOMM. BUILDING ");
        outtextxy(270,50," KMITL");
        setcolor(12);
        outtextxy(128,25,"**");
        setcolor(5);
        outtextxy(200,350," $ THE FIRST FLOOR $ ");
        setcolor(4);
        setlinestyle(0,0,3);
        line(40,100,600,100);
        line(600,100,600,200);
        line(600,245,600,350);
        line(40,100,40,180);
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(40,230,40,350);
line(40,350,600,350);
line(80,100,80,205);
line(140,100,140,200);
line(200,100,200,200);
line(260,100,260,200);
line(320,100,320,200);
line(400,100,400,200);
line(480,100,480,200);
line(520,100,520,200);
moveto(80,200);
lineto(520,200);
line(40,250,600,250);
line(80,245,80,250);
line(140,250,140,350);
line(210,250,210,350);
line(280,250,280,350);
line(340,250,340,350);
line(430,250,430,350);
line(520,250,520,350);
line(540,200,560,200);
line(560,200,560,150);
line(540,145,540,200);
line(560,150,560,100);
line(582,200,582,145);
line(582,145,540,145);
setcolor(0);
line(115,200,135,200);
line(145,200,165,200);
line(205,200,225,200);
line(280,200,300,200);
line(340,200,360,200);
line(370,200,390,200);
line(425,200,445,200);
line(455,200,475,200);
line(490,200,510,200);
line(115,250,135,250);
line(180,250,200,250);
line(235,250,255,250);
line(300,250,320,250);
line(390,250,410,250);
line(450,250,470,250);
line(550,250,570,250);
line(545,200,555,200);
setcolor(14);
setlinestyle(0,0,1);
line(115,200,135,200);
line(145,200,165,200);
line(205,200,225,200);
line(280,200,300,200);
line(340,200,360,200);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(370,200,390,200);
line(425,200,445,200);
line(455,200,475,200);
line(490,200,510,200);
line(115,250,135,250);
line(180,250,200,250);
line(235,250,255,250);
line(300,250,320,250);
line(390,250,410,250);
line(450,250,470,250);
line(550,250,570,250);
line(80,205,80,245);
line(600,200,600,245);
line(545,200,555,200);
    /*Made the report*/
    line(150,390,470,390);
    line(470,390,470,460);
    line(470,460,150,460);
    line(150,390,150,460);

    for(y=155;y<200;y+=5){
        line(562,y,580,y);
    }
line(560,145,580,150);
    for(x=20;x<45;x+=5){
        line(x,180,x,230);
    }
line(18,180,40,180);
line(18,230,40,230);
    for(x=600;x<625;x+=5){
        line(x,200,x,245);
    }
line(600,200,623,200);
line(600,245,623,245);
    for(y=110;y<160;y+=5){
        line(42,y,78,y);
    }

line(42,102,80,110);
setcolor(8);
circle(530,110,4);
circle(589,110,4);
setcolor(1);
line(60,100,60,170);
line(570,145,570,200);
settextstyle(0,1,1);
outtextxy(55,130,"UP");
outtextxy(555,155,"STORE");
outtextxy(570,170,"up");
outtextxy(15,185,"FRONT");
setcolor(1);
settextstyle(0,0,1);

```

```

outtextxy(538,110,"TOILET");
outtextxy(530,125,"MAN");
outtextxy(562,125,"WOMAN");
outtextxy(90,170,"T 102");
outtextxy(150,170,"T 104");
outtextxy(210,170,"T 106");
outtextxy(270,170,"T 108");
outtextxy(325,170,"T110 T112");
outtextxy(407,170,"T114 T116");
outtextxy(485,170,"T118");
outtextxy(65,280,"T 101");
outtextxy(155,280,"T 103");
outtextxy(225,280,"T 105");
outtextxy(290,280,"T 107");
outtextxy(360,280,"T 109");
outtextxy(450,280,"T 111");
outtextxy(540,280,"T 113");
    for(x=85;x<600;x+=5){
        putpixel(x,205,5);
        putpixel(x,210,5);
            putpixel(x,215,5);
            putpixel(x,220,5);
        putpixel(x,225,5);
        putpixel(x,230,5);
        putpixel(x,235,5);
        putpixel(x,240,5);
            putpixel(x,245,5);
        }
    for(y=160;y<250;y+=5){
        putpixel(45,y,5);
        putpixel(50,y,5);
        putpixel(55,y,5);
            putpixel(60,y,5);
        putpixel(65,y,5);
        putpixel(70,y,5);
        putpixel(75,y,5);
    }
drawpoly(7, polypoints);
setcolor(6);
setlinestyle(1,2,3);
line(220,70,360,70);
    for(i=0;i<5;i++)
    {
        setcolor(5);
        gotoxy(pox1,poy1);
        switch(Result.Status) {
            case NORMAL:
sprintf(String,"Temperature = %d-%d",Result.Tmp-1,Result.Tmp+1);
                outtextxy(190,440,String);
                outtextxy(pox1-22,poy1+20,"Normal");
                setcolor(8);

```

```

        outtextxy(200,400,"**REPORT FOR ROOM      **");
        outtextxy(345,400,AddressToRoom[Room].szName);
        outtextxy(190,420,"Status This Room = NORMAL");
        break;
    case FIRE:
        outtextxy(pox1-15,poy1+20,"Fire");
        outtextxy(190,440,"Temperature over64 celcius");
        setcolor(8);
        outtextxy(200,400,"**REPORT FOR ROOM      **");
        outtextxy(345,400,AddressToRoom[Room].szName);
        outtextxy(220,420,"Status This Room = FIRE");
        break;
    case OPEN:
        sprintf(String,"Temperature = %d-%d",Result.Tmp-1,Result.Tmp+1);
        outtextxy(190,440,String);
        outtextxy(pox1-27,poy1+20,"Open");
        setcolor(8);
        outtextxy(200,400,"**REPORT FOR ROOM      **");
        outtextxy(345,400,AddressToRoom[Room].szName);
        outtextxy(150,420,"Status This Room = OPEN");
        break;
    case ERROR:
        outtextxy(pox1-18,poy1+20,"Error");
        setcolor(8);
        outtextxy(200,400,"**REPORT FOR ROOM      **");
        outtextxy(345,400,AddressToRoom[Room].szName);
        outtextxy(190,420,"Status This Room = ERROR");
        outtextxy(200,440,"Can't check temperature");
        break;
    case NOTCONNECT:
        outtextxy(pox1-34,poy1+20,"Unconnect");
        setcolor(8);
        outtextxy(200,400,"**REPORT FOR ROOM      **");
        outtextxy(345,400,AddressToRoom[Room].szName);
        outtextxy(190,420,"Status This Room = UNCONNECT");
        outtextxy(200,440,"Can't check temperature");
        break;
}
circle(pox1,poy1,2);
delay(500);
setcolor(7);
gotoxy(pox1,poy1);
switch(Result.Status) {
    case NORMAL:
        outtextxy(pox1-22,poy1+20,"NONMAL");
        break;
    case FIRE:
        outtextxy(pox1-15,poy1+20,"FIRE");
        break;
    case OPEN:
        outtextxy(pox1-27,poy1+20,"OPEN");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Addressable Asynchronous Receiver/Transmitter CMOS

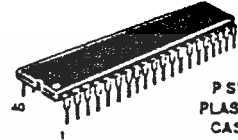
The MC14469 receives one or two eleven-bit words in a serial data stream. One of the incoming words contains the address and when the address matches, the MC14469 then transmits information in two eleven-bit-word data streams. Each of the transmitted words contains eight data bits, an even parity bit, and start and stop bits.

The received word contains seven address bits with the address of the MC14469 set on seven pins. Thus 27 or 128 units can be interconnected in simplex or full duplex data transmission. In addition to the address received, seven command bits may be received for general-purpose data or control use.

The MC14469 finds application in transmitting data from remote A-to-D converters, remote MPUs, or remote digital transducers to the master computer or MPU.

- Supply Voltage Range: 4.5 V to 18 V
- Low Quiescent Current: 75 μ A Maximum @ 5 V, 25°C
- Guaranteed Data Rates to 4800 Baud @ 5 V, to 9600 Baud @ 12 V
- Receive — Serial to Parallel
 Transmit — Parallel to Serial
- Transmit and Receive Simultaneously in Full Duplex
- Crystal or Resonator Operation for On-Chip Oscillator
- See Application Note AN-806A
- Chip Complexity: 1200 FETs or 300 Equivalent Gates

MC14469



P SUFFIX
 PLASTIC DIP
 CASE 711

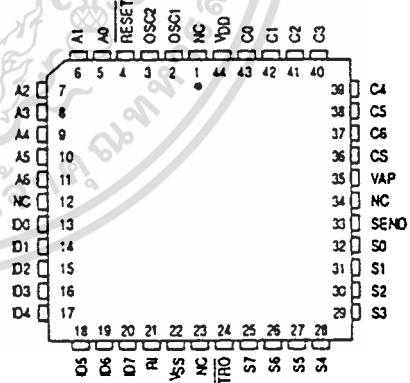
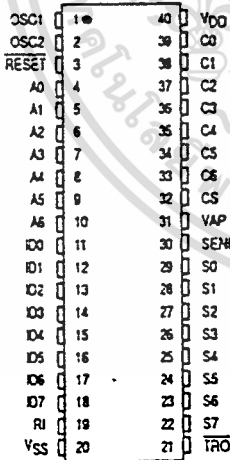


FN SUFFIX
 PLCC
 CASE 777

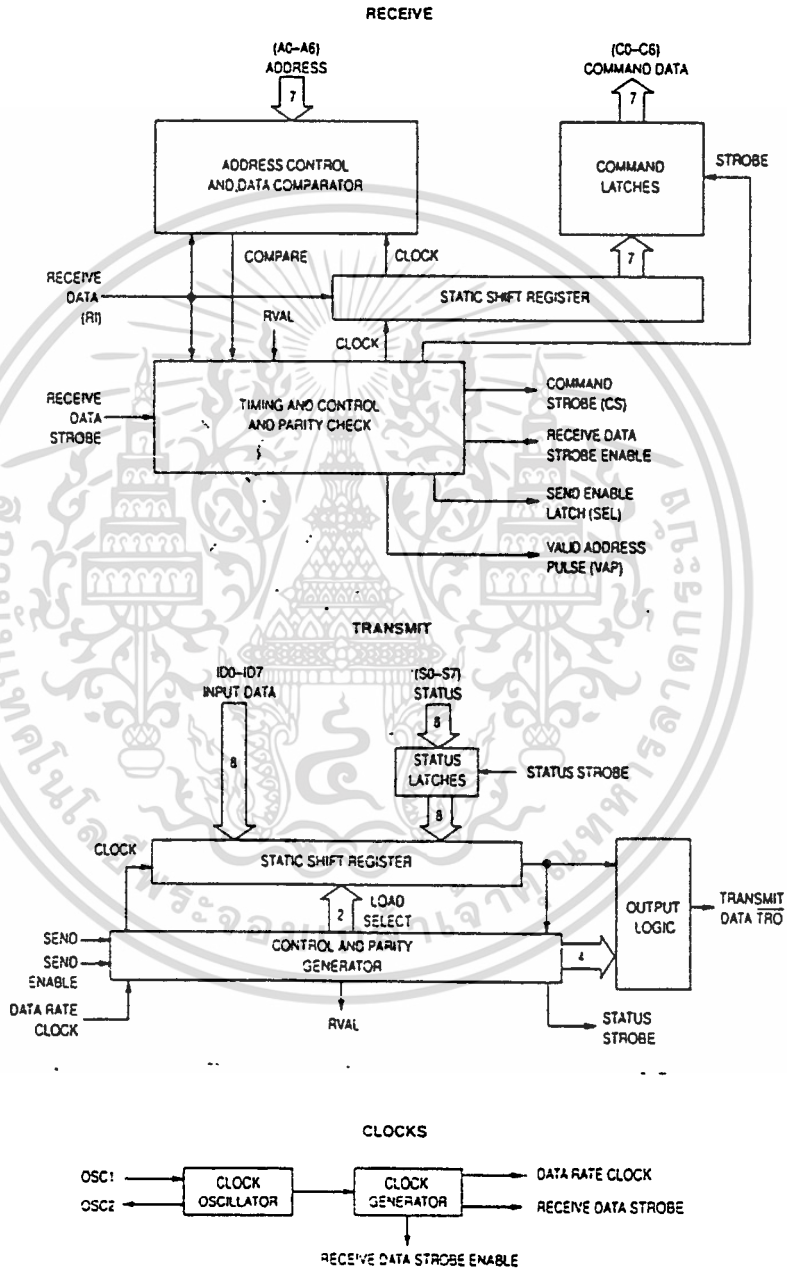
ORDERING INFORMATION

MC14469P Plastic DIP
 MC14469FN PLCC

PIN ASSIGNMENTS



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
        break;
    case ERROR:
        outtextxy(pox1-18,poy1+20,"ERROR");
        break;
    case NOTCONNECT:
        outtextxy(pox1-34,poy1+20,"UNCONNECT");
        break;
}
delay(500);
sound(100);
delay(700);
nosound();
if(kbhit())
    return(0);
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC1488

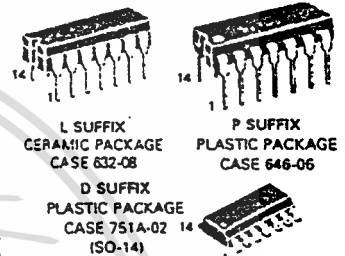
QUAD LINE DRIVER

The MC1488 is a monolithic quad line driver designed to interface data terminal equipment with data communications equipment in conformance with the specifications of EIA Standard No. RS-232C.

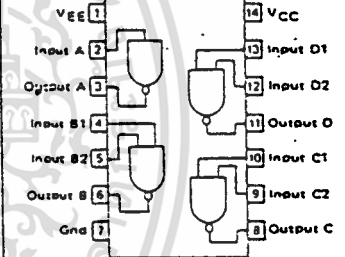
Features:

- Current Limited Output
 ± 10 mA typ
- Power-Off Source Impedance
 300 Ohms min
- Simple Slew Rate Control with External Capacitor
- Flexible Operating Supply Range
- Compatible with All Motorola MOTL and MTTL Logic Families

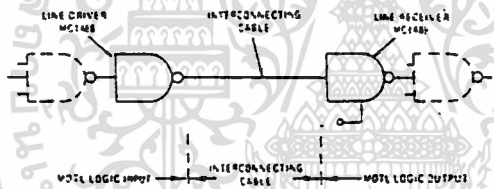
**QUAD MOTL LINE DRIVER
 RS-232C
 SILICON MONOLITHIC
 INTEGRATED CIRCUIT**



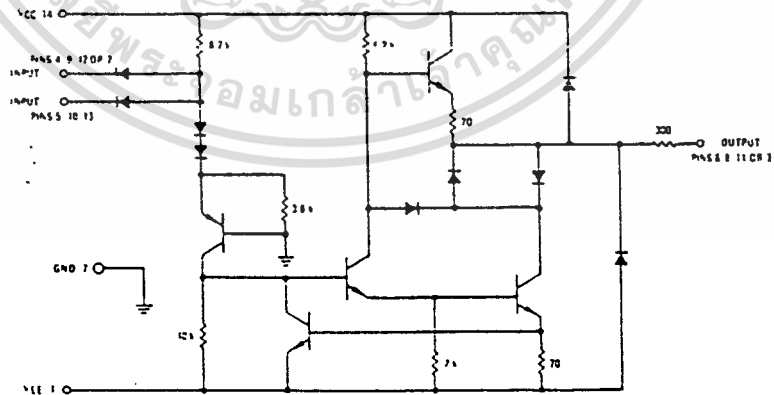
PIN CONNECTIONS



TYPICAL APPLICATION



**CIRCUIT SCHEMATIC
 (1/4 OF CIRCUIT SHOWN)**



MOTL and MTTL are trademarks of Motorola Inc.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC1489
MC1489A

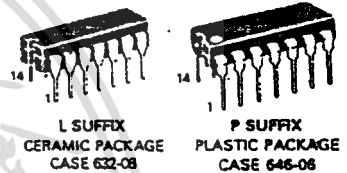
QUAD LINE RECEIVERS

The MC1489 monolithic quad line receivers are designed to interface data terminal equipment with data communications equipment in conformance with the specifications of EIA Standard No. RS-232C.

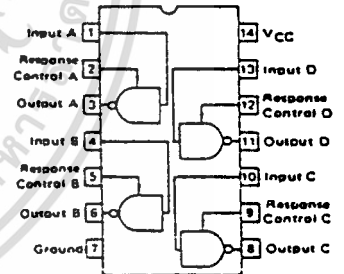
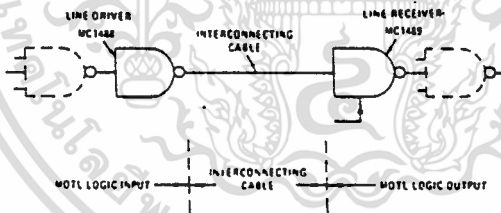
- Input Resistance — 3.0 k to 7.0 kilohms
- Input Signal Range — ± 30 Volts
- Input Threshold Hysteresis Built In
- Response Control
 - a) Logic Threshold Shifting
 - b) Input Noise Filtering

QUAD MOTL
LINE RECEIVERS
RS-232C

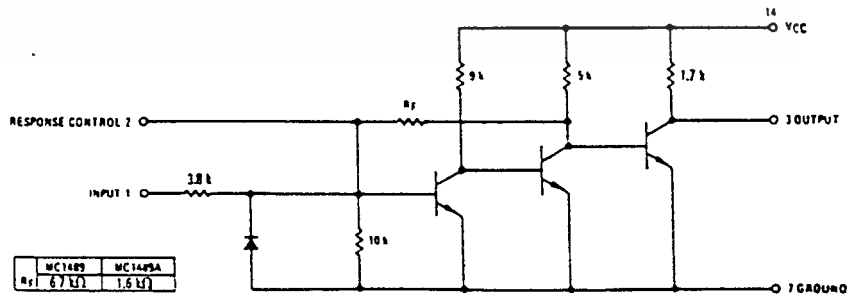
SILICON MONOLITHIC
INTEGRATED CIRCUIT



TYPICAL APPLICATION



EQUIVALENT CIRCUIT SCHEMATIC (1/4 OF CIRCUIT SHOWN)



MOTL and MTTL are trademarks of Motorola Inc.

MC1489, MC1489A

MAXIMUM RATINGS (T_A = +25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Power Supply Voltage	V _{CC}	10	Vdc
Input Voltage Range	V _{IR}	±30	Vdc
Output Load Current	I _L	20	mA
Power Dissipation (Package Limitation, Ceramic and Plastic Dual In-Line Package) Derate above T _A = -25°C	P _D T _{θJA}	1000 6.7	mW mW/°C
Operating Ambient Temperature Range	T _A	0 to +75	°C
Storage Temperature Range	T _{stg}	-65 to +175	°C

ELECTRICAL CHARACTERISTICS (Response control pin is open.) (V_{CC} = -5.0 Vdc ± 10%, T_A = 0 to -75°C unless otherwise noted)

Characteristics	Symbol	Min	Typ	Max	Unit
Positive Input Current (V _{IH} = +25 Vdc) (V _{IH} = -3.0 Vdc)	I _{IH}	3.6 0.43	—	8.3 —	mA
Negative Input Current (V _{IL} = -25 Vdc) (V _{IL} = -3.0 Vdc)	I _{IL}	-3.6 -0.43	—	-8.3 —	mA
Input Turn-On Threshold Voltage (T _A = +25°C, V _{OL} ≤ 0.45 V)	V _{IH}	1.0 1.75	— 1.95	1.5 2.25	Vdc
Input Turn-Off Threshold Voltage (T _A = -25°C, V _{OH} ≥ 2.5 V, I _L = -0.5 mA)	V _{IL}	0.75 0.75	— 0.8	1.25 1.25	Vdc
Output Voltage High (V _{IH} = 0.75 V, I _L = -0.5 mA) (Input Open Circuit, I _L = -0.5 mA)	V _{OH}	2.5 2.5	4.0 4.0	5.0 5.0	Vdc
Output Voltage Low (V _{IL} = 3.0 V, I _L = 10 mA)	V _{OL}	—	0.2	0.45	Vdc
Output Short-Circuit Current	I _{OS}	—	-3.0	-4.0	mA
Power Supply Current (All Gates "on," I _{out} = 0 mA, V _{IH} = +5.0 Vdc)	I _{CC}	—	16	26	mA
Power Consumption (V _{IH} = +5.0 Vdc)	P _C	—	80	130	mW

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 1%, T_A = -25°C. See Figure 1.)

Characteristics	Symbol	Min	Typ	Max	Unit
Propagation Delay Time (R _L = 3.9 kΩ)	t _{PLH}	—	25	85	ns
Rise Time (R _L = 3.9 kΩ)	t _{TLH}	—	120	175	ns
Propagation Delay Time (R _L = 390 kΩ)	t _{PHL}	—	25	50	ns
Fall Time (R _L = 390 kΩ)	t _{THL}	—	10	20	ns

TEST CIRCUITS

FIGURE 1 — SWITCHING RESPONSE

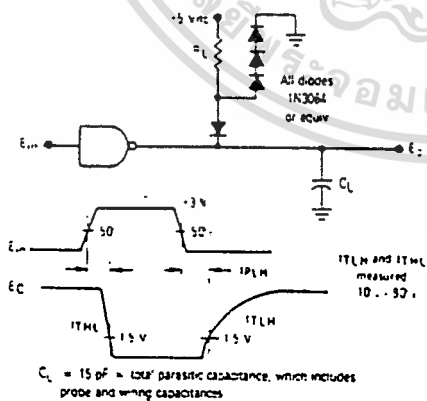
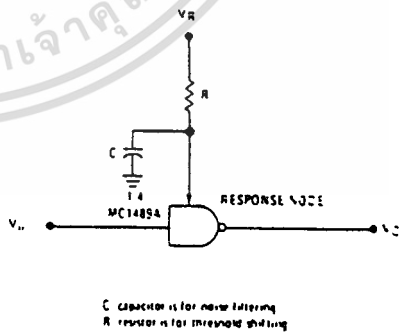


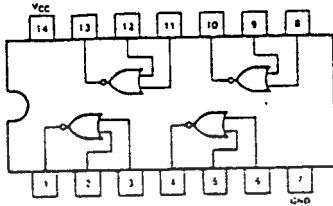
FIGURE 2 — RESPONSE CONTROL NODE





MOTOROLA

SN54/74LS02



J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-06 (Plastic)

QUAD 2-INPUT NOR GATE

LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74	0.35	0.5	V	
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total, Output HIGH Total, Output LOW			3.2	mA	V _{CC} = MAX
				5.4		

AC CHARACTERISTICS: T_A = 25°C

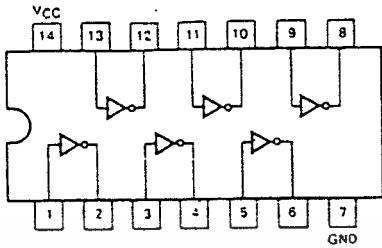
SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn Off Delay, Input to Output		10	15	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Turn On Delay, Input to Output		10	15	ns	



SN54/74LS04

HEX INVERTER

LOW POWER SCHOTTKY



J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-06 (Plastic)

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T _A	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54			4.0	mA
		74			8.0	

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5		
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN. V _{IN} = V _{IL} or V _{IH} per Truth Table
		74	0.35	0.5	V	
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total: Output HIGH Total: Output LOW			2.4	mA	V _{CC} = MAX
				6.6		

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn Off Delay, Input to Output		9.0	15	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Turn On Delay, Input to Output		10	15	ns	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA SEMICONDUCTOR TECHNICAL DATA

NPN
TIP100
TIP101*
TIP102*

PNP
TIP105
TIP106*
TIP107*

*Motorola preferred devices.

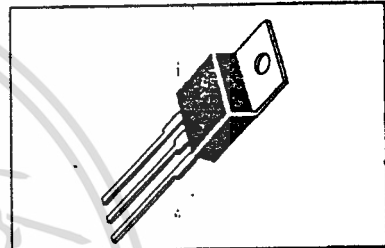
PLASTIC MEDIUM-POWER COMPLEMENTARY SILICON TRANSISTORS

... designed for general-purpose amplifier and low-speed switching applications.

- High DC Current Gain –
 $h_{FE} = 2500$ (Typ) @ $I_C = 4.0$ Adc
- Collector-Emitter Sustaining Voltage – @ 30 mAdc
 $V_{CE(sus)} = 60$ Vdc (Min) – TIP100, TIP105
 $= 80$ Vdc (Min) – TIP101, TIP106
 $= 100$ Vdc (Min) – TIP102, TIP107
- Low Collector-Emitter Saturation Voltage –
 $V_{CE(sat)} = 2.0$ Vdc (Max) @ $I_C = 3.0$ Adc
 $= 2.5$ Vdc (Max) @ $I_C = 8.0$ Adc
- Monolithic Construction with Built-In Base-Emitter Shunt Resistors
- TO-220AB Compact Package

DARLINGTON 8 AMPERE COMPLEMENTARY SILICON POWER TRANSISTORS

60-80-100 VOLTS
80 WATTS



***MAXIMUM RATINGS**

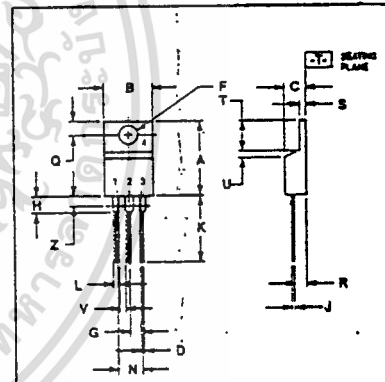
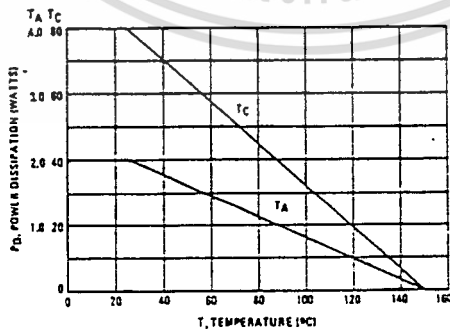
Rating	Symbol	TIP100, TIP105	TIP101, TIP106	TIP102, TIP107	Unit
Collector-Emitter Voltage	V_{CE}	60	80	100	Vdc
Collector-Base Voltage	V_{CB}	60	80	100	Vdc
Emitter-Base Voltage	V_{EB}	—	5.0	—	Vdc
Collector Current – Continuous	I_C	—	8.0	—	Adc
		—	15	—	
Base Current	I_B	—	1.0	—	Adc
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	—	80	—	Watts
		—	0.64	—	
Unclamped Inductive Load Energy (1)	E	—	30	—	mJ
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	—	2.0	—	Watts
		—	0.018	—	
Operating and Storage Junction Temperature Range	T_J, T_{stg}	—65 to +150			$^\circ\text{C}$

THERMAL CHARACTERISTICS

Characteristics	Symbol	Max	Unit
Thermal Resistance, Junction to Case	$R_{\theta JC}$	1.56	$^\circ\text{C/W}$
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	62.5	$^\circ\text{C/W}$

(1) $I_C = 1.1$ A, $L = 50$ mH, P.R.F. = 10 Hz, $V_{CC} = 20$ V, $R_{BE} = 100 \Omega$.

FIGURE 1 – POWER DERATING



NOTES

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION INCH.
3. DIM Z DEFINES A ZONE WHERE ALL BODY AND LEAD IRREGULARITIES ARE ALLOWED.

	MILLIMETERS	INCHES
DM	8.91 ± 0.13	0.351 ± 0.005
A	14.44 ± 0.15	0.572 ± 0.006
B	1.66 ± 0.28	0.065 ± 0.011
C	4.27 ± 0.42	0.168 ± 0.016
D	0.64 ± 0.08	0.025 ± 0.003
E	3.81 ± 0.25	0.150 ± 0.010
F	2.42 ± 0.25	0.095 ± 0.010
G	2.30 ± 0.25	0.091 ± 0.010
H	0.46 ± 0.04	0.018 ± 0.002
I	12.70 ± 0.27	0.500 ± 0.011
J	1.15 ± 0.12	0.045 ± 0.005
K	4.83 ± 0.13	0.190 ± 0.005
L	2.54 ± 0.25	0.100 ± 0.010
M	1.15 ± 0.25	0.045 ± 0.010
N	3.17 ± 0.25	0.125 ± 0.010
O	1.27 ± 0.25	0.050 ± 0.010
P	1.15 ± 0.25	0.045 ± 0.010
Q	1.15 ± 0.25	0.045 ± 0.010
R	1.15 ± 0.25	0.045 ± 0.010
S	1.15 ± 0.25	0.045 ± 0.010
T	1.15 ± 0.25	0.045 ± 0.010
U	0.20 ± 0.27	0.008 ± 0.011
V	1.15 ± 0.25	0.045 ± 0.010
W	—	—
Z	—	—

STYLE B
 1. BASE
 2. COLLECTOR
 3. EMITTER
 4. COLLECTOR

CASE 221A-06
TO-220AB

TIP100, TIP101, TIP102 NPN/TIP105, TIP106, TIP107 PNP

ELECTRICAL CHARACTERISTICS (T_C = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Sustaining Voltage (1) (I _C = 30 mA, I _B = 0)	TIP100, TIP105 TIP101, TIP106 TIP102, TIP107	V _{CE(sus)}	60 80 100	—	V _{dc}
Collector Cutoff Current (V _{CE} = 30 V _{dc} , I _B = 0) (V _{CE} = 40 V _{dc} , I _B = 0) (V _{CE} = 50 V _{dc} , I _B = 0)	TIP100, TIP105 TIP101, TIP106 TIP102, TIP107	I _{CEO}	— — —	50 50 50	μA _{dc}
Collector Cutoff Current (V _{CB} = 60 V _{dc} , I _E = 0) (V _{CB} = 80 V _{dc} , I _E = 0) (V _{CB} = 100 V _{dc} , I _E = 0)	TIP100, TIP105 TIP101, TIP106 TIP102, TIP107	I _{CBO}	— — —	50 50 50	μA _{dc}
Emitter Cutoff Current (V _{BE} = 5.0 V _{dc} , I _C = 0)		I _{EBO}	—	8.0	mA _{dc}

ON-CHARACTERISTICS (1)

DC Current Gain (I _C = 3.0 A _{dc} , V _{CE} = 4.0 V _{dc}) (I _C = 8.0 A _{dc} , V _{CE} = 4.0 V _{dc})	h _{FE}	1000 200	20,000 —	—
Collector-Emitter Saturation Voltage (I _C = 3.0 A _{dc} , I _B = 6.0 mA _{dc}) (I _C = 8.0 A _{dc} , I _B = 80 mA _{dc})	V _{CE(sat)}	—	2.0 2.5	V _{dc}
Base-Emitter On Voltage (I _C = 8.0 A _{dc} , V _{CE} = 4.0 V _{dc})	V _{BE(on)}	—	2.8	V _{dc}

DYNAMIC CHARACTERISTICS

Small-Signal Current Gain (I _C = 3.0 A _{dc} , V _{CE} = 4.0 V _{dc} , f = 1.0 MHz)	h _{fe}	4.0	—	—
Output Capacitance (V _{CB} = 10 V _{dc} , I _E = 0, f = 0.1 MHz)	C _{ob}	—	300 200	pF

(1) Pulse Test: Pulse Width < 300 μs, Duty Cycle < 2%.

FIGURE 2 - SWITCHING TIMES TEST CIRCUIT

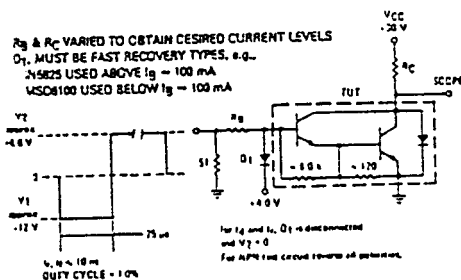
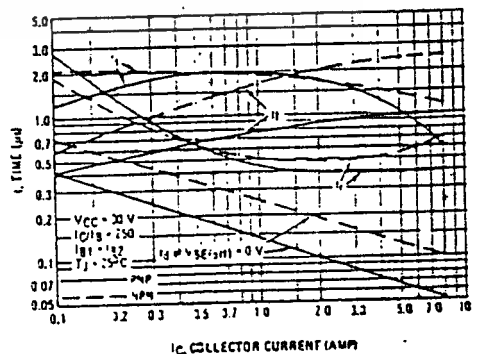


FIGURE 3 - SWITCHING TIMES



หนังสืออ้างอิง

จิตติ หนูแก้ว "เทคนิคการเชื่อมต่อ IBM PC กับอุปกรณ์ภายนอกเพื่อประยุกต์ใช้ในงานต่างๆ", ซีเอ็ดดูเคชั่น.

มนตรี พจนารถลาวัณย์ "การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี", ซีเอ็ดดูเคชั่น.

วิทยา เรืองพรพิสุทธ์ "คู่มือโปรแกรมภาษา C", ซีเอ็ดดูเคชั่น.

สหพัฒน์ รุ่งตะวันเรืองศรี "เรียนรู้คอมพิวเตอร์กราฟฟิค 2 มิติด้วยภาษาซี", ซีเอ็ดดูเคชั่น.

ธันวา ศรีประโม่ง "การเขียนโปรแกรมภาษา C สำหรับวิศวกรรม", วิทยาลัยมหานคร.

กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษา คืออาจารย์สมยศ จุณณะปิยะ ที่ช่วยให้คำปรึกษาแนะนำและให้ความช่วยเหลืออย่างดี ในการทำโครงการนี้จนสำเร็จลงได้

จึงขอขอบคุณมา ณ โอกาสนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้