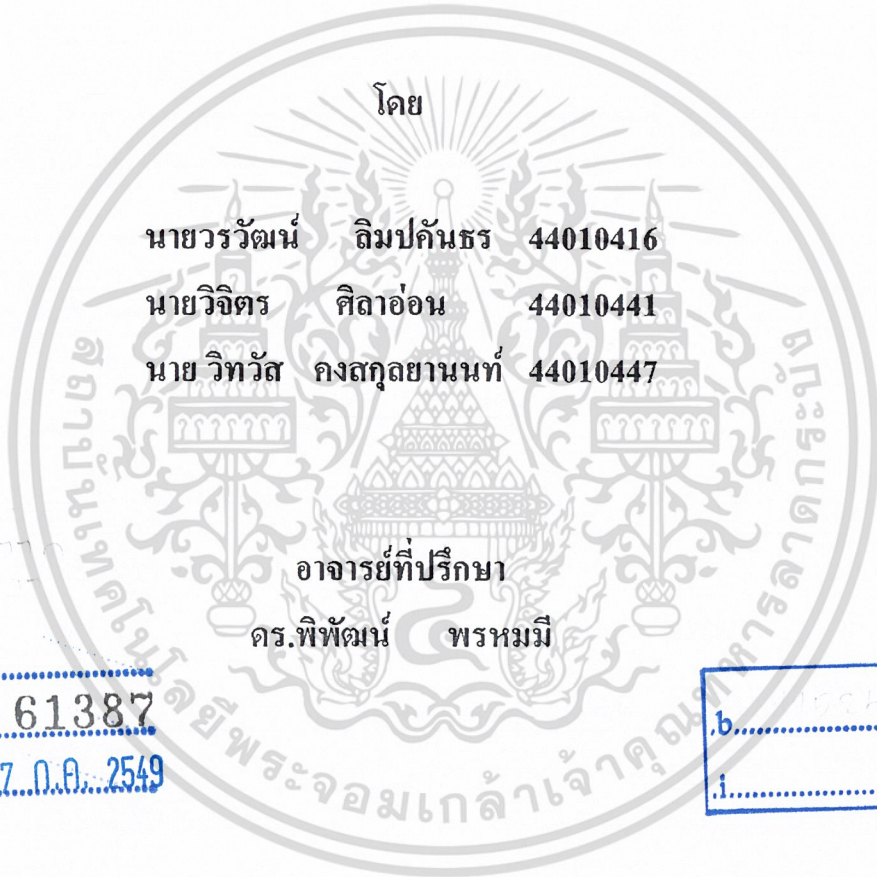


สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



คีย์บอร์ดและเมาส์ไร้สายโดยใช้ความถี่วิทยุ
RF Wireless Mouse and Keyboard



โดย

นายวรวัฒน์ ลิ้มปคันทร 44010416

นายวิจิตร ศิลาอ่อน 44010441

นาย วิทวัส คงสกุลยานนท์ 44010447

อาจารย์ที่ปรึกษา

ดร.พิพัฒน์ พรหมมี

เลขหมู่.....
เลขทะเบียน.....61387
วัน,เดือน,ปี.1.7.0.ศ. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กี๊ย์บอร์ด์และเมาส์ไร้สายโดยใช้ความถี่วิทยุ

RF Wireless Mouse and Keyboard



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชา
วิศวกรรมโทรคมนาคม

ปริญญาโทปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

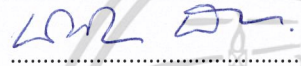
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

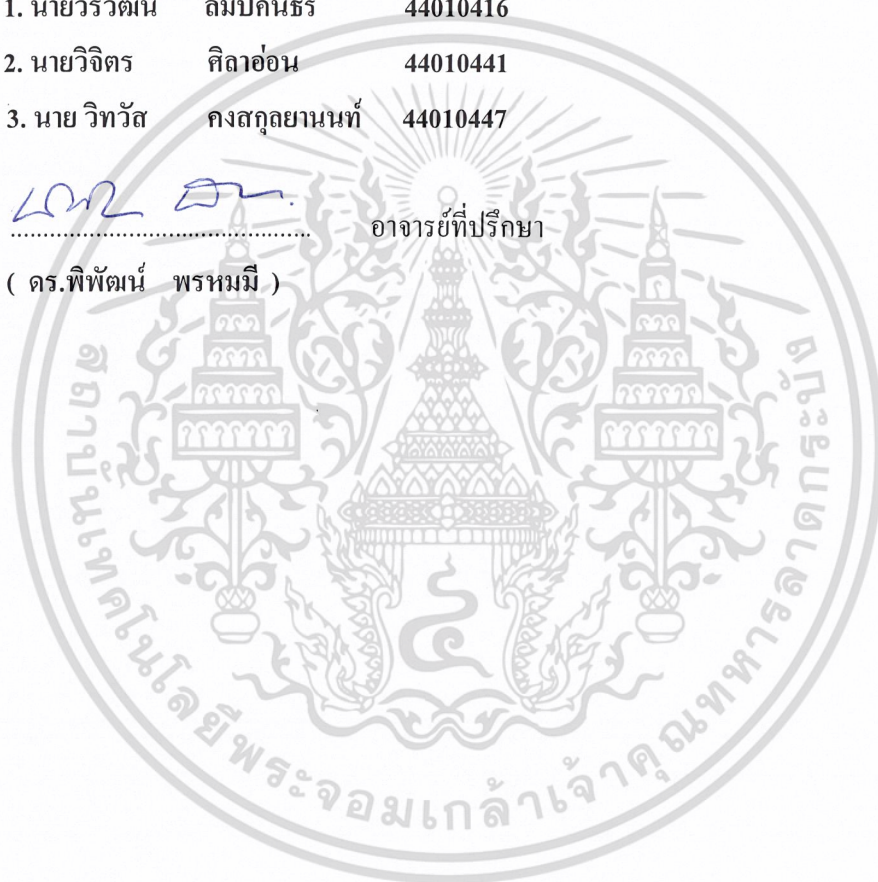
เรื่อง กีย์บอร์ดและเมาส์ไร้สายโดยใช้ความถี่วิทยุ

RF Wireless Mouse and Keyboard

ผู้จัดทำ

1. นายรวิวัฒน์ ลิ้มปลั้ณธร 44010416
2. นายวิจิตร ทิลาอ่อน 44010441
3. นาย วิทวัส กงสกุลยานนท์ 44010447


..... อาจารย์ที่ปรึกษา
(ดร.พิพัฒน์ พรหมมี)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์บอร์ดและเมาส์ไร้สายโดยใช้ความถี่วิทยุ

RF Wireless Mouse and Keyboard

โดย นายวรวัฒน์ ลิ้มปັນธร 44010416
นายวิจิตร ศีลาอ่อน 44010441
นายวิหวัศ คงสกุลยานนท์ 44010447

อาจารย์ที่ปรึกษา ดร. พิพัฒน์ พรหมมี

บทคัดย่อ

โครงการนี้นำเสนอคีย์บอร์ดและเมาส์แบบไร้สายโดยใช้ความถี่วิทยุ ซึ่งประกอบด้วยชุดเชื่อมต่อกับคอมพิวเตอร์และชุดเชื่อมต่อกับคีย์บอร์ดและเมาส์ โดยชุดเชื่อมต่อกับคีย์บอร์ดและเมาส์ แบบไร้สายจะใช้ไมโครคอนโทรลเลอร์ควบคุมการสื่อสารระหว่างชุดเชื่อมต่อโดยใช้โปรโตคอลที่กำหนดขึ้นและส่งผ่านตัวกลางที่เป็นความถี่วิทยุ

ABSTRACT

This project proposes a wireless mouse and keyboard interface based on radio frequency (RF). It consists of computer interface and mouse and keyboard interface subsystems. The interface subsystems are controlled by microcontroller for communicated by proprietary protocol. The protocols have sent over RF transmission media.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1		
บทนำ		1
บทที่ 2		
ส่วนของไมโครคอนโทรลเลอร์		
2.1 ทฤษฎีของไมโครคอนโทรลเลอร์		2
2.2 คุณสมบัติพิเศษของP89C51RD2 8-BitFlash microcontroller family		3
2.3 การเขียนโปรแกรมอินเทอร์พรีต		4
2.4 การโปรแกรมพอร์ทอนุกรม		8
2.5 การโปรแกรมไทม์เมอร์/เคาน์เตอร์		10
ส่วนของการอินเทอร์เฟซของเมาส์-คีย์บอร์ด		
2.6 ลักษณะทางกายภาพของพอร์ต PS/2		13
2.7 ทฤษฎีของ คีย์บอร์ด		13
2.8 ทฤษฎีของ PS/2 เมาส์		20
ส่วนของโปรโตคอลของเมาส์-คีย์บอร์ด		
2.9 รายละเอียดของแหล่งจ่ายไฟ		24
2.10 การติดต่อทั่วไป		25
2.11 ระบบการรับข้อมูล		28
2.12 ระบบการส่งข้อมูล		29
ส่วนของเครื่องรับ-ส่งวิทยุ		
2.13 การมอดูเลตแอมพลิฟูดแบบแถบข้างคู่ขั้วคลื่นพาห้		32
2.14 การมอดูเลตแอมพลิฟูดสำหรับวิทยุกระจายเสียง		37
2.15 วงจรตรวจจับกรอบสัญญาณ		38
2.16 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (D/A)		40
2.17 กำเนิดคลื่นเอ เอ็ม		43
2.18 การกู้สัญญาณเอเอ็ม		46
บทที่ 3		
การคำนวณ และ การสร้าง		49
3.1 รูปแบบของโปรแกรมที่ใช้ในระบบ		50
3.2 การเชื่อมต่อฮาร์ดแวร์ของระบบ		51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

การทดลองตอนที่ 1 และผลการทดลอง	57
การทดลองตอนที่ 2 และผลการทดลอง	58
การทดลองตอนที่ 3 และผลการทดลอง	63
การทดลองตอนที่ 4 และผลการทดลอง	66

บทที่ 5

บทวิจารณ์และสรุปผล

67

ภาคผนวก

กิตติกรรมประกาศ

บรรณานุกรม



สารบัญรูปลูกภาพ

รูปที่ 2.1	การจัดหน่วยความจำของไมโครคอนโทรลเลอร์	2
รูปที่ 2.2	การอินเทอร์รัพต์	5
รูปที่ 2.3	รูปแบบการรับส่งข้อมูลในโหมด 3	10
รูปที่ 2.4	แสดงเคาน์เตอร์	10
รูปที่ 2.5	แสดงการเปรียบเทียบข้อแตกต่างระหว่างไทม์เมอร์กับเคาน์เตอร์	11
รูปที่ 2.6	แสดงฐานเวลาของไทม์เมอร์	11
รูปที่ 2.7	ลักษณะทางกายภาพของพอร์ต PS/2	13
รูปที่ 2.8	โค้ดตัวอักษร	14
รูปที่ 2.9	โค้ด Extended	15
รูปที่ 2.10	แสดง Open-Collector Interface	25
รูปที่ 2.11	แสดงสัญญาณบนเส้นข้อมูล และ สัญญาณนาฬิกา ที่เกิดขึ้นเมื่อมีการสื่อสารในทิศทาง อุปกรณ์ไปยังโฮสต์	26
รูปที่ 2.12	ตัวอย่างการส่งข้อมูล 'A' จากคีย์บอร์ดไปยังพีซี	26
รูปที่ 2.13	แสดงสัญญาณบนเส้นข้อมูลและ สัญญาณนาฬิกา ที่เกิดขึ้นเมื่อมีการสื่อสาร ในทิศทาง โฮสต์ไปยังอุปกรณ์	28
รูปที่ 2.14	แสดงช่วงเวลาที่เกิดการสื่อสารระหว่างโฮสต์ไปอุปกรณ์	28
รูปที่ 2.15	เวลาต่างๆที่คอมพิวเตอร์ใช้ในการรับข้อมูล	29
รูปที่ 2.16	เวลาต่างๆ ที่คอมพิวเตอร์ใช้ในการส่งข้อมูล	30
รูปที่ 2.17	รูปแบบของการสื่อสารในการรับส่งสัญญาณ	31
รูปที่ 2.18	กระบวนการมอดูเลตสัญญาณดิจิตอล	34
รูปที่ 2.19	การดีมอดูเลตสัญญาณดิจิตอล และ สัญญาณต่างๆ พร้อมทั้งค่าความหนาแน่น สเปกตรัมของสัญญาณที่อินพุตของวงจรกรองความถี่ต่ำผ่าน	35
รูปที่ 2.20	สัญญาณเอเอ็ม ที่เปลี่ยนแปลงตามสถานะของสัญญาณข้อมูล	38
รูปที่ 2.21	ประกอบการทำงานของการทำงานของวงจรตรวจจับรอบสัญญาณ	39
รูปที่ 2.22	แผนผังระบบการสร้างสัญญาณเอเอ็มแบบง่าย	40
รูปที่ 2.23	การแปลงสัญญาณ D/A แบบ FSK	41
รูปที่ 2.24	การแปลงสัญญาณ D/A แบบ PSK	41
รูปที่ 2.25	รูปแบบของการมอดูเลตทางแอมพลิจูด	42
รูปที่ 2.26	ON-OFF ASK	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.27 การแปลงสัญญาณ D/A แบบ ASK	42
รูปที่ 2.28 วงจรคอลเลกเตอร์มอดูเลชัน	43
รูปที่ 2.29 วงจรคอลเลกเตอร์มอดูเลชัน สำหรับไฮเลเวลมอดูเลต	44
รูปที่ 2.30 วงจรเบสมอดูเลชันแบบอนุกรม	44
รูปที่ 2.31 วงจรเบสมอดูเลชันแบบอนุกรมแบบขนาน	45
รูปที่ 2.32 วงจรอิมิตเตอร์มอดูเลชัน	45
รูปที่ 2.33 อุปกรณ์ที่ไม่เป็นเชิงเส้นที่ถูกต้องใช้เป็นเครื่องรับ	47
รูปที่ 2.34 เครื่องรับแบบ ไดโอด (Diode detector)	48
รูปที่ 3.1 บล็อกไดอะแกรมด้านเม้าส์ และคีย์บอร์ด	49
รูปที่ 3.2 บล็อกไดอะแกรมด้านคอมพิวเตอร์	49
รูปที่ 3.3 แสดงโปรแกรม Keil uVision	50
รูปที่ 3.4 โปรแกรม P89C51RD2	51
รูปที่ 3.5 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับโฮสต์	52
รูปที่ 3.6 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์	52
รูปที่ 3.7 วงจรภาคส่งความถี่ 40.680MHz	53
รูปที่ 3.8 วงจรด้านรับ 40.680MHz	54
รูปที่ 3.9 แสดงวงจรคอมพิวเตอร์	54
รูปที่ 3.10 แสดงขาใน ไอซี 74F14 Hex inverter	55
รูปที่ 3.11 แสดงวงจรจ่ายไฟ	56
รูปที่ 4.1 ตัวอย่างข้อมูลที่ได้รับการกดปุ่ม "A" (1CH)	57
รูปที่ 4.2 ตัวอย่างข้อมูลที่ได้รับการกดปุ่ม "S" (1BH)	57
รูปที่ 4.3 ตัวอย่างข้อมูลที่ได้รับการกดปุ่ม "F6" (0BH)	57
รูปที่ 4.4 บล็อกไดอะแกรมการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์กับคีย์บอร์ด	58
รูปที่ 4.5 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม "A" (1CH) ก่อนเข้าไมโครคอนโทรลเลอร์	58
รูปที่ 4.6 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม "S" (1BH) ก่อนเข้าไมโครคอนโทรลเลอร์	59
รูปที่ 4.7 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม "Enter" (5AH) ก่อนเข้าไมโครคอนโทรลเลอร์	59
รูปที่ 4.8 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม "F6" (0BH) ก่อนเข้าไมโครคอนโทรลเลอร์	60
รูปที่ 4.9 ผลจากการกดปุ่ม "A" (1CH) สามารถแสดงผลได้ดังรูป (Print screen)	60
รูปที่ 4.10 ผลจากการกดปุ่ม "S" (1BH) สามารถแสดงผลได้ดังรูป (Print screen)	61
รูปที่ 4.11 ผลจากการกดปุ่ม "Enter" (5AH) สามารถแสดงผลได้ดังรูป (Print screen)	61
รูปที่ 4.12 รูปแสดงการส่งข้อมูลเมื่อกดปุ่มขวาของเม้าส์ก่อนเข้าไมโครคอนโทรลเลอร์	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.13 ผลจากการกดปุ่มขวาของเมาส์ สามารถแสดงผลได้ดังรูป	62
รูปที่ 4.14 รูปการวัดโดยวัดอินพุต (Ch1) เทียบกับเอาต์พุต (Ch2) ของภาคส่ง	63
รูปที่ 4.15 อินพุตที่ป้อนกับภาคส่งกับเอาต์พุตจากภาคส่งเทียบกัน	63
รูปที่ 4.16 รูปการวัด โดยวัดอินพุต (Ch1) เทียบกับเอาต์พุต (Ch2) ของภาครับ	64
รูปที่ 4.17 อินพุตที่ภาครับเทียบกับเอาต์พุตที่ภาครับ	64
รูปที่ 4.18 รูปการวัด โดยวัดเอาต์พุตภาคส่ง(Ch1) เทียบกับอินพุต (Ch2) ของภาครับ	65
รูปที่ 4.19 เอาต์พุตของภาคส่งเทียบกับอินพุตที่ภาครับ	65
รูปที่ 4.20 รูปการวัด โดยวัดอินพุตภาคส่ง(Ch1) เทียบกับเอาต์พุต (Ch2) ของภาครับ	66
รูปที่ 4.21 อินพุตที่ภาคส่งเทียบกับเอาต์พุตภาครับ	66



สารบัญตาราง

ตารางที่ 2.1	แสดงค่าการตั้งค่าระดับความสำคัญของอินเทอร์เน็ต	4
ตารางที่ 2.2	แสดงบิตภายใน IE	6
ตารางที่ 2.3	แสดงบิตภายใน IP	6
ตารางที่ 2.4	หมายเลขอินเทอร์เน็ตของอินเทอร์เน็ต	7
ตารางที่ 2.5	แสดงบิตภายใน TCON	7
ตารางที่ 2.6	แสดงบิตภายใน PCON	8
ตารางที่ 2.7	แสดงบิตภายใน SCON	9
ตารางที่ 2.8	แสดงการเลือกโหมด	9
ตารางที่ 2.9	บิตภายใน TMOD	12
ตารางที่ 2.10	หน้าที่บิตภายใน TMOD	12
ตารางที่ 2.11	แสดงโหมดภายใน TMOD	12
ตารางที่ 2.12	ตัวอย่างโค้ด	15
ตารางที่ 2.13	ไบต์แสดงสถานะต่างๆ	18
ตารางที่ 2.14	สถานะแฟลคของ Status Register	20
ตารางที่ 2.15	การรายงานค่าของเคาน์เตอร์	21
ตารางที่ 2.16	รายละเอียดแต่ละบิตของเมาส์	22
ตารางที่ 2.17	ไบต์แสดงสถานะต่างๆของเมาส์	24
ตารางที่ 2.18	รายละเอียดของเวลาในการรับข้อมูล	29
ตารางที่ 2.19	รายละเอียดของเวลาในการส่งข้อมูล	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันคอมพิวเตอร์เป็นสิ่งจำเป็นสำหรับชีวิตประจำวันของคนเราเป็นอย่างมาก จนแทบจะเรียกว่าเป็นปัจจัย 5 ที่เพิ่มเข้ามาอย่างหลีกเลี่ยงไม่ได้ และคนที่ใช้คอมพิวเตอร์จะต้องมีการติดต่อสื่อสารกับมันโดยใช้อุปกรณ์ อินพุต ที่เรารู้จักกันดีก็คือ เมาส์ และ คีย์บอร์ด เป็นของที่ต้องใช้ทำงานร่วมกันเพื่อใช้ติดต่อสื่อสารกับคอมพิวเตอร์นั่นเอง และยิ่งในการประชุมสัมมนา การเสนอชิ้นงาน โครงการ และการวิจัยต่างๆ อาจจะเป็นที่จะต้องมีการติดต่อกับคอมพิวเตอร์ในระยะทางไกลเพื่อความสะดวกในการนำเสนอแก่ผู้คนที่สนใจ โดยได้มีการคิดค้น ระบบสื่อสารไร้สายที่เรียกว่า wireless ขึ้น ซึ่งได้นำเอาความรู้ทางด้านโทรคมนาคมมาใช้ในการ โมดูเลต (modulate) สัญญาณ เพื่อให้สามารถส่งข้อมูลไปในอากาศในระยะทางที่ไกล ๆ ได้ แต่ในการรับ-ส่งกันนั้น อาจมีการผิดพลาด (error) ของสัญญาณจากปัจจัยต่าง ๆ ทั้งภายในและภายนอกของระบบ ดังนั้นการที่เราจะรู้ว่าข้อมูลที่เรารับมาถูกต้องหรือไม่นั้นจึงจำเป็นที่จะต้องมียระบบที่สามารถตรวจสอบการทำงานของมันเป็นได้ ปัจจุบัน โปรโตคอล ได้มีบทบาทในระบบการสื่อสารข้อมูล (data communication) เพราะมันสามารถที่จะใช้จัดการกับข้อมูลให้เป็นระบบแบบแผนแล้วยังสามารถที่จะตรวจสอบความผิดพลาดของข้อมูลได้เป็นอย่างดีอีกด้วย เพราะฉะนั้น โปรโตคอล จึงเป็นสิ่งจำเป็นอย่างมากในการติดต่อสื่อสารข้อมูลที่ต้องการความถูกต้อง

โครงการนี้ได้นำเอา เมาส์ และ คีย์บอร์ดมาทำเป็น การสื่อสารไร้สายโดยใช้คลื่นวิทยุ (RF wireless) เพื่อใช้กับงานที่กล่าวไปข้างต้น โดยการใช้ไมโครคอนโทรลเลอร์ (microcontroller) ในการประมวลผลข้อมูลต่าง ๆ ทางโปรโตคอล และใช้วงจรคลื่นวิทยุ ในการรับ-ส่งข้อมูล ระหว่างอุปกรณ์ อินพุต (เมาส์และคีย์บอร์ด) กับ คอมพิวเตอร์

บทที่ 2

ทฤษฎี และ หลักการ

ในบทนี้จะมี 4 ส่วนใหญ่ ๆ คือ ส่วนของไมโครคอนโทรลเลอร์ ส่วนของการอินเตอร์เฟซของเมมส์-คีย์บอร์ด ส่วนของโปรโตคอลของเมมส์-คีย์บอร์ด และส่วนของเครื่องรับส่งวิทยุ

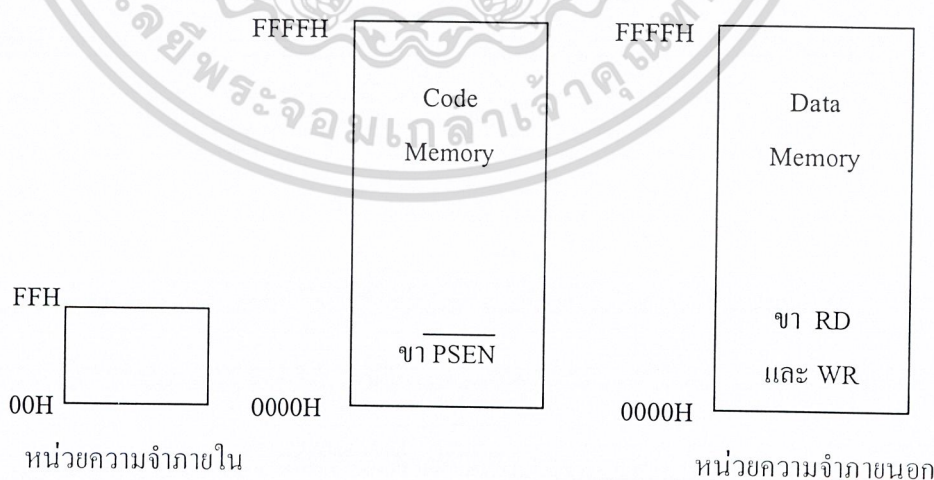
ส่วนของไมโครคอนโทรลเลอร์

2.1 ทฤษฎีของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ P89C51RD2 เป็นไมโครคอนโทรลเลอร์ที่มีหลักการทำงานเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ดังนั้นจึงเสนอหลักการทำงานของ MCS-51 บางประการที่สำคัญและใช้ในโครงการดังนี้

โครงสร้างหน่วยความจำ

หน่วยความจำสำหรับไมโครคอนโทรลเลอร์ จะมี 2 ชนิด คือ หน่วยความจำที่ใช้เก็บโปรแกรมหรือ รอม (ROM) กับหน่วยความจำที่ใช้เก็บข้อมูลในการประมวลผล หรือ แรม (RAM) ไมโครคอนโทรลเลอร์บางเบอร์เช่น 8051, 8052 จะมีหน่วยความจำโปรแกรมภายในไอซีและไมโครคอนโทรลเลอร์ ทุกเบอร์สามารถอ้างหน่วยความจำโปรแกรมภายนอกได้มากที่สุด 64 กิโลไบต์ สำหรับหน่วยความจำแรมภายใน จะประกอบด้วยพื้นที่ใช้งานทั่วไป, รีจิสเตอร์แบงก์, พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ เราอาจเขียนไออะแกรมของหน่วยความจำของ 8031 ได้ดังรูปที่ 2.1 โดยในรูปจะบอกด้วยว่าขาใดจะเอ็คที่ฟ



รูปที่ 2.1 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์

ใน 8031 จะมีหน่วยความจำภายในตั้งแต่ตำแหน่ง 00H ถึง FFH และสามารถอ้างหน่วยความจำ

โปรแกรมภายนอกได้ 64 กิโลไบต์ ตำแหน่ง ถ้าอ่านข้อมูลจากหน่วยความจำโปรแกรมขา PSEN จะแอกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พี นอกจากนี้ 8031 สามารถ อ้างหน่วยความจำภายนอกได้ 64 กิโลไบต์ ตำแหน่ง โดยการติดต่อกับ หน่วยความจำนี้ ขา RD และ WR จะแอกที่พี สำหรับหน่วยความจำข้อมูลภายในนั้นจะแบ่งออกได้ดังนี้

- 1) ชุดรีจิสเตอร์ 4 ชุด แต่ละชุดเรียกว่ารีจิสเตอร์เบงค์ ที่ตำแหน่ง 00H ถึง 1FH โดยแต่ละชุด ประกอบรีจิสเตอร์ R0 ถึง R7
- 2) หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ตำแหน่ง 20H ถึง 2FH
- 3) หน่วยความจำที่ใช้งานทั่วไปตำแหน่ง 30H ถึง 7FH
- 4) รีจิสเตอร์ ฟังก์ชันพิเศษ ตำแหน่ง 80H ถึง FFH

แผนผังการจัดหน่วยความจำข้อมูลแสดงได้ดังรูปที่ 2.2 จากแผนผังจะเห็นได้ว่าการอ้างตำแหน่ง หน่วยความจำภายในจะอ้างได้สองแบบ คือ การอ้างไปที่ตำแหน่งของไบต์ (เขียนหมายเลขตำแหน่งด้านนอก) หรืออ้างไปที่ตำแหน่งของบิต (เขียนหมายเลขตำแหน่งด้านใน) โดยตำแหน่งของหน่วยความจำที่ อ้างเป็นแบบบิตได้จะมีตำแหน่งบิตที่แน่นอน

รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

ในไมโครคอนโทรลเลอร์ รีจิสเตอร์จะใช้หน่วยความจำ แรม ภายในไอซี โดยส่วนหนึ่งเป็น รีจิสเตอร์พิเศษ (Special Function Register: SFR) ซึ่งมีทั้งหมด 21 ตัว โดยรีจิสเตอร์พิเศษต่างๆ จะเริ่มที่ หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษ เพียง 21 ตำแหน่ง แต่ถ้าเป็น 8032 / 8051 จะใช้ 26 ตำแหน่ง หรือมีรีจิสเตอร์พิเศษ 26 ตัว

2.2 คุณสมบัติพิเศษของP89C51RD2 8-BitFlash microcontroller family

ไมโครคอนโทรลเลอร์รุ่น P89C51RD2 เป็นไมโครคอนโทรลเลอร์ที่อยู่ในตระกูลของ MCS-51 เหมือนไมโครคอนโทรลเลอร์ทั่วไป แต่จะมีคุณสมบัติพิเศษซึ่งต่างจากไมโครคอนโทรลเลอร์ตัวอื่น ๆ ก็คือ สามารถทำการโปรแกรมแบบ อินซิทเต็ม โปรแกรมมิ่งได้นอกจากนั้นยังมีข้อดีอื่นๆ ที่เหนือกว่า ไมโครคอนโทรลเลอร์อื่นๆอีกเช่น

2.2.1 เลือกการทำงานได้ 2 แบบ คือ แบบ 6 คล็อกและ 12 คล็อก

P89C51RD2 สามารถเลือกที่จะทำงานเร็วขึ้นอีก 1 เท่า ด้วยคริสตอล (Xtal) ค่าเดิม กล่าวคือ สามารถเลือกได้ว่าจะทำงานในโหมด 12 คล็อกหรือ 6 คล็อกถ้าเลือก 12 คล็อกการทำคำสั่ง 1 คำสั่งก็จะใช้ 12 คล็อกตามปกติ ถ้าเลือกโหมด 6 คล็อกการทำคำสั่ง 1 คำสั่งก็จะใช้เพียง 6 นาฬิกาเท่านั้น โดยปกติทางโรงงานจะกำหนดไอซีทำงานในโหมด 6 คล็อกมาอยู่แล้ว แต่ผู้ใช้อย่างยังสามารถโปรแกรมให้ทำงานแบบ 12 คล็อกตามเดิมก็ได้ เพื่อที่จะนำมาใช้ กับเครื่องโปรแกรมแบบขนาน (Parallel) และที่สำคัญคือเมื่อเราเปลี่ยนให้กลายเป็นการโปรแกรมให้เป็นแบบ 12 คล็อกจะไม่สามารถกำหนดย้อนกลับไปเป็นแบบ 6 คล็อกได้อีก เพราะฉะนั้น ต้องตัดสินใจเลือกให้เด็ดขาดก่อน

2.2.2 มี ONCE™ Mode

ONCE ("On-Circuit Emulation) โหมดเป็นโหมดที่ให้การสนับสนุนในการทดสอบระบบ โดยที่ไม่ต้องถอดตัวไอซีออกมาจากวงจร ในการใช้ ONCE โหมดต้องกำหนดค่าต่างๆ ดังนี้

- 1) ให้ ALE เป็น LOW ขณะที่กำลังเซตวงจร และ ให้ PSEN เป็น HIGH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ค่า ALE ให้เป็น LOW ใ้จน RST ลึ้นสุดลง ขณะที่ยังจรออยู่ใน ONCE โหมด เราสามารถใช้อิมมูเลเตอร์ ในการขั้ววงจร การกับสู่การทำงานปกติทำได้โดยการ รีเซต

2.2.3 มีแหล่งกำเนิดอินเทอร์รัพต์ 7 แหล่ง 4 ระดับ (Interrupt Priority Structure 7Source4Level)

ปกติในไมโครคอนโทรลเลอร์ 8051 ทั่วไป จะมี 2 ระดับอินเทอร์รัพต์ แต่ในไมโครคอนโทรลเลอร์เบอร์ P89C51RD2 จะมีถึง 4 ระดับอินเทอร์รัพต์ โดยจะมี รีจิสเตอร์ฟังก์ชันพิเศษ 3 ตัวที่เป็นตัวสร้าง อินเทอร์รัพต์ คือ IE, IP และ IPH โดย IPH จะอยู่ที่ตำแหน่ง B7H ของ รีจิสเตอร์ฟังก์ชันพิเศษ IPH กับ รีจิสเตอร์ฟังก์ชันพิเศษ IP ดังตารางที่ 2.1

PRIORITY BITS		INTERRUPT PRIORITY LEVEL
IPH.x	IP.x	
0	0	Level 0 (Lowest Priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (Highest Priority)

ตารางที่ 2.1 แสดงค่าการตั้งค่าระดับความสำคัญของอินเทอร์รัพต์

2.2.4 มีค่าตำแหน่งเตอร์ (DPTR) 2 ตัว

ใน P89C51RD2 มีค่าตำแหน่งเตอร์ 2 ตัว โดยเราสามารถเลือกว่าจะใช้ตัวใดได้โดย เลือกเซตบิต DPS ในรีจิสเตอร์ฟังก์ชันพิเศษ AUXR1 (ตำแหน่ง A2H) โดย

เลือก DPTR0 ; DPS=0

เลือก DPTR1 ; DPS=1

2.2.5 แฟลช อีพรอม เมโมรี่ (Flash EPROM Memory)

ในไมโครคอนโทรลเลอร์ P89C51RD2 มีหน่วยความจำ แฟลชเมโมรี่ 64 กิโลไบต์ ซึ่งแบ่งเป็น 5 บล็อก โดยสองบล็อกแรกจะมีขนาด 8 กิโลไบต์ โดยมีแอดเดรสตั้งแต่ 0000F ถึง 3FFFH และ อีก 3 บล็อกที่เหลือมีขนาด 16 กิโลไบต์ โดยมีแอดเดรสตั้งแต่ 4000H ถึง FFFFH

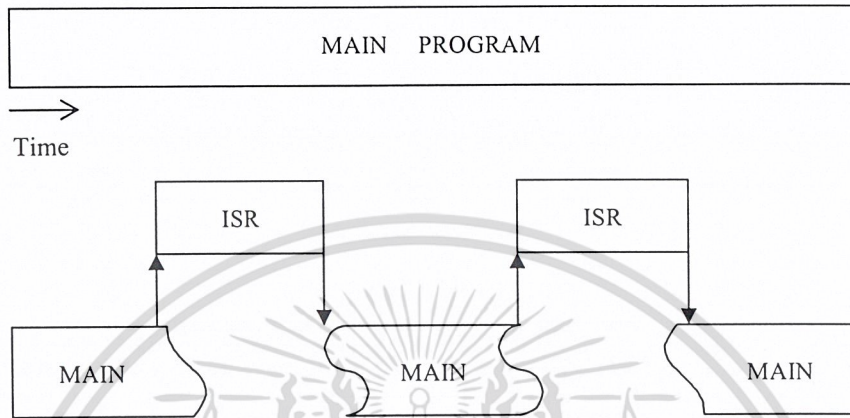
บูตโรม (Boot ROM) เป็นแฟลชเมโมรี่ ที่มีขนาด 1 กิโลไบต์ อยู่ที่แอดเดรส FC00H ถึง FFFFH ซึ่ง บูตโรม สามารถปิด (ไม่นำมาใช้งาน) ได้ดังนั้นเมื่อเราไม่ได้ใช้งาน บูตโรมแฟลชเมโมรี่ (Boot ROM Flash memory) 1 กิโลไบต์บนก็จะนำไปใช้ได้เป็น แฟลชเมโมรี่ ได้ตามปกติ

2.3 การเขียนโปรแกรมอินเทอร์รัพต์

การอินเทอร์รัพต์คือการขัดจังหวะการทำงานของโปรแกรมหลัก โดยให้โปรแกรมหลักหยุดชั่วคราวแล้วกระโดดไปทำงานในส่วนโปรแกรมบริการอินเทอร์รัพต์ (Interrupt Service Routine) เมื่อทำงานในโปรแกรมบริการเสร็จเรียบร้อยแล้วโปรแกรมจะกลับไปทำงานโปรแกรมหลักต่อไป

เหตุผลสำคัญที่ต้องมีการใช้การอินเทอร์รัพต์ เนื่องจากขณะที่มีการใช้อุปกรณ์ต่อพ่วง (Peripheral เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้หมายไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Device) หลายอุปกรณ์เข้ากับซีพียู จนซีพียูต้องทำการวนรอเพื่อตรวจสอบการร้องขอการบริการจากอุปกรณ์ต่อพ่วงเพื่อร้องขอการทำงาน จะทำให้เกิดการเสียเวลา ถ้าอุปกรณ์ต่อพ่วงมีจำนวนมาก เทคนิคการใช้วิธีการอินเตอร์รัพต์จึงช่วยลดเวลา การทำงานแบบวนรอได้โดยเปลี่ยนมาเป็นการบริการ เมื่อมีการร้องขอมาแบบขัดจังหวะมาเท่านั้น



รูปที่ 2.2 การอินเตอร์รัพต์

เมื่อเกิดการอินเตอร์รัพต์ที่เวลาใด ๆ จะทำให้การทำงานของโปรแกรมหลักหยุดไป แล้วกระโดดไปทำงานของโปรแกรมบริการอินเตอร์รัพต์ (ISR) สำหรับ MCS-51 จะมีแหล่งของอินเตอร์รัพต์อยู่ 5 แหล่งดังนี้

จากภายใน

1. Timer 0
2. Timer 1
3. Serial

จากภายนอก

1. INTO
2. INT1

สำหรับการอินเตอร์รัพต์จากภายในจะเป็นไทม์เมอร์ 0, ไทม์เมอร์ 1 และพอร์ทอนุกรม โดยจะตรวจสอบการเกิดอินเตอร์รัพต์จากแฟล็กโอเวอร์โฟลว์คือ TF1, TF0 ส่วนพอร์ทอนุกรมจะตรวจสอบจาก TI หรือ RI

การเกิดอินเตอร์รัพต์จากภายนอกจะใช้ขาสัญญาณ INTO และ INT1 ขาสัญญาณนี้จะทำงานที่ลอจิก “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมใช้งานอินเทอร์รัพต์

ในตัวไมโครคอนโทรลเลอร์ MCS-51 จะมีรีจิสเตอร์ สำหรับโปรแกรมการทำงานอินเทอร์รัพต์อยู่ 3 ตัว คือ IE (Interrupt Enable), IP (Interrupt Priority), TCON (Timer Control) ซึ่งรายละเอียดจะเป็นดังนี้

1. รีจิสเตอร์ IE (Interrupt Enable) อยู่ตำแหน่งแอดเดรส 0A8H สามารถอ้างตำแหน่งแบบบิตได้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

ตารางที่ 2.2 แสดงบิตภายใน IE

EA	ถ้าเป็นลอจิก “1” หมายความว่าให้อินเทอร์รัพต์ได้
ET2	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล Timer 2 (ใช้กับเบอร์ที่มี Timer 2)
ES	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล อินเทอร์รัพต์จากพอร์ทอนุกรม
ET1	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล Timer 1
EX1	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล สัญญาณอินเทอร์รัพต์จากภายนอกที่เข้ามาทางขา INT1
ET0	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล Timer 0
EX0	ถ้าเป็นลอจิก “1” จะเอนนาเบิ้ล สัญญาณอินเทอร์รัพต์จากภายนอกที่เข้ามาทางขา INT0

2. รีจิสเตอร์ IP (Interrupt Priority) อยู่ตำแหน่งแอดเดรส 0B8H สามารถอ้างตำแหน่งแบบบิตได้ ใช้กำหนดลำดับของการอินเทอร์รัพต์ กรณีที่เกิดการอินเทอร์รัพต์จากหลายแหล่งพร้อม ๆ กัน

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
-	-	PT2	PS	PT1	PX1	PT0	PX0

ตารางที่ 2.3 แสดงบิตภายใน IP

PT2	สงวนไว้ใช้งานภายใน
PS	การอินเทอร์รัพต์จาก พอร์ทอนุกรม
PT1	การอินเทอร์รัพต์จาก Timer1
PX1	การอินเทอร์รัพต์จาก INT1
PT0	การอินเทอร์รัพต์จาก Timer0
PX0	การอินเทอร์รัพต์จาก INT 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่มี การกำหนดลำดับความสำคัญของแหล่งการเกิดอินเทอร์รัพต์สูงสุดเท่ากัน ซึ่พียู จะเรียงลำดับความสำคัญ จากการโพล (Polling) ภายในโครงสร้างโดยมีการเรียงลำดับดังนี้

IE0	-	ลำดับความสำคัญสูงสุด
TF0	-	ลำดับความสำคัญสูงสุด
IE1	-	ลำดับความสำคัญสูงสุด
TF1	-	ลำดับความสำคัญสูงสุด
Serial	-	ลำดับความสำคัญสูงสุด

การเขียนโปรแกรมด้วยภาษาซีนั้น โปรแกรมตอบสนองการอินเทอร์รัพต์จะเขียนเหมือนกับ ฟังก์ชันทั่วไป แต่จะใช้คำว่า อินเทอร์รัพต์ และหมายเลขอินเทอร์รัพต์ต่อท้ายฟังก์ชันนั้น โดยหมายเลข อินเทอร์รัพต์ของอินเทอร์รัพต์แต่ละตัวจะเป็นดังนี้

แหล่งกำเนิดสัญญาณอินเทอร์รัพต์	หมายเลขอินเทอร์รัพต์
อินเทอร์รัพต์จากภายนอก (IE0)	0
อินเทอร์รัพต์ไทม์เมอร์ (TF0)	1
อินเทอร์รัพต์จากภายนอก (IE1)	2
อินเทอร์รัพต์ไทม์เมอร์ (TF1)	3
อินเทอร์รัพต์พอร์ตอนุกรม	4

ตารางที่ 2.4 หมายเลขอินเทอร์รัพต์ของอินเทอร์รัพต์

3. รีจิสเตอร์ TCON (Timer Control) เป็นรีจิสเตอร์ขนาด 8 บิตอยู่ตำแหน่งที่ 88H สามารถอ้างตำแหน่งแบบบิตได้ ใช้ในการควบคุมการทำงานของไทม์เมอร์และควบคุมการทำงานของอินเทอร์รัพต์ บิตต่างๆเป็นดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

ตารางที่ 2.5 แสดงบิตภายใน TCON

TF1 เป็นบิตโอเวอร์โฟลว์ของไทม์เมอร์ 1 จะเป็นลอจิก “1” เมื่อไทม์เมอร์เกิดโอเวอร์โฟลว์ และบิตนี้สามารถอินเทอร์รัพต์ไมโครคอนโทรลเลอร์ได้ เมื่อทำโปรแกรมตอบสนองการอินเทอร์รัพต์จบ บิต TF1 นี้จะกลับมาเป็นลอจิก “0”

TR1 ใช้ปิดเปิดไทม์เมอร์ 1

TF0 เหมือนกับ TF1 แต่ใช้กับไทม์เมอร์ 0

TR0 ใช้ปิดเปิดไทม์เมอร์ 0

IE1 เป็นบิตแสดงการอินเทอร์รัพต์ทางฮาร์ดแวร์ที่เข้ามาทางขา INT1 ว่ามีลักษณะใด ถ้าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น “1” หมายความว่า จะเกิดการอินเทอร์รัพต์เมื่อมีสัญญาณขอบขาลงเข้ามา ถ้าเป็นลอจิก “0” หมายความว่า จะเกิดการอินเทอร์รัพต์เมื่อมีระดับลอจิก “0”

IE0 ใช้งานเหมือน IE1 แต่จะใช้กับ INTO

IT0 ใช้งานเหมือนกับ IT1 แต่จะใช้กับ INTO

2.4 การโปรแกรมพอร์ทอนุกรม

โดยปกติแล้ว MCS-51 จะเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถในการรับข้อมูลจากภายนอกและนำมาประมวลผล พร้อมทั้งสามารถส่งสัญญาณเพื่อทำการควบคุมการทำงานของอุปกรณ์ต่างๆ ได้อย่างดี และในส่วนของ การติดต่อสื่อสารข้อมูล (Data Communication) กับระบบภายนอกอื่น ๆ ก็สามารถกระทำโดยผ่านทางพอร์ทอนุกรม (Serial Port) ซึ่งพอร์ทอนุกรมนี้ จะเป็นส่วนที่เหมาะสมในการรับ หรือส่งข้อมูลในระยะทางไกลได้ดีกว่าพอร์ทขนาน

รีจิสเตอร์ต่างๆที่เกี่ยวข้องในการใช้งานพอร์ทอนุกรม

1) รีจิสเตอร์ควบคุมไทม์เมอร์เนื่องจากว่า การใช้งานพอร์ทอนุกรมนั้นมีสิ่งที่จะต้องคำนึงถึงคือ อัตราการรับ-ส่งข้อมูล หรือ เรียกว่า อัตราบอด (Baud Rate) จริง ๆ แล้ว ก็คือจังหวะการเลื่อนข้อมูลเข้าหรือออกจาก MCS-51 นั่นเอง โดยอัตราบอดนี้สามารถสร้างขึ้นภายในชิพของ MCS-51 ได้จาก ไทม์เมอร์แซนแนล 1 โดยทำงานในโหมด 2 คือ โหมดค่ากลับอัตโนมัติ ดังนั้นรีจิสเตอร์ที่ต้องทำการโปรแกรม มีดังนี้

- TMOD ตำแหน่ง 89H ทำหน้าที่เลือกโหมดของไทม์เมอร์

- TCON ตำแหน่ง 88H ทำหน้าที่เริ่มต้นการสร้างอัตราบอด

- TH1 ตำแหน่ง 8CH ทำหน้าที่ใส่ข้อมูลการนับของไทม์เมอร์ 1 เพื่อสร้างอัตราบอด

2) รีจิสเตอร์ควบคุมการลดกำลัง เนื่องจากว่า การสร้างอัตราบอดนั้นจะต้องนำบิตในรีจิสเตอร์ PCON มาใช้ในการคำนวณข้อมูลของ TH1 ดังนั้นรีจิสเตอร์ที่ใช้ก็คือ

- PCON ตำแหน่ง 87H ทำหน้าที่ในการคำนวณข้อมูลที่จะใส่ในรีจิสเตอร์ TH1

ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SMOD	-	-	-	GF1	GF0	PD	IDL

ตารางที่ 2.6 แสดงบิตภายใน PCON

SMOD บิตกำหนดอัตราความเร็วการรับส่งข้อมูลอนุกรม UART

0 = อัตราความเร็ว 1 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1 = อัตราความเร็ว 2 เท่า
- GF1 แฟล็กใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ
- GF2 แฟล็กใช้งานทั่วไป ไม่เกี่ยวข้องกับการควบคุมใด ๆ
- PD Power down Bit
- 1 = หยุดออกสวิตลเตอร์ของซีพียูสัญญาณรีเซต หรือ อินเทอร์รัพต์ (83C154, 83C154D เท่านั้น) ที่จะยกเลิกโหมดนี้
- IDL Idle Mode Bit
- 1 = หยุดการทำงานของซีพียูสัญญาณรีเซต หรือ อินเทอร์รัพต์ เท่านั้นที่จะยกเลิกโหมดนี้

3) รีจิสเตอร์ควบคุมการอินเทอร์รัพต์ เนื่องจากว่า MCS-51 สามารถใช้งานพอร์ทอนุกรมในลักษณะของการอินเทอร์รัพต์ จึงมีรีจิสเตอร์ที่เกี่ยวข้องดังนี้

- IE ตำแหน่ง A8H ทำหน้าที่ยอมให้เกิดการอินเทอร์รัพต์จากพอร์ทอนุกรมได้หรือไม่
- IP ตำแหน่ง B8H ทำหน้าที่จัดลำดับความสำคัญของการอินเทอร์รัพต์

4) รีจิสเตอร์ควบคุมพอร์ทอนุกรม การใช้งานพอร์ทอนุกรมจะขึ้นอยู่กับรีจิสเตอร์โดยตรงคือ

- SBUF ตำแหน่ง 99H ทำหน้าที่เป็นบัฟเฟอร์การรับหรือส่งข้อมูลของพอร์ทอนุกรม
- SCON ตำแหน่ง 98H ทำหน้าที่ควบคุมและกำหนดโหมดการใช้งานพอร์ทอนุกรม

ทั้งหมดซึ่งมีรายละเอียดของบิตต่าง ๆ ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ตารางที่ 2.7 แสดงบิตภายใน SCON

SM0, SM1 บิตเลือกโหมดการทำงาน

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับส่งข้อมูลเท่ากับ (1/12) ของ OSC
0	1	1	8 บิต UART ความเร็วในการรับส่งข้อมูลกำหนดจาก Timer 1,2
1	0	2	9 บิต UART ความเร็วในการรับส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU OSC โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 บิต UART ความเร็วในการรับส่งข้อมูลกำหนดจาก Timer 1,2

ตารางที่ 2.8 แสดงการเลือกโหมด

SM2 บิตเลือกการทำงานแบบ Single Processor Environment หรือ Multiprocessor Environment

1: เลือก Multiprocessor Environment ใช้ได้กับโหมด 2, 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0: เลือก Single Processor Environment ใช้ได้กับทุกโหมด

REN บิตควบคุมให้รับหรือไม่รับข้อมูล

1: ให้รับข้อมูลได้

0: ห้ามรับข้อมูล

TB8 ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2, 3 ให้ใส่ในบิตนี้ได้เลย

RB8 ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมากเก็บในบิตนี้

TI แฟล็ก TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

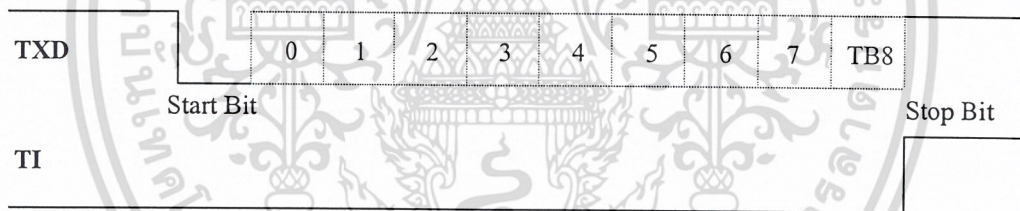
RI แฟล็ก RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

การส่งข้อมูลแบบอนุกรมในโหมด 3 (การส่งข้อมูลของกี้อย์บอร์ดและเมาส์)

พอร์ทสื่อสารอนุกรม 11 บิต UART โดย ข้อมูล (DATA) 8 บิต, 1 สตาร์ทบิต (Start Bit) และ 1 สต็อปปบิต (Stop Bit) เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1 สำหรับ 8051

$$BaudRateMode3 = \frac{2^{SMOD} \times CPUOSC}{32 \times 12 \times [256 - (TH1)]}$$

สมการแสดงการคำนวณหาอัตราบอดในโหมด 3

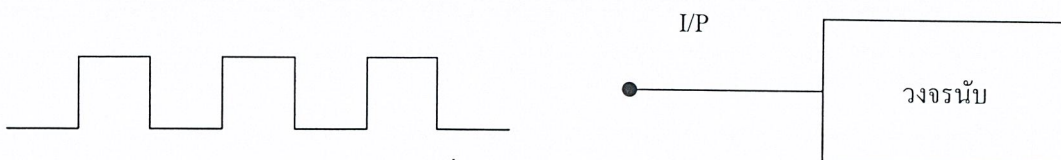


รูปที่ 2.3 รูปแบบการรับส่งข้อมูลในโหมด 3

2.5 การโปรแกรมไทม์เมอร์/เคาน์เตอร์

การใช้งานไทม์เมอร์/เคาน์เตอร์ภายใน MCS - 51 จะต้องมีการกำหนดรูปแบบการใช้งานต่าง ๆ เสียก่อนจึงจะทำงานได้อย่างถูกต้อง การกำหนดค่าเริ่มต้นของรีจิสเตอร์ในการใช้งานไทม์เมอร์/เคาน์เตอร์ จำเป็นต้องเข้าใจหลักการของ ไทม์เมอร์/เคาน์เตอร์ ดังนี้

เคาน์เตอร์ (Counter) ความหมายของเคาน์เตอร์ก็คือตัวนับสัญญาณ เช่นการนับของจำนวนพัลส์ของอินพุตที่มาจากภายนอก



รูปที่ 2.4 แสดงเคาน์เตอร์

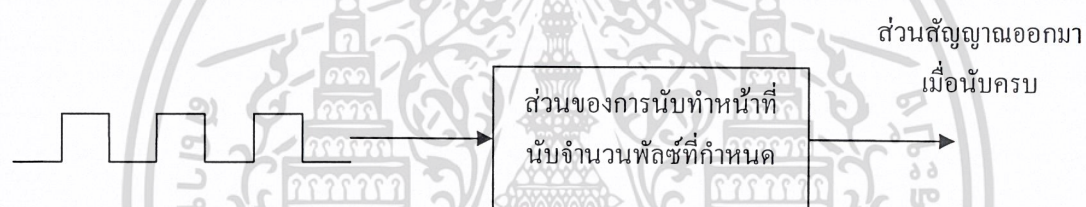
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไทม์เมอร์ (Timer) ความหมายของไทม์เมอร์ ก็คือการเป็นตัวตั้งเวลาด้วยการ โปรแกรมเมื่อถึงเวลาที่กำหนดจะแสดงผลออกมาให้รู้เหมือนกับนาฬิกาปลุกซึ่งแสดงออกให้รู้โดยส่งเสียงออกมา ใน MCS – 51 การใช้งานเป็นไทม์เมอร์จะแสดงผลออกมาให้รู้โดยแฟลค (TF) ถ้าเราเปรียบเทียบระหว่างไทม์เมอร์และแกนเตอร์จะเห็นข้อแตกต่างดังรูป

แกนเตอร์

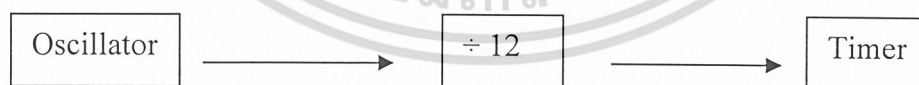


ไทม์เมอร์



รูปที่ 2.5 แสดงการเปรียบเทียบข้อแตกต่างระหว่างไทม์เมอร์กับแกนเตอร์

เมื่อเปรียบเทียบกับ MCS – 51 ในกรณีที่เป็นแกนเตอร์ สัญญาณพัลส์ที่มาจากภายนอกจะเข้ามาทางขา T0 และ T1 แต่ถ้าเป็นไทม์เมอร์สัญญาณที่ส่งออกมาเพื่อแสดงผลเมื่อไทม์เมอร์นับครบค่าของการนับที่ตั้งไว้ จะถือจากการเปลี่ยนแปลงของแฟลค TF0 และ TF1 ภายใน TCON สัญญาณพัลส์ที่มาจากภายใน MCS-51 เพื่อให้เกิดการนับของไทม์เมอร์นั้นจะได้จากออสซิลเลเตอร์ของ CPU หาร 12 ดังรูป



รูปที่ 2.6 แสดงฐานเวลาของไทม์เมอร์

ฐานเวลาของไทม์เมอร์คือ 1 หน่วยของการนับจะขึ้นอยู่กับ 1 คาบเวลาการทำงาน เพราะฉะนั้นการจะเริ่มต้นใช้งานไทม์เมอร์จึงต้องคำนึงถึงการป้อนค่าในรีจิสเตอร์ต่าง ๆ เพื่อกำหนดการนับ มีดังนี้

รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่งที่ 89H ไม่สามารถอ้างตำแหน่งแบบบิตได้ รายละเอียดของการกำหนดค่าบิตต่าง ๆ จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0



ตารางที่ 2.9 บิตภายใน TMOD

บิต	สัญลักษณ์	หน้าที่
7/3	Gate	การเอ็นนาเบิล(Enable) OR Gate เพื่อควบคุมการทำงานของไทม์เมอร์ Gate=1 ไทม์เมอร์จะทำงานถ้าสัญญาณขา INTn เป็นลอจิก 1 และ บิต TRn ในรีจิสเตอร์ TCON เป็น 1 ด้วย
6/2	C/T	ถ้าบิต C/T=0 เป็นการแสดงการทำงานแบบนับพัลส์จากภายในตัว MCS - 51 (Timer) เมื่อกำหนด C/T=1 เป็นการเลือกการทำงานแบบนับพัลส์จากภายนอกที่ป้อนเข้าทางขา P3.5(T1) หรือ P3.4(T0) (ตัวนับ)
5/1	M1	ใช้งานในการเลือกโหมดการทำงานดังตารางที่ 1
4/0	M0	ใช้งานในการเลือกโหมดการทำงานดังตารางที่ 1

ตารางที่ 2.10 หน้าที่บิตภายใน TMOD

M0	M1	โหมดการใช้งาน	ลักษณะการทำงาน
0	0	0	ไทม์เมอร์ขนาด 13 บิต
0	1	1	ไทม์เมอร์ขนาด 16 บิต
1	0	2	ไทม์เมอร์ขนาด 8 บิต แบบ โหลดค่าอัตโนมัติ
1	1	3	ไทม์เมอร์แบบใช้งานอิสระ

ตารางที่ 2.11 แสดงโหมดภายใน TMOD

รีจิสเตอร์ TH0, L0 และ TH1, TL1 รีจิสเตอร์เหล่านี้เป็นรีจิสเตอร์ขนาด 8 บิตใช้กำหนดค่าการนับของไทม์เมอร์ ตำแหน่งของ TH0, TL0 จะอยู่ 8CH, 8AH และตำแหน่งของ TH1, TL1 จะอยู่ที่ 8DH, 8BH และเนื่องจากว่าไทม์เมอร์ภายในนี้เป็นไทม์เมอร์แบบนับขึ้นจนถึงค่าสูงสุด คือ FFH (ในกรณี que เลือกการทำงานแบบ 8 บิต), FFFFH (ในกรณี que เลือกการทำงานแบบ 16 บิต) แล้วจะเกิดโอเวอร์โฟลว์ (เปลี่ยนจาก FFH หรือ FFFFH เป็นค่า 0) ฉะนั้นการกำหนดค่าการนับจะต้องนำค่าสูงสุดของโหมดการทำงานนั้น ๆ มาลบออกด้วยค่าที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการอินเตอร์เฟสของเมาส์-คีย์บอร์ด

2.6 ลักษณะทางกายภาพของพอร์ต PS/2

พอร์ต PS/2 มีลักษณะทางกายภาพอยู่ 2 รูปแบบคือ 5 - pin DIN และ 6 - pin mini DIN ทั้งคู่มีรูปแบบที่เหมือนกันทางด้านไฟฟ้า (Electrically) แต่ในทางปฏิบัติเราสามารถที่จะหาตัวแปลง (adaptor) มาทำการแปลงจาก 5 - pin DIN มาเป็น 6 - pin mini DIN และแปลงจาก 6 - pin mini DIN มาเป็น 5 - pin DIN โดยทั่วไปแล้วในคอมพิวเตอร์ในปัจจุบัน พอร์ต PS/2 จะเป็นแบบ 6 - pin mini DIN ซึ่งมีลักษณะทางกายภาพดังนี้



6 - Pin mini DIN (PS/2):

- 1) Data
- 2) Not Implemented
- 3) Ground
- 4) Vcc (+5V)
- 5) Clock
- 6) Not Implemented

จากรูปขาต่าง ๆ ของ อินเตอร์เฟส PS/2 จะมีขา Vcc / Ground สำหรับจ่ายไฟเลี้ยงแก่เมาส์และคีย์บอร์ดโดยที่กระแสที่อุปกรณ์ (Device) ไม่ควรเกิน 100 mA และควรหลีกเลี่ยงการถอดอุปกรณ์ ออกขณะที่โฮสต์ กำลังทำงานอยู่ ในเมนบอร์ด (main board) รุ่นเก่าๆ นั้นจะมีพินส์ต่ออยู่ที่ พอร์ตของตัวอุปกรณ์ (Device Port) ถ้าพินส์นี้ขาดเมนบอร์ดนั้นก็จะเป็นการใช้การไม่ได้ แต่เมนบอร์ดรุ่นใหม่ ๆ ได้ทำการแก้ปัญหานี้แล้วแต่ก็ควรหลีกเลี่ยงเพื่อความปลอดภัย

2.7 ทฤษฎีของ คีย์บอร์ด

สแกนโค้ด

ตัวประมวลผล (Processor) ของ คีย์บอร์ด นั้นเสียเวลาส่วนใหญ่ไปกับการสแกน (Scanning)

หรือการ “Monitoring” ถ้ามันสแกนหาทุกคีย์ที่ถูกกด , ถูกปล่อย หรือกดค้างไว้ คีย์บอร์ด จะส่งข้อมูลในเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแพ็คเกจ (Data Packet) ที่เรียกว่า สแกนโค้ด (Scan Code) ไปยังคอมพิวเตอร์ซึ่ง สแกนโค้ด นั้นมี 2 ชนิดคือเมคโค้ด (Make Code) และ เบรคโค้ด (Break Code) โดย เมคโค้ด จะส่งไปยังคอมพิวเตอร์เมื่อปุ่ม (Key) ถูกปล่อย (Released) ซึ่งทุก ๆ ปุ่ม (Key) จะมี เมคโค้ด และ เบรคโค้ดเฉพาะของแต่ละปุ่ม ซึ่งคอมพิวเตอร์จะรับไปคำนวณประมวลผลได้ว่าเรากดปุ่มใด โดยมีตารางของ เมคโค้ด และ เบรคโค้ด ของทุกปุ่มบนคีย์บอร์ด ประกอบกันเป็นตารางเรียกว่าสแกนโค้ด เซต (Scan Code Set) ซึ่งมีอยู่ 3 มาตรฐาน คือ Set1, Set2, Set3 ซึ่งคีย์บอร์ดในปัจจุบันจะถูกตั้งไว้เป็น Set2

สแกนโค้ด Set 1 : เป็นมาตรฐานคีย์บอร์ดแบบ XT

สแกนโค้ด Set 2 : คีย์บอร์ดในปัจจุบันใช้มาตรฐานนี้

สแกนโค้ด Set 3 : เป็นตารางเพิ่มเติมของคีย์บอร์ดแบบ PS/2 (ไม่ค่อยถูกใช้งาน)

Make Codes, Break Codes and Typematic Repeat:

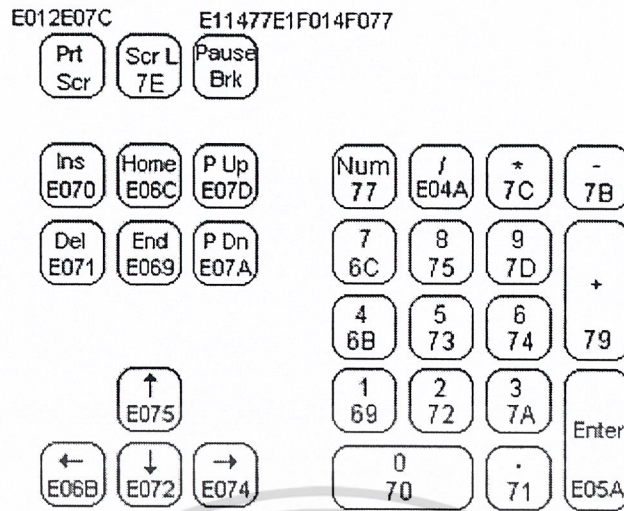
เมื่อใดก็ตามที่ปุ่มบนคีย์บอร์ดถูกกด เมคโค้ด จะถูกส่งไปยังคอมพิวเตอร์ซึ่ง เมคโค้ด ของแต่ละปุ่มจะต่างกัน และมีลักษณะเฉพาะสำหรับแต่ละปุ่ม และ สแกนโค้ด จะไม่สัมพันธ์กับรหัส ASCII เลขมันเป็นหน้าที่ของคอมพิวเตอร์ที่ทำการแปล สแกนโค้ด ที่ได้รับให้เป็น ตัวอักษร (Character) หรือ คำสั่ง (Command)

ESC 76	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 78	F12 07		
~ 0E	!@ 16	2@ 1E	3# 26	4\$ 25	5% 2E	6^ 36	7& 3D	8* 3E	9(46	0) 45	-= 4E	+= 55	\ 5D	← 66
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	I 43	O 44	P 4D	[54] 5B		
Caps 58	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	; 4C	"' 52	↵ 5A		
Shift 12	Z 1A	X 22	C 21	V 2A	B 32	N 31	M 3A	<, 41	>. 49	?/ 4A		Shift 59		
Ctrl 14	Alt 11	SPACE 29										Alt E0 11	Ctrl E0 14	

รูปที่ 2.8 โค้ดตัวอักษร

ดังนั้น เมคโค้ด ส่วนมากจะมีขนาด 1 ไบต์ ใน สแกนโค้ด Set 2 มีอยู่บางกลุ่มเท่านั้นที่ เมคโค้ด จะมีขนาด 2 หรือ 4 ไบต์พวกนี้ถูกเรียกว่า เอกซ์เทนคีย์ (Extended Keys) ซึ่ง เมคโค้ด ของเอกซ์เทนคีย์ (Extended Code) นั้นจะขึ้นต้นด้วย E0H เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 เอ็กซ์เทนส์โค้ด

เมื่อ เมคโค้ด ถูกส่งไปยังคอมพิวเตอร์ เมื่อใดก็ตามที่กดปุ่ม เบรคโค้ด ถูกส่งไป เมื่อใดก็ตามที่ปุ่มถูกปล่อย ซึ่งทุกปุ่มจะมี เมคโค้ด และ เบรคโค้ด เฉพาะตัวของใครของมัน โดยที่ เมคโค้ด กับ เบรคโค้ด สัมพันธ์กันโดยส่วนใหญ่ใน สแกนโค้ด Set 2 นั้น เมคโค้ด จะประกอบด้วยความยาว 1 ไบต์ แล้วเบรคโค้ด ประกอบด้วย 2 ไบต์ โดย ไบต์ แรกของ เบรคโค้ด จะเป็น F0 แล้ว ไบต์ ที่ 2 จะเป็น เมคโค้ดของปุ่มนั้น

ตัวอย่าง

Key	Make code	Break Code
“A”	1C	F0,1C
“5”	2E	F0,2E
“F10”	09	F0,09
“Right Arrow”	E0,74	E0,F0,74
“Ctrl” ขวา	E0,14	E0,F0,14

ตารางที่ 2.12 ตัวอย่างโค้ด

เช่น

ถ้าเราต้องการอักษร “G” ให้ไปปรากฏใน Microsoft Word จะส่งข้อมูลไปยังคอมพิวเตอร์อย่างไร

ในกรณีนี้ เราต้องทำการกดปุ่ม “Shift” ค้างไว้ แล้วกดปุ่ม “G” จากนั้นปล่อยปุ่ม “G” แล้วทำการปล่อยปุ่ม “G” แล้วทำการปล่อยปุ่ม “Shift” สแกนโค้ด เกี่ยวข้องกับเหตุการณ์ข้างต้นนี้ตามนี้

เมคโค้ด ของปุ่ม “Shift” (12H) ถูกส่งไปยังคอมพิวเตอร์

เมคโค้ด ของปุ่ม “G” (34H) ถูกส่งไปยังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมคโค้ด ของปุ่ม “G” (FOH, 34H) ถูกส่งไปยังคอมพิวเตอร์

เมคโค้ด ของปุ่ม “Shift” (FOH, 12H) ถูกส่งไปยังคอมพิวเตอร์

ดังนั้นสามารถสรุปข้อมูลที่ถูกส่งไปยังคอมพิวเตอร์ในกรณีนี้คือ : 12H, 34H, FOH, 34H, FOH, 12 ถ้าเรากดปุ่มแล้ว เมคโค้ด ของปุ่มนั้นก็จะส่งไปยังคอมพิวเตอร์ เมื่อกดปุ่มค้างไว้ปุ่มนั้นก็จะกลายเป็น “Typematic” หมายความว่าคีย์บอร์ด จะเก็บ เมคโค้ด นั้นไว้จนกว่าปุ่มจะถูกปล่อย หรือปุ่มอื่นจะถูกกด เราตรวจสอบได้โดยเปิดโปรแกรมประเภท Text Editor และกดปุ่ม“A” ค้างไว้ เมื่อเรากดครั้งแรกตัวอักษร“A” จะปรากฏบนหน้าจอ หลังจากช่วงเวลาคิเลย์ (Delay) สั้น ๆ ตัวอักษร“A” ตัวอื่นก็ปรากฏตามมา จนกระทั่งเราจะปล่อยปุ่ม “A” ในเหตุการณ์นี้มี 2 ตัวแปร (Parameter) ที่สำคัญ

1. Typematic Delay ซึ่งเป็นระยะหน่วงเวลาสั้นๆ (Short Delay) ระหว่างอักษร“A” ตัวที่ 1 และอักษร“A” ตัวที่ 2

2. Typematic Rateซึ่งเป็นค่าอัตราตัวอักษรต่อนาที (Character per second) ว่ามีจำนวนเท่าใดนี้ จะปรากฏบนจอหลังจาก Typematic Delay

Typematic Delay อยู่ในช่วง 0.25 นาที – 1.00 นาที และ Typematic Rateอยู่ในช่วง 2.0cps – 30.0cps (cps: Character per second) เราอาจสามารถเปลี่ยน Typematic Rateและ Typematic Delay โดยใช้คำสั่ง “Set Typematic Rate/Delay”

Typematic data จะไม่ถูกเก็บไว้ใน คีย์บอร์ด ในกรณีที่มากกว่า 1 ปุ่ม ถูกกดค้างไว้ซึ่งจะเป็นปุ่มสุดท้าย ที่ถูกกดเท่านั้นที่จะกลายเป็น typematic , typematic เกิดขึ้นซ้ำๆ แล้วหยุดเมื่อปุ่มนั้นถูกปล่อย

หมายเหตุ ปุ่ม “Pause/Break” จะไม่มี เมคโค้ด ใน สแกนโค้ด Set 1, 2 เมื่อปุ่มนี้ถูกกดจะส่ง เมคโค้ด ของมันไปยังคอมพิวเตอร์ เมื่อปล่อยปุ่มนี้มันจะไม่ส่งอะไร ไปยังคอมพิวเตอร์

รีเซต(Reset)

เมื่อเปิดเครื่องคอมพิวเตอร์หรือ Software Reset คีย์บอร์ดจะโหลด BAT (Basic Assurance Text) และโหลด ส่งมาดังนี้

Typematic Delay = 500 ms

Typematic Rate= 10.9 cps

สแกนโค้ดเซต 2 (Set 2)

เมื่อเข้าสู่กระบวนการ BAT คีย์บอร์ดจะเปิดการทำงานของ LED ทั้ง 3 (Num Lock, Caps Lock, Scroll lock) และปิดมันเพื่อกระบวนการ BAT เสร็จสิ้น ณ เวลานี้จะส่ง Code 0xAA แทน การทำ กระบวนการ BAT สำเร็จ และจะส่ง code 0xFC เมื่อเกิด error ไปยังคอมพิวเตอร์ โดยที่ code 0xAA (BAT successful) ต้องส่งไปในเวลา 500-700 ms หลังจาก power-on

คีย์บอร์ดหลายแบบที่ละทิ้งการตรวจสอบคล็อกและ DATA lines ของมัน จนกระทั่งหลังจาก กระบวนการ BAT เสร็จสิ้น (code 0xAA ถูกส่งไปยังคอมพิวเตอร์) ดังนั้นเงื่อนไขการยับยั้ง (Inhibit) (Clock line เป็น low) อาจจะไม่ขัดขวางจากการที่กำลังส่ง code 0xAA (BAT successful) ไป

คำสั่งควบคุมที่ถูกส่งจากพีซีไปยังคีย์บอร์ด (Command Set)

เป็นชุดคำสั่งที่ คอมพิวเตอร์ส่งไปยัง คีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

- คีย์บอร์ดจะเคลียร์บัพเฟอร์ของมันเมื่อมันรับ คำสั่ง (คำสั่ง)
- ถ้าคีย์บอร์ดรับคำสั่งมาผิด มันจะส่ง "0" x FE กลับไปยัง คอมพิวเตอร์
- คีย์บอร์ดต้องไม่ส่ง สแกนโค้ด ไปยังคอมพิวเตอร์ ขณะที่ยังมีกระบวนการของคำสั่ง

(Processing a command)

- ถ้าคีย์บอร์ดกำลังรอคอย Argument ไบต์ และมันได้รับ คำสั่ง ใหม่มาแทน มันจะทิ้งคำสั่ง ก่อนนั้นแล้วทำการประมวลผลคำสั่งใหม่แทน

ข้างล่างนี้เป็นคำสั่ง (คำสั่ง) ทั้งหมดที่ โฮสต์ อาจจะส่งไปยังคีย์บอร์ด

0xFF (Reset)

คีย์บอร์ดจะส่ง Response กลับมาด้วยสัญญาณ "ACK" โดยแทนด้วย 0xFA แล้วจะเข้าสู่ Reset Mode

0xFE (Resend)

คีย์บอร์ดจะส่ง response โดยไบต์ข้อมูลล่าสุดไปยังคอมพิวเตอร์อีกครั้ง (resending) ยกเว้น ไบต์ ก่อนหน้านั้นเป็นสัญญาณ "resend" ซึ่งแทนด้วย 0xFE คีย์บอร์ดจะส่งอีกครั้ง อีก 6 คำสั่ง สามารถเกิดขึ้นเมื่อคีย์บอร์ดอยู่ในทุก mode การทำงาน แต่มันมีผลเฉพาะ คีย์บอร์ด ใน โหมดที่ 3 (เช่นเมื่อ Set เป็น สแกนโค้ด Set 3)

0xFD (Set key Type Make)

จะเปิดการทำงานของ เบรกโค้ด และการพิมพ์ (typematic) ซ้ำ ๆ ของคีย์ที่ถูกระบุ คีย์บอร์ดจะ ส่ง respond ด้วยสัญญาณ "ACK" ซึ่งแทนด้วย 0xFA แล้วทำการปิดการทำงานการ Scan แล้ว ทำการอ่านลำดับข้อมูล ของปุ่ม ที่มาจากคอมพิวเตอร์คีย์นี้จะถูกระบุโดย Set 3 Make Code และ คีย์บอร์ดจะทำการตอบสนอง เมคโค้ด นั้นด้วยสัญญาณ "Acknowledge" โฮสต์ ลงท้าย list ด้วยการส่ง invalid Set 3 Make Code (เช่น valid คำสั่ง) แล้ว คีย์บอร์ด ทำการ enable การ Scan อีกครั้ง

0x FC (Set Keys Type Make/Break)

คล้ายกับคำสั่งข้างบน (0xFD) ยกเว้นจะทำการ disable typematic repeat อันเดียวเท่านั้น

0xFB (Set All Keys Type Typemaic)

คล้ายกับคำสั่งข้างบน (0xFD) ยกเว้นจะทำการ disable Break Code อันเดียวเท่านั้น

0xFA (Set All Keys Tpematic/Make/Break)

คีย์บอร์ดจะทำการตอบรับคำสั่งด้วยการส่งสัญญาณ "Acknowledge" (0xFA) แล้วทำการ Set ทุกคีย์ในการตั้งค่าตามปกติ (ทำการสร้าง สแกนโค้ด นั่นคือ make, break, typematic repeat)

0xF9 (Set All Keys Make)

คีย์บอร์ดตอบรับคำสั่งนี้ด้วยการส่ง "Acknowledge" (0xFA) เช่นเดียวกับ 0xFD ยกเว้น สามารถใช้กับทุกคีย์

0xF8 (Set All Keys Make/Break)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแต่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์บอร์ดตอบรับคำสั่งนี้ด้วยการส่ง “Acknowledge” (0xFA) เช่นเดียวกับ 0xFC ยกเว้นสามารถใช้กับทุกคีย์

0xF7 (Set all Keys Typematic)

คีย์บอร์ดตอบรับคำสั่งนี้ด้วยการส่ง “Acknowledge” (0xFA) เช่นเดียวกับ 0xFB ยกเว้นสามารถใช้กับทุกคีย์

0xF6 (Set Default)

ทำการ Load ค่า default ของค่า typematic rate/delay (10.9 cps/500ms), key type (ทุกค่าของ keys typematic/make break) และ สแกนโค้ด Set 2

0xF5 (Disable)

คีย์บอร์ดหยุดการ Scan แล้ว load ค่า default (ดู Set Default) แล้วคอยคำสั่งต่อไป

0xF4 (Enable)

ตั้ง Enable คีย์บอร์ดอีกครั้ง หลังจาก disable คีย์บอร์ด เพื่อใช้ในคำสั่งก่อนหน้า

0xF3 (Set typematic Rate/Delay)

โฮสต์ ทำตามคำสั่งหนึ่งตามตารางข้างล่าง เมื่อได้รับ Argument ไบต์ แล้วทำการกำหนด Typematic Rate และ delay ตามข้างล่าง

0xF2 (Read ID)

คีย์บอร์ดทำการตอบรับคำสั่งนี้ด้วยการส่งข้อมูล 2 ไบต์ของ Device ID คือ 0xAB, 0x83

0xF0 (Set Scan Code Set)

คีย์บอร์ดทำการตอบรับคำสั่งนี้ด้วยการส่ง “Acknowledge” (0xFA) ไปยัง โฮสต์ แล้วอ่านค่า Argument ไบต์ จาก โฮสต์ , Argument ไบต์ นี้ อาจเป็น 0x01, 0x02, 0x03 เพื่อเป็นการเลือก สแกนโค้ด Set 1, 2 หรือ 3 ตามลำดับ จากนั้นคีย์บอร์ดก็ทำการตอบรับด้วยการส่ง “Acknowledge” (0xFA) ถ้า Argument ไบต์ เป็น 0x00 คีย์บอร์ดจะตอบรับ Argument ไบต์ นี้ ด้วยการส่ง “Acknowledge” (0xFA) แล้วตามด้วย สแกนโค้ด Set ล่าสุด

0xEE (Echo)

คีย์บอร์ดทำการตอบรับคำสั่งนี้ด้วยการส่ง “Echo” (0xEE) ไปยัง โฮสต์

0xED (Set/Reset LED)

โฮสต์ ทำตามคำสั่งนี้ด้วย Argument ไบต์ อันหนึ่งซึ่งค่า Argument ไบต์ นี้ระบุสถานะของ Num Lock, Caps Lock, Scroll Lock LED Argument ไบต์ ที่ว่านี้ ถูกกำหนดตามด้านล่างนี้

MSB							LSB
0	0	0	0	0	Caps Lock	Num Lock	Scroll Lock

ตารางที่ 2.13 ไบต์แสดงสถานะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“Scroll Lock”	- Scroll Lock LED	off (0)/on (1)
“Num Lock”	- Num Lock LED	off (0)/on (1)
“Caps Lock”	- Caps Lock LED	off (0)/on (1)

คำสั่งควบคุมที่ถูกส่งจากคีย์บอร์ดไปยังพีซี

0xFF	-	เอาท์พุทบัฟเฟอร์โอเวอร์โฟลล์, คีย์บอร์ดตรวจพบข้อผิดพลาด
0xFE	-	Resend ร้องขอให้โฮสต์ส่งข้อมูลไบต์สุดท้ายอีกครั้ง
0xFA	-	ตอบรับ (Acknowledge : ACK)
0xF0 k	-	การปล่อยปุ่ม k โค้ดที่ถูกลบออก
0xAA	-	การตรวจสอบไฟเลี้ยงผ่าน
0xEE	-	Echo พีซีต้องตอบกลับมาด้วยคำสั่ง 0x EE เหมือนกัน เป็นการตรวจสอบ
0x00	-	เหมือน 0xFF

The i8042 คีย์บอร์ด Controller

การเขียนโปรแกรมเราไม่ต้องลงไปติดต่อดังเดิมกับ คีย์บอร์ด เราก็จะติดต่อกับตัว Controller ของ คีย์บอร์ด แทน ตัว controller นี้จะดูแลทุกระดับสัญญาณ และดูแลรายละเอียดของ โปรโตคอล ที่พ่วงกับการหาวิธีการแปล การแปล และการจัดการกับ สแกนโค้ดคำสั่ง (command)

Intel 8042/compatible microcontroller ถูกใช้เป็นตัว controller ใน คีย์บอร์ด ในคอมพิวเตอร์รุ่นใหม่ๆ ตัว microcontroller ที่ซ่อนอยู่ใน Motherboard's Chip Set ซึ่งเป็นการรวมหลายๆ controllers ใน single package กระนั้นอุปกรณ์นี้ก็ยังคงอยู่ และ controller ของ คีย์บอร์ด ยังคงหมายถึง “8042”

ขึ้นอยู่กับ motherboard คีย์บอร์ด controller อาจทำการใน 1 หรือ 2 mode “AT-compatible mode” หรือ “PS/2 compatible mode” PS/2 compatible mode ถูกใช้ถ้า motherboard นั้นรองรับ PS/2 เม้าส์ ถ้ากรณีนี้ 8042 จะเป็นตัว คีย์บอร์ด controller และ เม้าส์ controller , คีย์บอร์ด controller จะถูกตรวจพบโดยฮาร์ดแวร์ ซึ่ง mode มันจะใช้ตามสายที่ต่อกับ คีย์บอร์ด port

8042 จะประกอบด้วย Register ดังนี้

- A one-ไบต์ อินพุต buffer – ประกอบด้วย ไบต์ ที่อ่านจาก คีย์บอร์ด; read only
- A one-ไบต์ output buffer – ประกอบด้วย ไบต์ ที่อ่านจาก คีย์บอร์ด; write only
- A one-ไบต์ status registers – 8 status flags; read only
- A one-ไบต์ control registers – 7 status flags; read/write

3 register แรก (Input, output, status) เป็นการเข้าถึงข้อมูลโดยตรงโดยทาง port 0x60 และ 0x64 รีจิสเตอร์สุดท้าย (control) เป็นการอ่านโดยใช้คำสั่ง “Read command byte” และเขียนโดยใช้คำสั่ง “Write command byte” ตารางด้านล่าง แสดงพอร์ต (port) ที่ใช้ต่อกับ 8042 ทำการต่ออย่างไร

การเขียนไปยัง port 0x64 ไม่ต้องระบุ Register บัสดังคำสั่งไปให้ 8042 แปลได้เลย ถ้าคำสั่งอนุญาตตัวแปร ตัวแปรนี้จะส่งไปที่ port 0x60 เช่นเดียวกัน ผลลัพธ์อื่น จะกลับมาจากการใช้คำสั่งอ่านจาก Port 0x60 ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่ออธิบายถึง 8042 บางครั้งหมายถึง Physical I/O pins ของมัน pin ต่างๆของ 8042 จะถูกกำหนดตามข้างล่าง

สถานะของรีจิสเตอร์ (Status Register)

สถานะแฟล็ก ของ 8042 จะอ่านมาจาก port 0x64 มันจะประกอบด้วย error information, status information และ indicate data ไม่ถูกแสดงใน อินพุต และ output buffer, flags ถูกกำหนดตามข้างล่าง

	MSB							LSB
AT-compatible mode :	PERR	RxTo	TxTo	INH	A2	SYS	IBF	OBF
PS/2-compatible mode :	PERR	To	MOBF	INH	A2	SYS	IBF	OBF

ตารางที่ 2.14 สถานะแฟล็กของ Status Register

OBF (Output Buffer Full) ใช้แสดงเมื่อมันตกลงที่จะเขียน output buffer

0 : Output buffer ว่าง - ตกลงเขียนไปยัง Port 0x60

1 : Output buffer เต็ม - ไม่เขียนไปยัง Port 0x60

IBF (Input Buffer Full) ใช้แสดงเมื่อ อินพุต สามารถใช้ได้ ใน อินพุต buffer

0 : อินพุต buffer ว่าง - ไม่มี อินพุต ที่ต้องอ่านจาก Port 0x60

1 : อินพุต buffer เต็ม - อินพุตใหม่ สามารถอ่านจาก Port 0x60

SYS (System flag) ถูกอ่านเมื่อ power on Reset, หรือ software Reset

0 : Power Up value - ระบบอยู่ในกระบวนการ Power on Reset

1 : ได้รับ BAT code - ระบบพร้อมถูก Initialize

A2 (Address line A2) ใช้ภายใน

0 : A2 = 0 - Port 0x60 ถูกเขียนไปล่าสุด

1 : A2 = 1 - Port 0x64 ถูกเขียนไปล่าสุด

INH (Inhibit flag) ใช้แสดงว่า คีย์บอร์ด ไม่ถูกยับยั้งการสื่อสาร

0 : คีย์บอร์ดคล็อก= 0 - คีย์บอร์ดถูกยับยั้ง

0 : คีย์บอร์ดคล็อก= 1 - คีย์บอร์ดไม่ถูกยับยั้ง

2.8 ทฤษฎีของ PS/2 เม้าส์

การอินเตอร์เฟสทางไฟฟ้า/โปรโตคอล

PS/2 เม้าส์ ใช้ โปรโตคอล เดียวกับ PS/2(AT) คีย์บอร์ด เป็นมาตรฐานของ IBM technical reference manual PS/2 เม้าส์ สามารถรองรับ อินพุต ตามนี้

การเคลื่อนที่ที่แนวนอน (X Movement)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนที่แนวตั้ง (Y Movement)

ปุ่มซ้าย (Left Bottom)

ปุ่มกลาง (Middle Bottom)

ปุ่มขวา (Right Bottom)

เมาส์ จะอ่านค่านี้ตามจังหวะความถี่ และอ็อปเทค เคาท์เตอร์ (Counter) และ แฟลค หลายๆ ค่าเพื่อส่งกลับค่าของการเคลื่อนที่ และสถานะของปุ่ม ในการทดลองนี้จะครอบคลุมไมโครซอฟท์ IntelliMouse ซึ่งประกอบด้วยมาตรฐานการรองรับ อินพุตทั่วไป ทั้งแบบที่มีล้อเลื่อน (Scrolling wheel) และ ปุ่ม 2 ปุ่ม

มาตรฐาน เมาส์ มี 2 ตัวนับเก็บการเคลื่อนที่ คือ ตัวนับแนวนอน และ ตัวนับแนวตั้ง มีค่า 9 บิต 2's complement และแต่ละอันจะเกี่ยวข้องกับ โอเวอร์ โฟลว์ แฟลคตามด้วยสถานะของปุ่มกดทั้ง 3 ปุ่ม จะส่งไปยัง โฮสต์ (Computer) ในรูปแบบของชุดข้อมูล (Data packet) ขนาด 3 ไบต์, ตัวนับ ทั้ง 2 จะแทนจำนวนของการ การเคลื่อนที่ นั้นจะปรากฏ ตั้งแต่การ การเคลื่อนที่ ครั้งที่ล่าสุด จะถูกส่งไปยัง โฮสต์

เมื่อ เมาส์ อ่านค่า อินพุต ของมัน มันจะบันทึกสถานะล่าสุดของปุ่มของมัน แล้วตรวจสอบการ การเคลื่อนที่ ถ้ามีการเคลื่อนไหวเกิดขึ้น มันจะเพิ่ม (+x หรือ +y) หรือลด (-x หรือ -y) ค่าการเคลื่อนที่ของมัน ถ้าเกิด โอเวอร์ โฟลว์ มันจะ เซต (Set) ที่แฟลคโอเวอร์ โฟลว์

ตัวแปรที่คำนวณจำนวนของการ (+x หรือ +y) และ (-x หรือ -y) นั่นคือตัวแปรที่มีชื่อว่า "resolution" โดยค่าเริ่มต้น ของ resolution คือ 4 ตัว/ ม.ม. และ โฮสต์ อาจจะทำการเปลี่ยนค่านี้โดยใช้ คำสั่ง "Set Resolution" (0xE8)

มีตัวแปรที่ไม่มีผลกับ การเคลื่อนที่ ตัวนับ ทั้ง 2 แต่จะมีผลกับการรายงานค่าของเคาท์เตอร์ ตัวแปรนี้คือ "Scaling" ค่า default ของ Scaling ของเมาส์ จะใช้เป็น 1:1 scaling ซึ่งไม่มีผลกับการรายงาน เมาส์ การเคลื่อนที่ ใดๆก็ตาม โฮสต์ อาจจะเลือก 1:2 scaling โดยใช้คำสั่ง "Set Scaling" 2:1 (0xE7) ถ้า 2:1 scaling ทำงาน เมาส์ จะ apply ตาม algorithm ด้านล่างนี้ไปยัง ตัวนับ ก่อนจะส่ง ตัวนับ ของมัน ไปยัง โฮสต์

การเคลื่อนที่ ตัวนับ	Reported การเคลื่อนที่
0	0
1	1
2	1
3	3
4	6
5	9
N > 5	2 x N

ตารางที่ 2.15 การรายงานค่าของเคาท์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่ง Data ไปจาก อุปกรณ์ต่อร่วม ประกอบด้วย 11 บิต การไหลของข้อมูล ถูกส่งแบบอนุกรม บน คาต้าไลน์ (Data line) ตารางข้างล่างนี้ แสดงฟังก์ชันของแต่ละบิต, บิตพาริตี (Parity Bit) จะเป็น "1" หรือ "0" และ บิตข้อมูลมี 8 บิต บวกกับบิตพาริตี ทุกครั้งต้องมีเลขคี่ของ "1"

Bit	Function
11	Stop Bit(always 1)
10	Parity Bit(odd parity)
9	Data Bit7 (MSB)
8	Data Bit6
7	Data Bit5
6	Data Bit4
5	Data Bit3
4	Data Bit2
3	Data Bit1
2	Data Bit0
1	Start Bit(always 0)

ตารางที่ 2.16 รายละเอียดแต่ละบิตของเม้าส์

การเคลื่อนที่ Data Packet

PS/2 เม้าส์ จะส่ง การเคลื่อนที่แนวตั้งแนวนอน และสถานะของปุ่มตามชุดข้อมูล ด้านล่างนี้

การเคลื่อนที่ตัวนับจะเป็น 9 บิต 2's Complement integers ซึ่ง MSB(Most Significant Bit) จะปรากฏตาม Sign Bit ใน ไบต์ 1 ของ การเคลื่อนที่ data packet ตัว ตัวนับ นี้ จะถูก อัปเดต(update) เมื่อ เม้าส์ อ่านค่า อินพุต ของมัน และค้นหา การเคลื่อนที่ ที่เกิดขึ้น ค่าเหล่านี้จะเป็นจำนวนของการ การเคลื่อนที่ ที่ปรากฏตั้งแต่ การเคลื่อนที่ data packet ล่าสุด ถูกส่งไปยัง โฮสต์ (หลังจากแพคเกจหนึ่งถูกส่งไปยัง โฮสต์ การเคลื่อนที่ ตัวนับ ก็จะทำให้การ Reset) ช่วงพิสัยของค่านี้สามารถถูกแสดงโดย การเคลื่อนที่ ตัวนับ คือ 0-255 ถึง +255 ถ้าเกินกว่านี้ ก็ถือว่าเกิด overflow โดย overflow Bit จะถูก Set และ ตัว ตัวนับ จะไม่ถูกเพิ่มขึ้นหรือลดลง จนกระทั่งมันจะ Reset ใหม่อีกครั้ง

ตามที่กล่าวไปแล้วว่า การเคลื่อนที่ ตัวนับ จะ Reset เมื่อใดก็ตามที่ การเคลื่อนที่ data packet ถูกส่งไปยัง โฮสต์ เสร็จสิ้นแล้ว จะ Reset หลังจาก เม้าส์ ได้รับ คำสั่งจาก โฮสต์ (คำสั่ง "Reset" (0xFE))

Modes of Operation

ในการรายงานข้อมูล จะมี 4 โหมดมาตรฐาน ของการปฏิบัติการ (Operation)

Reset: เม้าส์ ทำการ Reset mode เมื่อเปิดเครื่อง (power up) หรือหลังจากกับคำสั่ง "Reset" (0xFF) จาก โฮสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stream: เป็น default mode (หลังจากทำ Reset Mode แล้ว)

Remote: เป็น mode ที่มีประโยชน์ในบางสถานการณ์ และอาจจะสั่งโดยใช้คำสั่ง “Set Remote Mode (0xF0) ไปยัง เมาส์

Wrap : โหมด นี้ไม่ได้ใช้ประโยชน์ทางใดทางหนึ่งเฉพาะ ยกเว้นใช้ในการทดสอบ การเชื่อมต่อระหว่าง เมาส์ กับ โฮสต์, Wrap mode อาจทำโดยการส่งคำสั่ง “Set Wrap Mode (0xFE) ไปยัง เมาส์ การออกจาก Wrap mode , โฮสต์ ต้องส่งคำสั่ง “Reset (0xFF) หรือ “Reset Wrap Mode (0xEC) ถ้าส่งคำสั่ง “Reset (0xFF) ไปแล้ว เมาส์ ได้รับคำสั่งแล้ว เมาส์ จะทำการเข้าสู่ Reset mode ถ้าใช้คำสั่ง “Reset Wrap Mode (0xEC) เมื่อ เมาส์ รับคำสั่งแล้ว เมาส์ จะเข้าสู่ Wrap Mode ก่อน

Reset Mode: เมาส์ ทำการ Reset mode เมื่อเปิดเครื่อง (Power up) หรือในการส่งสัญญาณ response กับคำสั่ง “Reset (0xFF) หลังจากเข้าสู่ mode นี้แล้ว เมาส์ ตรวจสอบด้วยตัวเองตาม BAT (Basic Assurance Test) และ Set ตามค่า default ด้านล่างนี้

Sample rate	- 100 samples/sec
Resolution	- 4 ตัว/มม.
Scaling	- 1:1
Data Reporting	- Disabled

แล้ว เมาส์ ก็จะส่ง BAT completion code ใดๆอย่างหนึ่งไป 0xAA หมายถึง BAT successful, 0xFC หมายถึง Error ถ้า โฮสต์ ได้รับ response อื่นนอกจาก 0xAA มันอาจจะวนรอบการ check ใหม่ และให้ เมาส์ Reset กระบวนการ BAT ใหม่

ตาม BAT completion code (0xAA หรือ 0x FC) เมาส์ ส่ง device ID ของมันคือ 0x00 นี้คือสิ่งที่แยกมันออกจาก คีย์บอร์ด หรือ เมาส์ ใช้งานใน Extended mode

หลังจาก เมาส์ ส่ง Device ID ของมันเสร็จ มันจะเข้าสู่ stream mode ด้านบนค่า default จะมี “Data Reporting Disabled” หมายถึง เมาส์ จะไม่ส่ง การเคลื่อนที่ data packet ใดๆไปยัง โฮสต์ จนกระทั่งจะได้รับคำสั่ง “Enable Data Reporting” (0xF4)

Stream Mode

ใน mode นี้ เมาส์ ส่ง การเคลื่อนที่ data เมื่อมันตรวจจับการเคลื่อนที่ หรือการเปลี่ยนแปลงสถานะของปุ่มกด 1 หรือมากกว่า 1 ปุ่ม ค่า maximum rate นี้ data reporting อาจจะมีปรากฏตามที่เรารู้จักคือ “Sample rate ตัวแปรนี้มีช่วงจาก 10 samples/sec ถึง 200 samples/sec ค่าเริ่มต้น ของมันคือ 100 samples/sec และ โฮสต์ อาจจะไปเปลี่ยนค่าโดยใช้คำสั่ง “Set Sample Rate” (0xF3) โหมด นี้เป็น โหมดเริ่มต้นของ เมาส์

Remote Mode

Mode นี้ เมาส์ จะอ่านค่า อินพุต ของมัน และ update ค่าของ ตัวนับ/flag ณ sampling rate ต่ำสุด แต่มันจะแจ้งเฉพาะการ การเคลื่อนที่ (และการเปลี่ยนสถานะของปุ่ม) เมื่อข้อมูลถูก

ร้องขอ โดย โฮสต์ โฮสต์ จะส่งคำสั่ง “Read Data” (0xEB) หลังจากรับคำสั่งนี้แล้ว เมาส์ จะส่ง การเคลื่อนที่ data packet และ Reset การเคลื่อนที่ ตัวนับของมัน

Wrap Mode

โหมด นี้เป็น การส่งกลับ(Echoing) โหมด คือ ทุกๆ ไบต์ ที่ เมาส์ ได้รับ จะส่งกลับ ไปยัง โฮสต์ เหตุการณ์นี้ถ้า ไบต์ แทนด้วย คำสั่งที่ผิด เมาส์ จะไม่ตอบรับคำสั่งนั้น (Response to that command) นั่นคือ มันจะส่งเฉพาะ ไบต์ นั้นกลับไปยัง โฮสต์ มีข้อยกเว้น 2 ประการ ใน กรณีนี้ คำสั่ง “Reset” (0xFF) และ “Reset Wrap Mode (0xEC) เมาส์ จะตอบรับกับ 2 คำสั่งนี้ แต่จะไม่ส่ง echo กลับไปยัง โฮสต์

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ไบต์ 1	Y overflow	X overflow	Y sign Bit	X sign Bit	Always 1	Middle Btn	Right Btn	Left Btn
ไบต์ 2	X การเคลื่อนที่							
ไบต์ 3	Y การเคลื่อนที่							
ไบต์ 4	Z การเคลื่อนที่							

ตารางที่ 2.17 ไบท์แสดงสถานะต่างๆ ของเมาส์

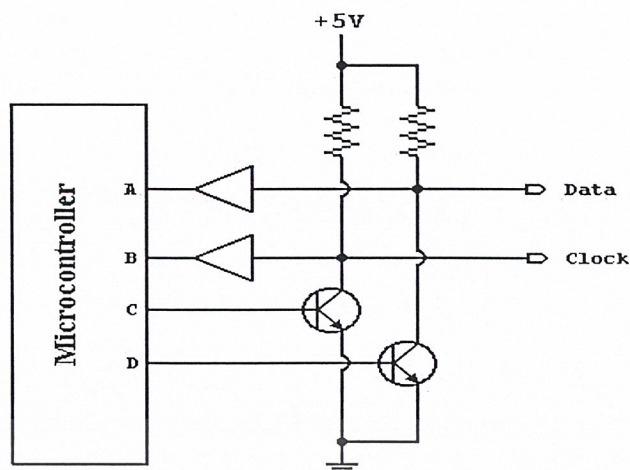
ส่วนของโปรโตคอลของเมาส์-กีย์บอร์ด

2.9 รายละเอียดของแหล่งจ่ายไฟ

ไฟ $V_{cc} = +5\text{ V}$ กระแสสูงสุด = 100 mA

คาต้าไลน์ (Data line) และคล็อกไลน์ (Clock line) เป็น Open – Collector ทั้งคู่ และ “ Open – Collector ” interface จะมีอยู่ 2 สถานะ คือ low หรือ high impedance ใน สถานะ “ low “ ทราานซิสเตอร์จะดึงให้คาต้าไลน์ ไปยังระดับGround ในสถานะ High Impedance นั้น จะ Open Circuit และจะไม่ทำให้คาต้าไลน์ มีสถานะ low หรือ high ยิ่งไปกว่านั้น ตัวต้านทาน pull up ที่อยู่ ระหว่างบัสและ V_{cc} ค่าของตัวต้านทานไม่สำคัญอาจมีค่าอยู่ประมาณ 1-10 k ohm โดย ถ้าขนาดของตัวต้านทานมีขนาดใหญ่ก็จะเปลืองไฟเลี้ยง แต่ถ้ามีค่าต่ำการทำงานของขอบขาขึ้นของสัญญาณก็จะเร็วรูป ด้านล่างแสดง open-collector interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดง Open-Collector Interface

2.10 การติดต่อทั่วไป

เมาส์และคีย์บอร์ดนั้นสนับสนุน Bidirectional synchronous serial protocol โดยเราจะสรุปสถานะของ Data Line และค็อกไลน์ไว้ดังนี้

สถานะของบิต (บิต State)

Data Line = high, Clock Line = high: **Idle State**

Data Line = high, Clock Line = low: **Communication Inhibited**

Data Line = low, Clock Line = high: **Host Request – to – Send**

ข้อมูลจะถูกส่งไป 1 ไบต์ จะเท่ากับ 11- 12 บิต ดังนี้:

1 start bit. : จะมีค่าเป็น “0” เสมอ

8 data bits : โดยจะส่งบิตที่มีลำดับความสำคัญน้อยที่สุด (LSB: Least Significant Bit) ไปก่อน

1 Odd parity bit: พาริตีบิตคี่

1 stop bit : จะเป็น “1” เสมอ

1 acknowledge bit: ใช้เฉพาะเวลาสื่อสารในทิศทาง โฮสต์ไปยังอุปกรณ์ เท่านั้น

parity bit ถูก set (= 1) เมื่อมี “1” ปรากฏเป็นจำนวนคู่ใน 8 data bits

parity bit ถูก clear (= 0) เมื่อมี “1” ปรากฏเป็นจำนวนคี่ใน 8 data bits

Parity bit นี้จะใช้ในกระบวนการ error detection

ข้อมูลที่ส่งจาก อุปกรณ์ไปยังโฮสต์ จะอ่านที่ขอบขาของค็อกไลน์ และ ข้อมูลส่งจากโฮสต์ไปยัง อุปกรณ์ (Host – to – device) จะถูกอ่านที่ขอบขาขึ้นของค็อกไลน์ โดยที่ความถี่ของค็อกไลน์ต้องอยู่ระหว่าง 10 – 16.7 kHz นั้นหมายความว่า ค็อกไลน์เป็น “high” เป็นเวลา 30 – 50 us (microsecond) และเป็น “low” เป็นเวลา 30 – 50 us สรุปสุดท้ายว่า ตัวอุปกรณ์ (device) จะเป็นตัว สร้าง (Generate) ค็อกไลน์แต่ โฮสต์ จะเป็นตัวควบคุมการสื่อสารทั้งหมด

การติดต่อจากอุปกรณ์ไปยังโฮสต์ (Device – to – Host)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายข้อมูล และ คล็อกไลน์ เป็น Open collector ทั้งคู่ ตัวต้านทานที่ต่ออยู่ระหว่างแต่ละสายกับ +5 V ดังนั้น Idle State ของบัสคือ high เมื่อเมาส์หรือคีย์บอร์ดต้องการส่งข้อมูล, อย่างแรกมันจะทำการตรวจสอบ คล็อกไลน์ ว่าอยู่ในสถานะ high หรือไม่ ถ้าไม่ใช่ก็แสดงว่า โฮสต์ กำลังทำขยับยั้งการส่งข้อมูลอยู่ และ device ต้องทำการรับและเก็บข้อมูลหรือคำสั่งที่ โฮสต์ ส่งมาให้หมด จนกว่า โฮสต์ จะทำการปล่อย คล็อกไลน์ คล็อกไลน์ ต้องอยู่ในสถานะ high ต่อเนื่องยาวนานอย่างน้อย 50 ms ก่อนตัวอุปกรณ์จึงจะสามารถเริ่มต้นการส่งข้อมูลของมันได้

เมาส์และคีย์บอร์ด ใช้ การสื่อสารแบบ อนุกรม (Serial protocol) โดยมีรูปแบบเป็น 11 เฟรมบิตตามนี้คือ:

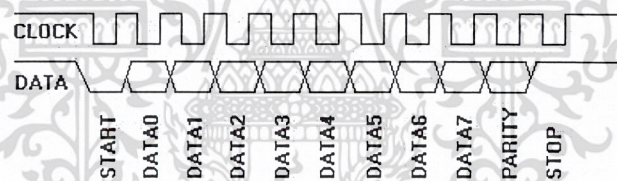
1 start bit : จะมีค่าเป็น “0” เสมอ

8 data bits : โดยจะส่งบิตที่มีลำดับความสำคัญน้อยที่สุด (LSB: Least Significant Bit) ไปก่อน

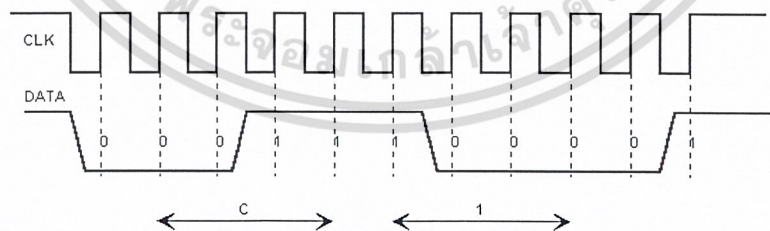
1 Parity bit : พาริตีบิต

1 stop bit : จะเป็น “1” เสมอ

เมาส์และคีย์บอร์ดทำการเขียนแต่ละบิตลงใน คาต้าไลน์เมื่อค็อกอยู่ในสถานะ High, และ มันจะทำการอ่านข้อมูลมาจาก โฮสต์ เมื่อค็อกมีสถานะเป็น low ตามรูปด้านล่างนี้



รูปที่ 2.11 แสดงสัญญาณบนคาต้าไลน์ และคล็อกที่เกิดขึ้นเมื่อมีการสื่อสารในทิศทางอุปกรณ์ไปโฮสต์ (Device-to-Host)



รูปที่ 2.12 ตัวอย่างการส่งข้อมูล ‘A’ จากคีย์บอร์ดไปยังพีซี

ความถี่ของคล็อกประมาณ 10 – 16.7 kHz เวลาจาก ขอบขาขึ้นของคล็อกพัลส์จนถึงการส่งข้อมูลต้องใช้เวลาน้อยกว่า 5 us และเวลาจากการส่งข้อมูลจนถึงขอบขาลงของคล็อกพัลส์ต้องใช้เวลาระหว่าง 5 – 25 us

โฮสต์อาจทำการขยับยั้งการส่งข้อมูลได้ตลอดเวลาโดยการ ดึง คล็อกไลน์ ลงมาเป็น low อย่างน้อย 100 ms ถ้าการส่งถูกขยับยั้งก่อนคล็อกพัลส์ลูกที่ 11 อุปกรณ์ (Device) ต้องยกเลิกการส่งข้อมูลที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังทำอยู่และทำการเตรียมส่งชุดของข้อมูลที่กำลังส่ง ก่อนถูกทำการยับยั้งจากโฮสต์อีกครั้ง เมื่อโฮสต์ทำการปล่อยคล็อก ชุดของข้อมูลที่กำลังส่ง ก่อนถูกทำการยับยั้งจากโฮสต์นั้นอาจจะเป็น make code , break code , device ID , mouse movement packet หรืออื่น ๆ ตัวอย่างเช่น ถ้าคีย์บอร์ดถูกทำการขัดจังหวะ (interrupt) ขณะที่กำลังส่งไบต์ ที่ 2 ของ เบริคโค้ด ซึ่งมีทั้งหมด 2 ไบต์ จะไม่ทำการส่งเฉพาะไบต์ที่ 2 แต่จะทำการส่งทั้ง 2 ไบต์ใหม่อีกครั้ง

ถ้าโฮสต์ทำการดึงคล็อกไลน์ (Clock line) เป็น low ก่อนเปลี่ยนของคล็อก (Clock) จาก high ไปยัง low ของการส่งครั้งแรก หรือ หลังจากขอบขาลงของ คล็อกพัลส์สุดท้าย ในกรณีนี้ อุปกรณ์ไม่ต้องทำการส่งข้อมูลใด ๆ ไปอีกครั้ง อย่างไรก็ตาม ถ้าข้อมูลใหม่ถูกสร้างและรอถูกส่ง มันจะต้องทำการกัน (buffer) ข้อมูลไว้ก่อนจนกระทั่งโฮสต์ จะทำการปล่อย คล็อกไลน์ เพื่อรองรับในกรณีนี้คีย์บอร์ดจึงมี 16 ไบต์ บัฟเฟอร์ และถ้าข้อมูลที่อยู่ในบัฟเฟอร์ (buffer) เต็มแล้วแต่เรายังต้องการส่งข้อมูลอีก ตัวอักษรที่ถูกกดหลังบัฟเฟอร์เต็มจะถูกละทิ้งจนกว่าบัฟเฟอร์จะว่าง

การติดต่อจากโฮสต์ไปยังตัวอุปกรณ์ (Host – to – Device)

ก่อนการส่งทุกครั้งตัวอุปกรณ์จะสร้างคล็อก ถ้า โฮสต์ต้องการส่งข้อมูล อย่างแรก มันต้องทำการดึง คล็อกไลน์ และ คาดำไลน์ให้อยู่ในสถานะ "Request – to- Send" ตามข้างล่างนี้

- ยับยั้งการส่งข้อมูลด้วยการดึง คล็อกไลน์ เป็น low เป็นเวลาอย่างน้อย 100 us
- จากนั้นทำการดึง คาดำไลน์เป็น low ก็จะอยู่ในสถานะ "Request – to- Send" แล้วจึงปล่อยคล็อก

อุปกรณ์ (Device) ควรทำการตรวจสอบสำหรับสถานะนี้ไม่ควรใช้เวลาเกิน 10 ms เมื่ออุปกรณ์ตรวจพบสถานะนี้ มันจะเริ่มสร้าง คล็อกและคล็อกใน ข้อมูล 8 บิต และ 1 – stop bit โฮสต์ จะทำการส่งบิตข้อมูลลงไปยัง คาดำไลน์เมื่อคล็อกไลน์ เป็น low เท่านั้นและข้อมูลจะถูกอ่านโดย device เมื่อคล็อกเป็น high กระบวนการนี้จะเกิดตรงข้ามกันเมื่อเป็นการสื่อสารในทิศทาง อุปกรณ์ไปโฮสต์

หลังจากอุปกรณ์ได้รับบิตสตอป (Stop bit) แล้ว อุปกรณ์จะทำการส่งสัญญาณ "acknowledge" ว่ารับข้อมูลเรียบร้อยแล้วโดยการทำให้ คาดำไลน์เป็น low และทำการสร้างคล็อกพัลส์สุดท้ายไป 1 ลูก ถ้า โฮสต์ ไม่ทำการปล่อย คาดำไลน์หลังจากคล็อกพัลส์ลูกที่ 11 อุปกรณ์จะทำการสร้างคล็อกพัลส์จนกระทั่ง คาดำไลน์ถูกปล่อย

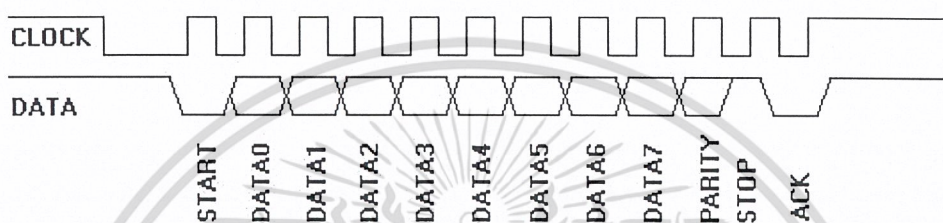
โฮสต์ อาจทำการยกเลิกการส่งก่อนคล็อกพัลส์ ลูกที่ 11 โดยการ ดึงคล็อก เป็น Low อย่างน้อย 100 us

ด้านล่างเป็นขั้นตอนการทำงานเมื่อส่งข้อมูลไปยัง Device

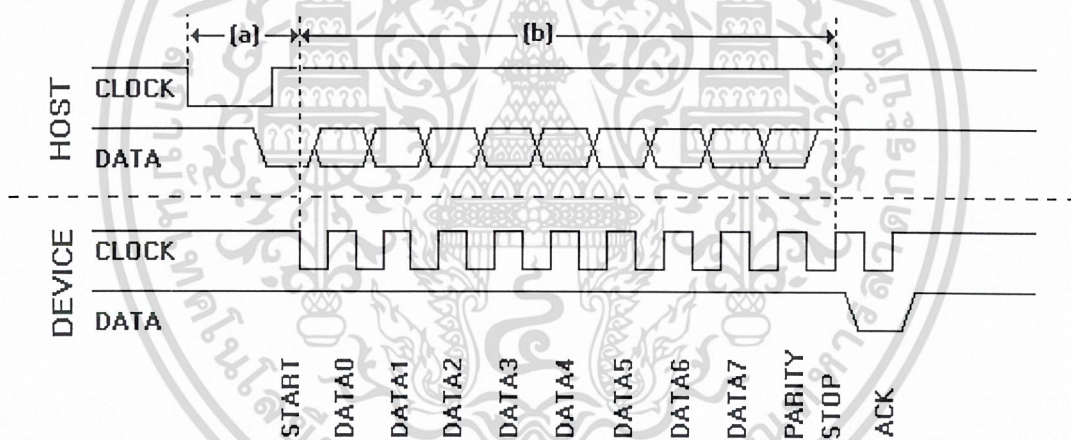
1. ดึงคล็อกให้เป็น low อย่างน้อย 100 us
2. ดึง คาดำไลน์ให้เป็น low
3. ปล่อย คล็อก line
4. กอຍการ acknowledge ของ device โดยรอรับคล็อกไลน์ เป็น low
5. Set / Reset คาดำไลน์เพื่อส่งข้อมูล บิต ที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คอยให้ device ส่งคล็อก high
7. คอยให้ device ส่งคล็อกเป็น low
8. ทำซ้ำข้อ 5-7 สำหรับอีก คาต่าบิต (data bit) ที่เหลืออีก 7 บิต และ parity bit
9. ปล่อย data line
10. คอยให้ device ส่ง data low มา
11. คอยให้ device ส่งคล็อกเป็น low มา
12. คอยให้ device ปล่อยคาต่า และ คล็อก



รูปที่ 2.13 แสดงสัญญาณบนคาต่าไลน์และคล็อก ที่เกิดขึ้นเมื่อมีการสื่อสารในทิศทางโฮสต์ไปยังอุปกรณ์



รูปที่ 2.14 แสดงช่วงเวลาที่เกิดการสื่อสารระหว่าง โฮสต์ไปอุปกรณ์

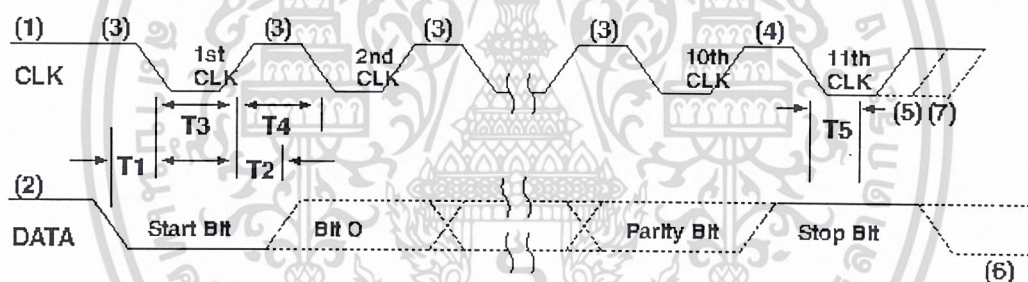
ตามรูป มี 2 จำนวนที่ โฮสต์ จะคำนึงถึง (a) คือเวลาที่ device เริ่มgenerate คล็อกพัลส์หลังจาก โฮสต์ ทำการ initial โดยทำให้คล็อก เป็น low อย่างน้อย 15 ms (b) เป็นเวลาการส่งชุดข้อมูล อย่างน้อยต้องใช้เวลา 2 ms ถ้าทั้ง (a)และ(b) ไม่เป็นตามนี้ โฮสต์ จะ สร้างสัญญาณผิดพลาด (generate error) จากนั้น หลังจาก “acknowledge” ได้รับเรียบร้อย ถ้าคำสั่ง(คำสั่ง)จาก โฮสต์ ต้องการ “response” , response ต้องไม่ได้รับช้ากว่า 20 ms หลังจาก โฮสต์ ปล่อยคล็อกไลน์ถ้าไม่มี response โฮสต์จะสร้าง สัญญาณผิดพลาด

2.11 ระบบการรับข้อมูล (System Receiving Data)

ค่าเวลา (Timing) ของการรับข้อมูลจากอุปกรณ์ต่อพ่วง (เมาส์/คีย์บอร์ด) ต่อไปจะอธิบายลำดับ การทำงานที่เกิดขึ้นเมื่อระบบกำลังทำการรับ data จากอุปกรณ์ต่อพ่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คีย์บอร์ด/เมาส์ จะตรวจสอบคล็อกถ้า คล็อกไลน์ (clock line) เป็น “0” output ที่มาจากคีย์บอร์ด/เมาส์ จะไม่ได้รับการอนุญาต
2. คีย์บอร์ด/เมาส์ ตรวจสอบค่าไลน์ (data line) ถ้า ค่าไลน์เป็น “0” ตัวควบคุม (Controller) ทำการรับข้อมูลจากคอมพิวเตอร์
3. คีย์บอร์ด/เมาส์ ตรวจสอบ คล็อกไลน์ (clock line) ในระหว่างทำการส่งถ้าใช้เวลาไม่เกิน 100 us ถ้า คีย์บอร์ด ค้นหา ระบบทำการครอบครองคล็อกไลน์ (clock line) ไว้ การส่งจะสิ้นสุดลง ระบบสามารถจบการส่งได้ทุกเวลา ตลอดเวลาในช่วงคล็อก 10 ลูกแรก
4. ชุดท้ายจะตรวจสอบการสิ้นสุดการส่ง โดยกระทำการหลังจากผ่าน 10 ลูกแรกไป อย่างน้อย 5 us
5. คอมพิวเตอร์สามารถให้คล็อกไลน์ (clock line) inactive (เป็น “0”) ไว้เพื่อยับยั้งการส่งครั้งต่อไปไว้ก่อน
6. คอมพิวเตอร์สามารถ Set ค่าไลน์ (Data line) ให้ inactive ถ้าไปต ถูกส่งไปยังอุปกรณ์ค่าไลน์ (data line) จะ Set เป็น inactive เมื่อ Start Bit (0 เสมอ) ปรากฏบน ค่าไลน์
7. ระบบปล่อยคล็อกไลน์ (clock line) อนุญาตให้ส่งครั้งต่อไป



รูปที่ 2.15 เวลาต่างๆที่คอมพิวเตอร์ใช้ในการรับข้อมูล

	TIMING PARAMETER	Min	Max
T1	เวลาการส่งข้อมูล ที่ขอบล่างของสัญญาณ CLK	5 us	25 us
T2	เวลาขอบขาขึ้นของสัญญาณ CLK ขณะที่มีการส่งข้อมูล	5 us	T4-5 us
T3	ช่วงเวลาที่สัญญาณ CLK เป็น “0”	30 us	50 us
T4	ช่วงเวลาที่สัญญาณ CLK เป็น “1”	30 us	50 us
T5	เวลาที่ใช้ในการยับยั้งอุปกรณ์ไม่ให้ส่งสัญญาณ หลังจากผ่านสัญญาณ CLK ลูกที่ 11 แล้ว	> 0 us	50 us

ตารางที่ 2.18 รายละเอียดของเวลาในการรับข้อมูล

2.12 ระบบการส่งข้อมูล (System Sending Data)

1. คอมพิวเตอร์ตรวจสอบว่า คีย์บอร์ด/เมาส์ กำลังดำเนินการขบวนกรส่งข้อมูลหรือไม่ ถ้าการส่งอยู่ในขั้นดำเนินการถัดจาก 10 ลูกแรก ระบบต้องทำการรับข้อมูล
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คีย์บอร์ด/เมาส์ จะตรวจสอบคล็อกไลน์ (clock line) ถ้า คล็อกเป็น “0” กระบวนการ I/O จะถูกยับยั้ง

3. คีย์บอร์ด/เมาส์ ตรวจสอบค่าไลน์ (data line) ถ้า ค่าไลน์เป็น “0” คอมพิวเตอร์ต้องทำการส่งข้อมูลค่าไลน์ (Data line) จะเป็น “0” เมื่อ บิตสตาร์ท (Start Bit) (“0” เสมอ) ปรากฏบน ค่าไลน์

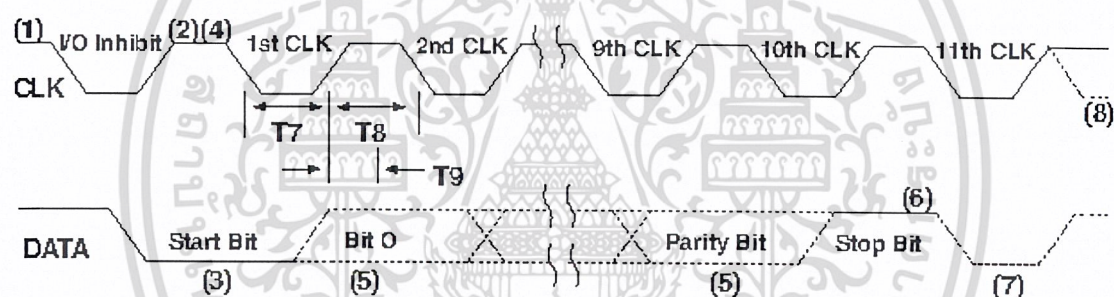
4. คีย์บอร์ด/เมาส์ จะ Setคล็อกไลน์เป็น “0” ระบบจะส่งบิตแรกลงบน ค่าไลน์แต่ละครั้งที่คีย์บอร์ด/เมาส์ Setคล็อกไลน์เป็น “0” ระบบจะส่งบิตต่อ ๆ ไปลงบนค่าไลน์ (Data line) จนกระทั่งทุกบิต ถูกส่งไปจนครบ

5. คีย์บอร์ด/เมาส์

6. คีย์บอร์ด/เมาส์ ตรวจสอบ positive-level ของ Stop-Bitหลังจากคล็อก10 ลูกแรก ถ้าเป็น “0” คีย์บอร์ด/เมาส์ จะเกิดที่ครั้งต่อไปเพื่อส่งหรือส่งซ้ำไปยังคอมพิวเตอร์

7. คีย์บอร์ด/เมาส์ ตั้ง ค่าไลน์ให้เป็น “0”

8. คอมพิวเตอร์สามารถดึงคล็อก (clock line) ให้เป็น “0” เพื่อยับยั้งการส่งข้อมูลมาจากคีย์บอร์ด



รูปที่ 2.16 เวลาต่างๆ ที่คอมพิวเตอร์ใช้ในการส่งข้อมูล

	TIMING PARAMETER	Min	Max
T7	ช่วงเวลาที่สัญญาณ CLK เป็น “0”	30 us	50 us
T8	ช่วงเวลาที่สัญญาณ CLK เป็น “1”	30 us	50 us
T9	เวลาจากสัญญาณ CLK เปลี่ยนจาก “0” เป็น “1”	30 us	50 us
T4	ช่วงเวลาที่สัญญาณ CLK เป็น “0”	30 us	50 us
T5	เวลาที่ใช้ในการยับยั้งอุปกรณ์ไม่ให้ส่งสัญญาณหลังจากผ่านสัญญาณ CLK ลูกที่ 11 แล้ว	> 0 us	50 us

ตารางที่ 2.19 รายละเอียดของเวลาในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

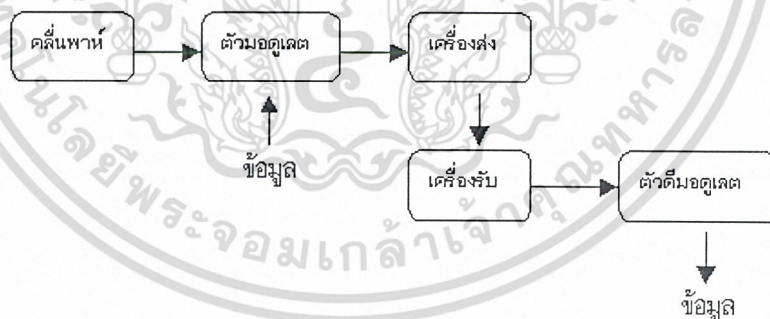
ส่วนของเครื่องรับส่งวิทยุ

สัญญาณอิเล็กทรอนิกส์ที่ใช้ในการสื่อสารสามารถแบ่งได้เป็น 2 ประเภทคือ สัญญาณอนาล็อก และสัญญาณดิจิทัล สัญญาณอนาล็อกได้แก่สัญญาณเสียง และสัญญาณในธรรมชาติทั้งหมด ปัญหาที่สำคัญของสัญญาณ อนาล็อกก็คือเรื่องสัญญาณรบกวน ซึ่งในบางครั้งอาจทำให้ระบบไม่สามารถใช้งานได้เลย ดังนั้นจึงมีการนำสัญญาณดิจิทัลเข้ามาแทนที่

1. สัญญาณแบบอนาล็อก (Analog Signal) จะเป็นสัญญาณแบบต่อเนื่องที่ทุกๆ ค่า ที่เปลี่ยนแปลงไป ของระดับสัญญาณจะมีความหมาย การส่งสัญญาณแบบนี้จะถูกรบกวนให้มีการแปลความหมายผิดพลาดได้ง่าย เนื่องจากค่าทุกค่าถูกนำมาใช้งาน ซึ่งสัญญาณแบบอนาล็อกนี้จะเป็นสัญญาณที่สื่อกลางใน การสื่อสารส่วนมากใช้อยู่เช่น สัญญาณเสียงในสายโทรศัพท์ เป็นต้น

2. สัญญาณแบบดิจิทัล (Digital Signal) จะประกอบขึ้นจากระดับสัญญาณเพียง 2 ค่าคือ สัญญาณ ระดับสูงสุด และสัญญาณระดับต่ำสุด ดังนั้นจะมีประสิทธิภาพ และความน่าเชื่อถือสูงกว่าแบบอนาล็อกเนื่องจากการใช้งานค่าสองค่า เพื่อนำมาตีความหมายเป็น on/off หรือ 0/1 เท่านั้น ซึ่งเป็นสัญญาณที่ คอมพิวเตอร์ใช้ในการติดต่อสื่อสารกัน

การมอดูเลต (Modulation) เป็นการผสมสัญญาณของข้อมูลเข้ากับสัญญาณอีกสัญญาณหนึ่ง เรียกว่า คลื่นพาห้ (carrier) ซึ่งสัญญาณนี้มีความถี่ที่เหมาะสมกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น การเลือกวิธีมอดูเลตขึ้นอยู่กับปัจจัยหลายประการ เช่น ชนิดของสัญญาณ แบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น



รูปที่ 2.17 รูปแบบของการสื่อสารในการรับส่งสัญญาณ

จากรูปแสดงรูปแบบของการสื่อสารในการรับส่งสัญญาณอย่างง่าย โดยคลื่นพาห้ผสมสัญญาณข้อมูลที่ตัวมอดูเลต (Modulator) แล้วส่งไปที่เครื่องส่ง จากเครื่องส่งไปยังเครื่องรับจะเป็นช่องสัญญาณสำหรับลำเลียงสัญญาณผสมนี้ สัญญาณผสมจากเครื่องรับจะไปเข้าตัวดีมอดูเลต (Demodulate) เพื่อแยกสัญญาณข้อมูลออกมา เพื่อให้เราสามารถสื่อสารไปได้ไกลยิ่งขึ้น จำเป็นต้องมอดูเลตสัญญาณข่าวสารลงพาหะ วิธีมอดูเลตแบ่งออกเป็น 2 แบบที่สำคัญคือ มอดูเลตทางแอมพลิจูด และมอดูเลตเชิงมุมซึ่งการมอดูเลตทั้งสองแบบนี้เป็นการมอดูเลตเชิงอนาล็อก (Analog)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารอนาล็อกเป็นระบบที่ออกแบบให้ส่งข้อมูลสัญญาณอนาล็อกเช่น สัญญาณเสียง แต่ได้มีการพัฒนาจนประยุกต์ให้สามารถส่งข่าวสารได้ด้วยในปัจจุบัน ปัญหาสำคัญสำหรับการสื่อสารอนาล็อกก็คือเรื่องสัญญาณรบกวน แต่เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณอนาล็อกจึงยังคงเห็นการพัฒนาของการสื่อสารแบบอนาล็อกในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) การมอดูเลตความถี่ (Frequency Modulation หรือ FM)

การมอดูเลตที่ใช้อยู่ในปัจจุบัน 3 วิธีได้แก่

1. การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) วิธีนี้แอมพลิจูดของคลื่นพาห้จะเปลี่ยนแปลงตามสัญญาณของข้อมูลเข้ามา การมอดูเลตแบบ AM เป็นวิธีที่ง่ายที่สุดในการมอดูเลตแต่คุณภาพของสัญญาณไม่ดี มีความต้านทานสัญญาณรบกวนต่ำ เหมาะกับข้อมูลที่ไม่ต้องการคุณภาพมากนัก เช่น สัญญาณเสียง เป็นต้น

2. การมอดูเลตความถี่ (Frequency Modulation หรือ FM) วิธีการนี้เป็นการเปลี่ยนแปลงความถี่ของคลื่นพาห้ตามสัญญาณของข้อมูลเข้ามา การมอดูเลตแบบความถี่ ให้คุณภาพที่ดีกว่าการมอดูเลตแบบแอมพลิจูด แต่ระบบจะซับซ้อนกว่า

3. การมอดูเลตเฟส (Phase Modulation หรือ PM) เป็นการมอดูเลตที่ใช้ในการเปลี่ยนแปลงเฟสของคลื่นพาห้ตามสัญญาณข้อมูลเข้ามา ทั้งคุณภาพของสัญญาณและความซับซ้อนไม่ค่อยแตกต่างจากการมอดูเลตแบบความถี่เท่าใดนัก ข้อแตกต่างระหว่างการมอดูเลตแบบความถี่กับการมอดูเลตแบบเฟสคือการมอดูเลตแบบเฟสใช้คลื่นพาห้เพียงความถี่เดียว การมอดูเลตและดีมอดูเลตสามารถทำได้ประหยัดกว่า แต่ไม่ได้หมายความว่าซับซ้อนน้อยกว่า

ในที่นี้จะกล่าวถึงการมอดูเลตแอมพลิจูดซึ่งใช้ในการรับส่งสัญญาณในการทดลองครั้งนี้

2.13 การมอดูเลตแอมพลิจูดแบบแถบข้างคู่จัดคลื่นพาห้

ในกระบวนการมอดูเลตแอมพลิจูดนั้น ขนาดของสัญญาณข่าวสารจะถูกใช้ไปบังคับค่าแอมพลิจูด A_c ของคลื่นพาห้ $A_c \cos(\omega_c t + \theta_c)$ ให้เปลี่ยนแปลงในขณะที่ความถี่ ω_c และเฟส θ_c จะยังมีค่าคงเดิม สมการของคลื่นที่มอดูเลตแล้ว $\phi_{DSB-SC}(t)$ จะเป็นดังนี้คือ

$$\phi_{DSB-SC}(t) = km(t)\cos(\omega_c t + \theta_c) \quad (2-1)$$

โดย k คือค่าคงที่ ที่กระบวนการมอดูเลตจัดการควบคุมสัดส่วนของ A_c ให้แปรผันตามสัญญาณข่าวสาร $m(t)$

เพื่อความสะดวกในการวิเคราะห์สัญญาณ เราสามารถสมมุติให้ $k = 1$ และ $\theta_c = 0$ ได้โดยไม่เสียความหมายของการวิเคราะห์สัญญาณโดยทั่วไปแต่อย่างไร เพียงแต่จะทำให้รูปของสมการที่กระชับขึ้น คือ จะได้สมการเป็น

$$\phi_{DSB-SC}(t) = m(t)\cos(\omega_c t) \quad (2-2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะทำความเข้าใจถึงการเปลี่ยนแปลงที่เกิดขึ้นใน โดเมนความถี่ จะสมมุติให้ $M(\omega)$ คือ ค่าฟังก์ชันสเปกตรัมของ $m(t)$ กล่าวคือ

$$m(t) \leftrightarrow M(\omega) \quad (2-3)$$

โดยอาศัยคุณสมบัติของการแปลงฟูเรียร์ จะได้

$$m(t)\cos(\omega_c t) \leftrightarrow \frac{1}{2}[M(\omega + \omega_c) + M(\omega - \omega_c)] \quad (2-4)$$

นั่นคือสัญญาณเอเอ็มในโดเมนความถี่ $\Phi_{DSB-SC}(\omega)$ จะมีค่าเป็น

$$\Phi_{DSB-SC}(\omega) = \frac{1}{2}[M(\omega + \omega_c) + M(\omega - \omega_c)] \quad (2-5)$$

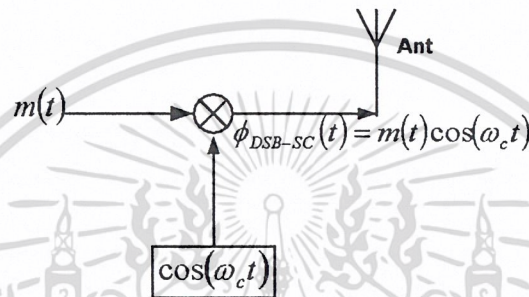
ถ้าแบนด์วิดท์ของ $M(\omega)$ ตามสมการ (2-3) มีค่าเป็น B เฮิร์ตซ์หรือ W เรเดียนต่อวินาที ($W = 2\pi B$) เราจะรู้จากสมการ (2-5) ได้ว่า $\Phi_{DSB-SC}(\omega)$ จะมีแบนด์วิดท์เป็น $2W$ เรเดียนต่อวินาทีดังรูปที่ 2.18 และเป็นที่น่าสังเกตว่า ค่าความถี่กึ่งกลางแบนด์ของสัญญาณเอเอ็มนั้นอยู่ที่ความถี่ของคลื่นพาห้คือ ω_c ซึ่งแกนตั้งที่ ω_c ในโดเมนของความถี่นี้จะแบ่งสเปกตรัมที่เกิดขึ้นออกเป็นสองส่วนที่สมมาตรกัน ส่วนของสัญญาณที่มีสเปกตรัมอยู่ที่ความถี่สูงกว่าและต่ำกว่าความถี่ ω_c นั้นมีชื่อว่า แถบข้างส่วนบนหรือ ไซด์แบนด์สูง (upper side band) นิยมเขียนย่อแทนด้วย ยูเอสบี (USB) และแถบข้างส่วนล่างหรือ ไซด์แบนด์ต่ำ (lower side band) นิยมเขียนย่อด้วย แอลเอสบี (LSB) ตามลำดับในทำนองเดียวกันบนแกนความถี่ด้านลบค่าความถี่ $-\omega_c$ ก็จะทำให้เกิด ไซด์แบนด์สูงและ ไซด์แบนด์ต่ำเช่นกันแต่ในกรณีของความถี่ด้านลบ ไซด์แบนด์สูงจะหมายถึงส่วนสเปกตรัมที่มีความถี่เป็นลบมากกว่า $-\omega_c$ และ ไซด์แบนด์ต่ำจะหมายถึงส่วนสเปกตรัมที่มีความถี่เป็นลบน้อยกว่า $-\omega_c$

ควรสังเกตอีกด้วยว่า ถ้าสัญญาณ $m(t)$ นั้นไม่มีส่วนประกอบเชิงความถี่ที่เป็นไฟตรง ไซด์แบนด์ทั้งสองที่เกิดจากการมอดูเลตก็จะอยู่ห่างกันอย่างเด่นชัด และสเปกตรัมของสัญญาณเอเอ็มก็จะไม่เกิดมีองค์ประกอบความถี่ที่ $\omega = \pm\omega_c$ ปรากฏ กล่าวคือ ส่วนประกอบสัญญาณที่มีความถี่ตรงกับ ความถี่ของคลื่นพาห้จะถูกขจัดออกไปในกรณีดังกล่าว ด้วยเหตุนี้เองทำกระบวนการมอดูเลตสัญญาณแบบนี้ได้ชื่อว่า การมอดูเลตแอมพลิจูดแบบ ไซด์แบนด์คู่ขจัดคลื่นพาห้ (AM double-sideband-suppressed carrier) หรือการมอดูเลตแอมพลิจูดแบบแถบข้างคู่ขจัดคลื่นพาห้ซึ่งเรียวย่อว่า ดีเอสบี เอสซี (DSB-SC) ตาม สมการ (2-5) จะบอกให้รู้ว่า การมอดูเลตแบบนี้จะทำให้เกิดการย้ายสเปกตรัมของสัญญาณข่าวสาร $M(\omega)$ ออกไปจากเดิม $\pm\omega_c$ (หมายถึงย้ายไปจากเดิม $+\omega_c$ และ $-\omega_c$ ตามลำดับ) ในการที่จะแยกสัญญาณข่าวสาร $m(t)$ ออกมาจาก $\phi_{DSB-SC}(t)$ นั้นลักษณะการทำหน้าที่นี้เรียกว่า การกู้สัญญาณ

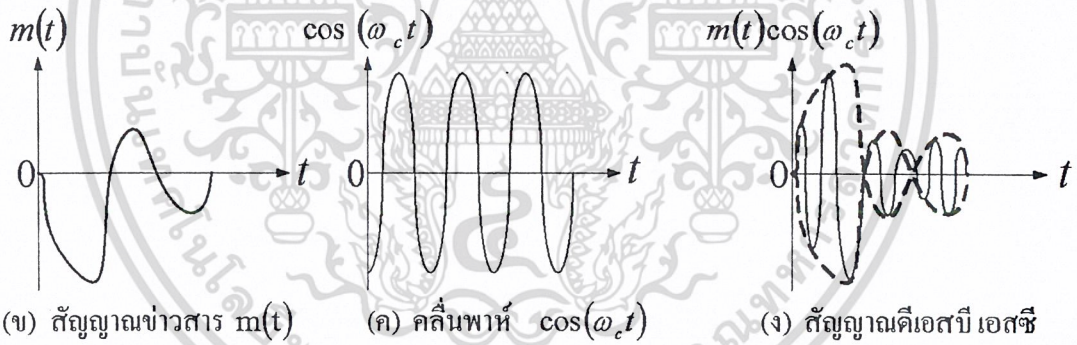
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ การดีมอดูเลต (Demodulate) สัญญาณหรือการตรวจจับอีกครั้งหนึ่ง ซึ่งเมื่อทำเช่นนี้แล้ว จะได้ผลลัพธ์ $v_d(t)$ เป็น

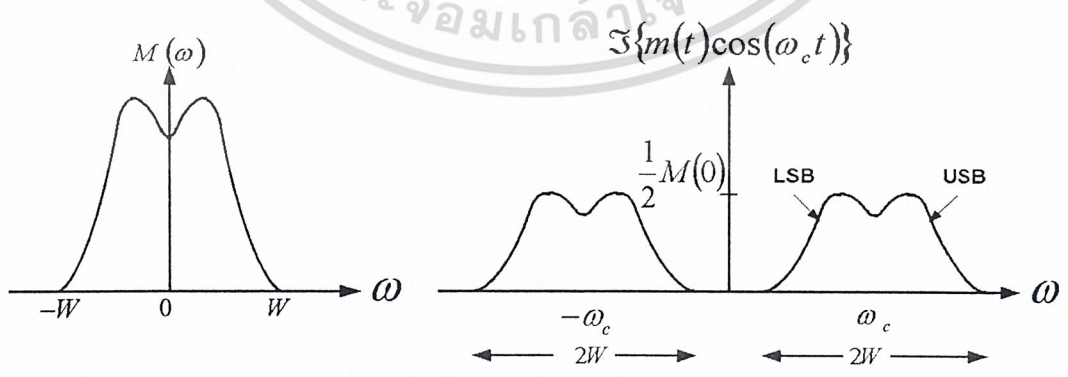
$$\begin{aligned}
 v_d(t) &= \phi_{DSB-SC}(t) \cos(\omega_c t) \\
 &= \{m(t) \cos(\omega_c t)\} \cos(\omega_c t) \\
 &= m(t) \cos^2(\omega_c t) \\
 &= \frac{1}{2} m(t) + \frac{1}{2} m(t) \cos(2\omega_c t)
 \end{aligned}
 \tag{2-6}$$



(ก) แผนภาพแสดงกระบวนการมอดูเลตสัญญาณดีเอสบี เอสซี



(ข) สัญญาณข่าวสาร $m(t)$ (ค) คลื่นพาห้ $\cos(\omega_c t)$ (ง) สัญญาณดีเอสบี เอสซี



(จ) ความหนาแน่นของสเปกตรัมของ $m(t)$ (ฉ) ความหนาแน่นของสัญญาณดีเอสบี เอสซี
 ในที่นี้ $W = 2\pi B$

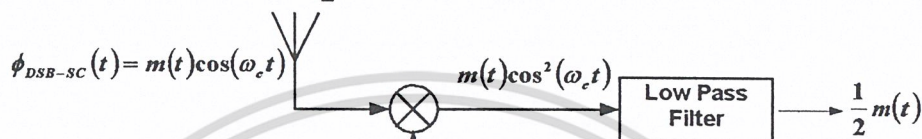
รูปที่ 2.18 กระบวนการมอดูเลตสัญญาณดีเอสบี เอสซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

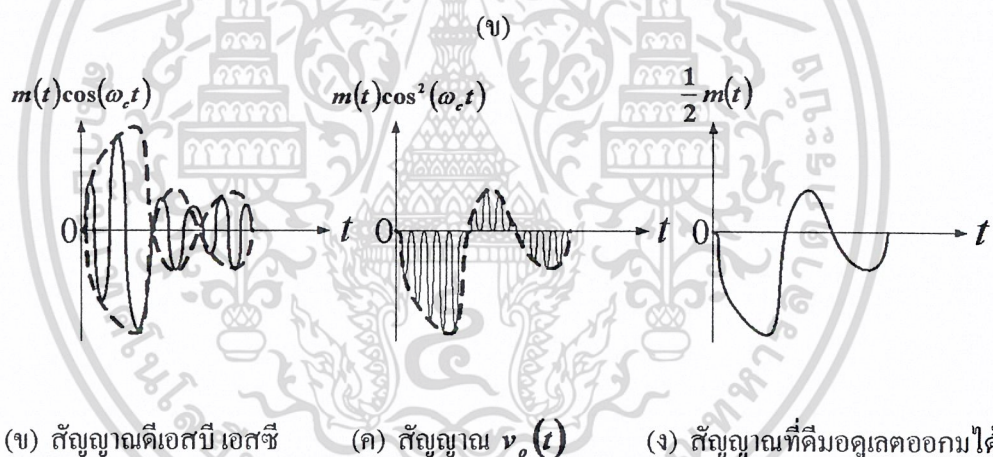
และโดยคุณสมบัติของการแปลงฟูเรียร์จะได้

$$\phi_{DSB-SC}(t)\cos(\omega_c t) \leftrightarrow \frac{1}{2}M(\omega) + \frac{1}{4}[M(\omega + 2\omega_c) + M(\omega - 2\omega_c)] \quad (2-7)$$

เมื่อนำสัญญาณ $\phi_{DSB-SC}(t)\cos(\omega_c t)$ นี้ไปผ่านวงจรกรองความถี่ต่ำผ่านเพื่อขจัดสัญญาณที่มีความถี่สูงคือพจน์ที่อยู่ในวงเล็บใหญ่ของ (2-4) ออก ก็จะเหลือเพียงแต่ส่วนของสเปกตรัม $\frac{1}{2}M(\omega)$ ออกมาซึ่งก็คือสเปกตรัมของสัญญาณเดิม $\frac{1}{2}m(t)$ นั่นเอง



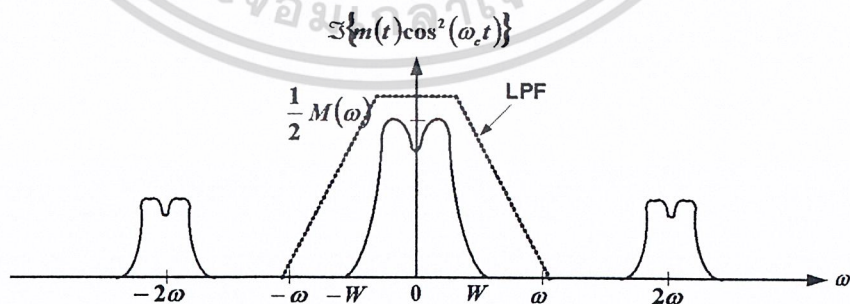
(ก) แผนภาพแสดงกระบวนการคืนมอดูเลตสัญญาณคือเอสบี เอสซี



(ข) สัญญาณเอสบี เอสซี

(ค) สัญญาณ $v_o(t)$

(ง) สัญญาณที่คืนมอดูเลตออกมาได้



(จ) สเปกตรัมของสัญญาณ $v_o(t)$

รูป 2.19 การคืนมอดูเลตสัญญาณคือเอสบี เอสซี และสัญญาณต่าง ๆ พร้อมทั้งค่าความหนาแน่นสเปกตรัมของสัญญาณที่อินพุตของวงจรกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามในการตรวจจับสัญญาณด้วยวิธีการดังกล่าวนี้ มักจะพบปัญหาอันเนื่องมาจากคลื่นพาห้ที่เครื่องรับสร้างขึ้นมานั้น มักจะมีความถี่และเฟสไม่เท่ากับความถี่และเฟสของคลื่นพาห้จากเครื่องส่ง การที่จะเข้าใจถึงผลของความคลาดเคลื่อนของคลื่นพาห้ที่นั่น ทำได้โดยสมมุติว่าเมื่อมีสัญญาณจากเครื่องส่งคือ $\phi_{DSB-SC}(t) = m(t)\cos(\omega_c t)$ นั้นคลื่นพาห้จากออสซิลเลเตอร์ทางเครื่องรับมีความคลาดเคลื่อนไปเล็กน้อยเป็น $\cos\{(\omega_c + \Delta\omega)t + \delta\}$ โดย $\Delta\omega$ และ δ คือค่าความคลาดเคลื่อนทางความถี่และเฟสตามลำดับ

ตามกระบวนการคิมอดูเลต เครื่องรับจะทำการคิมอดูเลตสัญญาณโดยใช้คลื่นพาห้ $\cos\{(\omega_c + \Delta\omega)t + \delta\}$ คูณกับ $\phi_{DSB-SC}(t)$ ทำให้ได้สัญญาณผลลัพธ์เป็น

$$\begin{aligned} v_d(t) &= \phi_{DSB-SC}(t)\cos\{(\omega_c + \Delta\omega)t + \delta\} \\ &= m(t)\cos(\omega_c t)\cos\{(\omega_c + \Delta\omega)t + \delta\} \\ &= \frac{1}{2}m(t)\cos(\Delta\omega t + \delta) + \frac{1}{2}m(t)\cos\{(2\omega_c + \Delta\omega)t + \delta\} \end{aligned} \quad (2-8)$$

วงจรกรองความถี่ต่ำผ่านในเครื่องรับจะขจัดส่วนความถี่สูง คือ พจน์หลังสุดของสมการ (2-8) ออก คงเหลือสัญญาณเอาต์พุต $v_o(t)$ ของวงจรตรวจจับสัญญาณ คือ

$$v_o(t) = \frac{1}{2}m(t)\cos(\Delta\omega t + \delta) \quad (2-9)$$

ถ้าเครื่องผลิตคลื่นพาห้ขึ้นมาโดยไม่มี ความคลาดเคลื่อนกล่าวคือ $\Delta\omega = 0$ และ $\delta = 0$ จากสมการ(2-9) จะพบว่าเพราะ $\cos(0) = 1$ ดังนั้น $v_o(t)$ จึงมีค่าเท่ากับ $\frac{1}{2}m(t)$ ซึ่งเป็นสัญญาณข่าวสาร $m(t)$ ที่ถูกต้อง แต่ถ้าเครื่องรับสร้างคลื่นพาห้ได้มีความถี่ถูกต้อง (คือ $\Delta\omega = 0$) แต่ทว่ายังมี ความคลาดเคลื่อนทางเฟสอยู่บ้าง (คือ $\delta \neq 0$) ในกรณีนี้เอาต์พุตตามสมการ (2-8) จะกลายเป็น

$$v_o(t) = \frac{1}{2}m(t)\cos(\delta) \quad (2-10)$$

เนื่องจาก $|\cos(\delta)| \leq 1$ ดังนั้นผลที่เกิดขึ้นก็คือ สัญญาณเอาต์พุตจะมีระดับลดลง และ ถ้า $\delta = \pm \frac{\pi}{2}$ ระดับของสัญญาณเอาต์พุตก็จะเป็นศูนย์ (เพราะ $\cos\left(\pm \frac{\pi}{2}\right) = 0$)

ถ้าสมมุติว่าเครื่องรับสร้างคลื่นพาห้ที่มีเฟสเริ่มต้นถูกต้อง ($\delta = 0$) แต่มีความถี่คลาดเคลื่อน ($\Delta\omega \neq 0$) จะพบว่าเอาต์พุตตามสมการ (2-9) จะได้เป็น

$$v_o(t) = \frac{1}{2} m(t) \cos(\Delta\omega t) \quad (2-11)$$

ซึ่งแสดงให้เห็นว่า $v_o(t)$ มีลักษณะผิดเพี้ยนไปจาก $m(t)$ ด้วยอิทธิพลของการบังคับขนาด (การคูณ) ด้วยคลื่นรูปไซน์ที่มีความถี่ $\Delta\omega$ ซึ่งเป็นความถี่ต่ำ ผลที่เกิดขึ้นนี้เรียกว่า มีการบีต (Beating) ด้วยความถี่ต่ำนั้น ทำให้เอาต์พุตไม่ใช่สัญญาณข่าวสาร $m(t)$ เดิม

เราได้รู้ว่าการตรวจจับสัญญาณดีเอสบี เอสซี นั้นทำด้วยการมอดูเลตคลื่นพาห์ที่สร้างขึ้นทางเครื่องรับเข้าเข้ากับสัญญาณ $(\phi_{DSB-SC}(t))$ นั่นอีกครั้งหนึ่ง แล้วใช้วงจรกรองความถี่ต่ำผ่านมาจัดส่วนประกอบที่เป็นความถี่สูงออก ก็จะได้สัญญาณข่าวสารเดิมกลับคืนมาประเด็นสำคัญในกระบวนการนี้คือ คลื่นพาห์ทางเครื่องรับจะต้องมีความถี่และเฟสตรงกับคลื่นพาห์จากเครื่องส่งพอดี การตรวจจับสัญญาณจึงจะมีประสิทธิภาพเต็มที่ จึงทำให้กระบวนการตรวจจับสัญญาณแบบนี้มีชื่อว่า การดีมอดูเลตแบบสัมพันธ์ (Synchronous demodulation) หรือการดีมอดูเลตแบบร่วมนัย (Coherent demodulation)

การสร้างวงจรมอดูเลตสัญญาณแบบแถบข้างคู่ขจัดคลื่นพาห์ อาจทำได้โดยใช้วงจรมอดูเลตแบบตัด (Chopper modulator) แบบไม่เชิงเส้น ฯลฯ และที่เคยได้รับความนิยมมากคือ ริงมอดูเลเตอร์ (Ring modulator) สำหรับวงจรรับนั้นวงจรเครื่องรับที่น่าสนใจคือเครื่องรับแบบคอสแทส ซึ่งเป็นเครื่องรับที่นิยมใช้กันอยู่ สามารถที่จะปรับค่าความถี่ของสัญญาณจากออสซิลเลเตอร์ภายในเครื่องรับให้ตรงกับความถี่ของคลื่นพาห์จากเครื่องส่งได้เลยอัตโนมัติ

2.14 การมอดูเลตแอมพลิจูดสำหรับวิทยุกระจายเสียง

จากที่ได้รู้ว่าการดีมอดูเลตสัญญาณดีเอสบี เอสซี และสัญญาณแอสเอสบี สามารถทำได้โดยวิธีการดีมอดูเลตแบบสัมพันธ์ ซึ่งเป็นวิธีการที่จะต้องให้การผลิตคลื่นพาห์ทางด้านเครื่องรับให้มีความถี่และเฟสให้เท่ากับค่าของคลื่นพาห์จากเครื่องส่ง และการที่จะควบคุมให้เครื่องรับสามารถสร้างคลื่นพาห์ที่ถูกต้องได้ จะต้องอาศัยวงจรที่มีความซับซ้อนพอควร ซึ่งย่อมเป็นการแน่นอนว่า เมื่อเป็นเช่นนี้ต้นทุนในการสร้างเครื่องรับดังกล่าวก็ย่อมจะต้องมีราคาสูงด้วย เพื่อลดปัญหาในเรื่องนี้จึงได้เกิดการคิดหาวิธีการที่จะดีมอดูเลตสัญญาณ ดีเอสบี เอสซี ชนิดที่ไม่จำเป็นต้องมีการสร้างคลื่นพาห์ทางด้านเครื่องรับขึ้น วิธีการนี้ทำได้โดยการใช้คลื่นพาห์ $\cos(\omega_c t)$ ซึ่งมีขนาดที่เหมาะสมส่งรวมไปกลับสัญญาณดีเอสบี เอสซี จากเครื่องส่ง การกระทำเช่นนี้ทำให้คลื่นสัญญาณที่ส่งมีรูปฟังก์ชันเป็น

$$\phi_{AM}(t) = A_c \cos(\omega_c t) + m(t) \cos(\omega_c t) \quad (2-12)$$

โดยในที่นี้ A_c คือ ค่าแอมพลิจูดของคลื่นพาห์ที่ใช้เพิ่มเข้ามาและเมื่อจัดรูป (2-12) ใหม่จะได้

$$\phi_{AM}(t) = A_c \left(1 + \frac{m(t)}{A_c} \right) \cos(\omega_c t) \quad (2-13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติค่าความถี่ของคลื่นพาห์ ω_c จะมีค่าสูงกว่าค่าความถี่ที่เป็นองค์ประกอบสูงสุดของสัญญาณ $m(t)$ มาก ดังนั้นถ้าขนาดของคลื่นพาห์ A_c มีค่ามากพอที่จะทำให้พจน์ในวงเล็บแรกของ (2-13) มีค่าไม่เป็นลบตลอดเวลาทุกค่า t ใดๆ แล้วสัญญาณ $\phi_{AM}(t)$ ตามสมการ (2-13) ก็จะมียอดคลื่นที่ค่อยๆ เปลี่ยนไปตามเอนVELOPE หรือกรอบที่มีลักษณะเหมือนกับสัญญาณ $m(t)$ ทุกประการ เพราะฉะนั้นถ้านำสัญญาณ $\phi_{AM}(t)$ นี้ไปป้อนให้กับ วงจรตรวจจับกรอบสัญญาณ (Envelope detector) แล้วก็จะทำให้เกิดการตรวจจับแยกสัญญาณ $m(t)$ กลับออกมาได้

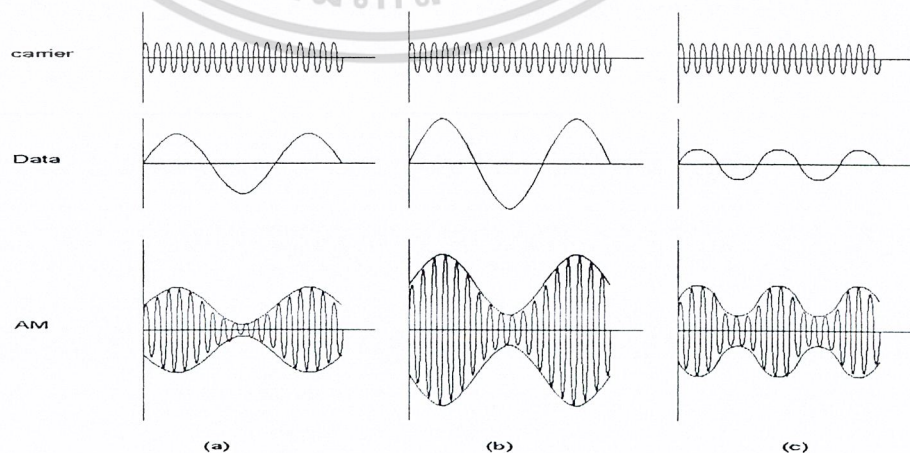
อย่างไรก็ตามสมการของสัญญาณเอเอ็ม (Standard AM) โดยทั่วไปนั้นนิยมเขียนคัดแปลงไปจาก (2-13) เล็กน้อยคือ

$$\phi_{AM}(t) = A_c(1 + k_a m(t)) \cos(\omega_c t) \quad (2-14)$$

โดยที่ k_a นั้นเทียบได้กับ $\frac{1}{A_c}$ และมีชื่อเรียกว่าความไวของการมอดูเลตแอมพลิจูด (AM modulation sensitivity) แต่อย่างไรก็ตาม k_a นี้ไม่จำเป็นต้องเท่ากับ $\frac{1}{A_c}$ แต่จะต้องมีค่าที่ไม่ทำให้ $1 + k_a m(t)$ มีค่าเป็นลบ

รูปคลื่นของสัญญาณเอเอ็ม

จากรูปที่ 2.20 (a) จะเห็นว่ารูปของสัญญาณเอเอ็ม นั้นจะมีความถี่ตามคลื่นพาห์แต่ความสูงจะเปลี่ยนตามสัญญาณข้อมูล ซึ่งเมื่อสัญญาณข้อมูลมีความสูงมากที่สุดทางด้านบวก รูปสัญญาณเอเอ็มจะมีความถี่มากที่สุดและคลื่นสัญญาณเอเอ็มมีความสูงต่ำสุด เมื่อสัญญาณข้อมูลมีค่าลบสูงสุดจากรูปที่ 2.20 (b) สัญญาณข้อมูลมีความสูงเพิ่มขึ้นจะทำให้รูปเอเอ็มมีค่าสูงสุดมากกว่าเดิมและค่าต่ำสุดต่ำกว่าเดิม ส่วนในรูปที่ 2.20 (c) สัญญาณข้อมูลมีขนาดเล็กลงแต่ความถี่มากขึ้น จะทำให้รูปคลื่นเอเอ็ม มีค่าสูงสุดและต่ำสุดลดลง และอัตราการแกว่งขึ้นลงเพิ่มขึ้นเท่ากับความถี่ของสัญญาณข้อมูล

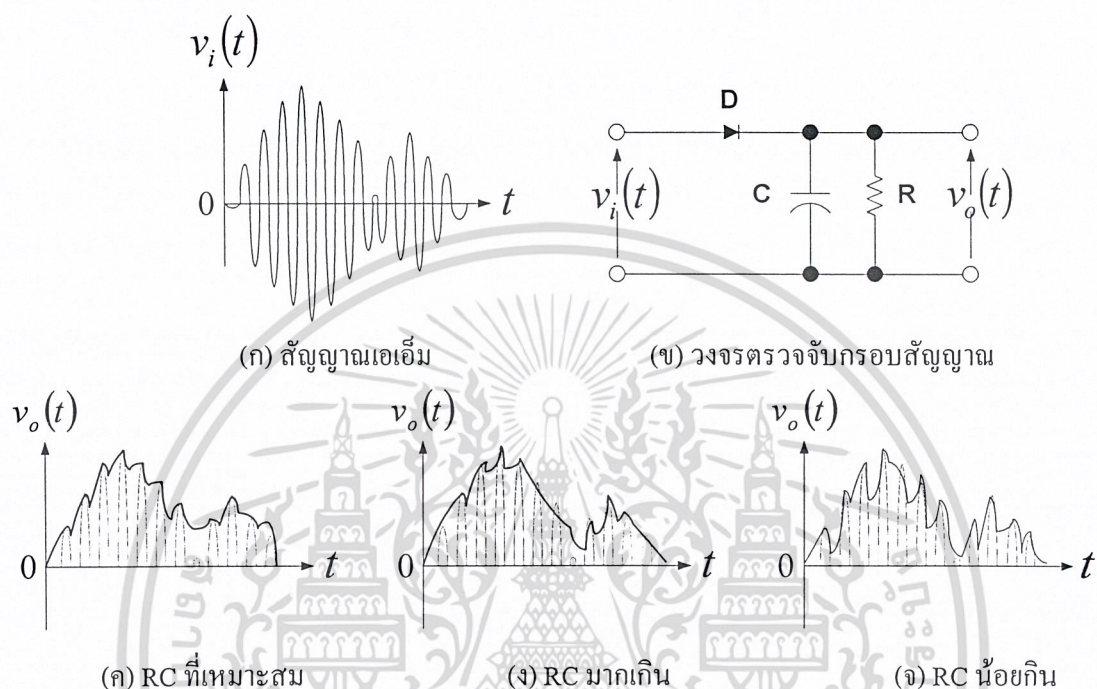


รูปที่ 2.20 รูปสัญญาณเอเอ็ม ที่เปลี่ยนแปลงตามสภาวะของสัญญาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 วงจรตรวจจับกรอบสัญญาณ

วงจรตรวจจับกรอบสัญญาณ คือวงจรที่ให้เอาต์พุตมีลักษณะเป็นไปตามกรอบหรือแนวทางเดินของยอดคลื่นของสัญญาณอินพุต วงจรตรวจจับกรอบสัญญาณแบบง่ายจะเป็นวงจรชนิดไม่เชิงเส้นดังแสดงในรูป



รูป 2.21 ประกอบการอธิบายการทำงานของวงจรตรวจจับกรอบสัญญาณ

การทำงานของวงจรนี้ เมื่อมีสัญญาณ $\phi_{AM}(t)$ เข้ามาสู่วงจรในตอนแรกนั้น ขณะที่สัญญาณมีค่าเพิ่มขึ้นในทางบวกไดโอด D จะนำกระแส ทำให้เกิดการประจุตัวเก็บประจุ C ให้มีแรงดันเอาต์พุตเกิดขึ้นคร่อม C ตามค่าการเพิ่มของสัญญาณอินพุต จนกระทั่งถึงค่าสูงสุด (Peak) หรือยอดของสัญญาณ แต่ต่อจากนี้เมื่ออินพุต $\phi_{AM}(t)$ ลดลงก็จะทำให้ค่าแรงดันไฟฟ้าทางเอาต์พุตของ D ต่ำกว่าคาโอด ในสถานการณ์เช่นนี้ไดโอด D จะหยุดนำกระแสและกั้นไม่ให้ C คายประจุผ่านตัวมันได้ ดังนั้นการคายประจุของ C จึงมีทางเดียวคือ คายประจุผ่าน R ไปอย่างช้า ๆ จนกระทั่งถึงเวลาที่ $\phi_{AM}(t)$ จะมีค่ากลับเป็นบวก และมีค่าเพิ่มขึ้นมากกว่าแรงดันไฟฟ้าคร่อม C ไดโอด D จึงตะนำกระแสอีก และจะเกิดการประจุของ C ทำให้เอาต์พุตเป็นไปตามค่าของสัญญาณ $\phi_{AM}(t)$ อีกไปจนถึงค่ายอดของคลื่นพาดถัดมา แล้วพฤติกรรมการทำงานจะเกิดซ้ำในลักษณะเดิมอีกกล่าวคือการคายประจุของ C ผ่าน R อีกครั้งหนึ่งพฤติกรรมเช่นนี้จะเกิดซ้ำแล้วซ้ำอีกเรื่อยไป ดังนั้นถ้าเราเลือกค่าคงตัวเชิงเวลา (Time constant) RC ที่ใช้ให้เหมาะสม ก็จะได้เอาต์พุตของวงจรเป็นไปตามเส้นทึบดังรูป ค แต่ถ้าค่าคงตัวเชิงเวลาของวงจรตรวจจับกรอบสัญญาณมีค่ามากเมื่อเทียบกับคาบเวลาที่สั้นที่สุดของสัญญาณที่ประกอบอยู่ใน $m(t)$ แล้วก็จะทำให้เอาต์พุตของวงจรเกิดลดตามยอดคลื่นของ $\phi_{AM}(t)$ ไม่ทันจะทำให้เกิดการผิดเพี้ยนของสัญญาณเอาต์พุตดังรูป และในกรณีที่เกิดการใช้ RC มีค่าน้อยเมื่อเทียบกับคาบเวลาของ

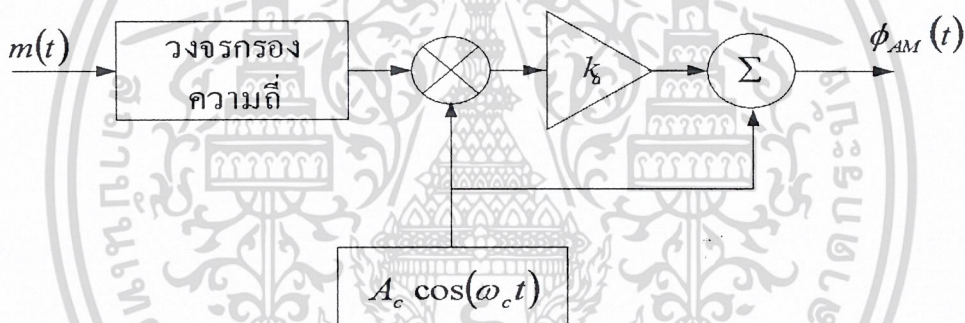
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทียบกับคาบเวลาของคลื่นพาห้ก็จะทำให้เกิดการกระเพื่อม (Ripple) บนสัญญาณเอาต์พุตมากดังรูปที่ 2.21 (จ) อย่างไรก็ตามค่าความถี่ของคลื่นพาห้ ω_c ที่ใช้ในทางปฏิบัตินั้นจะมีค่าสูงกว่าความถี่ของสัญญาณที่เป็นส่วนประกอบของ $m(t)$ มาก หรือกล่าวอีกนัยหนึ่งได้ว่า $m(t)$ จะเปลี่ยนแปลงช้ามากเมื่อเทียบกับคลื่นพาห้ที่ใช้จึงทำให้การเลือกค่าเวลาคงตัวของวงจร RC ทำได้อย่างเหมาะสมง่าย กล่าวคือจะทำให้การกระเพื่อมบนเอาต์พุตของวงจรมีค่าน้อยมากได้จนไม่ต้องคำนึงถึงในทางปฏิบัติ เงื่อนไขสำหรับการเลือกค่า RC ที่เหมาะสมสามารถสรุปได้ในรูปของอสมการดังนี้

$$\frac{1}{f_c} \ll RC \ll \frac{1}{f_m} \quad (2-15)$$

โดยในที่นี้ f_c และ f_m คือค่าความถี่ของคลื่นพาห้และค่าความถี่สูงสุดที่ประกอบอยู่ในสัญญาณข่าวสาร $m(t)$ ตามลำดับ

โดยอาศัยสมการที่ (2-13) เราสามารถสร้างสัญญาณเอเอ็มได้โดยระบบที่ง่ายดังรูป



รูปที่ 2.22 แผนผังระบบการสร้างสัญญาณเอเอ็มแบบง่าย

อย่างไรก็ดีเราสามารถที่จะสร้างสัญญาณเอเอ็ม โดยวิธีอื่น เช่น ใช้วงจรสวิตช์ หรือวงจรไม่เชิงเส้น ได้เช่นเดียวกับการสร้างสัญญาณดีเอสบี เอสซี

2.16 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก (D/A)

อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่แปลงสัญญาณข้อมูลดิจิตอลให้เป็นสัญญาณอนาล็อกย่านความถี่เสียง เราเรียกว่า โมเด็ม (MODEM หรือ Modulator Demodulator)

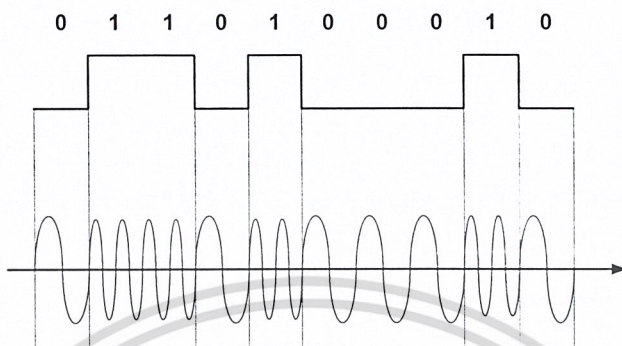
เทคนิคการแปลงสัญญาณข้อมูลดิจิตอลให้เป็นสัญญาณอนาล็อกนั้นมีอยู่ด้วยกัน 3 วิธีคือ

1. การมอดูเลตเชิงเลขทางแอมพลิจูด (Amplitude-Shift Keying หรือ ASK)
2. การมอดูเลตเชิงเลขทางความถี่ (Frequency-Shift Keying หรือ FSK)
3. การมอดูเลตเชิงเลขทางเฟส (Phase-Shift Keying หรือ PSK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมอดูเลตเชิงเลขทางความถี่ (FSK)

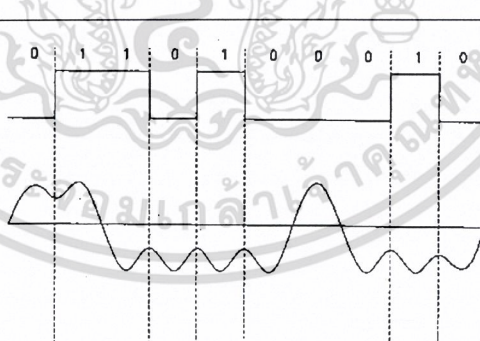
ในการมอดูเลตแบบ FSK ขนาดของคลื่นพาห์จะไม่เปลี่ยนแปลง ที่เปลี่ยนแปลงคือความถี่ของคลื่นพาห์ นั่นคือเมื่อบิตมีค่าเป็น “1” ความถี่ของคลื่นพาห์จะสูงกว่าปกติ และเมื่อบิตมีค่าเป็น “0” ความถี่ของคลื่นพาห์ก็จะต่ำกว่าปกติ



รูปที่ 2.23 การแปลงสัญญาณ D/A แบบ FSK

การมอดูเลตเชิงเลขทางเฟส (PSK)

หลักการของ PSK คือค่าของขนาดและความถี่ของคลื่นพาห์จะไม่มีการเปลี่ยนแปลง แต่ที่จะเปลี่ยนคือเฟสของสัญญาณ กล่าวคือเมื่อมีการเปลี่ยนแปลงสถานะของบิตจาก “1” ไปเป็น “0” หรือเปลี่ยนจาก “0” ไปเป็น “1” เฟสของคลื่นจะเปลี่ยน (Shift) ไป 180 องศาด้วย วิธี การ แบบ PSK จะมีสัญญาณรบกวนเกิดขึ้นน้อยที่สุด ได้สัญญาณที่มีคุณภาพดีที่สุด แต่วงจรการทำงานจะยุ่งยากกว่าและราคาสูงกว่า



รูปที่ 2.24 การแปลงสัญญาณ D/A แบบ PSK

การมอดูเลตเชิงเลขทางแอมพลิจูด (ASK)

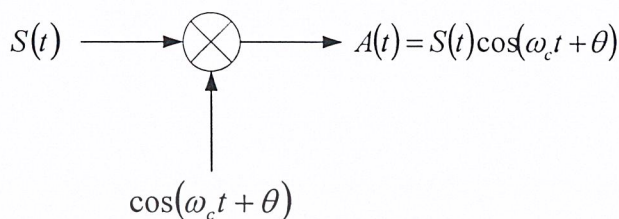
คลื่นที่ถูกมอดูเลตทางแอมพลิจูด ถูกทำให้แอมพลิจูดของคลื่นพาห์เปลี่ยนแปลงเป็นสัดส่วนกับสัญญาณข้อมูล ซึ่งทำได้โดยการคูณคลื่นพาห์ไซน์เข้ากับสัญญาณข้อมูล คลื่นที่ถูกมอดูเลตทางแอมพลิจูด $A(t)$ สามารถกำหนดโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A(t) = S(t) \times \cos(\omega_c t + \theta)$$

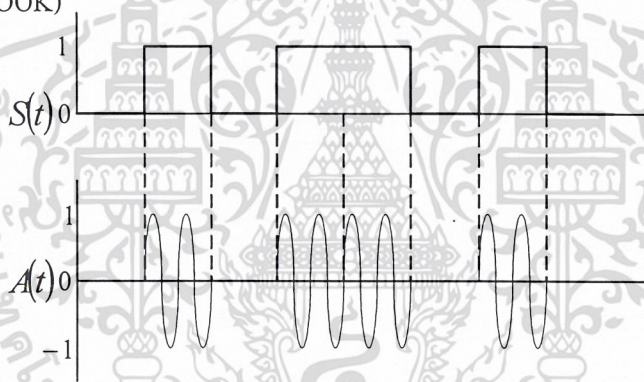
$$S(t) = \text{สัญญาณข้อมูล}$$

$$\cos(\omega_c t + \theta) = \text{คลื่นพาห้}$$



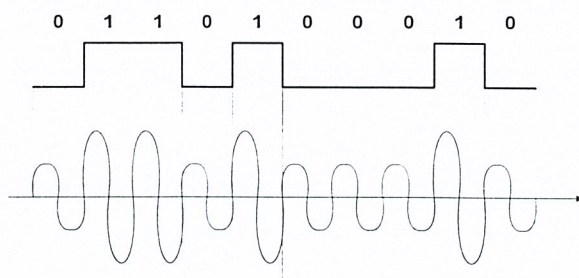
รูปที่ 2.25 รูปแบบของการมอดูเลตทางแอมพลิจูด

พิจารณาสัญญาณข้อมูล ซึ่งเป็นยูนิโพล่าพัลส์ เมื่อพัลส์เป็น 1 จะมีคลื่นพาห้ออกมา และเมื่อพัลส์เป็น 0 จะไม่มีคลื่นพาห้ออกมา การมอดูเลตทางแอมพลิจูดแบบนี้เรียกว่า ON-OFF ASK หรือ ON-OFF Keying (OOK)



รูปที่ 2.26 ON-OFF ASK

ความถี่ของคลื่นพาห้ (Carrier Wave) ซึ่งทำหน้าที่นำสัญญาณอนาล็อกผ่านตัวกลางสื่อสารนั้นจะคงที่ ลักษณะของสัญญาณมอดูเลตเมื่อค่าของบิตของสัญญาณข้อมูลดิจิทัลมีค่าเป็น "1" ขนาดของคลื่นพาห้จะสูงขึ้นกว่าปกติ และเมื่อบิตมีค่าเป็น "0" ขนาดของคลื่นพาห้จะตกลงกว่าปกติ การมอดูเลต ASK มักไม่ค่อยได้รับความนิยม เพราะจะถูกรบกวนจากสัญญาณอื่นได้ง่าย



รูปที่ 2.27 การแปลงสัญญาณ D/A แบบ ASK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

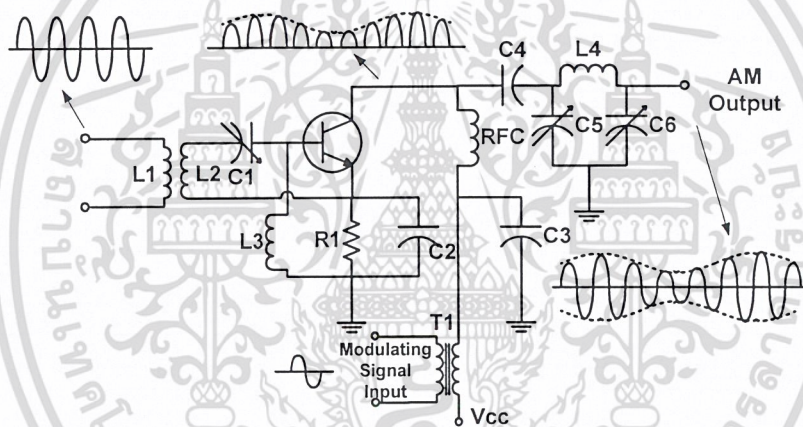
2.17 กำเนิดคลื่นเอเอ็ม

คลื่นเอเอ็ม เกิดจากการรวมคลื่นพาห์ กับคลื่นเสียงในอุปกรณ์ที่ไม่เป็นเชิงเส้นหรืออนลิเนียร์ เรียกว่า มอดูเลเตอร์ วงจรมอดูเลตมีหลายแบบเช่น ไดโอดมอดูเลเตอร์ โดยใช้การทำงานช่วงที่ไดโอดไม่เป็นลิเนียร์ แต่ไม่นิยมเพราะไดโอดไม่มีอัตราขยาย ทรานซิสเตอร์ทำงานในช่วงอนลิเนียร์ เมื่อจัดไบแอส ที่เหมาะสมให้จะเป็นวงจรขยายที่สามารถทำเป็นวงจรมอดูเลเตอร์ได้ดี

วงจรทรานซิสเตอร์มอดูเลเตอร์ ทำหน้าที่ขยายสัญญาณคลื่นพาห์ สามารถป้อนสัญญาณเสียงผ่านขาใดขาใน 3 ขา เป็นการมอดูเลตแบบต่าง ๆ คือ คอลเลคเตอร์มอดูเลชัน เบสมอดูเลชัน และอิมิเตอร์มอดูเลชัน

คอลเลคเตอร์มอดูเลชัน

วงจรที่นิยมใช้มาก คือ คอลเลคเตอร์มอดูเลเตอร์ ดังรูป สัญญาณเสียงจะต่ออนุกรมเข้ากับไฟดีซี ซึ่งเป็นไฟเลี้ยงของวงจรทรานซิสเตอร์ วงจรชนิดนี้เป็นวงจรขยายความถี่อาร์เอฟ คลาสซี (RF Class C Amplifier) และมักจะใช้ในวงจรไฮ-เลเวล มอดูเลชัน

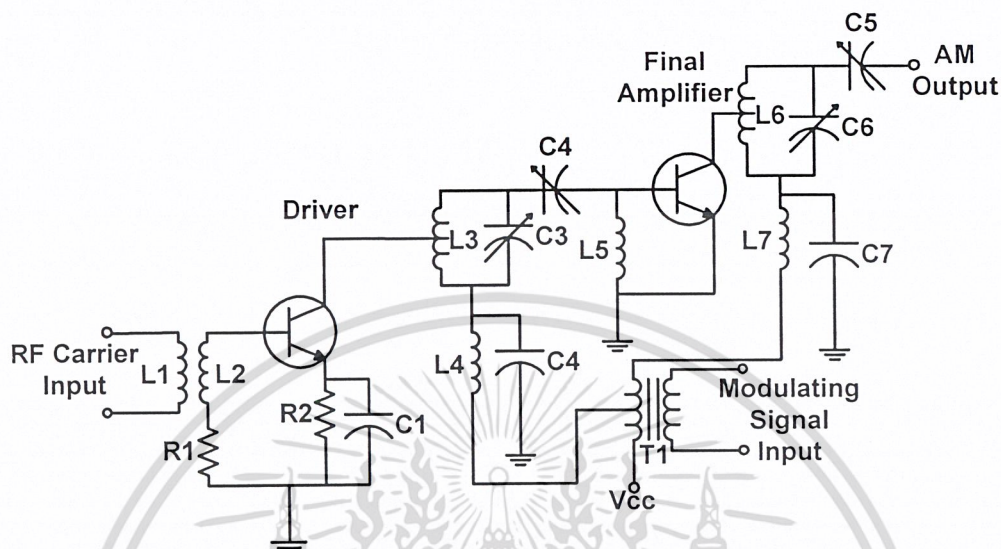


รูปที่ 2.28 วงจรคอลเลคเตอร์มอดูเลชัน

สัญญาณเสียงต่อผ่านทรานส์ฟอร์มเมอร์ เหนี่ยวนำสัญญาณ ไปรวมกับไฟเลี้ยง V_{cc} ความถี่อาร์เอฟ คลื่นพาห์ ต่อผ่านขาเบสของวงจรขยาย Q1 จากการเปลี่ยนแปลงคอลเลคเตอร์โวลเทจ ทำให้เอาต์พุตแอมพลิจูด มีการเปลี่ยนแปลงตามอัตราการเปลี่ยนแปลงของสัญญาณเสียง สัญญาณเอเอ็ม ที่เอาต์พุต มีลักษณะเพิ่มขึ้นและลดลงเมื่อเป็นเฟสเดียวกันและต่างเฟสกับไฟ V_{cc} หรือไฟ DC นั้นเอง ถ้าเฟสเดียวกัน ศักดาที่คอลเลคเตอร์เป็นบวกเพิ่มขึ้น ทำให้กระแสที่คอลเลคเตอร์สูงขึ้นเป็นผลทำให้เอาต์พุตแรงขึ้น เป็นช่วงยอดคลื่นของแอนเวลโลปเอ-เอ็ม ในทางตรงข้าม ถ้าสัญญาณต่างเฟสกับ V_{cc} ศักดาที่คอลเลคเตอร์จะเป็นบวกลดลง ทำให้กระแสและเอาต์พุตโวลเทจลดลง เป็นช่วงท้องคลื่นของแอนเวลโลปเอเอ็ม เนื่องจาก Q1 เป็นวงจรขยายคลาสิกซ์ ซึ่งมีกระแสเอาต์พุตเป็นพัลส์บวกเท่านั้น การทำให้ได้คลื่นเอเอ็ม ที่มีทั้งซีกบวกและซีกลบ จึงต้องผ่านวงจรรีโซแนนซ์ ซึ่งประกอบด้วย L4, C5 และ C6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

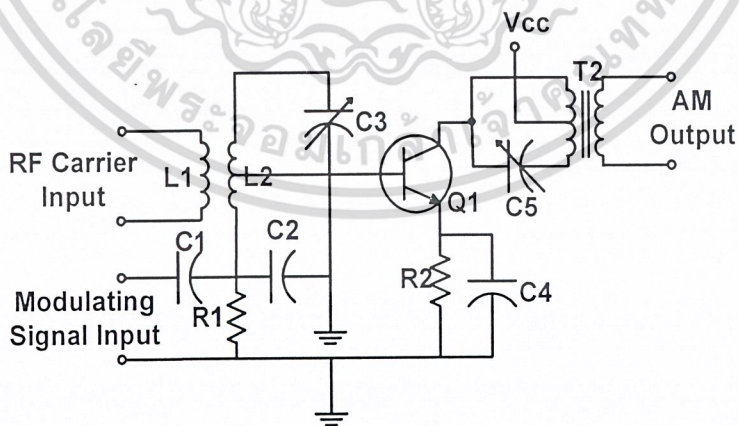
แต่วจรคอลเลกเตอร์มอดูเลชันไม่สามารถมอดูเลตได้ถึง 100% เพราะการอิ่มตัวของทรานซิสเตอร์ทำให้เอาต์พุตโวลเทจเพิ่มขึ้นไม่ถึงค่าสูงสุด ในกรณีใช้เป็นไฮ-เลเวลมอดูเลต แก้ปัญหานี้ด้วยการมอดูเลตทั้งภาคไดโอดและภาคขยายอาร์เอฟ ดังรูป



รูปที่ 2.29 วงจรคอลเลกเตอร์มอดูเลชัน สำหรับไฮเลเวลมอดูเลต

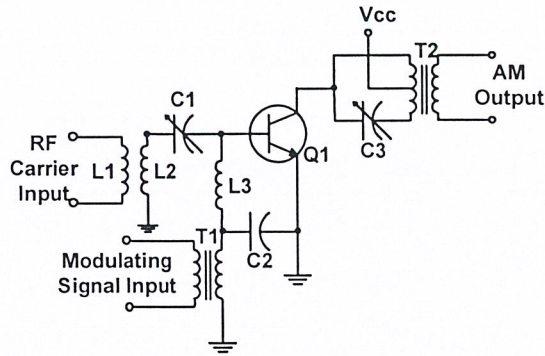
เบสมอดูเลชัน

วงจรเบสมอดูเลชัน สัญญาณเสียงต่อเข้าวงจรขยายอาร์เอฟ ผ่านทางขาเบสของทรานซิสเตอร์ กำลังงานของเสียงหรือสัญญาณมอดูเลตต่ำกว่า ที่ใช้กับวงจรคอลเลกเตอร์มอดูเลต สัญญาณเสียงสามารถต่อแบบอนุกรมและต่อแบบขนานดังรูป



รูปที่ 2.30 วงจรเบสมอดูเลชันแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 วงจรเบสมอดูละชันแบบอนุกรมแบบขนาน

การต่อแต่ละแบบจะเชื่อมโยงสัญญาณด้วย RC (RC coupling) หรือทรานสฟอร์มเมอร์ (Transformer Coupling) ก็ได้ แต่ทรานสฟอร์มเมอร์จะให้ค่าอิมพีแดนซ์ แมชชิงดีกว่า

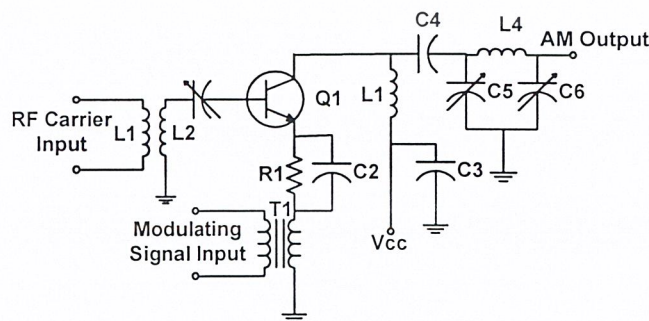
จากรูปทรานซิสเตอร์ทำงานเมื่อเบส อิมิตเตอร์โวลเทจ มีค่าสูงกว่า 0.6 โวลต์ และทรานซิสเตอร์ จะคัทออฟ หรือไม่ ทำงานในช่วงสัญญาณซีกลบ ต้องมีวงจรรีโซแนนซ์ มาต่อที่เอาต์พุตของ คอลเลคเตอร์ ของ Q1 ประกอบด้วย C5 และทรานสฟอร์มเมอร์ T1 เป็นวงจรรีโซแนนซ์ที่มีค่า Q ต่ำ เพื่อ เสริมสัญญาณครึ่งไซเคิลที่หายไปให้กลับคืนมา พัลส์ของกระแสแต่ละลูกจะทำให้วงจรแก๊งค์ออสซิลเลต ค่า Q ต่ำของวงจรมีผลให้ แอมพลิจูด ของสัญญาณครึ่งไซเคิลหลังยังคงอยู่และเท่ากับแอมพลิจูดครึ่ง ไซเคิลเดิม จึงได้เอาต์พุตเป็นคลื่นเอ-เอ็มสมบูรณ์ที่ขดทุติยภูมิของทรานสฟอร์มเมอร์ T1

เปรียบเทียบกับคอลเลคเตอร์มอดูละชัน เบสมอดูละชัน

1. ปรับแต่งวงจรยากกว่า
2. ประสิทธิภาพของคอลเลคเตอร์ต่ำกว่า
3. ถ้าใช้ทรานซิสเตอร์แบบเดียวกัน จะมีกำลังงานเอาต์พุตต่ำกว่า และ
4. มีความเป็นเชิงเส้นน้อยกว่า

อิมิตเตอร์มอดูละชัน

เป็นการต่อสัญญาณมอดูละชันเข้าที่ขาอิมิตเตอร์ของทรานซิสเตอร์ให้คุณสมบัติอยู่ระหว่างเบสและ คอลเลคเตอร์มอดูละชัน ดังรูป



รูปที่ 2.32 วงจรอิมิตเตอร์มอดูละชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นพาห้ต่อผ่าน C1 ไปยังขาเบสของทรานซิสเตอร์ ส่วนสัญญาณมอดูเลต หรือสัญญาณเสียง ต่อผ่านทรานสปอร์เมอร์ T1 ไปยังอิมิตเตอร์ ซึ่งช่วยให้อิมพิแดนซ์แมทชิงที่ดี R1, C2 เป็นวงจรอิมิตเตอร์ไบแอสมองผ่านคลื่นพาห้ วงจรมีลักษณะเป็นวงจรขยายชนิดคอมมอนอิมิตเตอร์ อัตราขยายของวงจรขึ้นอยู่กับสัญญาณเสียงที่ป้อนให้กับขาอิมิตเตอร์มอดูเลเตอร์ชนิดนี้ ขนาดสัญญาณเสียงมักจะสูงกว่า ขนาดของคลื่นพาห้ด้วยเหตุนี้ วงจรอิมิตเตอร์มอดูเลชัน จึงเหมาะกับการมอดูเลตระดับต่ำ (low level modulation)

2.18 การกู้สัญญาณเอเอ็ม (AM Detection)

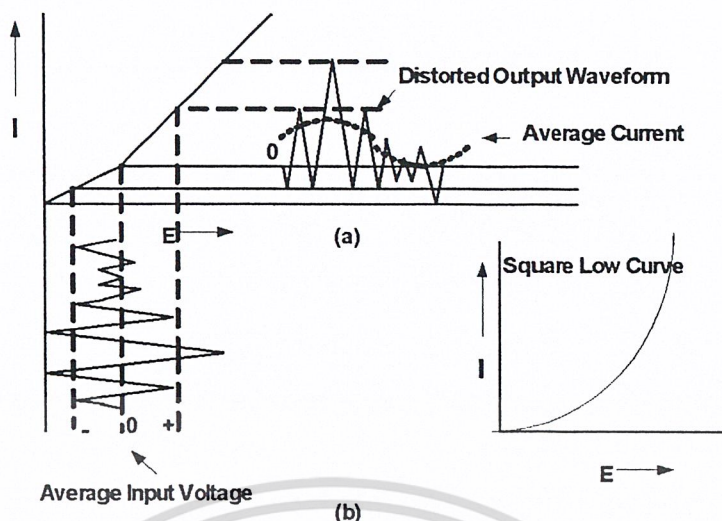
ในกระบวนการกู้สัญญาณข่าวสารออกจากคลื่นพาห้และไซด์แบนด์ สามารถทำได้โดยง่ายโดยจะย้อนกลับไปถึงการอธิบายเกี่ยวกับการกำเนิดสัญญาณ AM เรากล่าวว่าความถี่ที่แตกต่างกัน 2 ความถี่ จะผ่านอุปกรณ์ nonlinear เกิดการรวมและหักล้างกันของความถี่ระหว่างคลื่นพาห้และ Side band ของสัญญาณ AM ถูกแยกความถี่ให้มีค่าตามความถี่ของข่าวสาร (intelligence frequency) ถ้าสัญญาณ AM ผ่านไปยังอุปกรณ์ nonlinear ความต่างของความถี่ระหว่างคลื่นพาห้และ Side band จะเกิดขึ้นและความถี่ที่เกิดขึ้นนี้คือ ข่าวสารที่ส่งมานั้นเองถ้าเรานำสัญญาณ AM ไปผ่านอุปกรณ์ nonlinear จะเกิดกระบวนการกู้สัญญาณขึ้น

การกู้สัญญาณ เอเอ็มต้องการอุปกรณ์ไฟฟ้าแบบ Nonlinear ในทางอุดมคติ curve ของสัญญาณ อุปกรณ์ linear จะมีผลให้ครึ่งบวกของคลื่นมอดูเลตมากกว่าที่จะให้ครึ่งลบของสัญญาณ ซึ่ง distorts ที่เกิดใน voltage wave จะอยู่ที่ระดับประมาณศูนย์ ดังนั้นการเปลี่ยนแปลงของกระแสก็จะมีผลกับแอมพลิจูดของสัญญาณข่าวสาร โดย curve เรียกว่า ideal curve เพราะว่าเส้นตรงบริเวณจุด P จะทำให้เกิดความถี่ฮาร์โมนิกได้

เมื่อมีอินพุตมายังอุปกรณ์ Nonlinear แบบอุดมคติพบว่าเอาท์พุตจะได้ความถี่ดังนี้

1. ความถี่คลื่นพาห้
2. upper side band
3. lower side band
4. กระแส dc
5. ความถี่ที่มีค่าเท่ากับคลื่นพาห้ต่ำกว่า side band เล็กน้อยและสูงกว่า side band เล็กน้อย ก็คือสัญญาณความถี่เดิม

การสร้างสัญญาณใหม่ของตัวกู้สัญญาณความถี่โดยการผลิตเอาท์พุต ที่ต้องการเอาท์พุตที่ได้จะผ่าน Low pass filter ที่ความถี่สูงจะถูกขจัดออกไป และจะมีเพียงสัญญาณข่าวสาร และองค์ประกอบของไฟ dc เท่านั้นที่ผ่านออกมาได้ ดังแสดงในการเกิด curve ของกระแสเฉลี่ยดังรูป



รูปที่ 2.33 อุปกรณ์ที่ไม่เป็นเชิงเส้นที่ถูกใช้เป็นเครื่องรับ (Nonlinear device used as detector)

วงจรของเครื่องรับในทางปฏิบัติ Curve จะเป็นในทางอุดมคติคือเป็น curve แบบ square law parabola เอาที่พูดที่ออกมาจากอุปกรณ์ จะอาศัยจาก curve นี้ ฮาร์โมนิกของความถี่สูงสามารถกรองออกไปได้ แต่ฮาร์โมนิกของผลรวมและผลต่างของความถี่จะทำให้เกิด distortion ขึ้น เครื่องรับแบบไดโอด (Diode Detector)

ตัวดีเทคเตอร์ ชนิดที่มีประสิทธิภาพสูงและนิยมใช้กันมากมีคุณสมบัติใกล้เคียงกับความต้านทานแบบชนิดไม่เป็นเชิงเส้น (Nonlinear) คือวงจรดีเทคเตอร์แบบ Diode ดังแสดงดังรูปที่ 2.33 (a) ให้ลองสังเกตกราฟ I-V curve ในรูปที่ 2.33 (a) ที่ Diode detector สามารถทำงานได้ ส่วนของเส้นโค้งที่มีการตอบสนองคือ บริเวณที่มี current ต่ำๆ และนั่นแสดงว่าสัญญาณเอาต์พุตเล็กๆ ที่ออกมาจากดีเทคเตอร์จะเป็นไปตามกฎของ square law สำหรับสัญญาณอินพุตที่มีขนาดแอมพลิจูดขนาดใหญ่ เอาต์พุตที่ผ่านออกมาจำเป็นต้องเป็นแบบ linear ดังนั้น ฮาร์โมนิก เอาต์พุตก็ถูกจำกัด การเปลี่ยนแปลงแบบ nonlinear ที่เกิดแบบทันทีทันใดจะเกิดขึ้นสำหรับครึ่งลบดังแสดงในรูปที่ 2.33 (b)

คลื่นพาห์ที่มอดูเลตแล้วจะถูกเหนี่ยวนำให้เกิดในวงจร Tune ทำให้เกิดโดยใช้ LC1 ในรูปที่ 2.33 (a) รูปร่างของคลื่นอินพุตที่เข้าไปยัง Diode แสดงดังรูปที่ 2.33 (c) เนื่องจากว่า diode จะทำงานเพียงครึ่ง cycle เท่านั้น ในที่นี้วงจรนี้จะขจัดครึ่งลบของ cycle ออกไปทั้งหมดและจะให้ผลดังแสดงในรูปที่ 2.33 (d) ค่าเอาต์พุตโดยเฉลี่ยแสดงดังรูปที่ 2.33 (e) ถึงแม้ว่าสัญญาณเฉลี่ยของอินพุตมีค่าเท่ากับศูนย์ แต่สัญญาณเฉลี่ยของเอาต์พุตที่ตกคร่อม R จะมีค่ามากกว่าศูนย์เสมอ

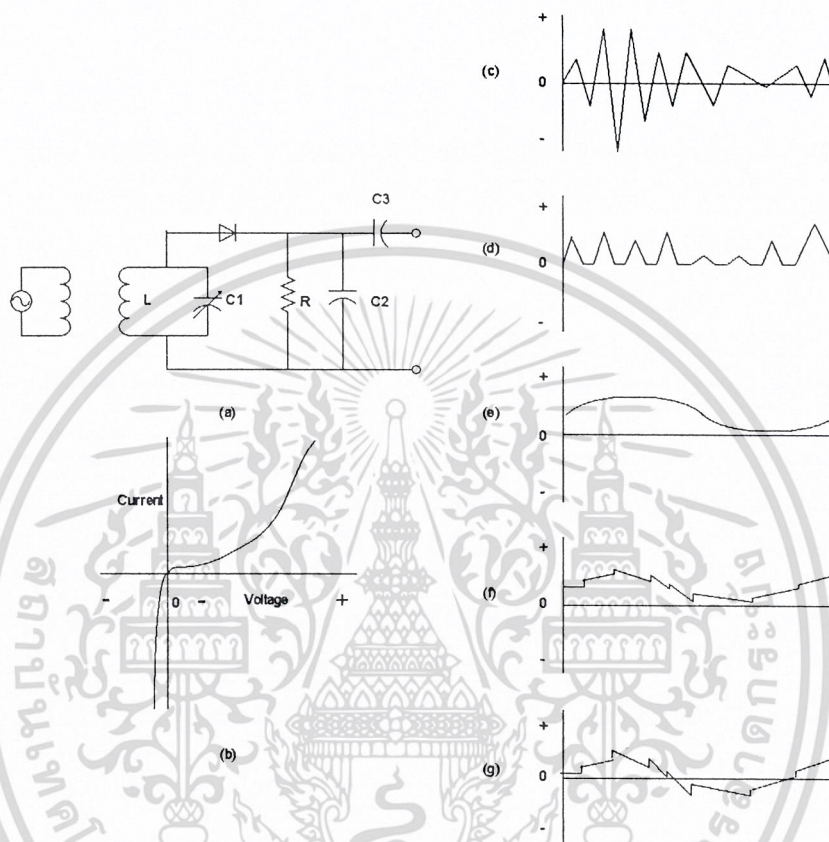
Low pass filter ที่ประกอบด้วยตัวเก็บประจุ $C2$ และตัวต้านทาง R จะขจัดความถี่คลื่นพาห์ (RF) ออกไป ตัวเก็บประจุ $C2$ ก็จะทำการ charge ประจุไปจนสุด peak voltage อย่างรวดเร็วให้ตกคร่อมค่าความต้านทานเล็กๆ ภายใน diode แต่จะ discharge อย่างช้าๆ ผ่านความต้านทาน สูงของ R

ขนาดของ R และ $C2$ โดยปรกติจะกำหนดจาก short time constant ที่ความถี่ของสัญญาณข่าวสาร (Audio frequency) และมี long time constant เมื่อเทียบกับความถี่คลื่นพาห์ (RF) ผลที่ออกมาจาก $C2$ ในวงจรคือการเปลี่ยนแปลงตามการเปลี่ยนแปลง peak ของการมอดูเลต ของคลื่นพาห์ พิจารณารูปที่ 2.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (f) จากเหตุผลดังกล่าวนี้จึงเรียกววงจรแบบนี้ว่า envelope detector กระแส dc จะถูกผลิตขึ้น โดยวงจรดีเท็คเตอร์ และถูกนำออกโดยตัวเก็บประจุ C3 ซึ่งเมื่อผ่าน C3 ไปแล้วจะได้สัญญาณคลื่นดังแสดงในรูปที่ 2.33
- (g) เครื่องรับในการสื่อสารจะใช้องค์ประกอบของไฟ dc สำหรับผลิต Automatic volume (gain) control (AGC)



รูปที่ 2.34 เครื่องรับแบบไดโอด (Diode detector)

ข้อดีของตัว ดีเท็คเตอร์แบบไดโอด มีดังนี้

1. สามารถใช้เป็นเครื่องมือในการรับสัญญาณที่มี power สูงๆ ซึ่งตัวดีเท็คเตอร์แบบ diode จะไม่มีข้อจำกัดของแอมพลิจูดของสัญญาณอินพุต
2. ระดับของการเกิด distortion อยู่ในระดับที่ยอมรับได้สำหรับการส่งในหลายๆระบบแบบ AM ซึ่ง distortion จะลดลงเมื่อแอมพลิจูดของอินพุตเพิ่มขึ้น
3. ประสิทธิภาพสูงเมื่อมีการออกแบบอย่างเหมาะสมซึ่งสามารถมีประสิทธิภาพสูง ถึง 90%
4. สามารถพัฒนาให้ใช้ได้กับไฟ dc สำหรับวงจร AGC (automatic gain control)
5. พลังงานจะถูก absorbed ไปกับวงจร tune โดยตัว diode ที่อยู่ในวงจรซึ่งผลที่เกิดขึ้นนี้จะไปลดค่าและ Selectivity ของวงจรจูน ภายใน
6. จะไม่มีการขยายเกิดขึ้นในวงจร ไดโอดดีเท็คเตอร์

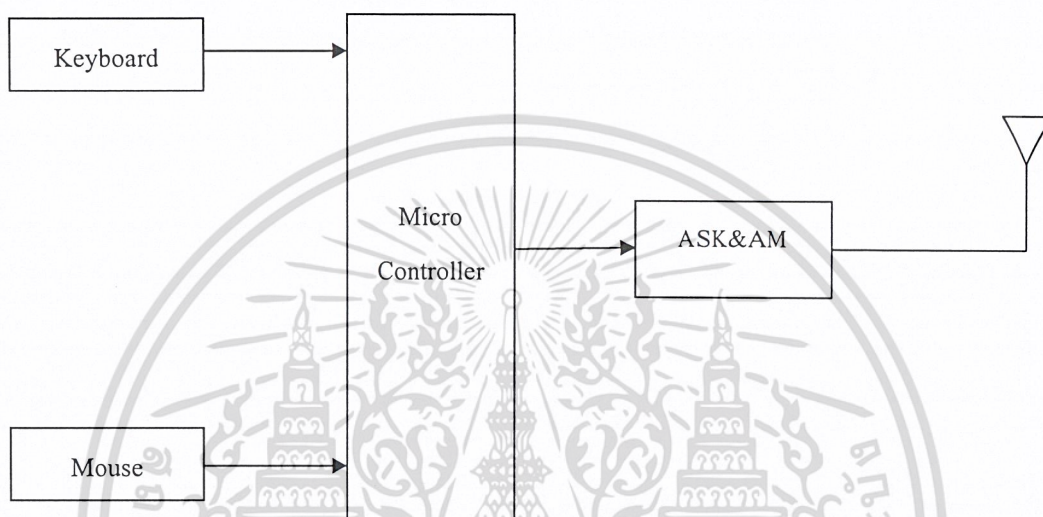
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

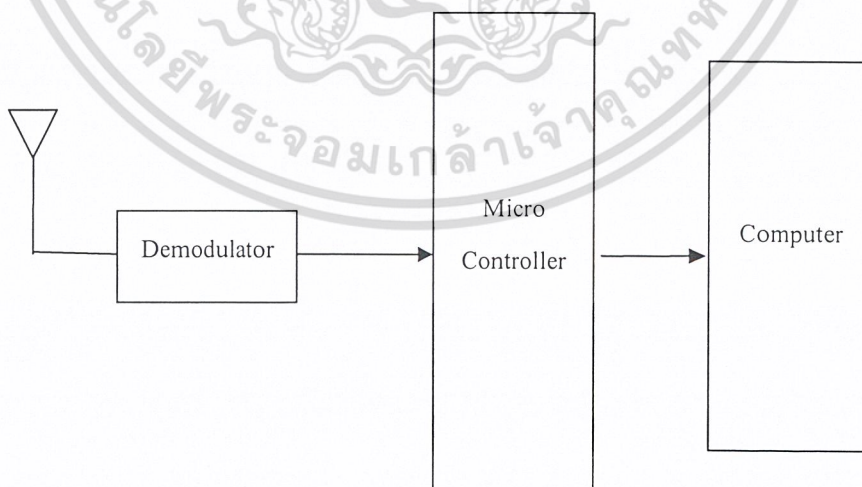
การคำนวณ และ การสร้าง

ภาพรวมของระบบ

ประกอบด้วยบล็อกไดอะแกรม 2 ด้านดังนี้



รูปที่ 3.1 บล็อกไดอะแกรมด้านเม้าส์ และคีย์บอร์ด



รูปที่ 3.2 บล็อกไดอะแกรมด้านคอมพิวเตอร์

โดยระบบจะแบ่งออกเป็น 2 ด้าน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ด้านเมาส์ และคีย์บอร์ด เชื่อมต่อกับคอนโทรลเลอร์

2. ด้านคอมพิวเตอร์เชื่อมต่อกับคอนโทรลเลอร์

โดยจะมอดูเลตด้วยคลื่นพาห์ที่ 40.680MHz

ไมโครคอนโทรลเลอร์ด้านคีย์บอร์ดและเมาส์จะมามีการทำงานดังนี้

เมื่อเมาส์ และคีย์บอร์ด ต้องการส่งข้อมูล มีการทำงานดังนี้

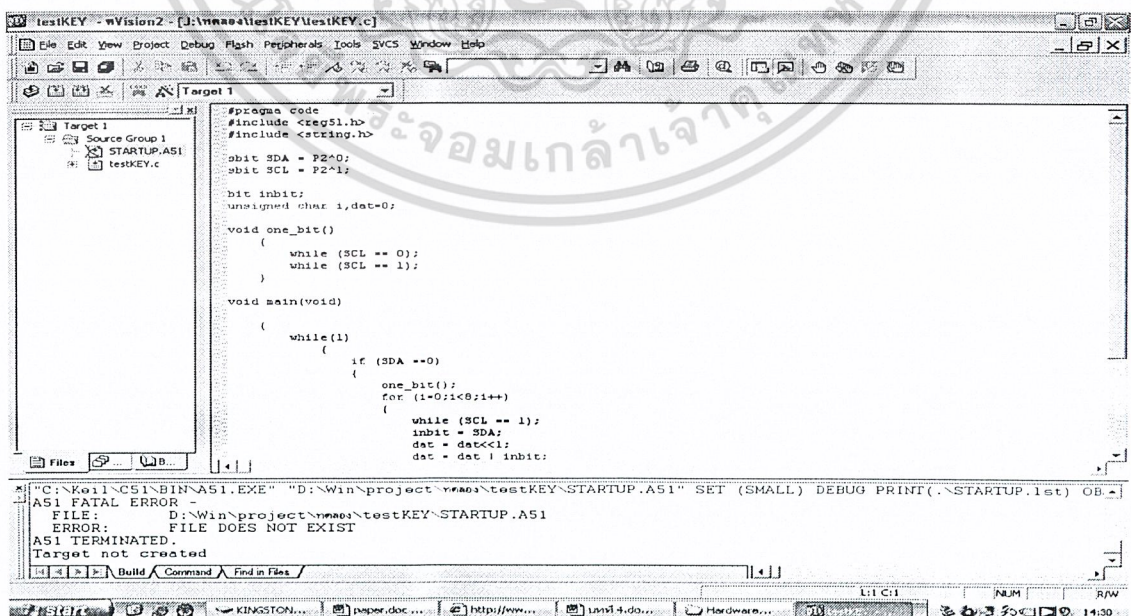
เมาส์และคีย์บอร์ดจะส่งข้อมูลมาเก็บและประมวลผลที่คอนโทรลเลอร์ แล้วจัดการกับข้อมูลในรูปของโปรโตคอลเพื่อส่งข้อมูลต่อไปยังภาคส่ง ASK เพื่อส่งสัญญาณไปทางฝั่งคอมพิวเตอร์โดยเมื่อต้องการจำลองให้ไมโครคอนโทรลเลอร์ให้เสมือนเป็นคอมพิวเตอร์เพื่อส่งข้อมูลสั่งให้เมาส์และคีย์บอร์ดเริ่มทำงานได้โดยวิธีการนี้เรียกว่าการทำแฮนด์เช็คคิง (Hand-checking)

ไมโครคอนโทรลเลอร์ด้านคอมพิวเตอร์จะมามีการทำงานดังนี้

เมื่อภาครับหรือภาคีเท็คสัญญาณ จะทำการตีเท็คสัญญาณ ASK ที่ส่งมาแปลงเป็นจุดข้อมูลแล้วส่งเข้าไปทางคอนโทรลเลอร์เพื่อทำการประมวลผลข้อมูลและแปลงให้เหมาะสมจึงทำการส่งข้อมูลไปยังเพื่อใช้ในการควบคุมเมาส์และคีย์บอร์ด โดยในด้านนี้จะทำการแฮนด์เช็คคิงเพื่อใช้ติดต่อกับคอมพิวเตอร์เสมือนว่ามีมการต่อเมาส์และคีย์บอร์ดอยู่

3.1 รูปแบบของโปรแกรมที่ใช้ในระบบ

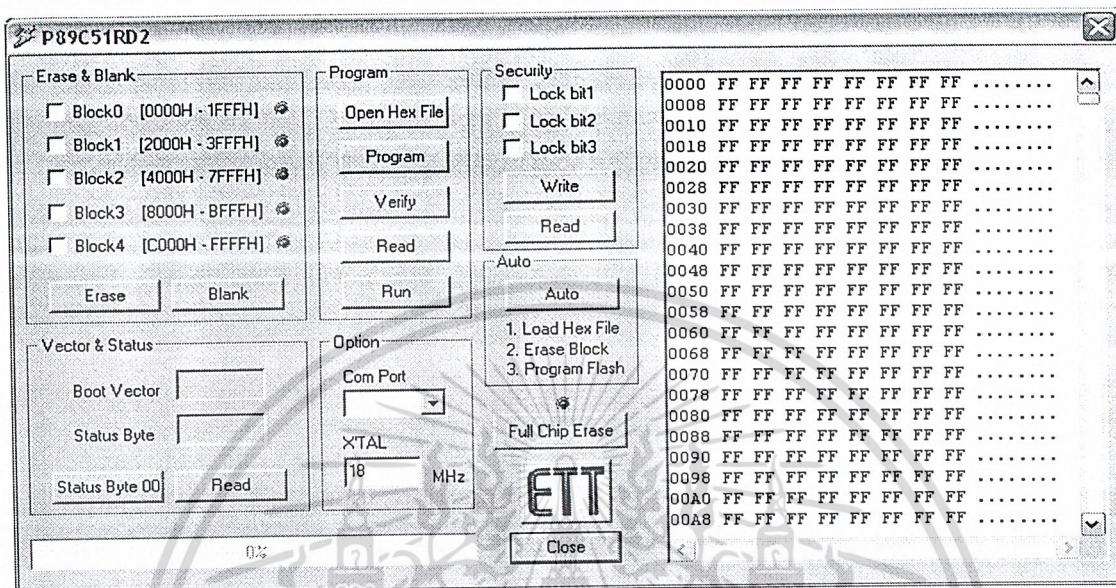
ในการเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์เพื่อใช้ในการจัดการข้อมูล เราจะใช้ภาษาซี สำหรับโปรแกรมที่จะใช้งานนั้นเราเลือกใช้โปรแกรม Keil uVision การคอมไพล์ภาษาซีให้เป็นข้อมูลในรูปแบบ HEX ไฟล์ เนื่องจากการโปรแกรมบนไมโครคอนโทรลเลอร์จำเป็นต้องใช้ข้อมูลที่อยู่ในรูป HEX



รูปที่ 3.3 แสดงโปรแกรม Keil uVision

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากผ่านการคอมไพล์แล้วจะมีไฟล์จุด Hex ออกมาให้ นำไฟล์นั้นไปโปรแกรมลงบน ไมโครคอนโทรลเลอร์โดยผ่านโปรแกรม P89C51RD2.EXE



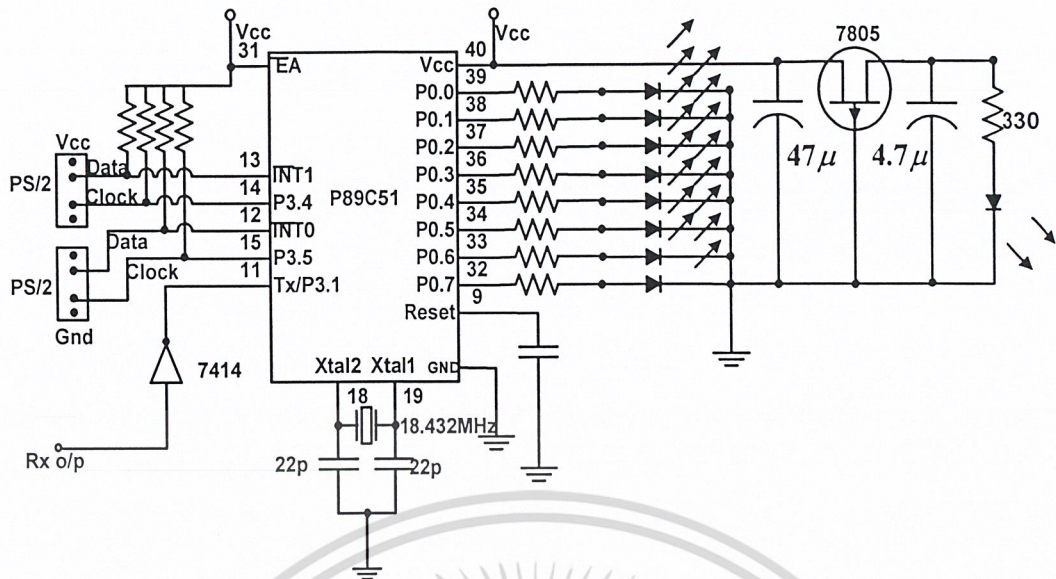
รูปที่ 3.4 โปรแกรม P89C51RD2

ในส่วนของการทำงานของโปรแกรมนั้น โดยจากรูปที่ 3.1 ในเวลาปกติโปรแกรมจะวนรอรับข้อมูลจากคีย์บอร์ดและเมาส์ การส่งข้อมูลนั้นจะนำข้อมูลที่ได้จากคีย์บอร์ดและเมาส์ผ่านวงจร ASK & Amplitude Modulator โดยใช้ความถี่ 40.680 MHz ซึ่งเป็นความถี่เดียวกันกับที่ใช้ในรถกระป๋อง โดยที่จะใช้ความถี่เดียวในการส่งข้อมูล

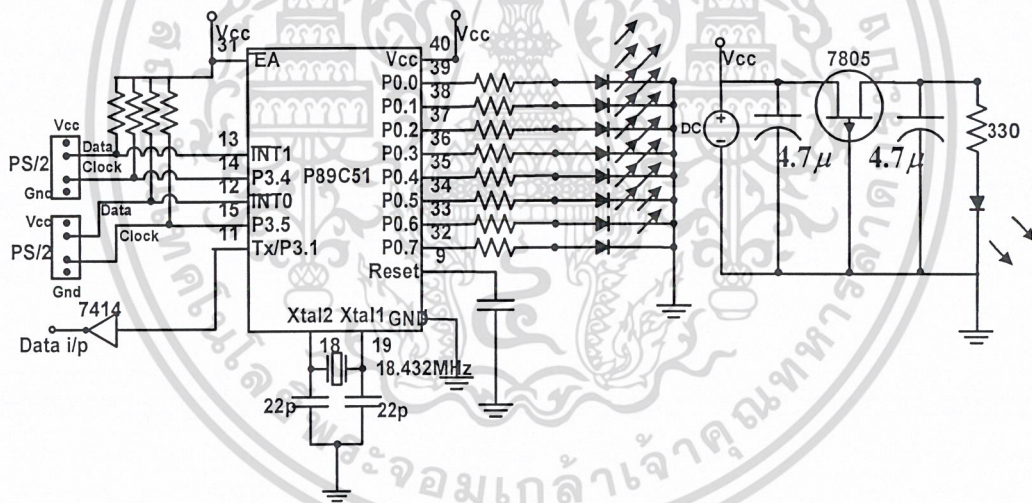
จากรูปที่ 3.2 โปรแกรมจะวนรอรับข้อมูลจากเมาส์และคีย์บอร์ดที่ส่งมาจากพอร์ตอนุกรม และส่งไปยังคอมพิวเตอร์ ถ้ามีการส่งข้อมูลจากคอมพิวเตอร์เข้ามาจากพอร์ตอินเทอร์รัพท์จะทำการหยุดการส่งข้อมูลและใช้ไมโครคอนโทรลเลอร์ได้ต่อกับคอมพิวเตอร์แทน ไมโครคอนโทรลเลอร์ที่เราใช้คือ ชิปชื่อ Philips เบอร์ P89C51RD2HBP ซึ่งมาพร้อมกับบอร์ดไมโครคอนโทรลเลอร์ CP-SPI/S8252 Version 1

3.2 การเชื่อมต่อฮาร์ดแวร์ของระบบ

การเชื่อมต่อก่อนถึงภาคเครื่องรับเครื่องส่ง เราใช้ไมโครคอนโทรลเลอร์เป็นตัวเชื่อมต่อข้อมูลจากคอมพิวเตอร์กับภาครับ-ส่งASKดังรูป



รูปที่ 3.5 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับโฮสต์ และอีกด้านหนึ่งเราใช้ไมโครคอนโทรลเลอร์เป็นตัวเชื่อมต่อข้อมูลจากอุปกรณ์อินพุตกับภาครับส่งASKดังรูป



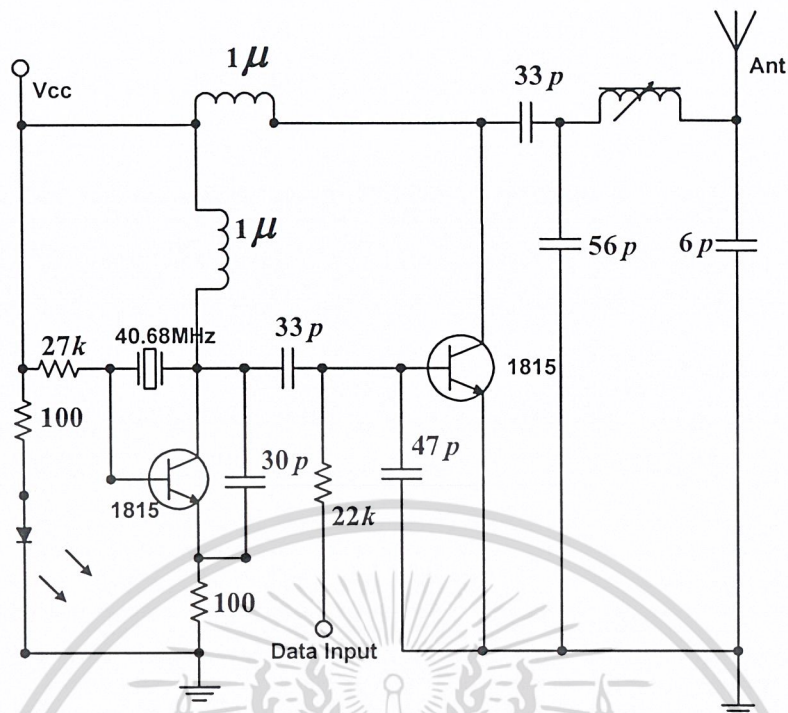
รูปที่ 3.6 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์

ในส่วนของภาครับ-ส่งASK

วงจรภาคส่ง

ในภาคส่งนี้เรานำข้อมูลที่ได้จากคอนโทรลเลอร์ซึ่งอยู่ในรูปแพ็คเกจมาใช้ในการส่งข้อมูล ซึ่งวงจรที่เราใช้ส่งนี้จะประกอบด้วยส่วนต่างๆดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรภาคส่งความถี่ 40.680MHz

หลักการทำงานดังนี้

สัญญาณที่มาจากคอนโทรลเลอร์ จะเข้ามาที่ขา Data Input แล้วจะมาทำการมอดูเลตกับสัญญาณ Carrier ความถี่ 40.680MHz ข้อมูลจะมาทำการมอดูเลตกันโดยทรานซิสเตอร์ตัวซ้ายจะสร้างสัญญาณคล็อก มาให้สัมพันธ์กับ อินพุตของข้อมูล (Data Input) ที่ส่งเข้ามา แล้วทรานซิสเตอร์ตัวขวาก็จะเหมือนสวิตซ์ที่ให้ข้อมูลที่ถูกรมอดูเลตแบบเอเอสเค (ASK) ถูกส่งไปที่เสาอากาศส่งออกอากาศต่อไป โดยรูปแบบการมอดูเลตจะเป็นแบบแอมพลิจูดมอดูเลชัน (AM) โดยใช้วงจรดังรูปในการส่ง

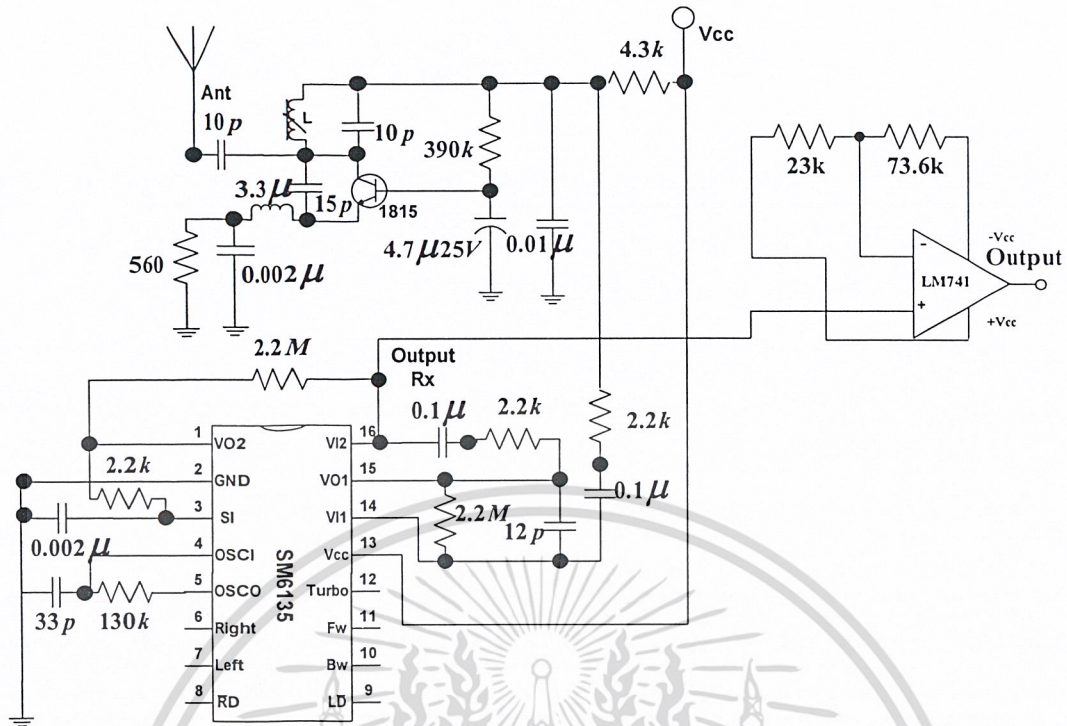
วงจรถอดรับ

โดยทางด้านรับเราต้องทำการดีมอดูเลตสัญญาณที่เป็นเอเอ็ม โดยรับข้อมูลเข้ามาในรูปแบบของ เอเอ็มแล้ว นำวงจรดังรูปมาใช้ในการดีเทก (Detect) ข้อมูลออกมา ก่อนที่จะนำข้อมูลไปใช้ในส่วนต่อไป

หลักการทำงานเป็นดังนี้

รับข้อมูลที่ที่เป็น AM มาทาง Antenna แล้วมาเข้าวงจรจูน (Tune) ก่อน เพื่อนำมาทำการดีเทกแบบ เอนVELOP ดีเทกเตอร์ (envelop detector) หรือ วงจรตรวจจับกรอบสัญญาณ ก่อนเข้าไป ที่ไอซี และ o/p จะออกมาที่ขา 16 ของไอซีแล้วนำสัญญาณที่ออกจากขา 16 ไปต่อกับวงจรคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

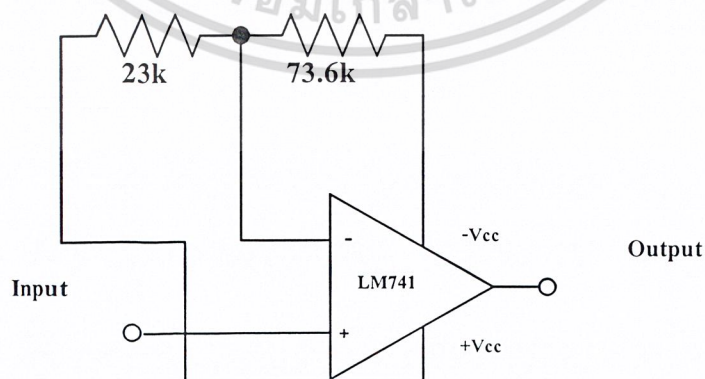


รูปที่ 3.8 วงจรด้านรับ 40.680MHz

วงจรคอมพาราเตอร์ (Comparator)

ในบางครั้งสัญญาณดิจิทัลที่ได้จากวงจรดีเท็คต่าง ๆ อาจจะมีไฟดีซี (DC) ผสมมา กับสัญญาณที่ ดีเท็คออกมาด้วย ซึ่งบางครั้ง ถ้าเราเอาสัญญาณนั้นไปใช้เลยอาจจะทำให้ไม่เข้ากับวงจรที่ไม่สนับสนุนไฟ ดีซี เช่น วงจรชมิททริกเกอร์ ซึ่งจะทำให้สัญญาณที่ได้จากวงจรดังกล่าวเกิดความผิดพลาดขึ้นได้

โดยวงจรที่จะช่วยทำให้ปัญหาสัญญาณดีซีหมดไปก็คือ วงจรคอมพาราเตอร์ ซึ่งจะช่วยให้ได้ สัญญาณดิจิทัลที่มีสัญญาณดีซีผสมมาหมดไปแล้วยังช่วยปรับความสวยงามของสัญญาณดิจิทัลที่เกิด ปัญหาการลดลงด้วย หรือที่เราเรียกกันว่าการกู้สัญญาณกลับมาใหม่ (Recovery) นั่นเอง



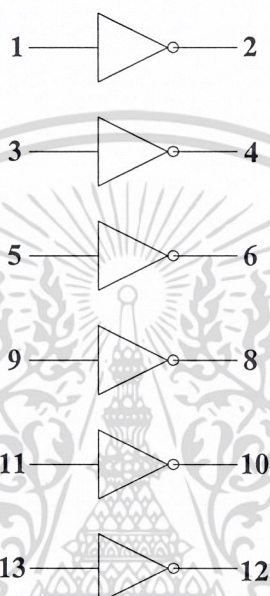
รูปที่ 3.9 แสดงวงจรคอมพาราเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรบัฟเฟอร์ (Buffer)

ในการต่อวงจรหลายๆวงจรเข้าด้วยกัน บางครั้งอาจจะเกิดปัญหาการลดทอนหรือการลดลงของสัญญาณในระหว่างส่วนรอยต่อของวงจรมันๆ ซึ่งเราจำเป็นต้องมีวงจรบัฟเฟอร์เข้าไปแทนที่ระหว่างส่วนนั้นๆ โดยในที่นี้เราได้ใช้วงจรชmitt trigger (Schmitt trigger) มาใช้เป็นวงจรบัฟเฟอร์

โดยเราใช้ไอซี 74F14 Hex inverter ของบริษัท Philips โดยมีขาทั้งหมด 14 ขา ไฟเลี้ยงขา 14 กราวด์ขา 7 ที่เหลือเป็นคู่ของอินพุตและเอาต์พุตอีก 6 คู่



รูปที่ 3.10 แสดงวงจรบัฟเฟอร์

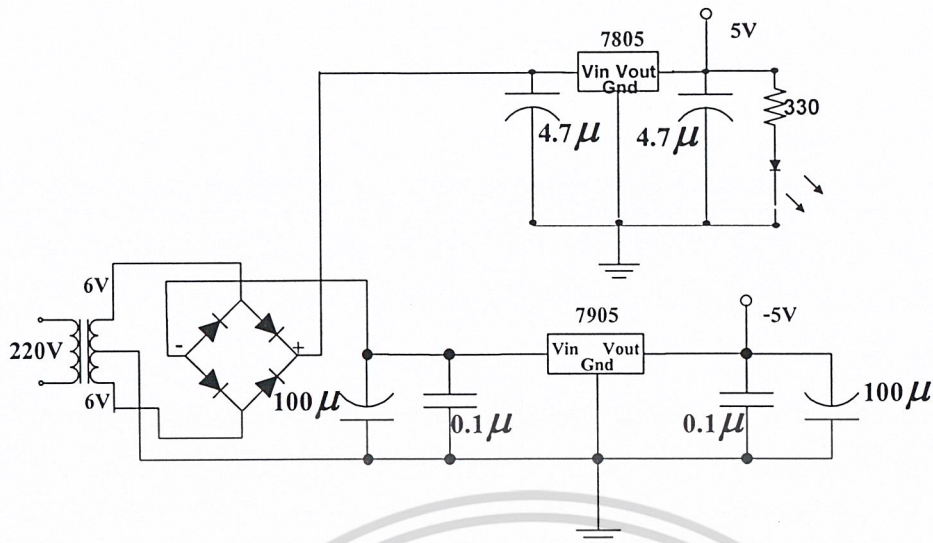
หลักการต่อเข้ากับระบบเป็นดังนี้

เราใส่บัฟเฟอร์ที่ 2 ส่วนของระบบคือ ระหว่างคอนโทรลเลอร์กับภาคส่งของสัญญาณที่ด้านส่ง และ ระหว่างภาครับกับคอนโทรลเลอร์ของสัญญาณที่ด้านรับ

วงจรจ่ายไฟ (Power Supply)

เป็นวงจรแปลงแรงดันกระแสสลับ (220VAC) ให้เป็นไฟกระแสตรง $\pm 5VDC$ ออกแบบโดยใช้ไอซีเร็กกูเลเตอร์ 7905 และ 7805 ซึ่งแสดงดังวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงวงจรจ่ายไฟ

หลักการต่อเข้ากับระบบเป็นดังนี้

การทำงานเริ่มจากกระแสสลับ (220VAC) ซึ่งผ่านหม้อแปลงขนาด 6 โวลต์ แปลงขนาดแรงดันให้เป็น 6VAC ผ่านวงจรเรกติไฟ์กลายเป็นไฟกระแสตรงแบบแมกนีน ผ่านคาปาซิเตอร์ทำการกรองความดันให้เรียบขึ้น ซึ่งจะนำไปเป็นอินพุตให้กับไอซีเรกูเลเตอร์ 7905 และ 7805 ทำการแปลงแรงดันจาก 6VAC เป็น $\pm 5\text{VDC}$ เพื่อนำไปใช้งานเป็นแหล่งจ่ายไฟให้กับวงจรคอมพิวเตอร์ต่อไป

บทที่ 4

การทดลองและผลการทดลอง

การทดลองที่ 1

เป็นการศึกษารูปแบบข้อมูลที่ส่งมาจากคีย์บอร์ดและเมาส์ โดยทำการเขียนโปรแกรมเก็บข้อมูลที่ได้มา แสดงผลผ่านหลอดไฟ LED

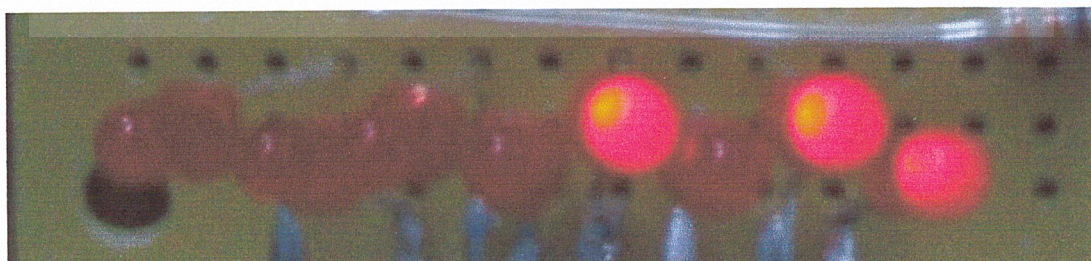
ผลการทดลอง



รูปที่ 4.1 ตัวอย่างข้อมูลที่ได้รับจากการกดปุ่ม “A” (ICH)



รูปที่ 4.2 ตัวอย่างข้อมูลที่ได้รับจากการกดปุ่ม “S” (IBH)

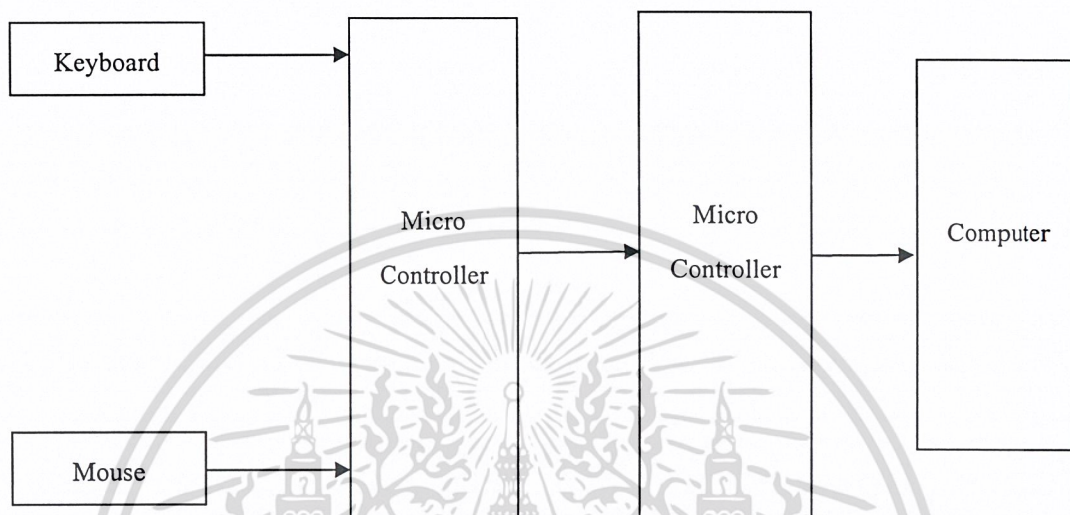


รูปที่ 4.3 ตัวอย่างข้อมูลที่ได้รับจากการกดปุ่ม “F6” (OBH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

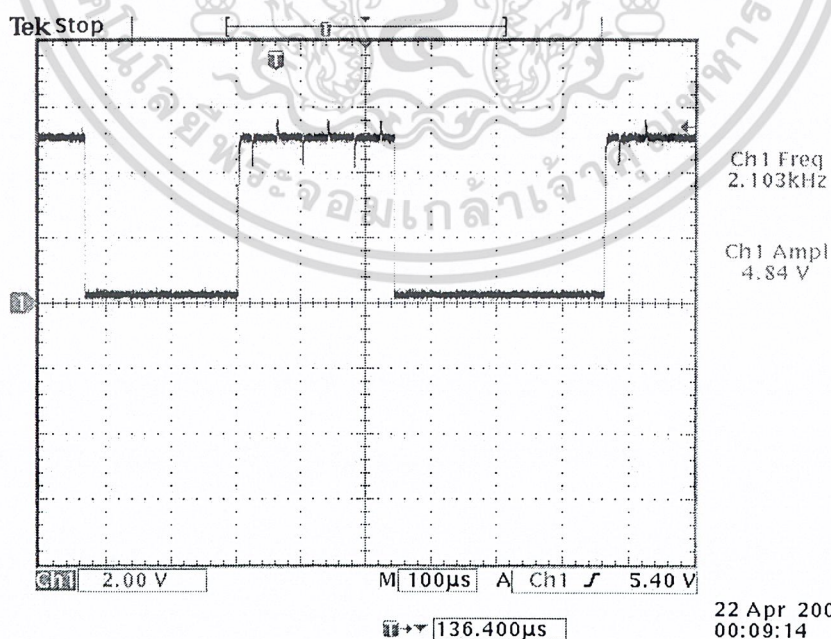
ทำการเชื่อมต่อระหว่างบอร์ดทดลองกับคีย์บอร์ด และ บอร์ดทดลองกับคอมพิวเตอร์โดยยังไม่ต้องส่งออกอากาศ เพื่อจำลองการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์กับคีย์บอร์ด โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวกลางในการจัดการส่งผ่านข้อมูลระหว่างกัน



รูปที่ 4.4 บล็อกไดอะแกรมการรับ-ส่งข้อมูลระหว่างคอมพิวเตอร์กับคีย์บอร์ด

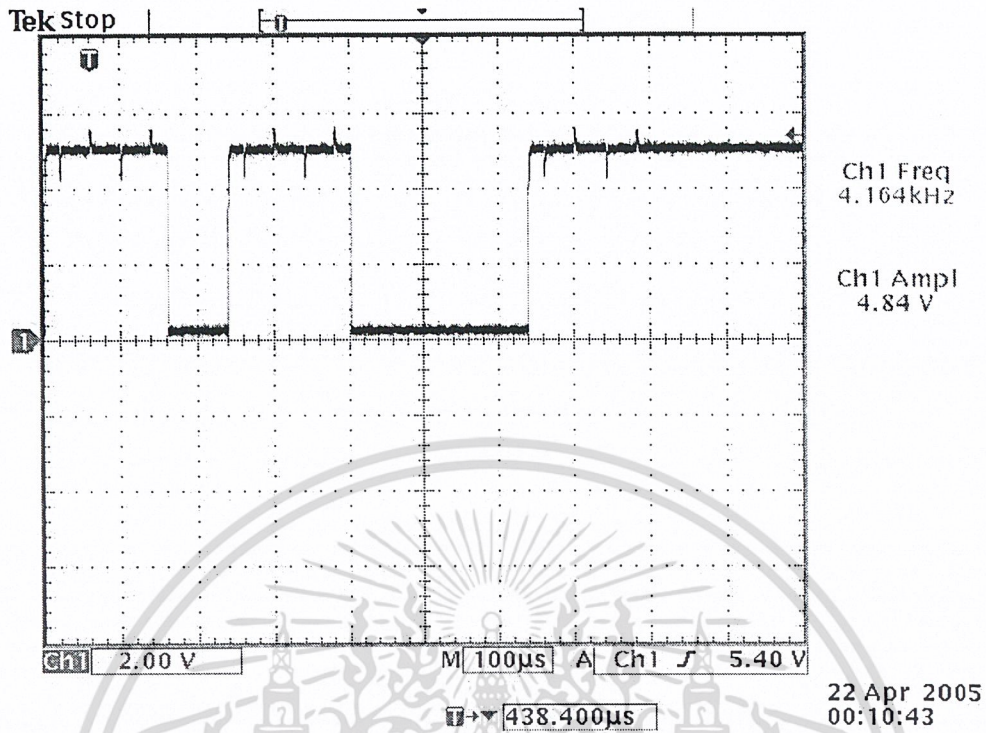
ผลการทดลอง

สามารถที่จะส่งข้อมูลจากเมาส์และคีย์บอร์ดไปแสดงผลยังหน้าจอคอมพิวเตอร์ได้

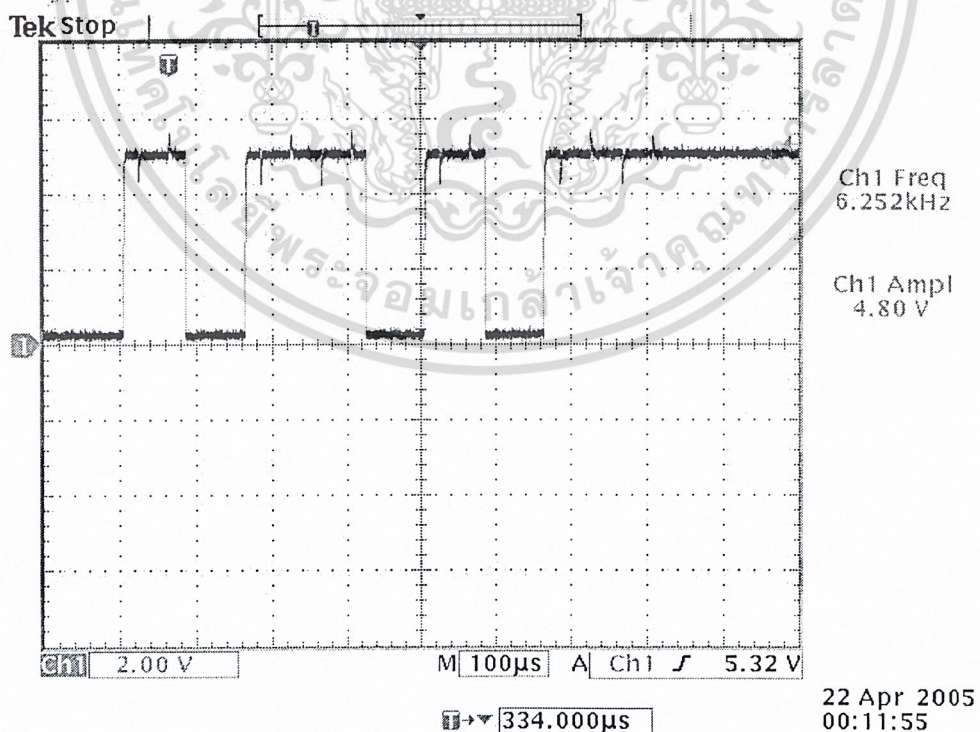


รูปที่ 4.5 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม “A” (1CH) ก่อนเข้าไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

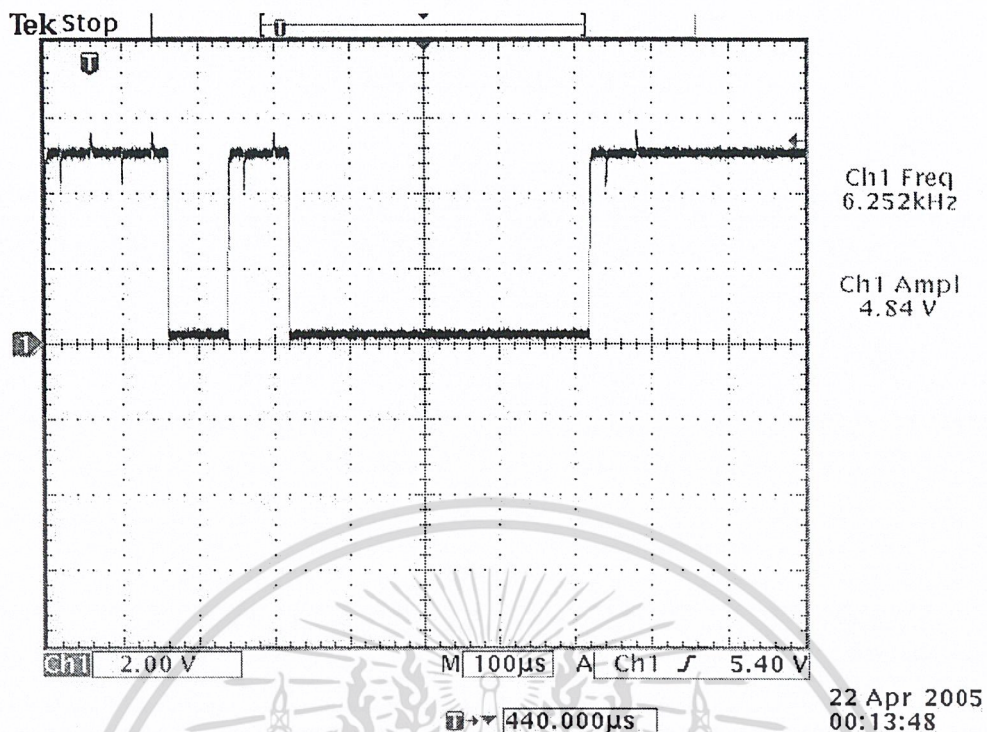


รูปที่ 4.6 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม “S” (IBH) ก่อนเข้าไมโครคอนโทรลเลอร์

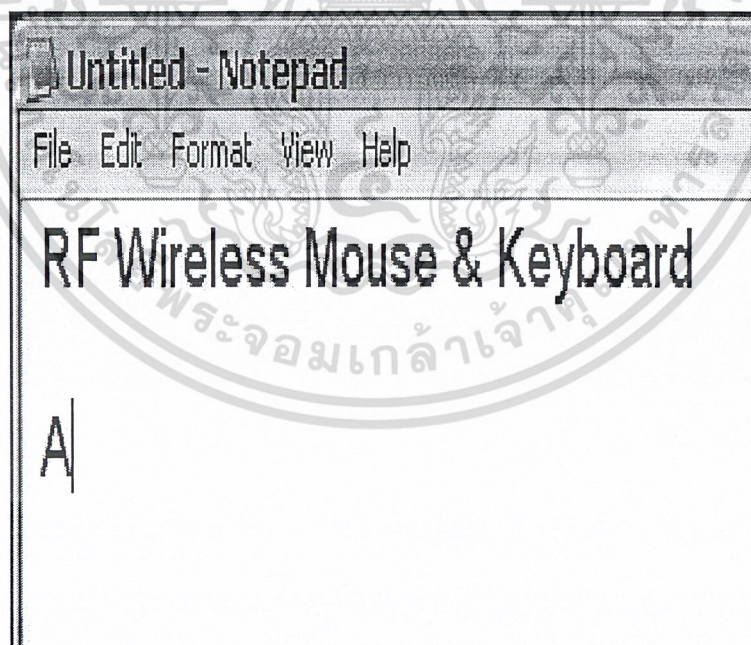


รูปที่ 4.7 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม “Enter” (5AH) ก่อนเข้าไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

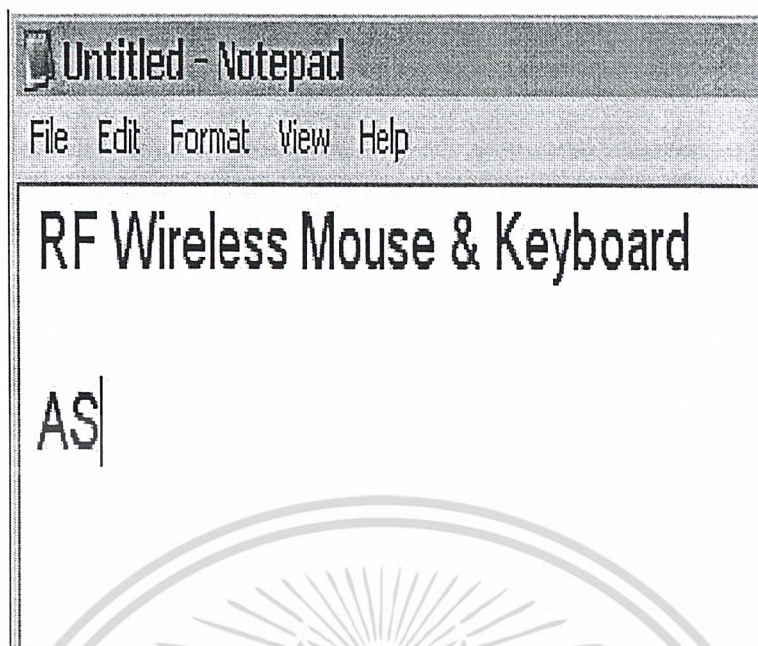


รูปที่ 4.8 รูปแสดงการส่งข้อมูลเมื่อกดคีย์บอร์ดปุ่ม “F6” (OBH) ก่อนเข้าไมโครคอนโทรลเลอร์



รูปที่ 4.9 ผลจากการกดปุ่ม “A” (1CH) สามารถแสดงผลได้ดังรูป (Print screen)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

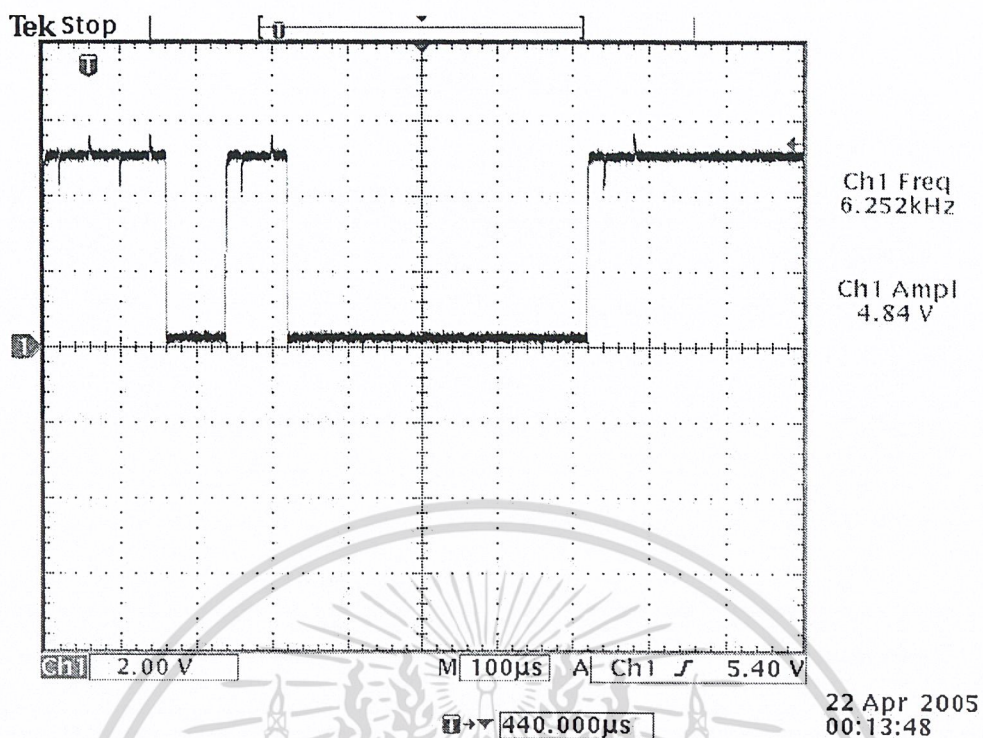


รูปที่ 4.10 ผลจากการกดปุ่ม “S” (1BH) สามารถแสดงผลได้ดังรูป (Print screen)



รูปที่ 4.11 ผลจากการกดปุ่ม “Enter” (5AH) สามารถแสดงผลได้ดังรูป (Print screen)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 รูปแสดงการส่งข้อมูลเมื่อกดปุ่มขวาของเมาส์ก่อนเข้าไมโครคอนโทรลเลอร์



รูปที่ 4.13 ผลจากการกดปุ่มขวาของเมาส์ สามารถแสดงผลได้ดังรูป (Print screen การกดเอา)

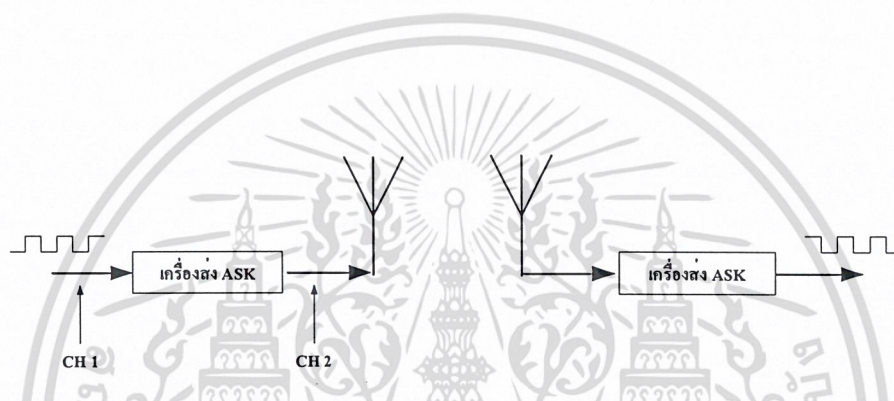
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

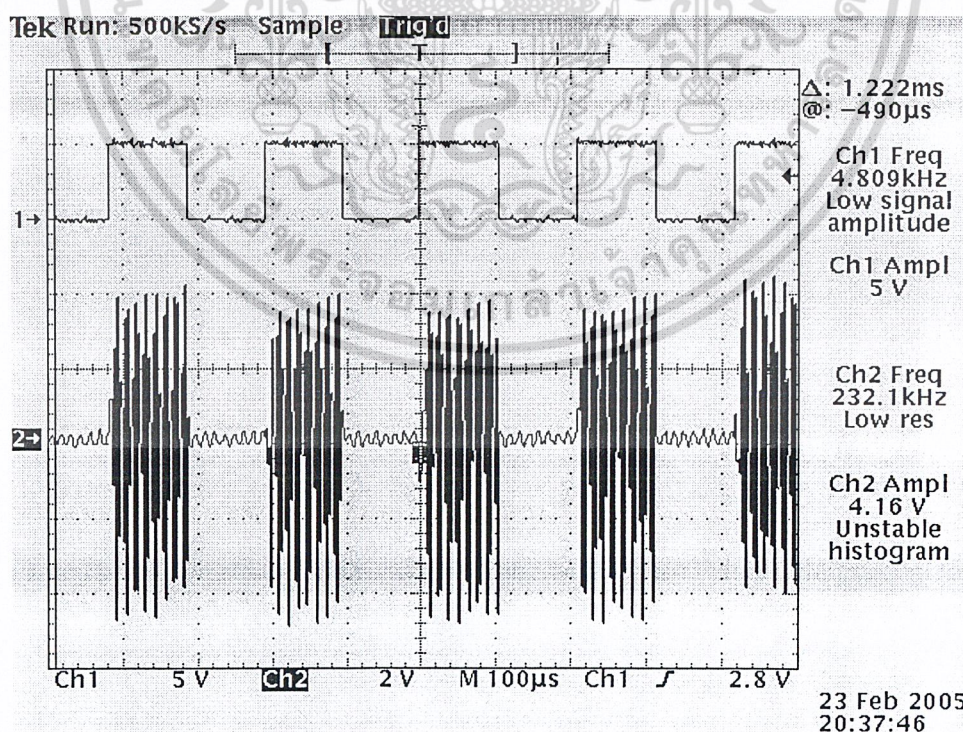
ทดลองส่งข้อมูลระหว่างเครื่องรับ-ส่งแบบแอมพลิจุ่มมอดูเลชัน โดยส่งสัญญาณที่เป็นสัญญาณสี่เหลี่ยม (Square Wave) ที่ความถี่ต่างๆ เพื่อหาค่าอัตราบอร์ตสูงสุดที่สามารถส่งได้และกู้สัญญาณออกมาได้

ผลการทดลอง

ทำการวัดสัญญาณโดยใช้เครื่องวัดสัญญาณวัดที่จุดต่างๆ ดังรูป

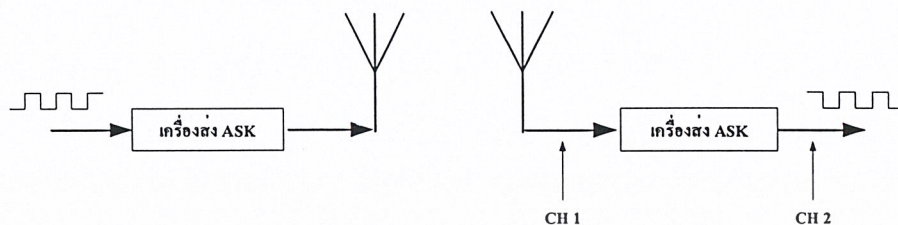


รูปที่ 4.14 รูปการวัดโดยวัดอินพุต (Ch1) เทียบกับเอาต์พุต (Ch2) ของภาคส่ง

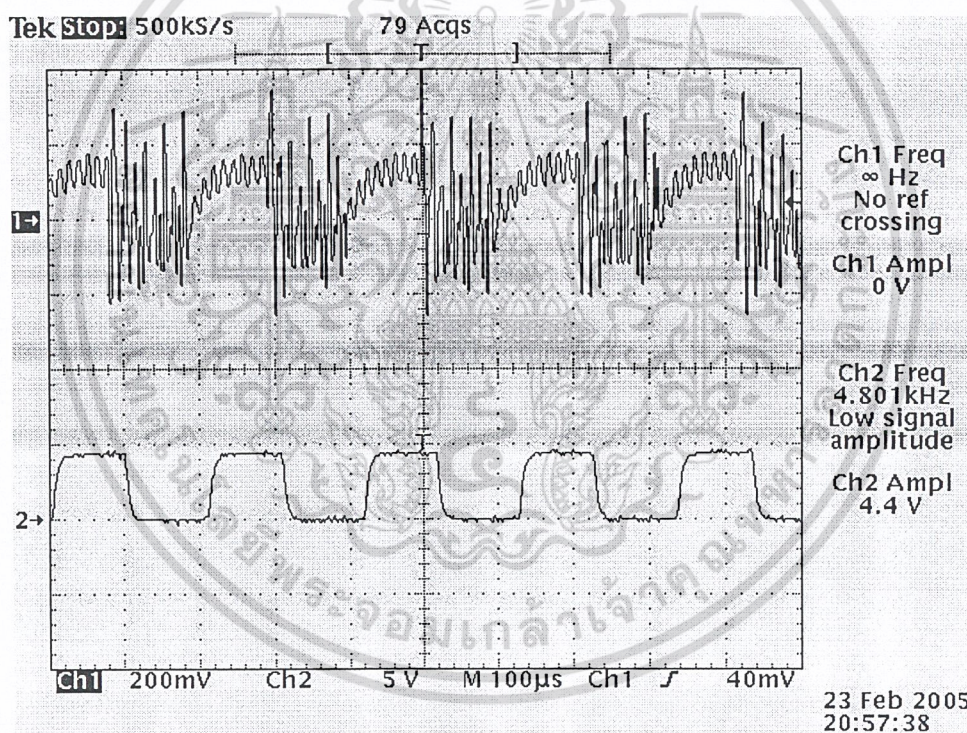


รูปที่ 4.15 อินพุตที่ป้อนกับภาคส่งกับเอาต์พุตจากภาคส่งเทียบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

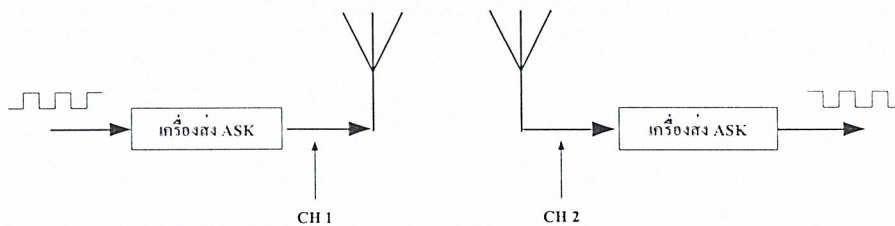


รูปที่ 4.16 รูปการวัด โดยวัดอินพุต (Ch1) เทียบกับเอาต์พุต (Ch2) ของภาครับ

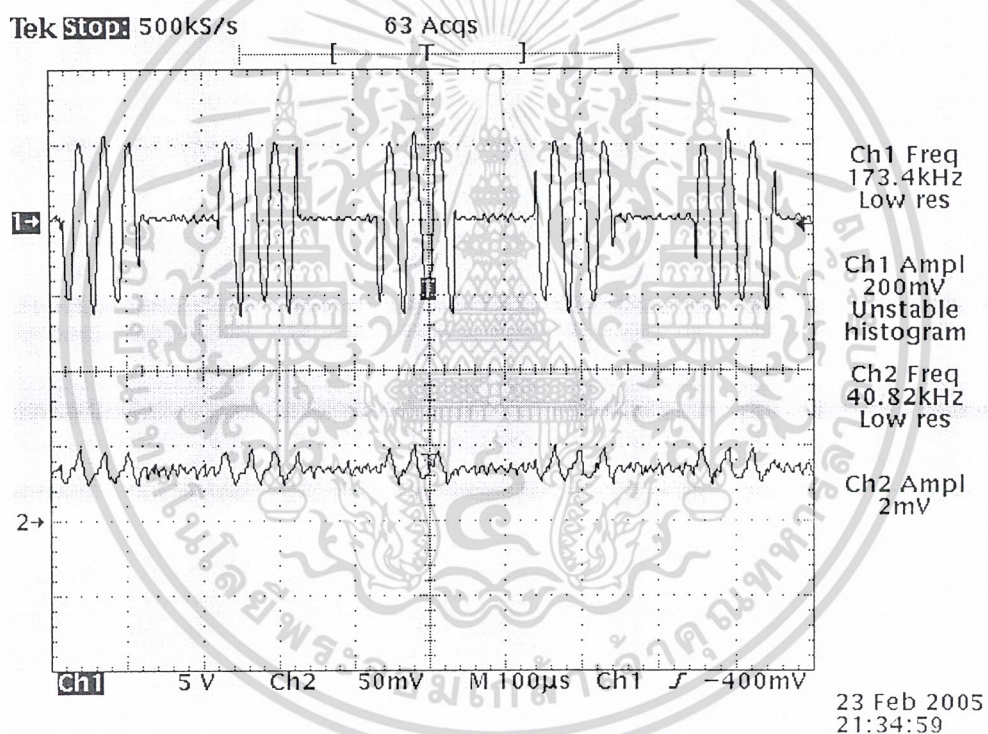


รูปที่ 4.17 อินพุตที่ภาครับเทียบกับเอาต์พุตที่ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

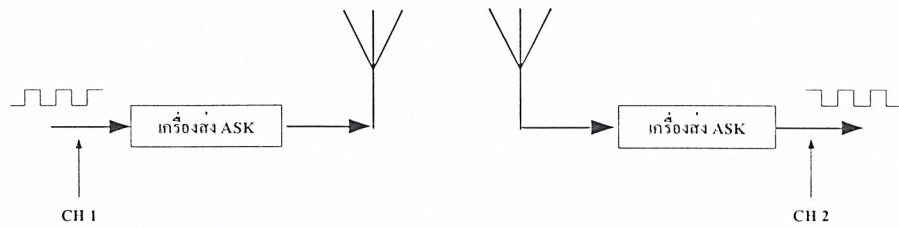


รูปที่ 4.18 รูปการวัดโดยวัดเอาต์พุตภาคส่ง(CH1) เทียบกับอินพุต (Ch2) ของภาครับ

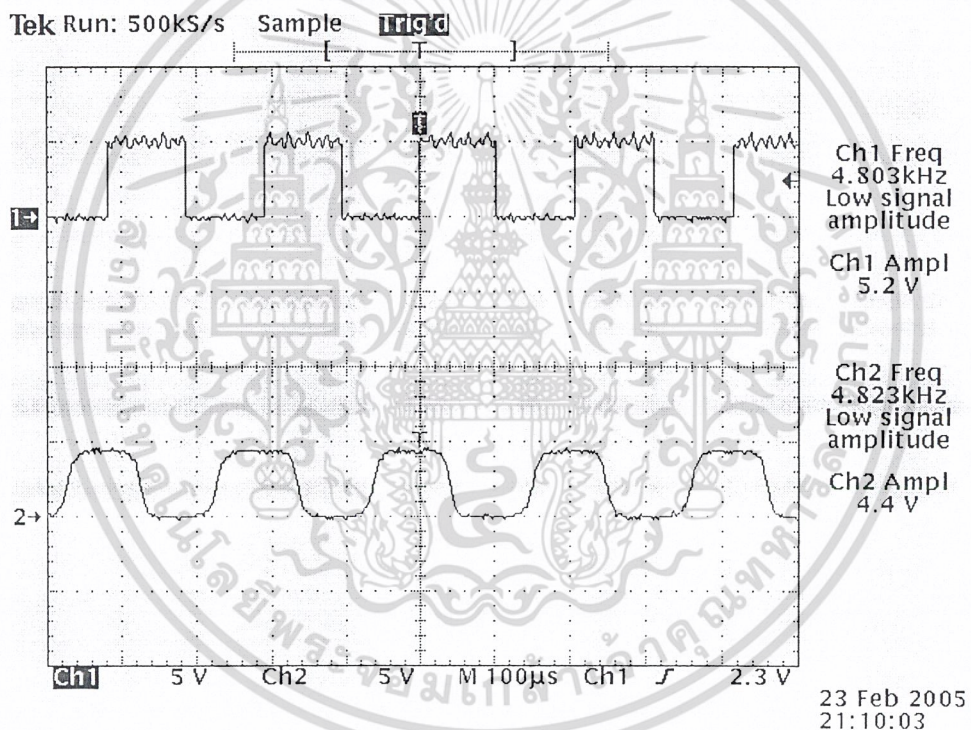


รูปที่ 4.19 เอาท์พุทของภาคส่งเทียบกับอินพุทที่ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 รูปการวัดโดยวัดอินพุตภาคส่ง (Ch1) เทียบกับเอาต์พุต (Ch2) ของภาครับ



รูปที่ 4.21 อินพุตที่ภาคส่งเทียบกับเอาต์พุตภาครับ

การทดลองที่ 4

ทำการรวมชุดรับ-ส่ง ทั้งส่วนไมโครคอนโทรลเลอร์และส่วนเครื่องรับ-ส่งเอเอ็มเข้าด้วยกัน

ผลการทดลอง

สามารถส่งผ่านข้อมูลจากคีย์บอร์ดและเมาส์ไปยังคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุปผล

จากการเขียนโปรแกรมในการทดลองที่ 1 นั้น ข้อมูลที่รับมาได้มีค่าตรงกับรหัสประจำปุ่ม และเราก็ใช้สโคปจับภาพสัญญาณที่ได้เทียบเคียงด้วยเพื่อตรวจสอบว่าข้อมูลที่ได้มีค่าตรงกันจริงๆ ปัญหาที่เกิดจากการทดลองนี้คือ เราต้องหาสโคปที่มีความไวพอที่จะจับภาพของข้อมูลที่เรากดออกมาจากเมาส์ และคีย์บอร์ดได้ ถ้าเราใช้สโคปที่มีความไวน้อยจะไม่สามารถจับภาพชุดข้อมูลที่กดบนคีย์บอร์ดและเมาส์ได้เลย

จากการทดลองที่ 2 เราสามารถส่งข้อมูลจากคีย์บอร์ดมาแสดงที่หน้าคอมได้ โดยเราเขียนโปรแกรมให้คอลโทรลเลอร์รับค่ามาจากเมาส์และคีย์บอร์ดและนำข้อมูลส่งออกไปหาคอนโทรลเลอร์อีกตัวทางด้านคอมพิวเตอร์ เพื่อรับข้อมูลแล้วส่งต่อไปยังคอมพิวเตอร์เพื่อประมวลผลแสดงที่หน้าจอต่อไป ปัญหาที่เกิดขึ้นในการทดลองนี้คือเราต้องแปลงความเร็วบิตเรตที่ปกติเมาส์และคีย์บอร์ดจะติดต่อกับคอมพิวเตอร์ด้วยอัตราบิต 9600 bps แต่เราต้องลดสเปกของมันลงเหลืออัตราบิต 4800 bps เพื่อที่จะนำมาใช้กับชุดรับส่ง ASK ที่มีสเปกอัตราบิต 4800 bps เท่านั้น ไม่เช่นนั้นถ้าข้อมูลที่ส่งมาถึงชุดรับส่งถ้าส่งด้วยอัตราบิต 9600 bps จากองข้อมูลที่อยู่ต่อระหว่างชุดรับส่งทำให้เกิดการซ้ำซ้อนของข้อมูลขึ้นทำให้เกิดความผิดพลาดขึ้นได้

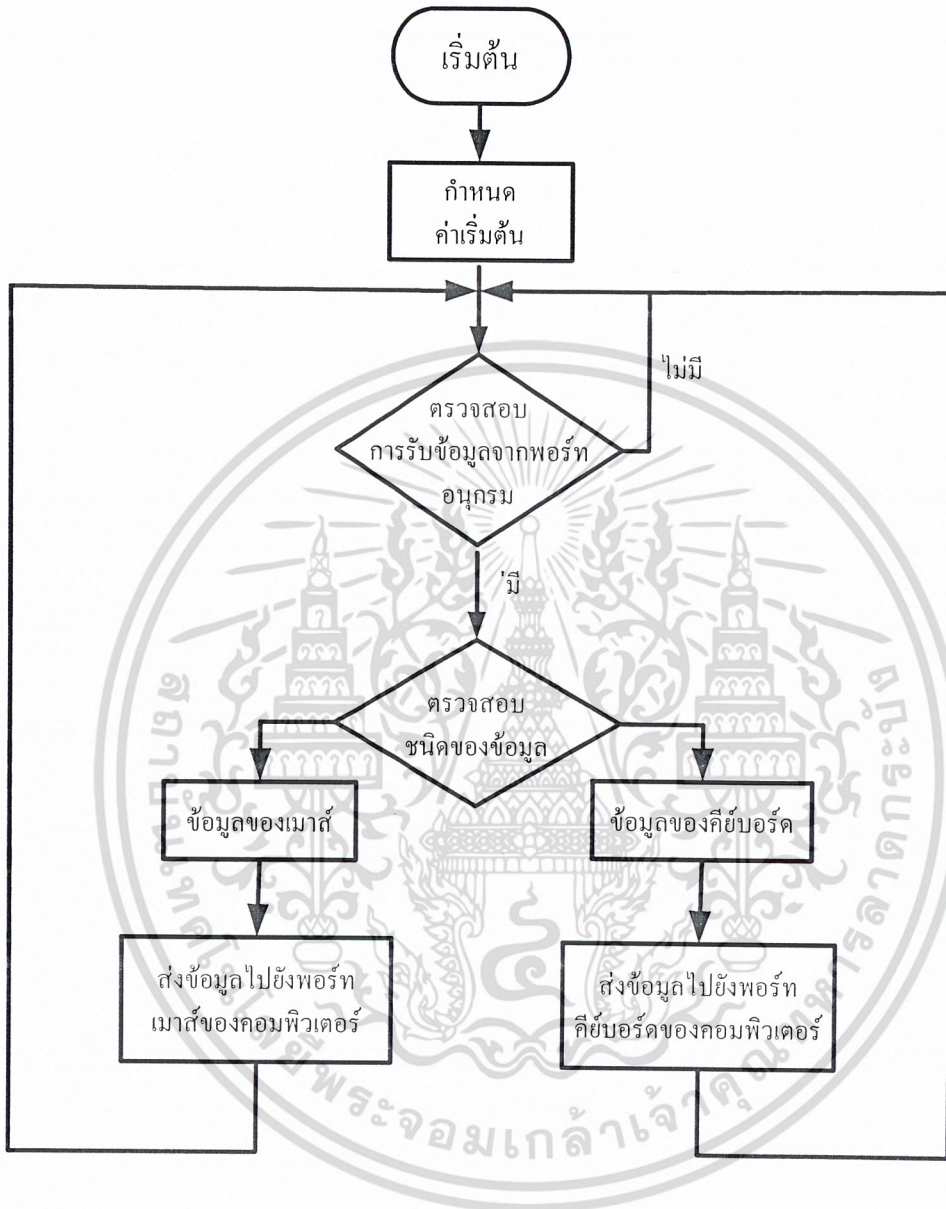
จากการทดลองที่ 3 เรานำวงจรภาครับ-ส่งแบบ ASK มาทำการทดลองส่งสัญญาณ Square Wave โดยปัญหาที่เกิดขึ้นคือ ถ้าส่งสัญญาณที่มีความถี่มากกว่า 4800 Bit/Sec จะทำให้เอาต์พุตที่ได้มีความผิดเพี้ยนไปจากที่ส่งมา และถ้าเราส่งข้อมูลด้วยชุดรับส่ง ASK นี้ด้วยระยะทางมากกว่า 3 เมตรไปสัญญาณจะเกิดความผิดเพี้ยนขึ้น

จากการทดลองที่ 4 เรารวบวงจรจากการทดลองที่ 2 กับ 3 เข้าด้วยกันแล้วเกิดปัญหาการคอบของสัญญาณที่อยู่ต่อของวงจรระหว่างคอนโทรลเลอร์กับภาคส่ง-รับ เราจึงแก้โดยใส่วงจรบัฟเฟอร์เข้าไปทำให้สามารถลดการคอบของสัญญาณลง และสามารถที่จะส่งข้อมูลที่มีความแรงของสัญญาณพอที่จะมาถึงทางด้านรับ และปัญหาอีกที่ก็คือเราใช้วงจรหมิตริกเกอร์ในการทำเป็นบัฟเฟอร์แต่มันต้องการสัญญาณอินพุตเป็นสัญญาณเอชไอทีเท่านั้น แต่สัญญาณเอาต์พุตที่ออกจากชุดรับ ASK สัญญาณจะมีไฟดิซิผสมเข้ามาด้วยเราจึงต้องหาวงจรที่นำค่าดิซิดังกล่าวออกไป โดยเราได้ใช้วงจรคอมพาราเตอร์มาใช้ในการตัดไฟดิซิออกและทำให้เราได้สัญญาณที่ออกมามีความคมชัดมากกว่าเดิมด้วย



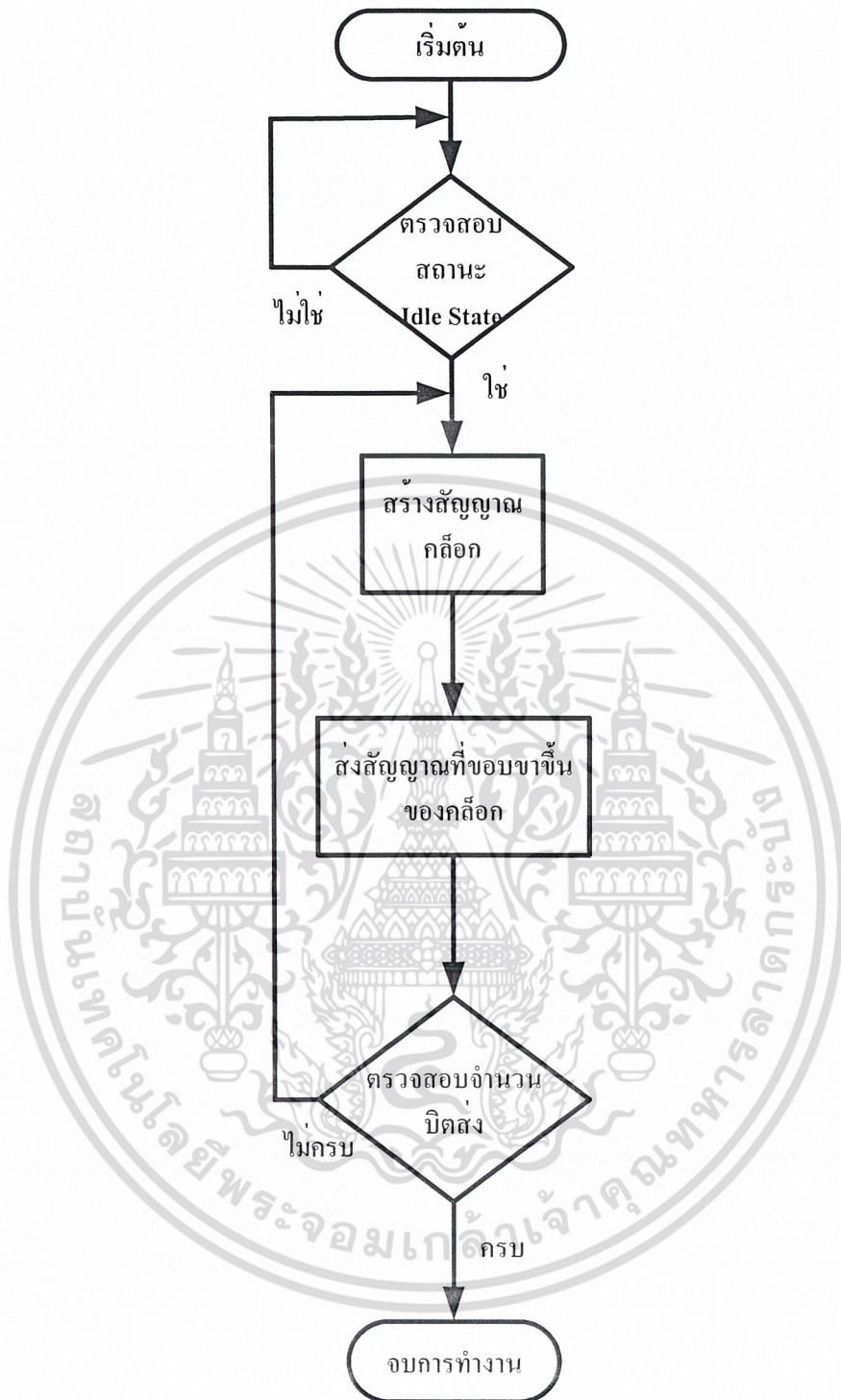
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลว์ชาร์ตการทำงานของระบบ



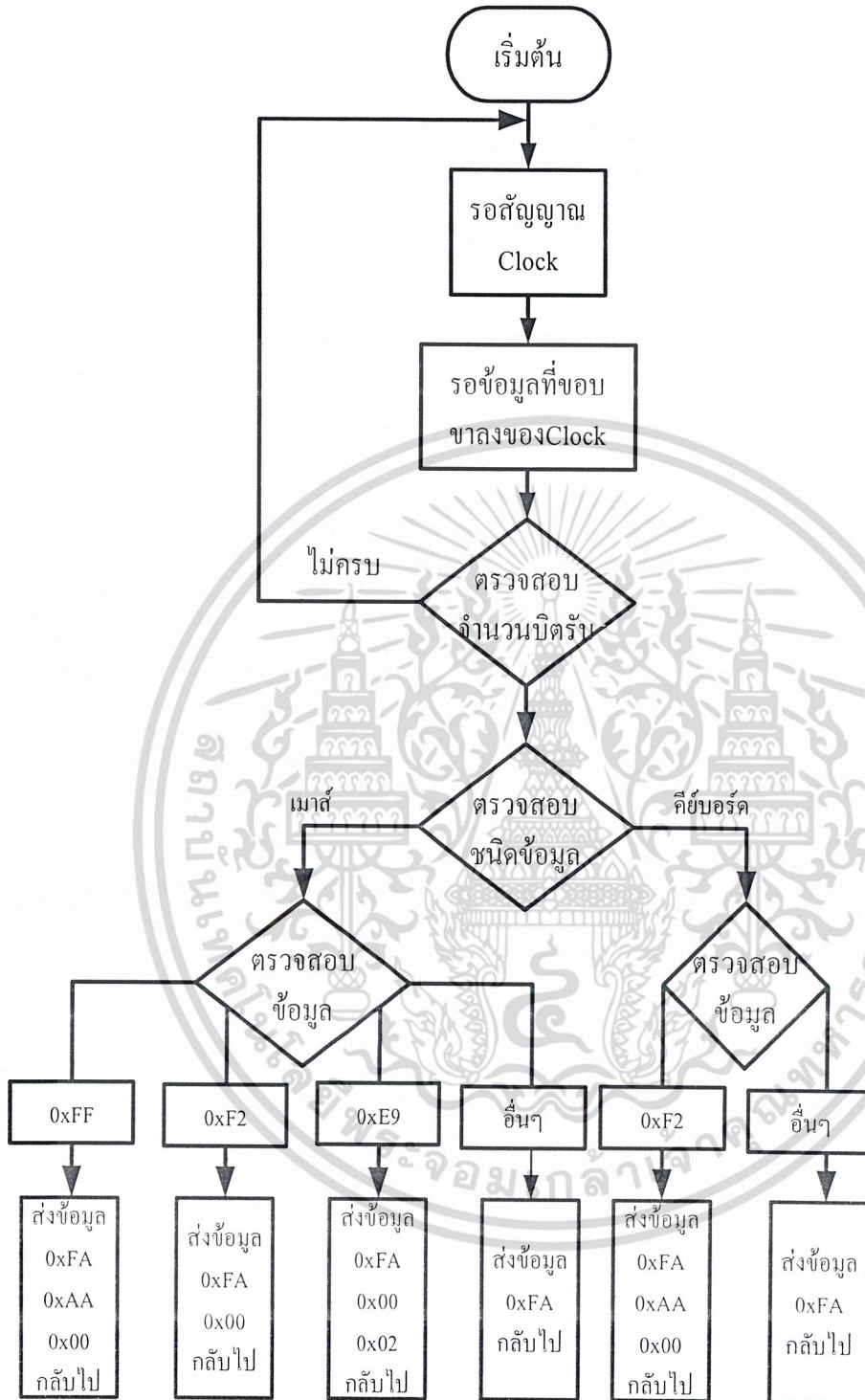
โฟลว์ชาร์ตด้านรับ (ด้านคอมพิวเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



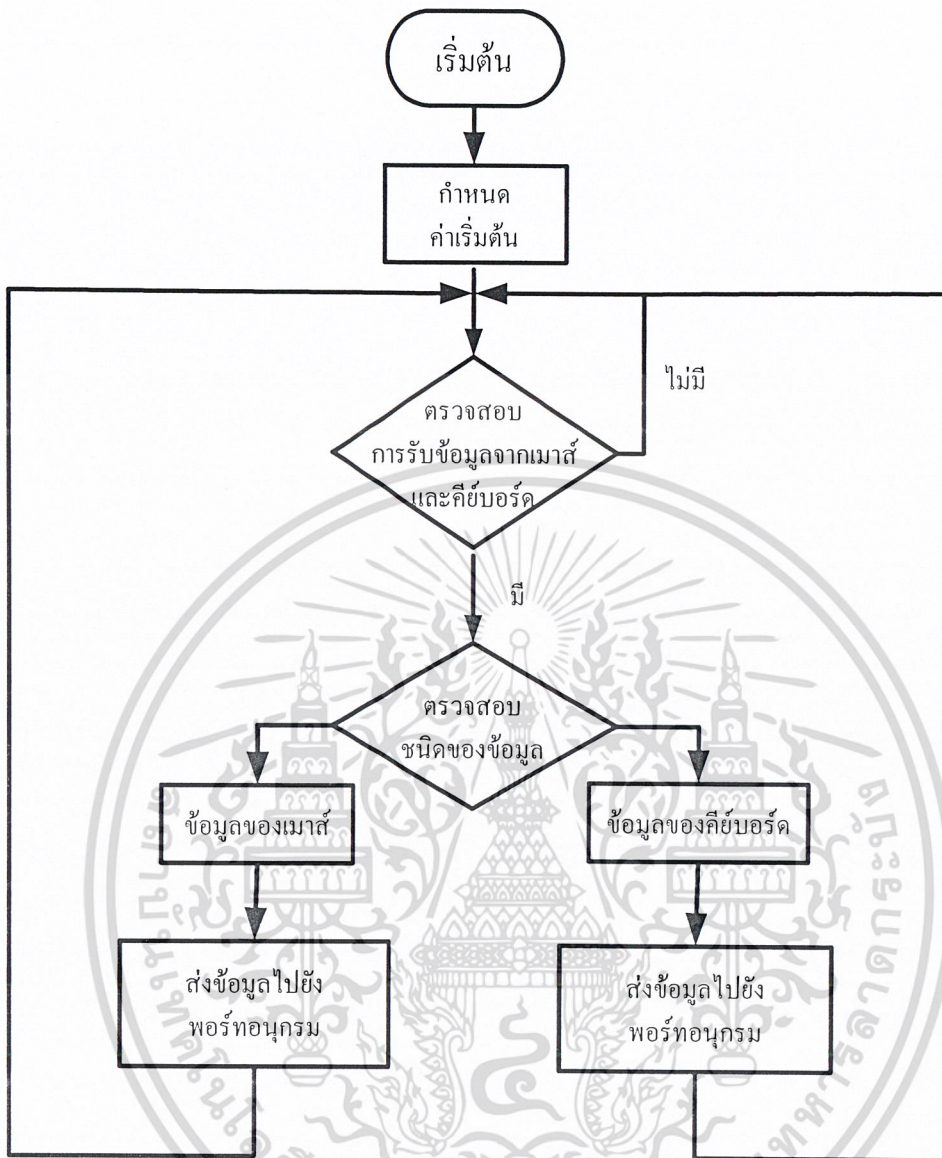
โฟลว์ชาร์ตแสดงวิธีการส่งข้อมูลไปยังพอร์ทเม้าส์และคีย์บอร์ดของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



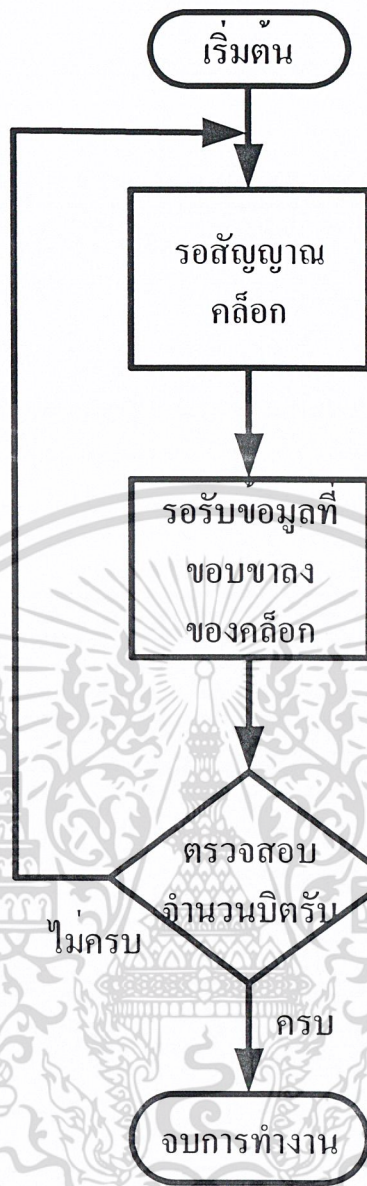
ไฟล์วีซาร์ตแสดงวิธีการรับข้อมูลจากคอมพิวเตอร์ผ่านขาอินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



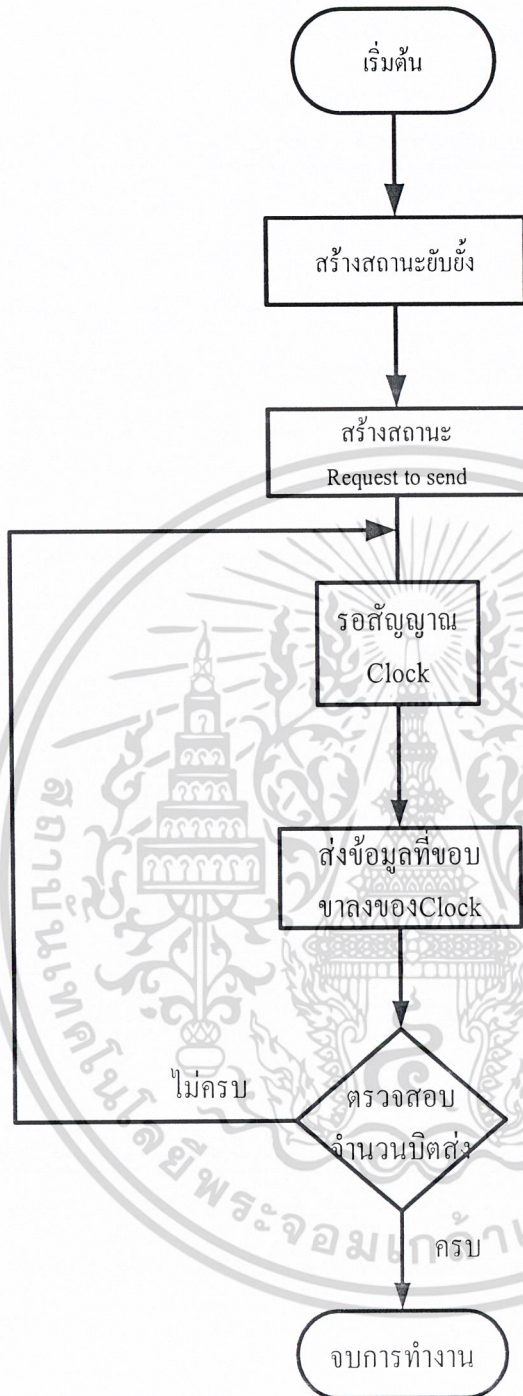
โฟลว์ชาร์ตด้านส่ง (ด้านเม้าส์และคีย์บอร์ด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟลว์ชาร์ตแสดงวิธีการรับข้อมูลจากเมาส์และคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟลว์ชาร์ตแสดงวิธีการส่งข้อมูลจากมาส์และกีบอร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code Program

คำสั่ง

```
#pragma code
#include <reg51.h>
#include <string.h>
#include <intrins.h>
sbit SDA_key = P3^3;
sbit SCL_key = P3^4;
sbit SDA_mouse = P3^2;
sbit SCL_mouse = P3^5;

bit inbit,parity_bit;
unsigned char i,dat,temp=0;
unsigned char x[10];
int count;
unsigned char led;
bit scroll_bit,num_bit,cap_bit;
/*Keyboard & Mouse both use*/
/*****
*****/
void delay() /*delay
50 us*/
{
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();
}
void delay25() /*delay
25 us*/
{
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();
}
void delay2(int count)
{
    int i,j;
    for (i = 0 ; i < count ; i++)for (j = 0 ; j < 500 ; j++);
}
void gen_parity(unsigned char temp_dat)
{
    bit outbit;
    int i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

parity_bit = 1;
for (i=0;i<8;i++)
{
    outbit = temp_dat & 0x80;
    parity_bit ^= outbit;
    temp_dat = temp_dat<<1;
}
}
/*****
*****/
/*keyboard use only*/
/*****
*****/
void gen_led()
{
    led = 0x00;
    led = led<<1;
    led = led | scroll_bit;
    led = led<<1;
    led = led | num_bit;
    led = led<<1;
    led = led | cap_bit;
    for (i=0;i<5;i++) led = led<<1;
}
void one_bit()
{
    while (SCL_key == 1);
    while (SCL_key == 0);
}
void send_to_key(unsigned char old_dat)
{
    bit outbit;
    int i;
    gen_parity(old_dat);
    SCL_key = 0;          /*inhibit
communication*/
    delay();
    delay();
    delay25();          /*delay 125 us*/
    SDA_key = 0;          /*Request-to-Send +
Send Start bit*/
    SCL_key = 1;
    while (SCL_key == 1);
    for (i=0;i<8;i++)
    {
        outbit = old_dat & 0x80;
        SDA_key = outbit;
        old_dat = old_dat<<1;
        while (SCL_key == 0);
        while (SCL_key == 1);
    }
    SDA_key = parity_bit;
    while (SCL_key == 0);
    while (SCL_key == 1);
    SDA_key = 1;
    while (SCL_key == 0);
    while (SCL_key == 1);          /*wait for ACK*/
    while (SCL_key == 0);
    while (SDA_key == 0);
}
}
void key_serial_send(unsigned char old_dat)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    while (!TI);
    TI=0;
    TB8 = 0;
    SBUF = old_dat;
}
/*****
*****/
/*Mouse use only*/
/*****
*****/
void one_bit_mouse()
{
    while (SCL_mouse == 1);
    while (SCL_mouse == 0);
}
void send_to_mouse(unsigned char old_dat)
{
    bit outbit;
    int i;
    gen_parity(old_dat);
    SCL_mouse = 0; /*inhibit
communication*/
    delay();
    delay();
    delay25(); /*delay 125 us*/
    SDA_mouse = 0; /*Request-to-Send +
Send Start bit*/
    SCL_mouse = 1;
    while (SCL_mouse == 1);
    for (i=0;i<8;i++)
    {
        outbit = old_dat & 0x80;
        SDA_mouse = outbit;
        old_dat = old_dat<<1;
        while (SCL_mouse == 0);
        while (SCL_mouse == 1);
    }
    SDA_mouse = parity_bit;
    while (SCL_mouse == 0);
    while (SCL_mouse == 1);
    SDA_mouse = 1;
    while (SCL_mouse == 0);
    while (SCL_mouse == 1); /*wait for ACK*/
    while (SCL_mouse == 0);
    while (SDA_mouse == 0);
}
void mouse_serial_send(unsigned old_dat)
{
    while (!TI);
    TI=0;
    TB8 = 1;
    SBUF = old_dat;
}
void main(void)
{
    SCON = 0xCA; /*serial port mode 1*/
    TMOD = 0x20; /*Timer1 = mode 2*/
    TH1 = 0xEC; /*baud rate 4800 (EC) Fosc =
18.432 MHz*/
    TR1 = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count = 0;
/*initial mouse*/
delay2(5000);
send_to_mouse(0x2F);
/*initial keyboard*/
scroll_bit = 0; num_bit = 1; cap_bit = 0;
gen_led();
send_to_key(0xB7);
send_to_key(led);
while(1)
{
    SDA_mouse = 1;
    SCL_mouse = 1;
    SDA_key = 1;
    SCL_key = 1;
    /*****
    *****/
    /*Mouse use only*/
    /*****
    *****/
    if (SDA_mouse ==0)
    {
        one_bit_mouse();          /*delay start
bit*/
        for (i=0;i<8;i++)
        {
            while (SCL_mouse == 1);
            delay25();
            inbit = SDA_mouse;
            dat = dat<<1;
            dat = dat | inbit;
            while (SCL_mouse == 0);
        }
        one_bit_mouse();          /*delay parity
bit*/
        one_bit_mouse();          /*delay stop
bit*/
        x[count] = dat;
        count = count+1;
        if (dat == 0x5F) count = 0;
    }
    else
    {
        if (count == 3)
        {
            send_to_mouse(0xAF);
            for (i=0;i<count;i++)
            {
                mouse_serial_send(x[i]);
            }
            count = 0;
            send_to_mouse(0x2F);
        }
    }
}

/*keyboard use only*/
/*****
*****/
if (SDA_key ==0)
{
    one_bit();          /*delay start bit*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=0;i<8;i++)
{
    while (SCL_key == 1);
    delay25();
    inbit = SDA_key;
    dat = dat<<1;
    dat = dat | inbit;
    while (SCL_key == 0);
}
one_bit();          /*delay parity bit*/
one_bit();          /*delay stop bit*/
if (dat != 0x5F || dat != 0x00 || dat !=
0x55)  key_serial_send(dat);
if (dat == 0xEE || dat == 0x1A || dat ==
0x7E)
    {
        while (count < 2)
        {
            if (SDA_key ==0)
            {
                one_bit();
                for
(i=0;i<8;i++)
                {
                    while (SCL_key == 1);
                    delay25();
                    inbit = SDA_key;
                    dat = dat<<1;
                    dat = dat | inbit;
                    while (SCL_key == 0);
                }
                one_bit();
                one_bit();
                key_serial_send(dat);
                x[count] = dat;
                count = count+1;
            }
        }
        switch (x[1])
        {
            case 0xEE : num_bit =
~num_bit;
                        break;
            case 0x1A : cap_bit =
~cap_bit;
                        break;
            case 0x7E : scroll_bit =
~scroll_bit;
                        break;
        }
        gen_led();
        send_to_key(0xB7);
        send_to_key(led);
        count = 0;
    }
}
/*****
*****/
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านรับ

```
#pragma code
#include <reg51.h>
#include <string.h>
#include <intrins.h>
sbit SDA_key = P3^3;           /*INT1*/
sbit SCL_key = P3^4;
sbit SDA_mouse = P3^2;       /*INT0*/
sbit SCL_mouse = P3^5;
bit inbit,parity_bit;
unsigned char i,dat=0;
unsigned char test;
/*Keyboard & Mouse both use*/
/*****
*****/
void delay()                  /*delay 50 us*/
{
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();
}
void delay25()               /*delay 25 us*/
{
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_();_nop_()
;_nop_();
    _nop_();
}
void delay2(int count)
{
    int i,j;
    for (i = 0 ; i < count ; i++)for (j = 0 ; j < 500 ; j++);
}
void gen_parity(unsigned char temp_dat)
{
    bit outbit;
    int i;
    parity_bit = 1;
    for (i=0;i<8;i++)
    {
        outbit = temp_dat & 0x80;
        parity_bit ^= outbit;
        temp_dat = temp_dat<<1;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****
*****/
/*keyboard use only*/
/*****
*****/
void Clock_high()
{
    SCL_key = 1;
    delay();
}
void Clock_low()
{
    SCL_key = 0;
    delay();
}
void key_to_pc_state()
{
    SCL_key = 0;
    delay();
    SCL_key = 1;
}
void key_send_data_pc(unsigned char old_dat) /*send at high
rise*/
{
    bit outbit;
    gen_parity(old_dat);
    while (SCL_key == 0);
    delay(); /*check clock high at least
50us*/
    delay25();
    while (SCL_key == 0);
    key_to_pc_state();
    SDA_key = 0; /*SEND START BIT*/
    delay();
    for (i=0;i<8;i++)
    {
        key_to_pc_state();
        outbit = old_dat & 0x80;
        SDA_key = outbit;
        old_dat = old_dat<<1;
        delay(); /*delay 50us*/
    }
    key_to_pc_state();
    SDA_key = parity_bit; /*send parity bit*/
    delay();
    key_to_pc_state();
    SDA_key = 1; /*send stop bit*/
    delay();
    Clock_low();
    SCL_key = 1;
}
/*****
*****/
/*Mouse use only*/
/*****
*****/
void mouse_Clock_high()
{
    SCL_mouse = 1;
    delay();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void mouse_Clock_low()
{
    SCL_mouse = 0;
    delay();
}
void mouse_to_pc_state()
{
    SCL_mouse = 0;
    delay();
    SCL_mouse = 1;
}
void mouse_send_data_pc(unsigned char old_dat)          /*mouse_send at
high rise*/
{
    bit outbit;
    gen_parity(old_dat);
    while (SCL_mouse == 0);
    delay();                                           /*check mouse_Clock
high at least 50us*/
    delay25();
    while (SCL_mouse == 0);
    mouse_to_pc_state();
    SDA_mouse = 0;                                     /*mouse_send START
BIT*/
    delay();
    for (i=0;i<8;i++)
    {
        mouse_to_pc_state();
        outbit = old_dat & 0x80;
        SDA_mouse = outbit;
        old_dat = old_dat<<1;
        delay();                                     /*delay 50us*/
    }
    mouse_to_pc_state();
    SDA_mouse = parity_bit;                           /*mouse_send parity
bit*/
    delay();
    mouse_to_pc_state();
    SDA_mouse = 1;                                    /*mouse_send stop
bit*/
    delay();
    mouse_Clock_low();
    SCL_mouse = 1;
}

void main(void)
{
    /*serial*/
    SCON = 0xD4;                                     /*serial port mode 1*/
    TMOD = 0x20;                                     /*Timer1 = mode 2*/
    TH1 = 0xEC;                                       /*baud rate 4800(EC) Fosc = 18.432
MHz*/
    TR1 = 1;
    /*interrupt*/
    IT1 = 0x1;                                       /*interrupt when low rise*/
    IT0 = 0x1;                                       /*interrupt when low rise*/
    IE = 0x85;                                       /*enable interrupt from INT1&INT0*/
    IP = 0x01;                                       /*INT0(mouse) priority over INT1(key)*/
    mouse_send_data_pc(0x55);
    mouse_send_data_pc(0x00);
    delay2(5000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(1)
{
    SDA_mouse = 1;
    SCL_mouse = 1;
    SDA_key = 1;
    SCL_key = 1;
    while(!RI);
    RI = 0;
    test = SBUF;
    if (RB8 == 1) mouse_send_data_pc(test);
        if (RB8 == 0) key_send_data_pc(test);
}
}

void mouse_receive_from_pc() interrupt 0          /*interrupt2 from
INT1, interrupt0 from INTO*/
{
    if (SDA_mouse == 0 && SCL_mouse == 0)
    {
        while (SCL_mouse == 0);
        mouse_Clock_high();          /*mouse_send start
bit*/
        mouse_Clock_low();
        for (i=0;i<8;i++)
        {
            delay25();          /***sample data im middle
cell***/
            inbit = SDA_mouse;
            dat = dat<<1;
            dat = dat | inbit;
            mouse_Clock_high();
            mouse_Clock_low();
        }
        mouse_Clock_high();
        mouse_Clock_low();          /*mouse_send parity
bit*/
        mouse_Clock_high();
        mouse_Clock_low();          /*mouse_send stop
bit*/
        SDA_mouse = 0;          /*ACK*/
        mouse_Clock_high();
        mouse_Clock_low();
        SDA_mouse = 1;
        SCL_mouse = 1;
        if (dat == 0xFF)
        {
            mouse_send_data_pc(0x5F);
            mouse_send_data_pc(0x55);
            mouse_send_data_pc(0x00);          /*00 = 2
button*/
        }
        if (dat == 0x4F)
        {
            mouse_send_data_pc(0x5F);
            mouse_send_data_pc(0x00);
        }
        if (dat == 0x97)
        {
            mouse_send_data_pc(0x5F);
            mouse_send_data_pc(0x00);
            mouse_send_data_pc(0x40);
            mouse_send_data_pc(0x26);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่มีอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else mouse_send_data_pc(0x5F);
}
}
void key_receive_from_pc() interrupt 2          /*interrupt2
from INT1 P3^3*/
{
    if (SDA_key == 0 && SCL_key == 0)
    {
        while (SCL_key == 0);
        Clock_high();                          /*send start bit*/
        Clock_low();
        for (i=0;i<8;i++)
        {
            delay25();                          /**sample data im middle
cell***/
            inbit = SDA_key;
            dat = dat<<1;
            dat = dat | inbit;
            Clock_high();
            Clock_low();
        }
        Clock_high();                          /*send parity bit*/
        Clock_low();
        Clock_high();                          /*send stop bit*/
        SDA_key = 0;                            /*ACK*/
        Clock_high();
        Clock_low();
        SDA_key = 1;
        SCL_key = 1;
        if (dat == 0x4F)
        {
            key_send_data_pc(0x5F);
            key_send_data_pc(0xD5);
        }
        else key_send_data_pc(0x5F);
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN5414, SN54LS14, SN7414, SN74LS14 HEX SCHMITT-TRIGGER INVERTERS

SDLS049B – DECEMBER 1983 – REVISED FEBRUARY 2002

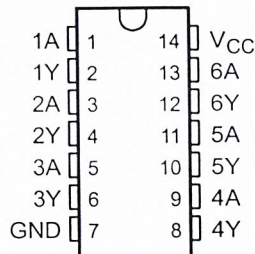
- Operation From Very Slow Edges
- Improved Line-Receiving Characteristics
- High Noise Immunity

description

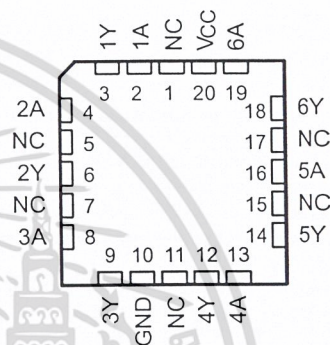
Each circuit functions as an inverter, but because of the Schmitt action, it has different input threshold levels for positive-going (V_{T+}) and negative-going (V_{T-}) signals.

These circuits are temperature compensated and can be triggered from the slowest of input ramps and still give clean, jitter-free output signals.

SN5414, SN54LS14 . . . J OR W PACKAGE
SN7414 . . . D, N, OR NS PACKAGE
SN74LS14 . . . D, DB, OR N PACKAGE
(TOP VIEW)



SN54LS14 . . . FK PACKAGE
(TOP VIEW)



ORDERING INFORMATION

TA	PACKAGE†	ORDERABLE PART NUMBER	TOP-SIDE MARKING	
0°C to 70°C	PDIP – N	Tube	SN7414N	
		Tube	SN74LS14N	
	SOIC – D	Tube	SN7414D	7414
		Tape and reel	SN7414DR	
		Tube	SN74LS14D	LS14
		Tape and reel	SN74LS14DR	
	SOP – NS	Tape and reel	SN7414NSR	SN7414
SSOP – DB	Tape and reel	SN74LS14DBR	LS14	
–55°C to 125°C	CDIP – J	Tube	SN5414J	SN5414J
		Tube	SNJ5414J	SNJ5414J
		Tube	SN54LS14J	SN54LS14J
		Tube	SNJ54LS14J	SNJ54LS14J
	CFP – W	Tube	SNJ5414W	SNJ5414W
		Tube	SNJ54LS14W	SNJ54LS14W
	LCCC – FK	Tube	SNJ54LS14FK	SNJ54LS14FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

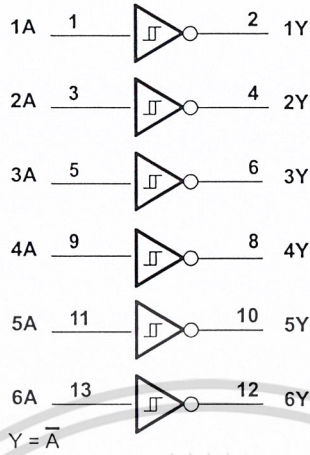
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN5414, SN54LS14,
SN7414, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS
SDLS049B – DECEMBER 1983 – REVISED FEBRUARY 2002

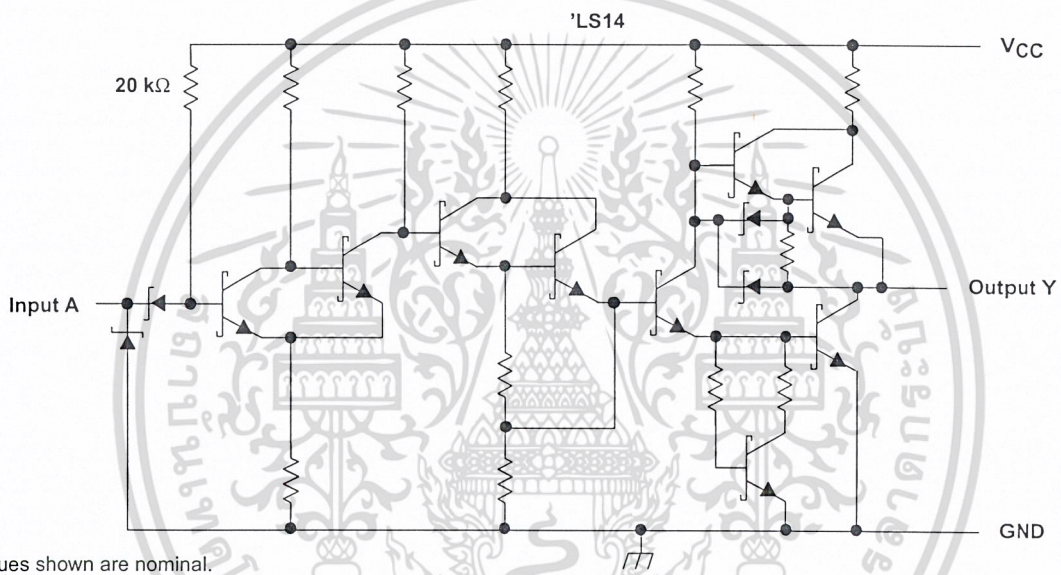
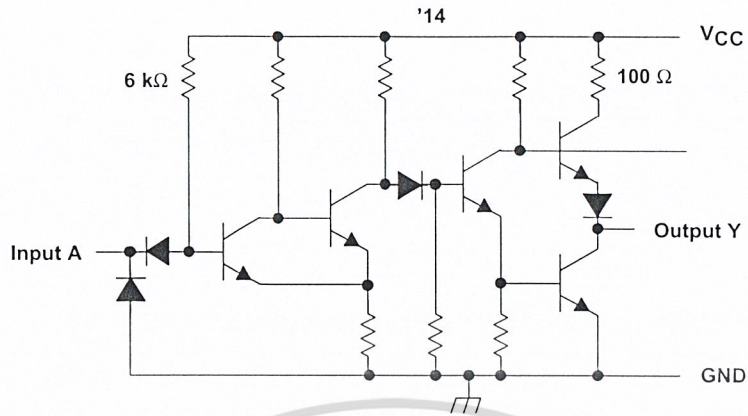
logic diagram (positive logic)



Pin numbers shown are for the D, DB, J, N, NS, and W packages.



schematic



Resistor values shown are nominal.

**SN5414, SN54LS14,
SN7414, SN74LS14
HEX SCHMITT-TRIGGER INVERTERS**
SDLS049B – DECEMBER 1983 – REVISED FEBRUARY 2002

absolute maximum ratings over operating free-air temperature (unless otherwise noted)†

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage: '14	5.5 V
'LS14	7 V
Package thermal impedance, θ_{JA} (see Note 2): D package	86°C/W
DB package	96°C/W
N package	80°C/W
NS package	76°C/W
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. Voltage values are with respect to network ground terminal.
2. The package thermal impedance is calculated in accordance with JESD 51-7

recommended operating conditions

	SN5414			SN7414			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH} High-level output current			-0.8			-0.8	mA
I_{OL} Low-level output current			16			16	mA
T_A Operating free-air temperature	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS‡	SN5414 SN7414			UNIT
		MIN	TYP§	MAX	
V_{T+}	$V_{CC} = 5\text{ V}$	1.5	1.7	2	V
V_{T-}	$V_{CC} = 5\text{ V}$	0.6	0.9	1.1	V
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = 5\text{ V}$	0.4	0.8		V
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -12\text{ mA}$			-1.5	V
V_{OH}	$V_{CC} = \text{MIN}$, $V_I = 0.6\text{ V}$, $I_{OH} = -0.8\text{ mA}$	2.4	3.4		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_I = 2\text{ V}$, $I_{OL} = 16\text{ mA}$		0.2	0.4	V
I_{T+}	$V_{CC} = 5\text{ V}$, $V_I = V_{T+}$	-0.43			mA
I_{T-}	$V_{CC} = 5\text{ V}$, $V_I = V_{T-}$	-0.56			mA
I_I	$V_{CC} = \text{MAX}$, $V_I = 5.5\text{ V}$			1	mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_{IH} = 2.4\text{ V}$			40	µA
I_{IL}	$V_{CC} = \text{MAX}$, $V_{IL} = 0.4\text{ V}$		-0.8	-1.2	mA
$I_{OS}¶$	$V_{CC} = \text{MAX}$	-18		-55	mA
I_{CCH}	$V_{CC} = \text{MAX}$		22	36	mA
I_{CCL}	$V_{CC} = \text{MAX}$		39	60	mA

‡ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

§ All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

¶ Not more than one output should be shorted at a time.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	SN5414 SN7414			UNIT
				MIN	TYP	MAX	
t_{PLH}	A	Y	$R_L = 400\ \Omega$, $C_L = 15\ \text{pF}$		15	22	ns
t_{PHL}					15	22	

recommended operating conditions

		SN54LS14			SN74LS14			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH}	High-level output current			-0.4			-0.4	mA
I_{OL}	Low-level output current			4			8	mA
T_A	Operating free-air temperature	-55		125	0		70	$^\circ\text{C}$

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54LS14			SN74LS14			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{T+}	$V_{CC} = 5\text{ V}$	1.4	1.6	1.9	1.4	1.6	1.9	V
V_{T-}	$V_{CC} = 5\text{ V}$	0.5	0.8	1	0.5	0.8	1	V
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = 5\text{ V}$	0.4	0.8		0.4	0.8		V
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -18\ \text{mA}$			-1.5			-1.5	V
V_{OH}	$V_{CC} = \text{MIN}$, $V_I = 0.5\text{ V}$, $I_{OH} = -0.4\ \text{mA}$	2.5	3.4		2.7	3.4		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_I = -1.9\text{ V}$, $I_{OL} = 4\ \text{mA}$		0.25	0.4		0.25	0.4	V
						0.35	0.5	
I_{T+}	$V_{CC} = 5\text{ V}$, $V_I = V_{T+}$		-0.14			-0.14		mA
I_{T-}	$V_{CC} = 5\text{ V}$, $V_I = V_{T-}$		-0.18			-0.18		mA
I_I	$V_{CC} = \text{MAX}$, $V_I = 7\text{ V}$			0.1			0.1	mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_{IH} = 2.7\text{ V}$			20			20	μA
I_{IL}	$V_{CC} = \text{MAX}$, $V_{IL} = 0.4\text{ V}$			-0.4			-0.4	mA
I_{OS}^{\S}	$V_{CC} = \text{MAX}$	-20		-100	-20		-100	mA
I_{CCH}	$V_{CC} = \text{MAX}$		8.6	16		8.6	16	mA
I_{CCL}	$V_{CC} = \text{MAX}$		12	21		12	21	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ (see Figure 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 2\ \text{k}\Omega$, $C_L = 15\ \text{pF}$		15	22	ns
t_{PHL}					15	22	



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM341/LM78MXX Series 3-Terminal Positive Voltage Regulators

General Description

The LM341 and LM78MXX series of three-terminal positive voltage regulators employ built-in current limiting, thermal shutdown, and safe-operating area protection which makes them virtually immune to damage from output overloads.

With adequate heatsinking, they can deliver in excess of 0.5A output current. Typical applications would include local (on-card) regulators which can eliminate the noise and degraded performance associated with single-point regulation.

Features

- Output current in excess of 0.5A
- No external components
- Internal thermal overload protection
- Internal short circuit current-limiting
- Output transistor safe-area compensation
- Available in TO-220, TO-39, and TO-252 D-PAK packages
- Output voltages of 5V, 12V, and 15V

Connection Diagrams

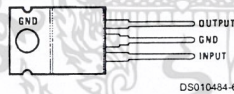
TO-39 Metal Can Package (H)



Bottom View

Order Number LM78M05CH, LM78M12CH or LM78M15CH
See NS Package Number H03A

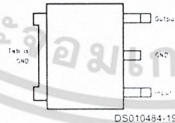
TO-220 Power Package (T)



Top View

Order Number LM341T-5.0, LM341T-12, LM341T-15, LM78M05CT, LM78M12CT or LM78M15CT
See NS Package Number T03B

TO-252



Top View

Order Number LM78M05CDT
See NS Package Number TD03B

Absolute Maximum Ratings (Note 1)

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

Lead Temperature (Soldering, 10 seconds)

TO-39 Package (H)	300°C
TO-220 Package (T)	260°C

Storage Temperature Range -65°C to +150°C

Operating Junction Temperature Range -40°C to +125°C

Power Dissipation (Note 2) Internally Limited

Input Voltage 5V ≤ V_O ≤ 15V 35V

ESD Susceptibility TBD

Electrical Characteristics

Limits in standard typeface are for T_J = 25°C, and limits in **boldface type** apply over the -40°C to +125°C operating temperature range. Limits are guaranteed by production testing or correlation techniques using standard Statistical Quality Control (SQC) methods.

LM341-5.0, LM78M05C

Unless otherwise specified: V_{IN} = 10V, C_{IN} = 0.33 μF, C_O = 0.1 μF

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V _O	Output Voltage	I _L = 500 mA	4.8	5.0	5.2	V
		5 mA ≤ I _L ≤ 500 mA	4.75	5.0	5.25	
		P _O ≤ 7.5W, 7.5V ≤ V _{IN} ≤ 20V				
V _{R LINE}	Line Regulation	7.2V ≤ V _{IN} ≤ 25V			50	mV
		I _L = 100 mA			100	
V _{R LOAD}	Load Regulation	5 mA ≤ I _L ≤ 500 mA			100	
I _Q	Quiescent Current	I _L = 500 mA		4	10.0	mA
ΔI _Q	Quiescent Current Change	5 mA ≤ I _L ≤ 500 mA			0.5	
		7.5V ≤ V _{IN} ≤ 25V, I _L = 500 mA			1.0	
V _n	Output Noise Voltage	f = 10 Hz to 100 kHz		40		μV
$\frac{\Delta V_{IN}}{\Delta V_O}$	Ripple Rejection	f = 120 Hz, I _L = 500 mA		78		dB
V _{IN}	Input Voltage Required to Maintain Line Regulation	I _L = 500 mA	7.2			V
ΔV _O	Long Term Stability	I _L = 500 mA			20	mV/khrs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

Limits in standard typeface are for $T_J = 25^\circ\text{C}$, and limits in **boldface type** apply over the -40°C to $+125^\circ\text{C}$ operating temperature range. Limits are guaranteed by production testing or correlation techniques using standard Statistical Quality Control (SQC) methods. (Continued)

LM341-12, LM78M12C

Unless otherwise specified: $V_{IN} = 19\text{V}$, $C_{IN} = 0.33\ \mu\text{F}$, $C_O = 0.1\ \mu\text{F}$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_O	Output Voltage	$I_L = 500\ \text{mA}$	11.5	12	12.5	V
		$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$	11.4	12	12.6	
		$P_D \leq 7.5\text{W}$, $14.8\text{V} \leq V_{IN} \leq 27\text{V}$				
$V_{R\ \text{LINE}}$	Line Regulation	$14.5\text{V} \leq V_{IN} \leq 30\text{V}$	$I_L = 100\ \text{mA}$		120	mV
			$I_L = 500\ \text{mA}$		240	
$V_{R\ \text{LOAD}}$	Load Regulation	$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$			240	
I_Q	Quiescent Current	$I_L = 500\ \text{mA}$		4	10.0	mA
ΔI_Q	Quiescent Current Change	$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$			0.5	
		$14.8\text{V} \leq V_{IN} \leq 30\text{V}$, $I_L = 500\ \text{mA}$			1.0	
V_n	Output Noise Voltage	$f = 10\ \text{Hz to } 100\ \text{kHz}$		75		μV
$\frac{\Delta V_{IN}}{\Delta V_O}$	Ripple Rejection	$f = 120\ \text{Hz}$, $I_L = 500\ \text{mA}$		71		dB
V_{IN}	Input Voltage Required to Maintain Line Regulation	$I_L = 500\ \text{mA}$	14.5			V
ΔV_O	Long Term Stability	$I_L = 500\ \text{mA}$			48	mV/khrs

LM341-15, LM78M15C

Unless otherwise specified: $V_{IN} = 23\text{V}$, $C_{IN} = 0.33\ \mu\text{F}$, $C_O = 0.1\ \mu\text{F}$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_O	Output Voltage	$I_L = 500\ \text{mA}$	14.4	15	15.6	V
		$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$	14.25	15	15.75	
		$P_D \leq 7.5\text{W}$, $18\text{V} \leq V_{IN} \leq 30\text{V}$				
$V_{R\ \text{LINE}}$	Line Regulation	$17.6\text{V} \leq V_{IN} \leq 30\text{V}$	$I_L = 100\ \text{mA}$		150	mV
			$I_L = 500\ \text{mA}$		300	
$V_{R\ \text{LOAD}}$	Load Regulation	$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$			300	
I_Q	Quiescent Current	$I_L = 500\ \text{mA}$		4	10.0	mA
ΔI_Q	Quiescent Current Change	$5\ \text{mA} \leq I_L \leq 500\ \text{mA}$			0.5	
		$18\text{V} \leq V_{IN} \leq 30\text{V}$, $I_L = 500\ \text{mA}$			1.0	
V_n	Output Noise Voltage	$f = 10\ \text{Hz to } 100\ \text{kHz}$		90		μV
$\frac{\Delta V_{IN}}{\Delta V_O}$	Ripple Rejection	$f = 120\ \text{Hz}$, $I_L = 500\ \text{mA}$		69		dB
V_{IN}	Input Voltage Required to Maintain Line Regulation	$I_L = 500\ \text{mA}$	17.6			V
ΔV_O	Long Term Stability	$I_L = 500\ \text{mA}$			60	mV/khrs

Note 1: Absolute maximum ratings indicate limits beyond which damage to the component may occur. Electrical specifications do not apply when operating the device outside of its rated operating conditions.

Note 2: The typical thermal resistance of the three package types is:

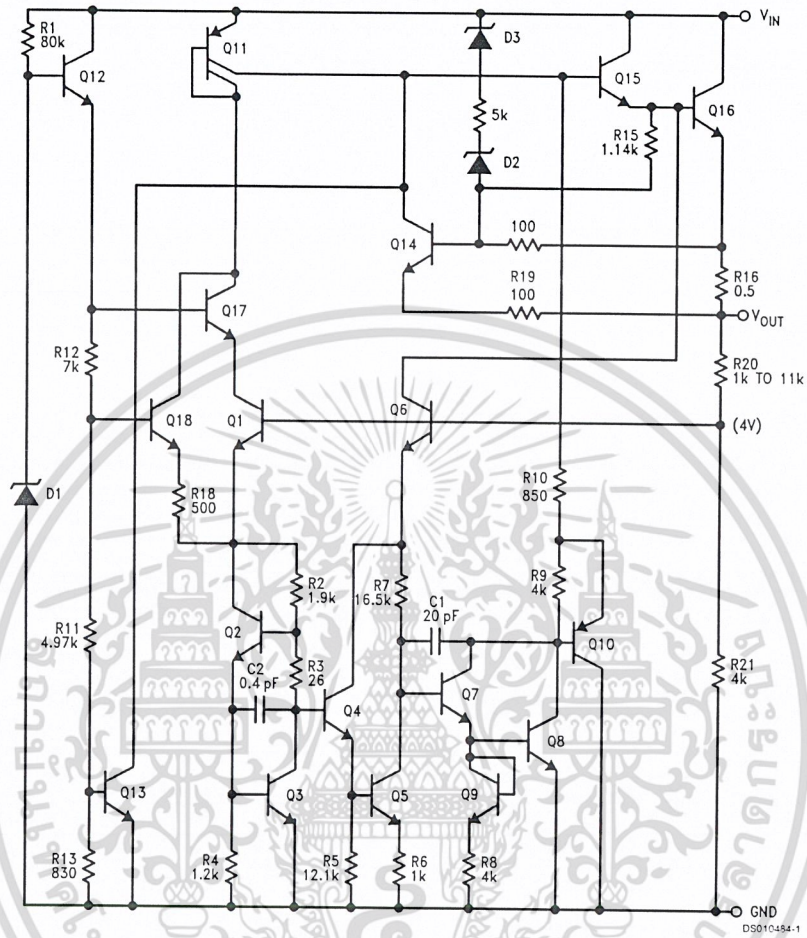
T (TO-220) package: $\theta_{(JA)} = 60^\circ\text{C/W}$, $\theta_{(JC)} = 5^\circ\text{C/W}$

H (TO-39) package: $\theta_{(JA)} = 120^\circ\text{C/W}$, $\theta_{(JC)} = 18^\circ\text{C/W}$

DT (TO-252) package: $\theta_{(JA)} = 92^\circ\text{C/W}$, $\theta_{(JC)} = 10^\circ\text{C/W}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

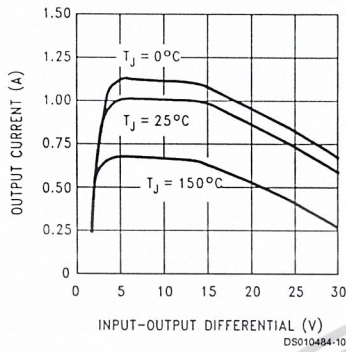
Schematic Diagram



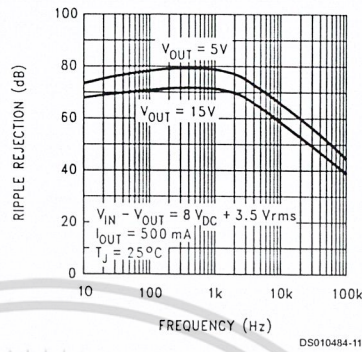
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics

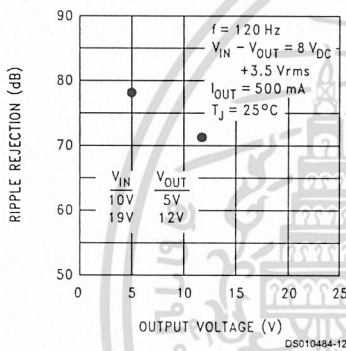
Peak Output Current



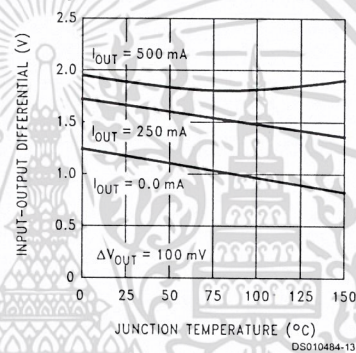
Ripple Rejection



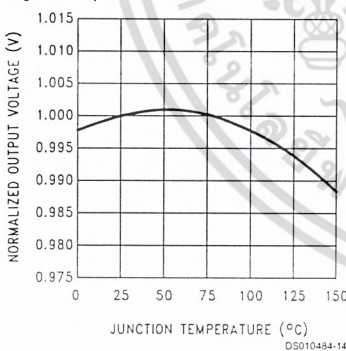
Ripple Rejection



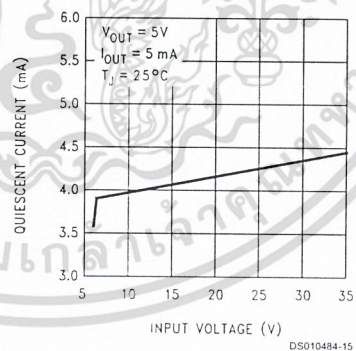
Dropout Voltage



Output Voltage (Normalized to 1V at $T_J = 25^\circ\text{C}$)



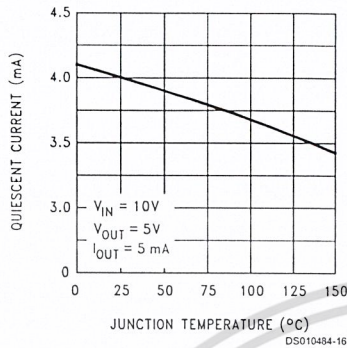
Quiescent Current



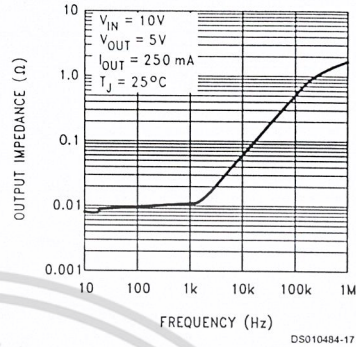
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)

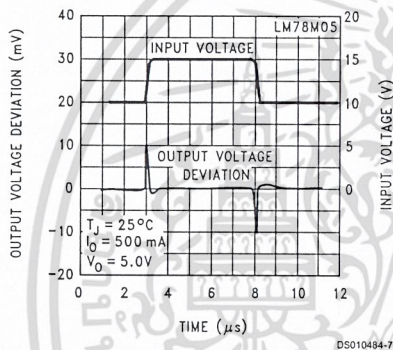
Quiescent Current



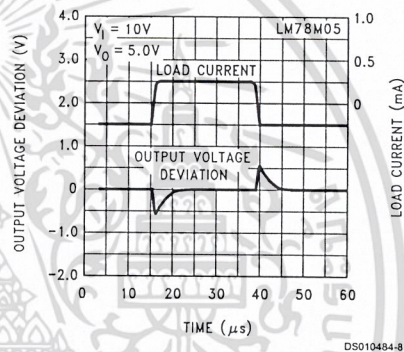
Output Impedance



Line Transient Response



Load Transient Response



Design Considerations

The LM78MXX/LM341XX fixed voltage regulator series has built-in thermal overload protection which prevents the device from being damaged due to excessive junction temperature.

The regulators also contain internal short-circuit protection which limits the maximum output current, and safe-area protection for the pass transistor which reduces the short-circuit current as the voltage across the pass transistor is increased.

Although the internal power dissipation is automatically limited, the maximum junction temperature of the device must be kept below +125°C in order to meet data sheet specifications. An adequate heatsink should be provided to assure this limit is not exceeded under worst-case operating conditions (maximum input voltage and load current) if reliable performance is to be obtained.

1.0 Heatsink Considerations

When an integrated circuit operates with appreciable current, its junction temperature is elevated. It is important to quantify its thermal limits in order to achieve acceptable performance and reliability. This limit is determined by summing the individual parts consisting of a series of temperature rises from the semiconductor junction to the operating environment. A one-dimension steady-state model of conduction heat transfer is demonstrated in The heat generated at the

device junction flows through the die to the die attach pad, through the lead frame to the surrounding case material, to the printed circuit board, and eventually to the ambient environment. Below is a list of variables that may affect the thermal resistance and in turn the need for a heatsink.

R^{jC} (Component Variables) R^{jCA} (Application Variables)

Leadframe Size & Material Mounting Pad Size, Material, & Location

No. of Conduction Pins Placement of Mounting Pad

Die Size PCB Size & Material

Die Attach Material Traces Length & Width

Molding Compound Size and Material Adjacent Heat Sources

Volume of Air

Air Flow

Ambient Temperature

Shape of Mounting Pad

Design Considerations (Continued)

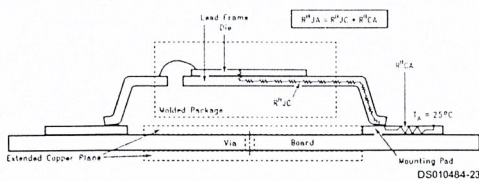


FIGURE 1. Cross-sectional view of Integrated Circuit Mounted on a printed circuit board. Note that the case temperature is measured at the point where the leads contact with the mounting pad surface

The LM78MXX/LM341XX regulators have internal thermal shutdown to protect the device from over-heating. Under all possible operating conditions, the junction temperature of the LM78MXX/LM341XX must be within the range of 0°C to 125°C. A heatsink may be required depending on the maximum power dissipation and maximum ambient temperature of the application. To determine if a heatsink is needed, the power dissipated by the regulator, P_D , must be calculated:

$$I_{IN} = I_L + I_G$$

$$P_D = (V_{IN} - V_{OUT}) I_L + V_{IN} I_G$$

shows the voltages and currents which are present in the circuit.

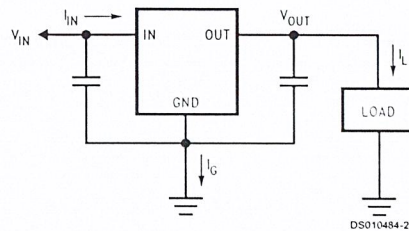


FIGURE 2. Power Dissipation Diagram

The next parameter which must be calculated is the maximum allowable temperature rise, $T_{IK(max)}$:

$$\theta_{JA} = TR(max)/P_D$$

If the maximum allowable value for θ_{JA} °C/W is found to be ≥ 60 °C/W for TO-220 package or ≥ 92 °C/W for TO-252 package, no heatsink is needed since the package alone will dissipate enough heat to satisfy these requirements. If the calculated value for θ_{JA} fall below these limits, a heatsink is required.

As a design aid, Table 1 shows the value of the θ_{JA} of TO-252 for different heatsink area. The copper patterns that we used to measure these θ_{JA} are shown at the end of the Application Note Section. reflects the same test results as what are in the Table 1

shows the maximum allowable power dissipation vs. ambient temperature for the TO-252 device. shows the maximum allowable power dissipation vs. copper area (in²) for the TO-252 device. Please see AN1028 for power enhancement techniques to be used with TO-252 package.

TABLE 1. θ_{JA} Different Heatsink Area

Layout	Copper Area		Thermal Resistance (θ_{JA} , °C/W) TO-252
	Top Side (in ²)*	Bottom Side (in ²)	
1	0.0123	0	103
2	0.066	0	87
3	0.3	0	60
4	0.53	0	54
5	0.76	0	52
6	1	0	47
7	0	0.2	84
8	0	0.4	70
9	0	0.6	63
10	0	0.8	57
11	0	1	57
12	0.066	0.066	89
13	0.175	0.175	72
14	0.284	0.284	61
15	0.392	0.392	55
16	0.5	0.5	53

*Tab of device attached to topside copper

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Design Considerations (Continued)

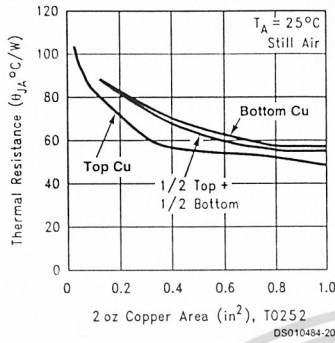


FIGURE 3. θ_{JA} vs. 2oz Copper Area for TO-252

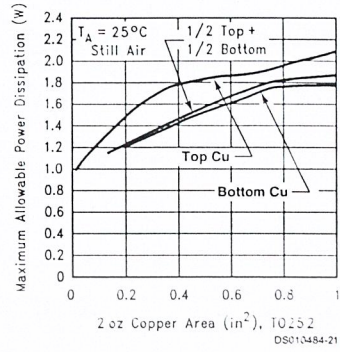


FIGURE 5. Maximum Allowable Power Dissipation vs. 2oz. Copper Area for TO-252

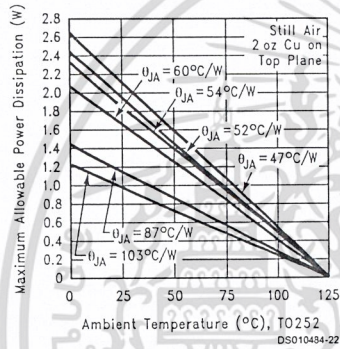
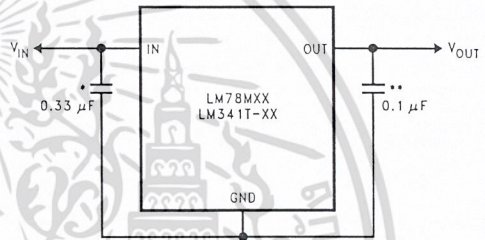


FIGURE 4. Maximum Allowable Power Dissipation vs. Ambient Temperature for TO-252

Typical Application



*Required if regulator input is more than 4 inches from input filter capacitor (or if no input filter capacitor is used).
 **Optional for improved transient response.

DS010484-9

LM79XX Series

3-Terminal Negative Regulators

General Description

The LM79XX series of 3-terminal regulators is available with fixed output voltages of $-5V$, $-12V$, and $-15V$. These devices need only one external component—a compensation capacitor at the output. The LM79XX series is packaged in the TO-220 power package and is capable of supplying 1.5A of output current.

These regulators employ internal current limiting safe area protection and thermal shutdown for protection against virtually all overload conditions.

Low ground pin current of the LM79XX series allows output voltage to be easily boosted above the preset value with a

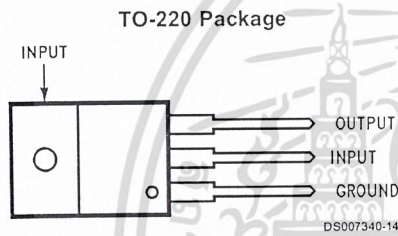
resistor divider. The low quiescent current drain of these devices with a specified maximum change with line and load ensures good regulation in the voltage boosted mode.

For applications requiring other voltages, see LM137 datasheet.

Features

- Thermal, short circuit and safe area protection
- High ripple rejection
- 1.5A output current
- 4% tolerance on preset output voltage

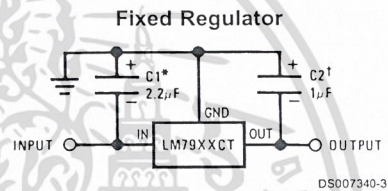
Connection Diagrams



Front View

Order Number LM7905CT, LM7912CT or LM7915CT
See NS Package Number TO3B

Typical Applications



*Required if regulator is separated from filter capacitor by more than 3". For value given, capacitor must be solid tantalum. 25µF aluminum electrolytic may be substituted.

†Required for stability. For value given, capacitor must be solid tantalum. 25µF aluminum electrolytic may be substituted. Values given may be increased without limit.

For output capacitance in excess of 100µF, a high current diode from input to output (1N4001, etc.) will protect the regulator from momentary input shorts.

Absolute Maximum Ratings (Note 1)

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

Input Voltage

$(V_o = -5V)$

$(V_o = -12V \text{ and } -15V)$

-25V

-35V

Input-Output Differential

$(V_o = -5V)$

25V

$(V_o = -12V \text{ and } -15V)$

30V

Power Dissipation (Note 2)

Internally Limited

Operating Junction Temperature Range

0°C to +125°C

Storage Temperature Range

-65°C to +150°C

Lead Temperature (Soldering, 10 sec.)

230°C

Electrical Characteristics

Conditions unless otherwise noted: $I_{OUT} = 500\text{mA}$, $C_{IN} = 2.2\mu\text{F}$, $C_{OUT} = 1\mu\text{F}$, $0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, Power Dissipation $\leq 1.5\text{W}$.

Part Number			LM7905C			Units
Output Voltage			-5V			
Input Voltage (unless otherwise specified)			-10V			
Symbol	Parameter	Conditions	Min	Typ	Max	
V_o	Output Voltage	$T_J = 25^\circ\text{C}$	-4.8	-5.0	-5.2	V
		$5\text{mA} \leq I_{OUT} \leq 1\text{A}$,	-4.75		-5.25	V
		$P \leq 15\text{W}$				V
ΔV_o	Line Regulation	$T_J = 25^\circ\text{C}$, (Note 3)		8	50	mV
						V
				2	15	mV
ΔV_o	Load Regulation	$T_J = 25^\circ\text{C}$, (Note 3)				mV
		$5\text{mA} \leq I_{OUT} \leq 1.5\text{A}$		15	100	mV
		$250\text{mA} \leq I_{OUT} \leq 750\text{mA}$		5	50	mV
I_Q	Quiescent Current	$T_J = 25^\circ\text{C}$		1	2	mA
ΔI_Q	Quiescent Current Change	With Line			0.5	mA
		With Load, $5\text{mA} \leq I_{OUT} \leq 1\text{A}$			0.5	mA
V_n	Output Noise Voltage	$T_A = 25^\circ\text{C}$, $10\text{Hz} \leq f \leq 100\text{Hz}$		125		μV
	Ripple Rejection	$f = 120\text{Hz}$	54	66		dB
	Dropout Voltage	$T_J = 25^\circ\text{C}$, $I_{OUT} = 1\text{A}$		1.1		V
I_{OMAX}	Peak Output Current	$T_J = 25^\circ\text{C}$		2.2		A
	Average Temperature Coefficient of Output Voltage	$I_{OUT} = 5\text{mA}$, $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$		0.4		mV/°C

Electrical Characteristics

Conditions unless otherwise noted: $I_{OUT} = 500\text{mA}$, $C_{IN} = 2.2\mu\text{F}$, $C_{OUT} = 1\mu\text{F}$, $0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, Power Dissipation $\leq 1.5\text{W}$.

Part Number			LM7912C			LM7915C			Units
Output Voltage			-12V			-15V			
Input Voltage (unless otherwise specified)			-19V			-23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	
V_o	Output Voltage	$T_J = 25^\circ\text{C}$	-11.5	-12.0	-12.5	-14.4	-15.0	-15.6	V
		$5\text{mA} \leq I_{OUT} \leq 1\text{A}$,	-11.4		-12.6	-14.25		-15.75	V
		$P \leq 15\text{W}$							V
ΔV_o	Line Regulation	$T_J = 25^\circ\text{C}$, (Note 3)		5	80		5	100	mV
									V
				3	30		3	50	mV
ΔV_o	Load Regulation	$T_J = 25^\circ\text{C}$, (Note 3)							V
									V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

Conditions unless otherwise noted: $I_{OUT} = 500\text{mA}$, $C_{IN} = 2.2\mu\text{F}$, $C_{OUT} = 1\mu\text{F}$, $0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, Power Dissipation $\leq 1.5\text{W}$.

Part Number			LM7912C			LM7915C			Units
Output Voltage			-12V			-15V			
Input Voltage (unless otherwise specified)			-19V			-23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	
		$5\text{mA} \leq I_{OUT} \leq 1.5\text{A}$		15	200		15	200	mV
		$250\text{mA} \leq I_{OUT} \leq 750\text{mA}$		5	75		5	75	mV
I_Q	Quiescent Current	$T_J = 25^\circ\text{C}$		1.5	3		1.5	3	mA
ΔI_Q	Quiescent Current Change	With Line			0.5			0.5	mA
		With Load, $5\text{mA} \leq I_{OUT} \leq 1\text{A}$			0.5			0.5	V
V_n	Output Noise Voltage	$T_A = 25^\circ\text{C}$, $10\text{Hz} \leq f \leq 100\text{Hz}$		300			375		μV
	Ripple Rejection	$f = 120\text{Hz}$	54	70		54	70		dB
				$(-25 \leq V_{IN} \leq -15)$			$(-30 \leq V_{IN} \leq -17.5)$		V
	Dropout Voltage	$T_J = 25^\circ\text{C}$, $I_{OUT} = 1\text{A}$		1.1			1.1		V
I_{OMAX}	Peak Output Current	$T_J = 25^\circ\text{C}$		2.2			2.2		A
	Average Temperature Coefficient of Output Voltage	$I_{OUT} = 5\text{mA}$, $0^\circ\text{C} \leq T_J \leq 100^\circ\text{C}$		-0.8			-1.0		$\text{mV}/^\circ\text{C}$

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee Specific Performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics.

Note 2: Refer to Typical Performance Characteristics and Design Considerations for details.

Note 3: Regulation is measured at a constant junction temperature by pulse testing with a low duty cycle. Changes in output voltage due to heating effects must be taken into account.

Design Considerations

The LM79XX fixed voltage regulator series has thermal overload protection from excessive power dissipation, internal short circuit protection which limits the circuit's maximum current, and output transistor safe-area compensation for reducing the output current as the voltage across the pass transistor is increased.

Although the internal power dissipation is limited, the junction temperature must be kept below the maximum specified temperature (125°C) in order to meet data sheet specifications. To calculate the maximum junction temperature or heat sink required, the following thermal resistance values should be used:

θ_{JA} = Junction-to-Ambient Thermal Resistance

θ_{JC} = Junction-to-Case Thermal Resistance

θ_{CA} = Case-to-Ambient Thermal Resistance

θ_{CS} = Case-to-Heat Sink Thermal Resistance

θ_{SA} = Heat Sink-to-Ambient Thermal Resistance

Package	Typ	Max	Typ	Max
	θ_{JC} °C/W	θ_{JC} °C/W	θ_{JA} °C/W	θ_{JA} °C/W
TO-220	3.0	5.0	60	40

$$P_{D\text{MAX}} = \frac{T_{J\text{MAX}} - T_A}{\theta_{JC} + \theta_{CA}} \text{ or } \frac{T_{J\text{MAX}} - T_A}{\theta_{JA}}$$

$$\theta_{CA} = \theta_{CS} + \theta_{SA} \text{ (without heat sink)}$$

Solving for T_J :

$$T_J = T_A + P_D (\theta_{JC} + \theta_{CA}) \text{ or } \\ = T_A + P_D \theta_{JA} \text{ (without heat sink)}$$

Where:

T_J = Junction Temperature

T_A = Ambient Temperature

P_D = Power Dissipation

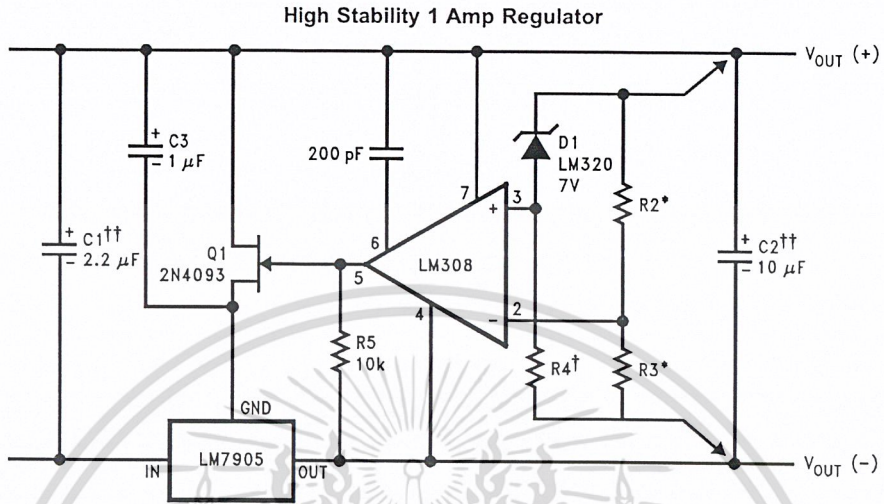
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications

Bypass capacitors are necessary for stable operation of the LM79XX series of regulators over the input voltage and output current ranges. Output bypass capacitors will improve the transient response by the regulator.

The bypass capacitors, (2.2μF on the input, 1.0μF on the output) should be ceramic or solid tantalum which have good

high frequency characteristics. If aluminum electrolytics are used, their values should be 10μF or larger. The bypass capacitors should be mounted with the shortest leads, and if possible, directly across the regulator terminals.



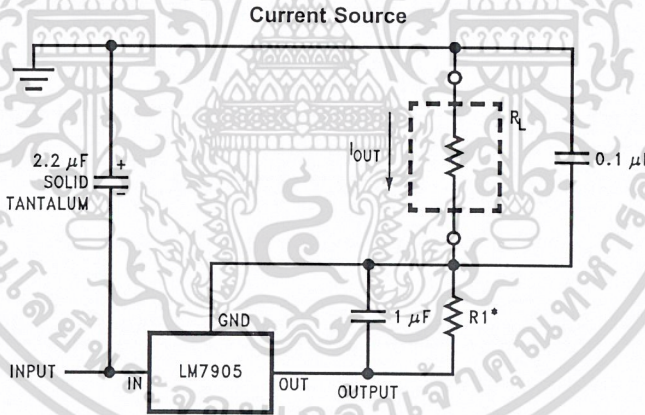
DS007340-5

Load and line regulation < 0.01% temperature stability ≤ 0.2%

†Determine Zener current

††Solid tantalum

*Select resistors to set output voltage. 2 ppm/°C tracking suggested

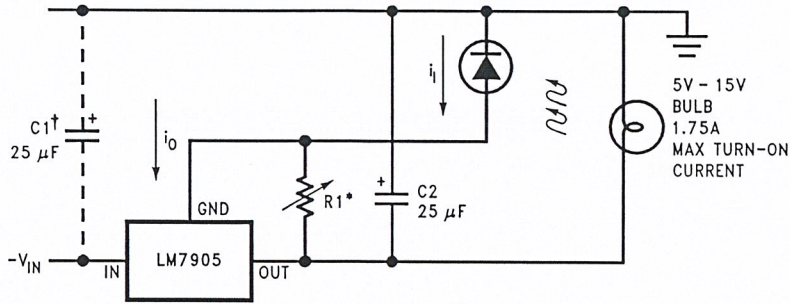


DS007340-7

$$I_{OUT} = 1 \text{ mA} + \frac{5V}{R1}$$

Typical Applications (Continued)

Light Controller Using Silicon Photo Cell

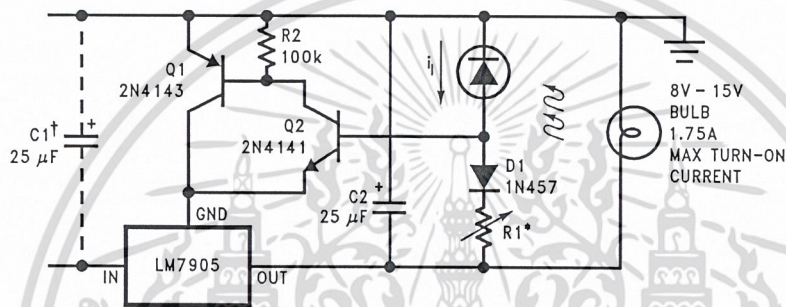


DS007340-8

*Lamp brightness increase until $i_1 = i_0 (= 1 \text{ mA}) + 5\text{V}/R_1$.

†Necessary only if raw supply filter capacitor is more than 2" from LM7905CT

High-Sensitivity Light Controller

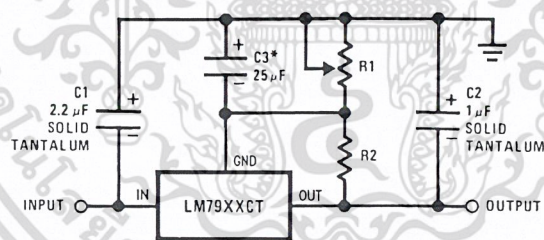


DS007340-9

*Lamp brightness increases until $i_1 = 5\text{V}/R_1$ (i_1 can be set as low as 1 μA)

†Necessary only if raw supply filter capacitor is more than 2" from LM7905

Variable Output



DS007340-2

*Improves transient response and ripple rejection. Do not increase beyond 50 μF .

$$V_{\text{OUT}} = V_{\text{SET}} \left(\frac{R_1 + R_2}{R_2} \right)$$

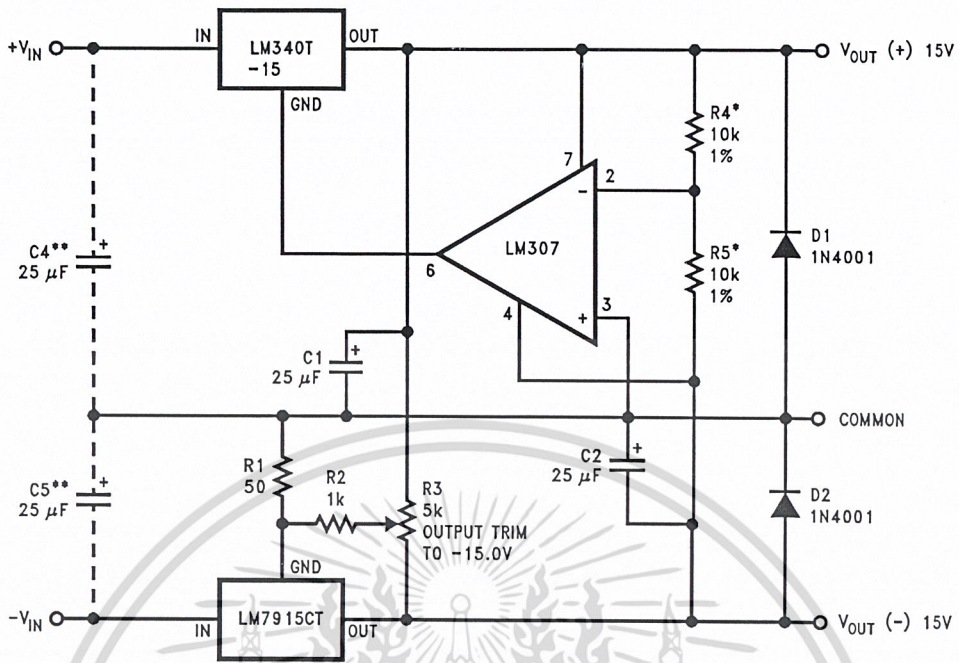
Select R2 as follows:

LM7905CT	300 Ω
LM7912CT	750 Ω
LM7915CT	1k

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

±15V, 1 Amp Tracking Regulators

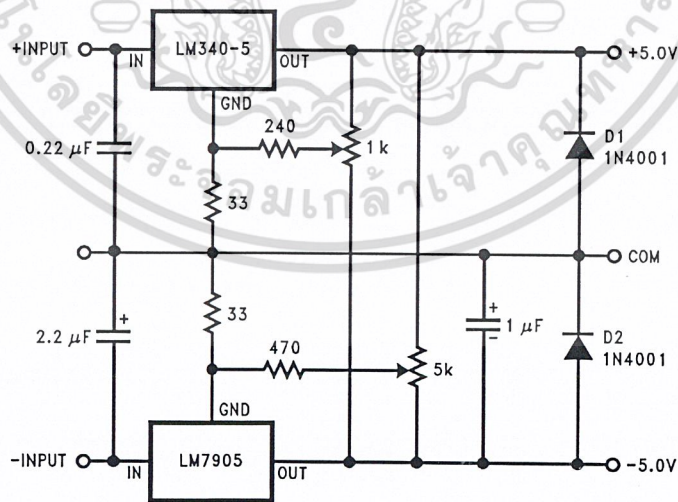


DS007340-1

	(-15)	(+15)
Load Regulation at $\Delta I_L = 1A$	40mV	2mV
Output Ripple, $C_{IN} = 3000\mu F, I_L = 1A$	100 μVms	100 μVms
Temperature Stability	50mV	50mV
Output Noise $10Hz \leq f \leq 10kHz$	150 μVms	150 μVms

*Resistor tolerance of R4 and R5 determine matching of (+) and (-) outputs.
 **Necessary only if raw supply filter capacitors are more than 3" from regulators.

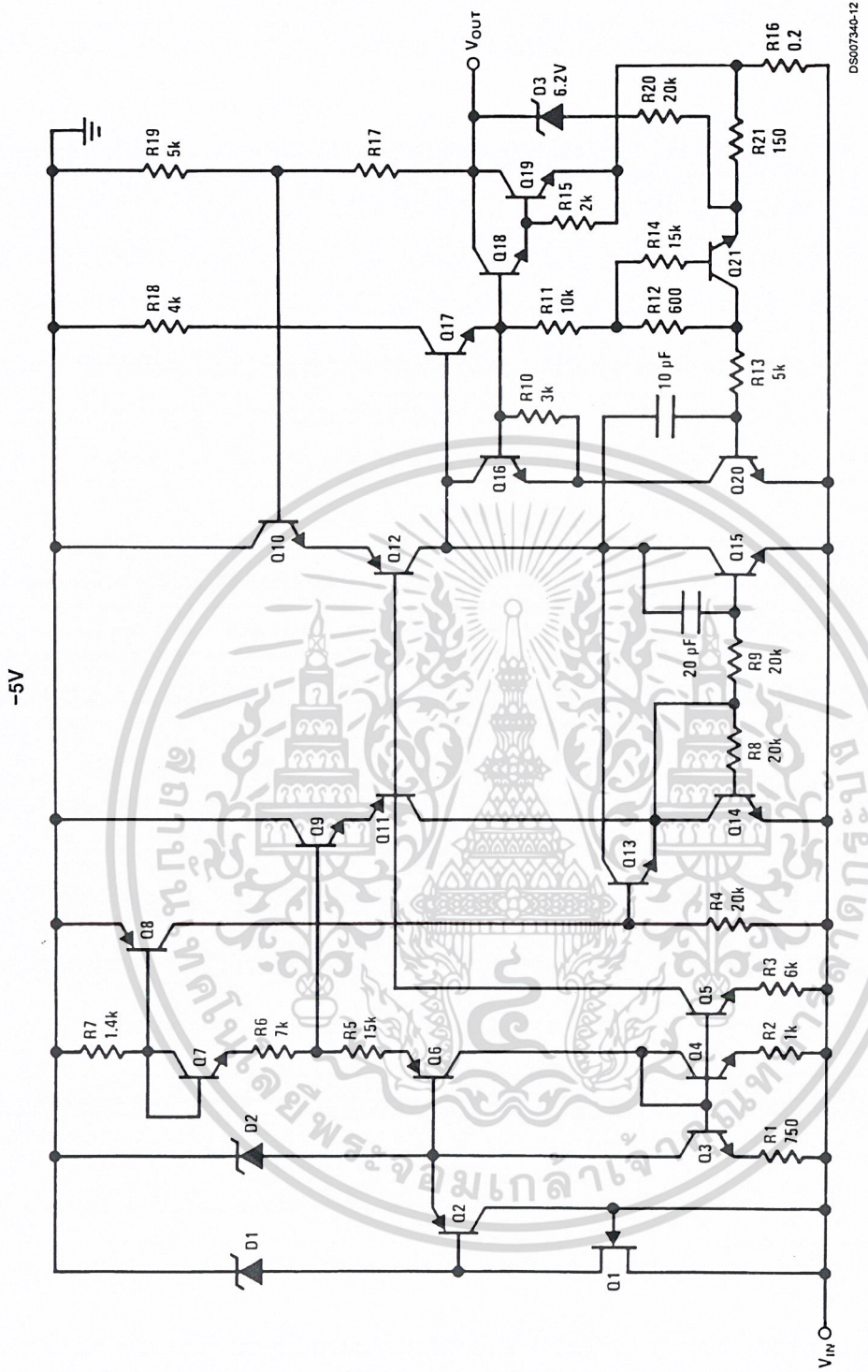
Dual Trimmed Supply



DS007340-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schematic Diagrams



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการนี้ประสบความสำเร็จไปได้ด้วยดีเนื่องจากได้รับการแนะนำ และช่วยเหลือจากอาจารย์ที่ปรึกษาโครงการ ดร.พิพัฒน์ พรหมมี อีกทั้งเพื่อน ๆ ทุกคนที่ให้กำลังใจเรื่อยมา ตลอดจน คณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ และพระคุณของบิดามารดาผู้ให้กำเนิด

นาย วรวัฒน์ ลิ้มปคันทร

นาย วิจิตร สีลาอ่อน

นาย วิทวัฑ คงสกุลยานนท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ผศ.ธีรวัฒน์ ประกอบผล , “การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี” ส.ศ.ท. ,2545
- [2] รศ.สมยศ จุณณะปิยะ , “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์” สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546
- [3] <http://www.beyondlogic.org/keyboard/keybrd.htm>
- [4] <http://www.hut.fi/~then/mytexts/mouse.html>
- [5] http://www.simandl.cz/stranky/elektro/keyboard/keyboard_a.htm
- [6] <http://www.networktechinc.com/ps2-prots.html>
- [7] <http://panda.cs.ndsu.nodak.edu/~achapwes/PICmicro/PS2/ps2.htm>
- [8] <http://www.epanorama.net/documents/pc/mouse.html>
- [9] <http://panda.cs.ndsu.nodak.edu/~achapwes/PICmicro/mouse/mouse.html>
- [10] http://www.uni.net.th/~08_2543/chap04.html
- [11] http://www.fareastern.ac.th/acad/bc/pichate/network/net_c2/chapter2.htm
- [12] วิวัฒน์ กิรานนท์ , “ วิศวกรรมการสื่อสาร “ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2544
- [13] สุชาติ กังวารจิตต์ , “เครื่องรับส่งวิทยุและระบบวิทยุสื่อสาร” แผนกวิศวกรรม กองบังคับการตำรวจสื่อสาร กรมตำรวจ , 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้