การ์ดอินเตอร์เฟสสัญญาณโทรศัพท์

TELEPHONE INTERFACE CARD

โดย

นาย สุรชัย วิมลแก้ว

วัน เดือน ปี 1 ต.ค. 2539
เลขทะเบียน 038079
เลขเรียกหนังสือ T 390.99 ส814ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038079

การ์ดอินเตอร์เฟสสัญญาณโทรศัพท์

TELEPHONE INTERFACE CARD

โดย

นาย สุรชัย วิมลแก้ว 36013272

อาจารย์ที่ปรึกษา

อาจารย์ อุทัย ศรีธีระวิโรจน์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2539

ภาควิชาเทคนิคอุตสาหกรรม

สาขาเทคโนโลยีโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การ์ดอินเตอร์เฟสสัญญาณโทรศัพท์ (TELEPHONE INTERFACE CARD)

ผู้จัดทำ นาย สุรชัย วิมลแก้ว 36013272

_____ อาจารย์ที่ปรึกษา

(อ. อุทัย ศรีธีระวิโรจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ์ดอินเตอร์เฟสสัญญาณโทรศัพท์

สุรชัย วิมลแก้ว
อาจารย์ที่ปรึกษา อ.อุทัย ศรีธีระวิโรจน์
ปีการศึกษา 2539

บทคัดย่อ

การ์ดอินเตอร์เฟสสัญญาณโทรศัพท์ เป็นโครงงานที่ประดิษฐ์ขึ้นเพื่อใช้ใน การเชื่อมต่อเครื่องคอมพิวเตอร์กับระบบโทรศัพท์ เพื่อให้คอมพิวเตอร์สามารถรับข้อมูล และทำงานโดยการควบคุมทางโทรศัพท์ ซึ่งประกอบด้วย วงจรอินเตอร์เฟสกับคอมพิวเตอร์ ใช้ไอซีเบอร์ 8255 วงจรตรวจสอบสัญญาณเรียกโทรศัพท์ ใช้ไอซี MC34012 และ NE555 วงจรตรวจสอบการวางหูและยกหูใช้ไอซีเบอร์ LM567 และวงจรถอดรหัสสัญญาณ DTMF ใช้ไอซีเบอร์ MT8870 ซึ่งการ์ดอินเตอร์เฟสสัญญาณโทรศัพท์นี้จะทำการรับข้อมูล เพื่อส่งให้คอมพิวเตอร์นำไปประมวลผล ซึ่งควบคุมโดยโปรแกรมที่เขียนขึ้นให้เหมาะสมกับงาน และการ์ดนี้ยังสามารถส่งสัญญาณจากคอมพิวเตอร์ออกมาควบคุมอุปกรณ์ภายนอกได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TELEPHONE INTERFACE CARD

Mr. Surachai Wimonkaew

Advisor : Uthai Sriteraviroj

1996

Abstract

Telephone interface card is project. which created to using for interface between micro-computer and telephone. Computer can be received data and controled by telephone. Consist of main card, that install on slot of computer to be media in communication between computer and interface card. This project used IC 8255 in Interface part, IC MC34012 and NE 555 in ringing tone checking part, IC LM567 in hook on-off part and IC MT8870 in Decoder part .

Operation of this project can be coltrollable by computer program which checking data form Telephone interface card then take data to process and present output data to output device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1	บทนำ	1
บทที่ 2	ระบบโทรศัพท์	3
	2.1 ระบบโทรศัพท์แบบส่งสัญญาณพัลซ์	3
	2.2 ระบบโทรศัพท์แบบส่งสัญญาณคู่	4
	2.2.1 ระบบสัญญาณ DTMF	5
	2.2.2 ข้อดีของระบบ DTMF	7
บทที่ 3	หลักการทำงานและการออกแบบส่วนต่างๆ	9
	3.1 ส่วนตรวจจับสัญญาณกระดิ่ง	9
	3.1.1 หลักการทำงานของ IC MC34012	9
	3.1.2 หลักการทำงานของ Optocouple	10
	3.1.3 ส่วนประกอบของ IC LM567	11
	3.2 ส่วนโมโนสเตเบิล	13
	3.2.1 การทำงานของ IC NE555	13
	3.3 วงจรแปลงความถี่ของระบบโทน	15
	3.3.1 รายละเอียดและหน้าที่การทำงาน ของ MT8870	16
	3.3.2 ขั้นตอนการทำงาน ของ MT8870	17
	3.3.3 อธิบายคำศัพท์	17
บทที่ 4	หลักการทำงานและออกแบบ Interface Card โดยใช้ IC 8255	20
	4.1 การใช้งาน 8255	20
	4.1.1 รายละเอียดขาต่างๆ ของ 8255	20
	4.1.2 รีจีสเตอร์ภายใน 8255	21
	4.1.3 การกำหนดโหมดการทำงาน	22
	4.2 รายละเอียดเกี่ยวกับ Slot มาตรฐานของ IBM/PC	24
	4.3 การใช้งาน รีเลย์	27
	4.3.1 คุณสมบัติของรีเลย์	28
บทที่ 5	สรุปและวิจารณ์ผลการทดลอง	30
บรรณานุกรม		
กิตติกรรมประกาศ		
ภาคผนวก		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

ปัจจุบัน เทคโนโลยีการสื่อสารและคอมพิวเตอร์ได้เข้ามามีบทบาท ในชีวิตประจำวันมากขึ้น ไม่ว่าจะเป็นทางด้านการศึกษา ทางด้านธุรกิจการค้าขาย ทางอุตสาหกรรม และโดยเฉพาะองค์การต่าง ๆ ตั้งแต่ขนาดเล็กไปจนถึงองค์การระหว่างประเทศ จะต้องอาศัยอุปกรณ์การสื่อสาร เช่น โทรศัพท์ โทรเลข โทรสาร และคอมพิวเตอร์ เพื่อใช้ในการติดต่อสื่อสาร

ซึ่งมองในจุดหลักก็คือ การใช้งานระบบโทรศัพท์ เพื่อประโยชน์ ในการติดต่อสื่อสารกัน ในระยะไกลแทนการเดินทางไป ซึ่งนอกจากจะลดภาระในการเดินทาง เวลา ค่าใช้จ่าย แล้วยังเพิ่มความรวดเร็ว และสะดวกสบายได้มากขึ้น

ส่วนการใช้งานคอมพิวเตอร์ก็ เพื่อประโยชน์ในการป้อนเก็บข้อมูล ประมวลผลข้อมูล นอกจากที่กล่าวมาแล้ว สิ่งที่เห็นได้ชัด ก็คือ "การค้นหาข้อมูล" และ การนำคอมพิวเตอร์ไปควบคุมระบบต่าง ๆ

จากประโยชน์ที่กล่าวมาจึงเป็นที่มาของโครงงานนี้ โดยระบบของโครงงานจะมีลำดับการทำงานหลัก ๆ คือ

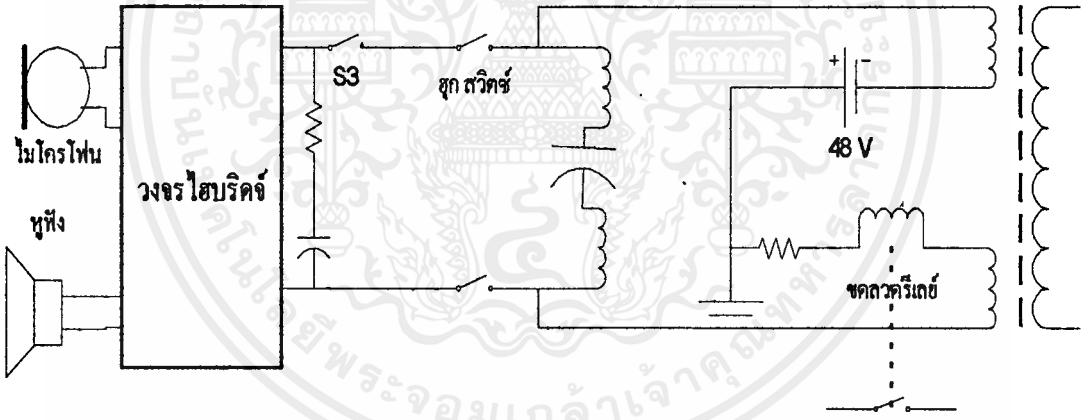
1. ตรวจสอบการโทรเข้าและวางสายของผู้ใช้
2. ถอดรหัสสัญญาณโทรศัพท์
3. เชื่อมต่อสัญญาณเข้ากับคอมพิวเตอร์
4. โปรแกรมนำข้อมูลที่ได้มาประมวลผล
5. คอมพิวเตอร์แสดงผล อาจจะแสดงออกโดย
    - แสดงออกทางจอภาพ
    - ส่งสัญญาณออกมาควบคุมอุปกรณ์
    - ส่งสัญญาณเสียงออกทางลำโพง
    - ส่งงานพิมพ์ทาง Printer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบโทรศัพท์

2.1 ระบบโทรศัพท์แบบส่งสัญญาณพัลซ์

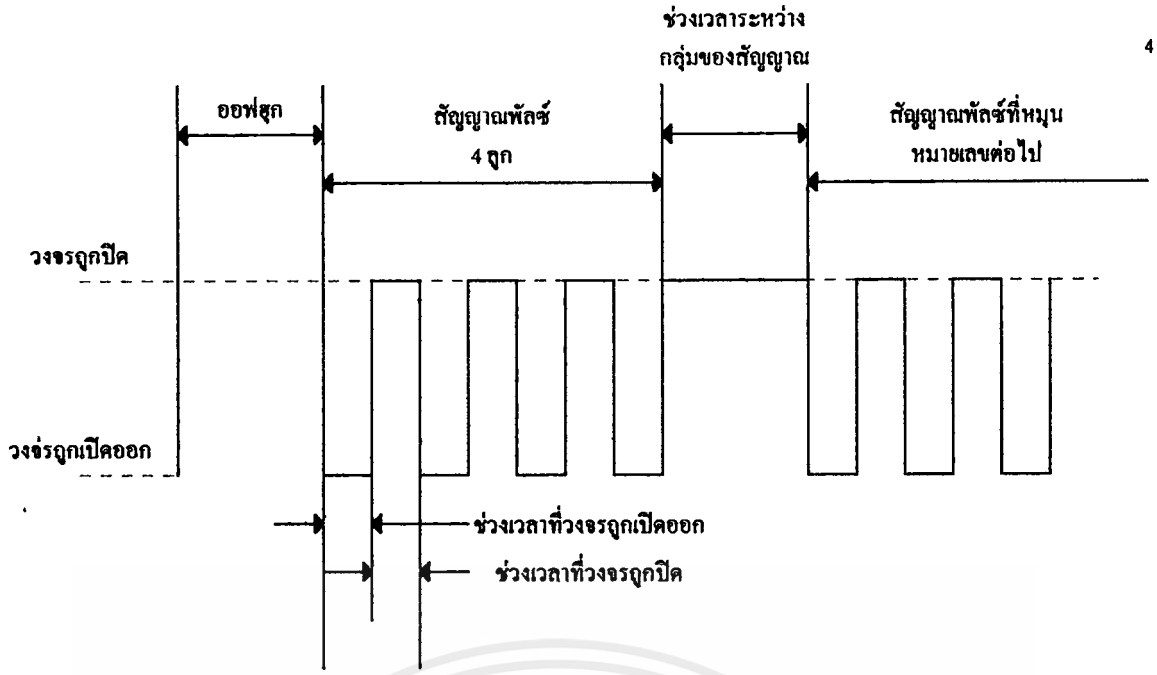
จากรูปที่ 2.1 เป็นวงจรที่ใช้ส่งหมายเลขโทรศัพท์ในแบบพัลซ์ เห็นได้ว่าสวิตซ์ S3 จะถูกเปิดวงจรออกเมื่อมีการหมุนหมายเลขโทรศัพท์ เมื่อสวิตซ์ S3 ถูกเปิดวงจรออกก็จะไม่มีกระแสไหลผ่านเข้าไปในวงจรส่วนที่อยู่ถัดไปได้ จึงเสมือนว่าเป็น การขัดจังหวะ (interruption) การไหลของกระแส สำหรับจำนวนครั้งที่สวิตซ์ S3 ถูกเปิดกระแสออกจะขึ้นอยู่กับระยะห่างของ แป้นหมุด (dialer) ที่ถูกหมุนไป กับตำแหน่งปกติในขณะที่ไม่มีการหมุนหมายเลขใด ๆ เป็นต้นว่า ถ้าหมุนหมายเลข 4 สวิตซ์ S3 ถูกทำให้เปิดออก 4 ครั้ง ซึ่งสวิตซ์ S3 จะถูกเปิดวงจรในช่วงที่ปล่อยให้แป้นหมุนกลับสู่ตำแหน่งเดิมเท่านั้น ไม่ได้เกิดขึ้นในระหว่างที่ทำการหมุนหมายเลขอยู่



รูปที่ 2.1 แสดงวงจรหมุนหมายเลขแบบพัลซ์อย่างง่าย

รูปที่ 2.2 จะแสดงถึงลักษณะของรูปสัญญาณเมื่อมีการหมุนหมายเลขโทรศัพท์ จากรูปนี้จะเห็นว่าในตอนแรก โทรศัพท์อยู่ในสภาวะ On hook คือ หูโทรศัพท์วางอยู่บนที่วางหูโทรศัพท์ตามปกติ ไม่มีกระแสจากชุมสายเข้าสู่โทรศัพท์ แต่เมื่อมีการยกหูโทรศัพท์ขึ้น โทรศัพท์จะอยู่ในสภาวะ off hook ชุดสวิตซ์จะถูกปิดวงจรลง ทำให้มีกระแสไหลครบวงจรได้ และเมื่อมีการหมุนหมายเลข 4 จะทำให้มีการเปิดวงจรออกด้วยสวิตซ์ S3 จำนวน 4 ครั้ง ได้รูปสัญญาณตามที่เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงไดอะแกรมของเวลาคร่าวๆ ของสัญญาณที่เกิดจากการหมุนหมายเลข 4

ในระบบโทรศัพท์แบบที่ส่งสัญญาณแบบพัลซ์นี้ จะถูกกำหนดให้สามารถส่งสัญญาณในอัตรา 10 พัลซ์ต่อวินาที (10 pps, 10 pulse per sec) หรือ 20 พัลซ์ต่อวินาที (20 pps) และ เพื่อความเข้าในที่ตรงกันในการพิจารณาสัญญาณที่เกิดขึ้น จึงควรทราบความหมายต่อไปนี้

- อัตราการส่งสัญญาณพัลซ์ (pulse rate) = จำนวนพัลซ์ที่ถูกส่งออกไปใน 1 วินาที = 1000/คาบเวลาของสัญญาณพัลซ์ (เป็นมิลลิวินาที)
- เปอร์เซ็นต์ของการเปิดวงจร (percent break) = 100 * อัตราส่วนการเปิดวงจร (break ratio) = 100 * ช่วงเวลาที่วงจรถูกเปิดออก/คาบของสัญญาณพัลซ์
- ช่วงเวลาระหว่างกลุ่มของสัญญาณ (interdigit jinterval) ถูกกำหนดให้มีค่าอย่างต่ำ 700 มิลลิวินาที

2.2 ระบบโทรศัพท์แบบส่งสัญญาณความถี่คู่

ระบบโทรศัพท์แบบเก่าที่มีใช้อยู่โดยทั่วไปนั้น ไม่ว่าจะเป็นแบบ Strowger หรือ ระบบ ครอสบาร์ การต่อโทรศัพท์จากผู้เช่าไปยังอีกเลขหมายหนึ่งนั้น ยังอาศัยการหมุนหน้าปัทม์เพื่อทำการส่งพัลซ์ให้สวิตซ์ทำงาน การส่งพัลซ์ยังคงใช้อยู่ทั้งระบบ 10 pps หรือ 20 pps (ในปัจจุบันระบบ 10 pps ไม่ได้รับการรับรองในความถูกต้องระหว่างการส่งข้อมูล)

วงการโทรศัพท์ได้พัฒนาไปอย่างรวดเร็วมาก จากการใช้ระบบ mechanics, relay จนถึงระบบ common control ซึ่งเป็นระบบกึ่ง electronics จนถึงระบบ electronic ฉะนั้นตัวเครื่องโทรศัพท์ก็ได้รับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาตามไปด้วย เพื่อให้ผู้ใช้ได้รับความสะดวกสบายมากยิ่งขึ้น จากเดิมที่เป็นแบบหมุนหน้าปัทม์ ก็เปลี่ยนเป็นการกดปุ่มแทน การกดปุ่มก็เพื่อจะส่งความถี่ออกไป ซึ่งเป็นสัญญาณ multi frequency signal

การที่เราใช้วิธีนี้ก็เพื่อ

- ลดเวลาในการหมุนหน้าปัทม์ลง
- ส่งเลขหมายได้ง่ายขึ้นเพียงการกดปุ่ม
- มีความผิดพลาดในการส่งเลขหมายน้อยลง
- สามารถที่จะเพิ่มปุ่มอื่น ๆ นอกจากที่มีอยู่แล้ว เพื่อใช้งานอย่างอื่นได้ด้วย
- ใช้ความถี่ในระดับคลื่นเสียง

ระบบนี้เป็นระบบการส่งสัญญาณอีกแบบหนึ่ง ซึ่งจะพบได้มากกว่าในระบบการส่งแบบพัลซ์ ระบบนี้มีชื่อว่าย่อว่า DTMF (Dual Tone Multi Frequency) มีวิธีการส่งหมายเลขของผู้ที่ต้องการจะติดต่อโดยการส่งสัญญาณความถี่ 2 ความถี่มอดดูเลทไป ความถี่นี้อยู่ในย่านความถี่เสียงพูด (0-4 กิโลเฮิรตซ์) กลุ่มความถี่ต่ำกว่าจะเป็นของแนว ส่วนกลุ่มความถี่สูงกว่าจะเป็นของแถวค่าต่าง ๆ ของความถี่ได้แสดงไว้ในตารางที่ 2.1

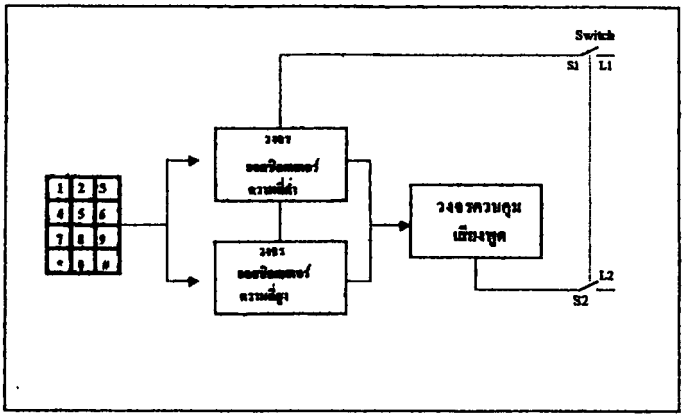
ความถี่ (Hz)	รหัสหรือหมายเลข			
679	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D
ความถี่ (Hz)	1209	1336	1477	1633

ตารางที่ 2.1 แสดงความถี่ต่างๆ ที่ใช้ในระบบ DTMF

2.2.1 <u>ระบบสัญญาณ DTMF</u>

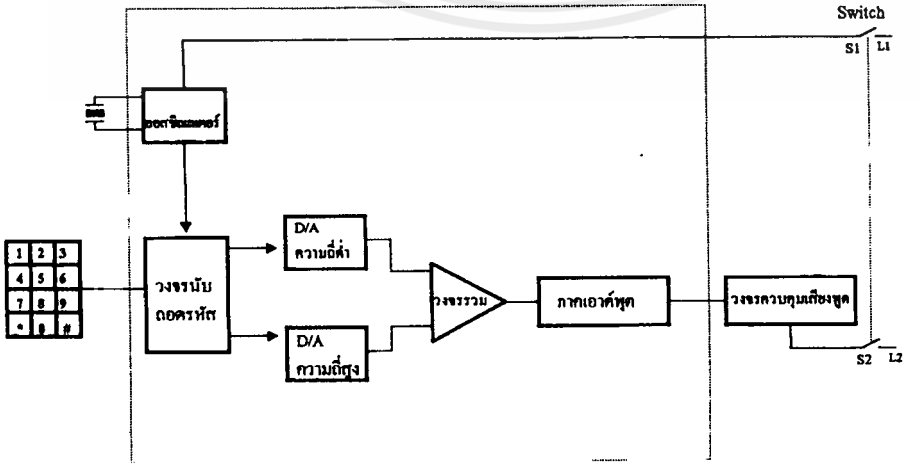
การส่งสัญญาณแบบ DTMF ด้วยการใช้ไอซีสำเร็จรูปในปัจจุบันดูจะเหมาะสมกว่าการนำอุปกรณ์มาต่อกันเพื่อทำการผลิตความถี่ต่าง ๆ ของเลขหมายที่ต้องการจะติดต่อ หลักการของไอซีจำพวกนี้ก็ใช้หลักการเดิม คือ การนำความถี่ที่แตกต่าง 2 ความถี่ที่เกิดจากการกดปุ่มหมายเลขโทรศัพท์มามอดดูเลทกัน แล้วส่งไปยังชุมสายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงบล๊อกไดอะแกรมของระบบ DTMF แบบเก่า

ในรูปที่ 2.3 เป็นบล็อกไดอะแกรมของการส่งสัญญาณแบบ DTMF ซึ่งในระบบนี้ยังคงต้องใช้อุปกรณ์พาสซีฟ (passive elements) ในการวงจรออสซิลเลเตอร์ ซึ่งแน่นอนปัญหาที่จะพบในวงจรที่ใช้อุปกรณ์เหล่านี้ คือ ความคลาดเคลื่อนเนื่องจากสภาวะแวดล้อมที่เปลี่ยนแปลงไปตลอดจนอายุการใช้งาน ซึ่งผลที่ตามมาคือความถี่ที่ผลิตออกมาย่อมเปลี่ยนแปลงความไปด้วยผลสุดท้ายก็จะทำให้ชุมสายเกิดการทำงานผิดพลาดในการติดต่อกันผู้ที่ถูกเรียกใช้ ดังนั้นการสร้างไอซีสำเร็จรูปมาใช้งานแทนอุปกรณ์พาสซีฟย่อมที่จะแก้ไขปัญหาเหล่านี้ได้ในระดับหนึ่งใน รูปที่ 2.4 เป็นบล็อกไดอะแกรมของไอซีที่นำมาใช้สร้างสัญญาณในระบบ DTMF ซึ่งวงจรภายในจะประกอบด้วย วงจรนับและถอดรหัส (counter and Decoder) ซึ่งก็จะแยกแยะว่าการกดหมายเลขแต่ละครั้งจะตรงกับตำแหน่งใดบ้างในแต่ละ แถวหรือแนว เมื่อทำการถอดรหัสจากการกดได้แล้ว จะนำค่าในแถวและแนวไปหาจากค่างความถี่หลัก สัญญาณที่ออกจากวงจรนับและถอดรหัสก็จะออกมาเป็นสัญญาณดิจิตอล 2 สัญญาณที่มีความถี่แตกต่างกัน หลังจากนั้นก็นำสัญญาณทั้ง 2 ไปผ่านวงจรแปลงสัญญาดิจิตอลไปเป็นอะนาลอก (D/A converter และนำมารวมกันโดยการนำไปผ่านวงจรรวมและขยายสัญญาณ (summing amplrfier แล้วจึงถูกส่งผ่านต่อไปยังวงจรควบคุมเสียงพูด (speech network) และผ่านต่อไปยังชุมสายโทรศัพท์ในที่สุด



รูปที่ 2.4 เป็นบล๊อกไดอะแกรมโดยทั่วไปของไอซีที่สร้างสัญญาณ DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ข้อดีของระบบ DTMF เมื่อเทียบกับระบบพัลซ์

ในการส่งสัญญาณแบบพัลซ์ 1 ลูก ใช้เวลาอย่างน้อย 100 มิลลิวินาที (60 มิลลิวินาทีในช่วงการเปิดวงจร และ 40 มิลลิวินาทีในช่วงการปิดวงจร) และยังต้องมีช่วงเวลาที่แยกสัญญาณแต่ละกลุ่มออกจากกันอีกอย่างน้อย 700 มิลลิวินาที ดังนั้นถ้าหมายเลขที่ต้องการติดต่อมีความยาวมากขึ้นเท่าใด ย่อมจะต้องเสียเวลาในการส่งสัญญาณมากขึ้นเท่านั้น ยกตัวอย่างเช่น ในการส่งสัญญาณ 555-5555 ใช้เวลาในการส่ง

= 5(พัลซ์/มิลลิวินาที) * 1000 (มิลลิวินาที/พัลซ์) * 7 (หมายเลข)

= 3.5 วินาที

ระยะเวลาของช่องว่างระหว่างกลุ่มสัญญาณ

= 700 (มิลลิวินาที) *6

= 4.2 วินาที

ดังนั้นจะใช้เวลาในการส่งทั้งหมด

= 3.5 + 4.2 วินาที

= 7.7 วินาที

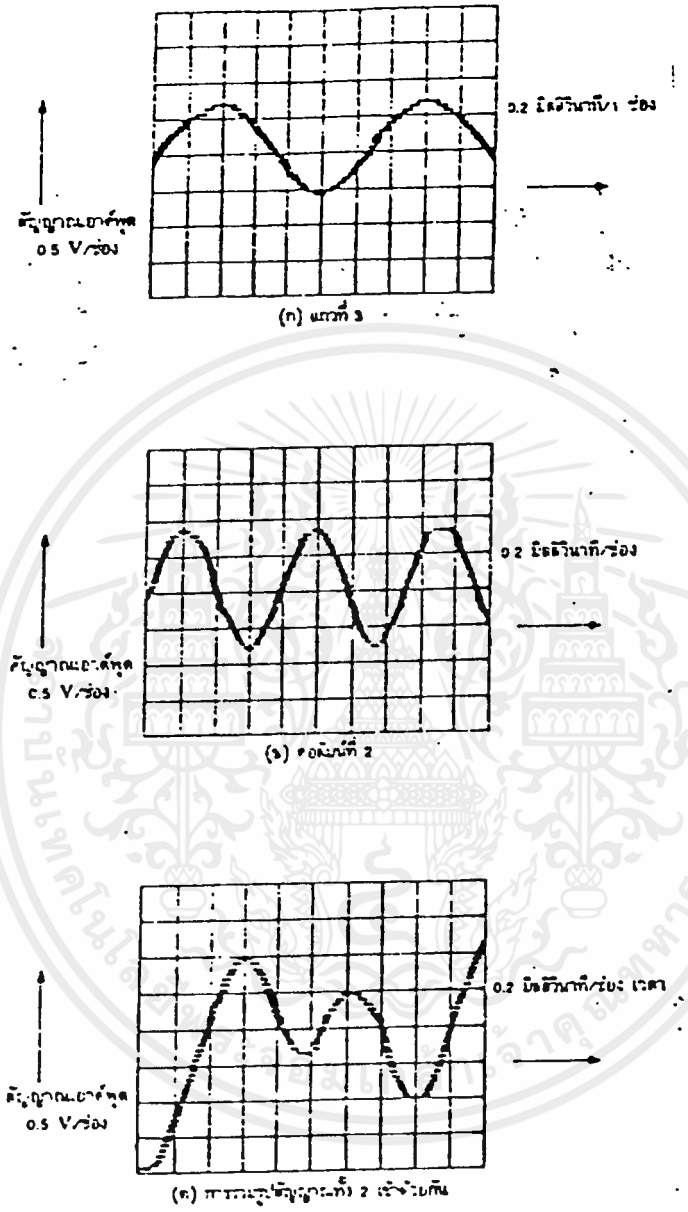
แต่ถ้าเป็นโทรศัพท์ที่ใช้การส่งระบบ DTMF จะใช้เวลาเท่ากับ

= 7* 100 มิลลิวินาที

= 0.7 วินาที

ดังนั้นจะเห็นได้ชัดเจนว่าข้อดีของระบบ DTMF คือ จะสามารถประหยัดเวลาในการส่งหมายเลขไปยังชุมสายได้มากกว่าระบบพัลซ์ สามารถใช้วงจรที่เป็นโซลิดเสตทได้ ทำให้เกิดความประหยัดและสะดวก ซึ่งเป็นผลให้ชุมสายใช้อุปกรณ์หน่วยความจำภายในชุมสายได้อย่างมีประสิทธิภาพมากขึ้นตามไปด้วย นอกจากนั้นยังสามารถนำไปเชื่อมต่อกับอุปกรณ์ภายในชุมสายได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงสัญญาณของระบบ DTMF เมื่อมีการกดปุ่มหมายเลข 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงานและการออกแบบส่วนต่างๆ

ส่วนตรวจจับสัญญาณกระดิ่ง

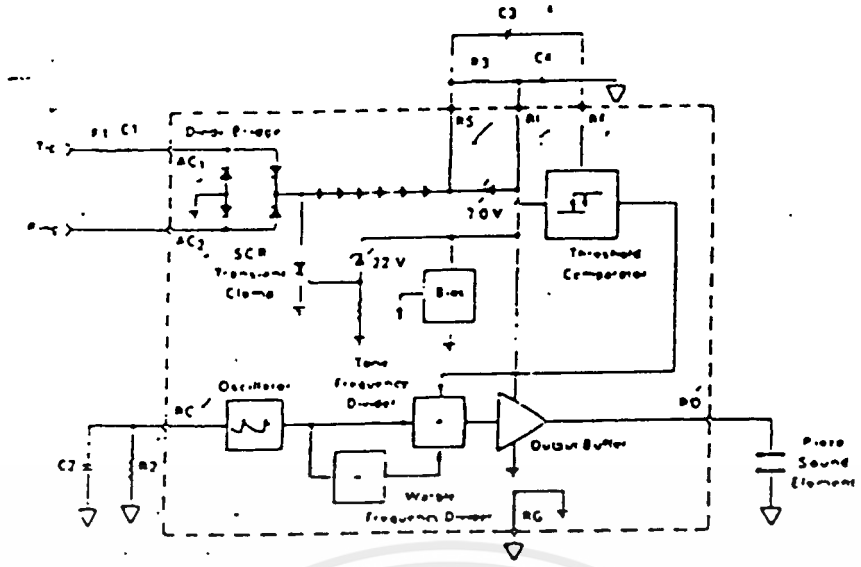
เป็นส่วนที่ใช้สำหรับตรวจจับสัญญาณกระดิ่งที่มาตามสาย ในขณะที่มีผู้โทรเข้ามาอาศัยหลักพื้นฐานของสัญญาณโทรศัพท์ คือ ในขณะที่สายว่าง คู่สายโทรศัพท์จะมีแรงดันประมาณ 48 โวลต์ ซึ่งจ่ายมาจากชุมสายโทรศัพท์ และเมื่อมีผู้เรียกเลขหมายเข้ามาทางชุมสายจะจ่ายสัญญาณกระดิ่งมาเป็นแรงดันกระแสสลับที่มีแรงดันประมาณ 100 Vc-c ส่วนตรวจจับสัญญาณจะให้พัลซ์ออกมาเมื่อมีกระดิ่งดังเข้ามา

หลักการทำงานของ IC MC34012

สัญญาณเรียกจากคู่สายโทรศัพท์ภายนอก ซึ่งมีลักษณะเป็นสัญญาณไฟฟ้าสลับขนาด 100 โวลย์จะถูกนำมาผ่านวงจรเรียกว่า (Full Wave Diode Budge) ทางขา AC1 และ AC2 ของ IC MC3412 (Telephone Tone Ringing) โดยมี R1 ทำหน้าที่ควบคุมอินพุทอิมพีแดนซ์และจำกัดกระแสทรานเซียนต์จากสายโทรศัพท์ และ C1 ทำหน้าที่ควบคุมอินพุทอิมพีแดนซ์ที่มีความถี่ต่ำ C นี้ได้ไฟเลี้ยงมาจากการแรคติไฟร์สัญญาณ AC ที่ป้อนเข้ามาทาง AC1 และ AC2 และจะทำให้กำเนิดความถี่ fc ที่ถูกกำหนดโดย R2 และ C2 ที่ต่อเข้าที่ขา R2 โดย fc จะมีค่าตั้งแต่ 1.0 KHz ถึง 10 KHz ขึ้นอยู่กับการเลือกใช้อุปกรณ์ ในที่นี้ fc มีค่า 4 KHz และทำให้มีสัญญาณเอาท์พุทที่ขา tone Ringer output (Ro มีขนาด fc/4 ถึง fc/5 Hz (ประมาณ 800-1000 Hz) 20 V

สำหรับความต้านทาน R3 ในวงจรจะเป็นตัวกำหนดตัวแอมปลิจูดของสัญญาณ Ringing ซึ่งเป็นสัญญาณเอาท์พุทออกที่ Ro โดยจะเริ่มจากสัญญาณที่เรคติไฟร์จากบริดจ์ไดโอดที่ขา AC1 และ AC2 จะทำให้เกิดค่ากระแสค่าหนึ่งที่ไหลผ่าน R3 จะถูกกรองโดย C3 ณ จุดอินพุทที่เข้าสู่วงจร Threshold เมื่อโวลท์เตจที่ตกคร่อม C3 มีค่าเกิน 1.7 โวลท์ จะทำให้ (Threshold comparator) อีนาเบิลสัญญาณ Tone Ringing ออกมา ค่า Line Transfer ที่เกิดจากการหมุนโทรศัพท์ที่ชาร์จ C3 จะยังมีค่าไม่มากพอที่ทำให้เกิดสัญญาณ Tone Gen ออกมา ส่วน C4 จะทำหน้าที่กรองไฟเลี้ยงให้วงจร Tone Generator สัญญาณเอาท์พุทที่ขา Ro จะถูกเรียงทิศกระแสโดยไดโอดบริดจ์แล้วส่งสัญญาณให้ออปโตคัปเปลอร์ (Opto Coupler) เปลี่ยนเป็นสถานะของลอจิกส่งให้ไมโครโปรเซสเซอร์ทราบ โดยถ้ามีสัญญาณเรียกจะให้ค่าทางสถานะลอจิกเป็น "1" แต่ถ้าไม่มีสัญญาณเรียกจะส่งสถานะลอจิก "0" ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 บล๊อกไดอะแกรมภายในไอซี MC34012

หลักการทำงานของ OPTOCOUPLE

การนำอาออปโต้คัปเปลอร์เข้ามาใช้วงจร ก็เพื่อจุดประสงค์ในการแยกส่วนของแรงดันไฟสูงที่เกิดจากสายโทรศัพท์ซึ่งมีค่าประมาณ 200 Vac ในขณะที่มีสัญญาณกริ่งดังเข้ามาให้เชื่อมต่อกับวงจรที่มีระดับไฟเลี้ยงขนาด 5 Vcc ซึ่งตัวออปโต้คับปปลอร์นี้จะมีความสำคัญอย่างยิ่งที่จะต้องนำมาใช้ในงานเพื่อให้วงจรที่มีระดับแรงดันต่างกันมาก สามารถทำงานร่วมกันได้ ในส่วนของวงจรนี้ใช้ ออปโต้คัปเปลอร์ เบอร์ 4N25

สำหรับรายละเอียดของ 4N25 เป็นดังนี้

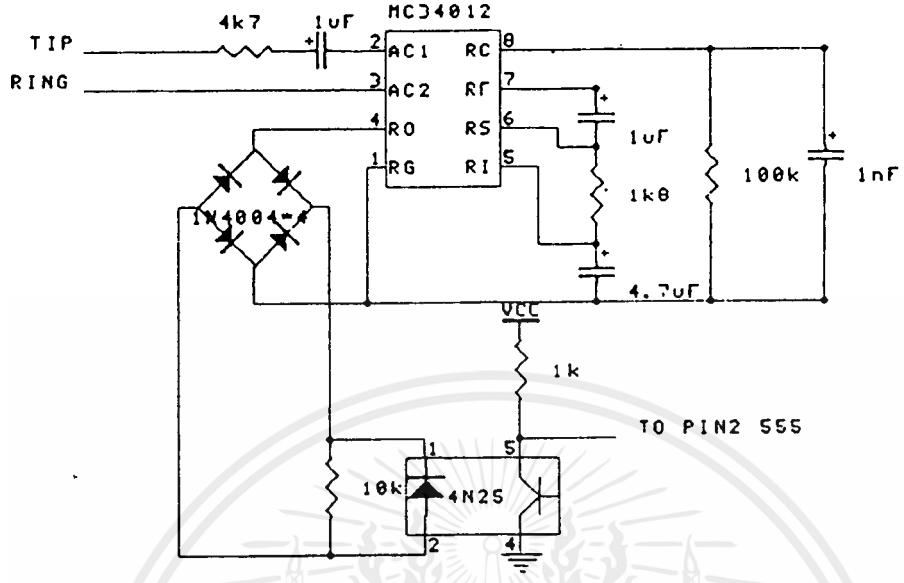
- **วงจรมีโครงสร้างคล้ายกับ TTL มาตรฐาน**
- **ใช้ GaAS ไดโอดซึ่งปล่องแสดงอินฟาเรดไปยัง Silicon NPN Phototransister**
- **การแปลง (Transfer) ของไฟกระแสตรงที่มีค่าสูง**
- **สามารถแยกวงจรที่มีระดับความดันที่ต่างกันได้ถึง + 25 KV**
- **ความเร็วในการสวิทช์ชิงค์สูง โดยมีค่า**

  Rise time (tr) = 2 uSec

  Fall time (tf) = 2 uSec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร DETECT RINGING



รูปที่ 3.2 แสดงวงจร Detect Ringing

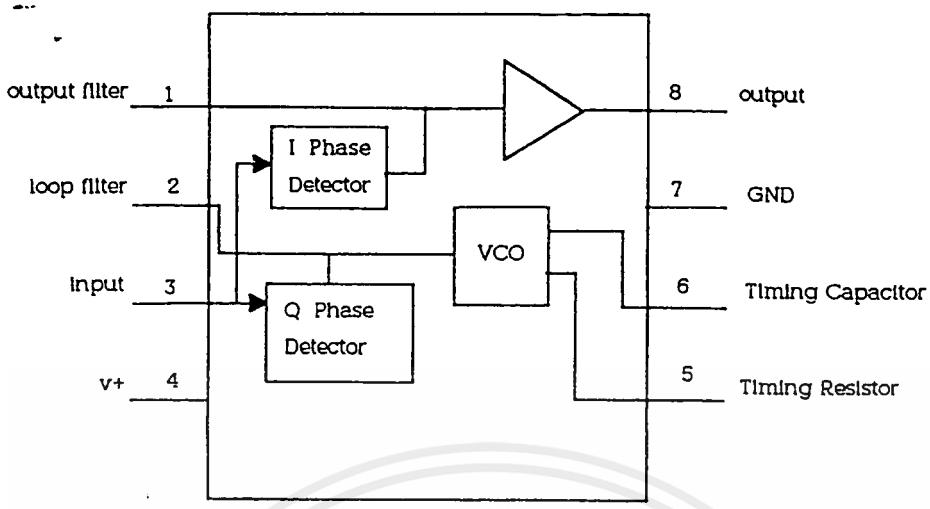
- สัญญาณโทรศัพท์จะเข้าที่ขา 2 และ 3 เพื่อเป็นการเปลี่ยนสัญญาณที่ได้มาเป็นสัญญาณระดับต่ำ โดย IC MC34012

- เอาท์พุท จะเป็นสัญญาณระดับต่ำ โดยจะอยู่ที่ 1 และ 4 ซึ่งจะออกมาเป็นรูป PLUS โดยมีขนาด 20 โวลท์ 1000 เฮิรตซ์ เรียกว่าสัญญาณ TONE RINGER และนำสัญญาณนี้ไปให้แก่ OPTO-COUPLE 4N25 ซึ่งจะรับสัญญาณจาก MC34012 แล้วทำให้เปิดสวิทช์ของ Transister ทำให้กระแสไหล เกิดสัญญาณลอจิก ขนาด "1" เมื่อมีสัญญาณมาและเป็น "0" เมื่อไม่มีสัญญาณจาก MC 34012 แล้วส่ง OUTPUT ไปยังส่วน MONO-STABLE

ส่วนประกอบของ IC LM567

เป็นวงจร Phase Lock Loop คือ เป็นไอซีทให้สัญญาณ OUTPUT เป็น Low เมื่อความถี่ INPUTมีค่าเท่ากับความถี่ศูนย์กลาง (fo) โดยสามารถตั้งความถี่ศูนย์กลาง และ Bandwidth เราสามารถตรวจจับความถี่ได้ตั้งแต่ 0.01 Hz - 500 kHz โดยมี Diagram ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 บล๊อกไดอะแกรมของ IC LM567

<u>รายละเอียดและหน้าที่ของแต่ละขา</u>

- OUTPUT FILTER เป็นขาที่ใช้ใส่ C เพื่อเป็นตัวกรองสัญญาณที่จะส่งออกไป โดยค่า C จะกำหนดดังนี้

$$C3 = 2C2$$

โดย C3 = ค่า C ที่ขา 1 C2 = ค่า C ที่ขา 2

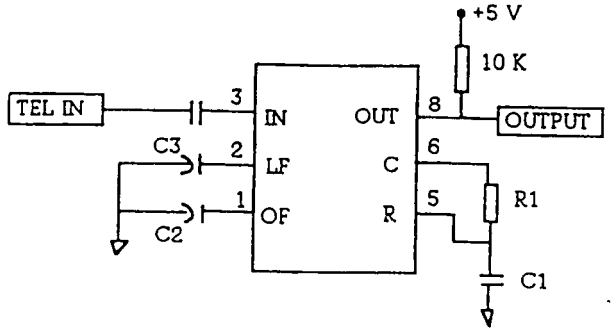
- LOOP FILER เป็นขาที่ใช้ใส่ C เพื่อเป็นตัวกรองสัญญาณที่จะรับเข้ามา โดยค่า C จะกำหนดได้ดังนี้

$$C2 = n/fo \qquad \text{ซึ่ง } 500 < n < 62000$$

fo = ค่าความถี่ศูนย์กลาง

- INPUT เป็นขาซึ่งใช้รับสัญญาณความถี่เพื่อนำมาเปรียบเทียบกับความถี่ศูนย์กลาง
- V+ ไฟเลี้ยงบวกมีค่าได้ดังนี้ $4.75 < V+ < 9$
- TIMING RESISTOR เป็นขาที่ใช้กำหนดความถี่ศูนย์กลางร่วมกับขา Timing Capacitor
- TIMING CAPACITOR เป็นขาที่ใช้กำหนดความถี่ศูนย์กลาง โดยจะกำหนดค่าได้จาก fo =?
- GND ขา Ground ของ IC
- OUTPUT เป็นขาเอาท์พุทที่จะส่ง GND ออกมาเมื่อสัญญาณ INPUT มีค่าความถี่เท่ากับค่าความถี่ศูนย์กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรของ IC LM567 ที่ใช้ในโครงงานนี้

ซึ่งได้ออกแบบค่าต่างๆที่ใช้งาน ดังนี้

- ค่าความถี่ศูนย์กลาง (fo) = $1/1.1R_1C_1$

$fo = 400$ Hz

$C_1 = 0.47\ \mu F$

$R_1 \cong 5\ K\Omega$

- ค่า C ที่ขา 2 (Loop Filter)

$C_2 = n / fo$ ซึ่ง $500 < n < 62000$

$C_2 = 2.2\ \mu F$

- ค่า C ที่ขา 1 (Output Filter)

$C_3 = 2C_2$

$C_3 \cong 4.7\ \mu F$

ส่วนโมโนสเตเบิล

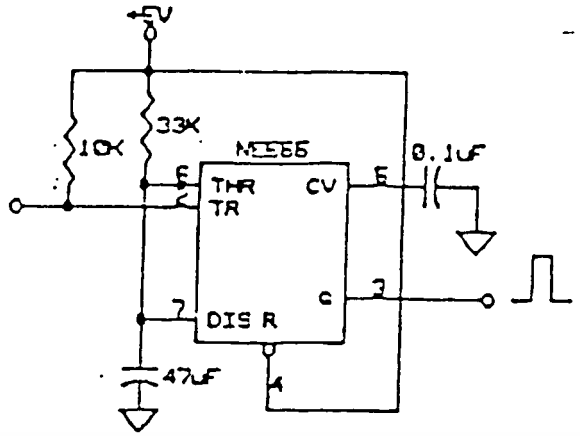
<u>การทำงานของ IC NE555</u>

เป็นวงจรที่เชื่อมต่อจาก 567 เพื่อเป็นการทำให้สัญญาณเรียบขึ้น ซึ่งรายละเอียดได้กล่าวมาแล้วข้างต้น

<u>ค่าที่นำมาใช้งาน</u>

ค่าของความต้านทานที่ขา DIs และตัวเก็บประจุที่ขา THR เป็นตัวกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรโมโนสเตเบิลที่ใช้ในโครงงาน

ความกว้างของพัลซ์ (Pulse) โดยมีค่า

$$t = 1.1RC$$

- จับสัญญาณจากสายนอก

$$t \cong 1 \text{ Sec}$$

- จับสัญญาณจาก PABX

$$t \cong 0.1 \text{ Sec}$$

- OUTPUT FILTER เป็นขาที่ใช้ใส่ C เพื่อเป็นตัวกรองสัญญาณที่จะส่งออกไปโดยค่า C จะกำหนดได้ดังนี้

$$C_3 = 2C$$

โดย C_3 = ค่า C ที่ขา 1

C_2 = ค่า C ที่ขา 2

- LOOP FILTER เป็นขาที่ใช้ใส่ C เพื่อเป็นตัวกรองสัญญาณที่จะรับเข้ามาโดยค่า C จะกำหนดได้ดังนี้

$$C_2 = n / fo$$

ซึ่ง $1,300 < n < 52.00$

fo = ค่าความถี่ศูนย์กลาง

- INPUT เป็นขาซึ่งใช้สัญญาณความถี่เพื่อนำมาเปรียบเทียบกับความถี่ศูนย์กลาง

- V+ ไฟเลี้ยงบวก มีค่าได้ดังนี้

$$4.75 < V+ < 9$$

- TIMING REGISTER เป็นขาที่ใช้กำหนดความถี่ศูนย์กลางรวมกับขา TIMING CAPACITOR หรือ ขา 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TIMING CAPACITOR เป็นขาที่ใช้กำหนดความถี่ศูนย์กลาง โดยจะกำหนดค่าได้ดังนี้

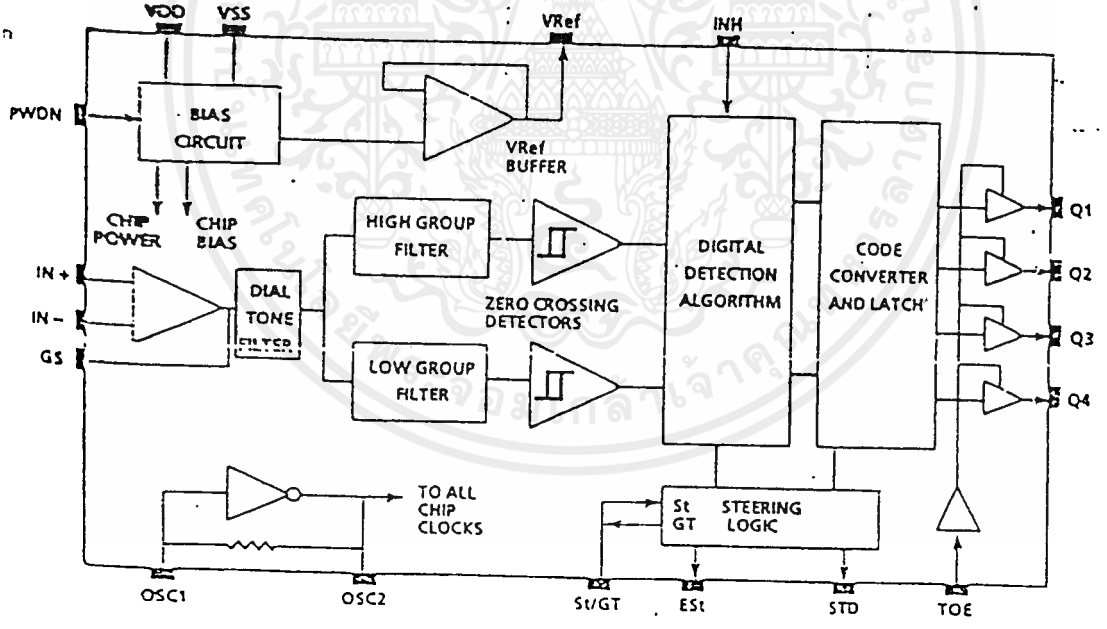
$$fo = 1 / 1.1(R1C1)$$

- GND ขา Ground ของ IC

- OUTPUT เป็นขาเอาท์พุท ที่จะส่ง GND ออกมาเมื่อสัญญาณ INPUT มีค่าความถี่เท่ากับ ค่าความถี่ศูนย์กลาง

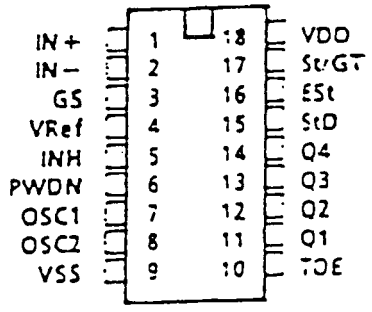
วงจรแปลงความถี่ของระบบโทน (TONE) ของโทรศัพท์เป็นตัวเลขฐานสอง 4 บิท

เนื่องจากโครงการชิ้นนี้ต้องมีการติดต่อระหว่างคอมพิวเตอร์และในการที่ ผู้ใช้จะติดต่อกับ เครื่องคอมพิวเตอร์ จึงต้องทำการส่งผ่านทาง ปุ่มโทรศัพท์ โดยที่เราสามารถใช้ IC 8870 เป็นตัวที่ทำการถอดรหัสให้กลายเป็นตัวเลขฐานสอง 4 บิท



รูปที่ 3.6 แสดงบล๊อกไดอะแกรมภายในของ IC MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ลักษณะขาสัญญาณต่างๆของ IC MT8870

<u>การทำงานภายใน MT8870 มีส่วนสำคัญ 5 ส่วน</u> ดังนี้

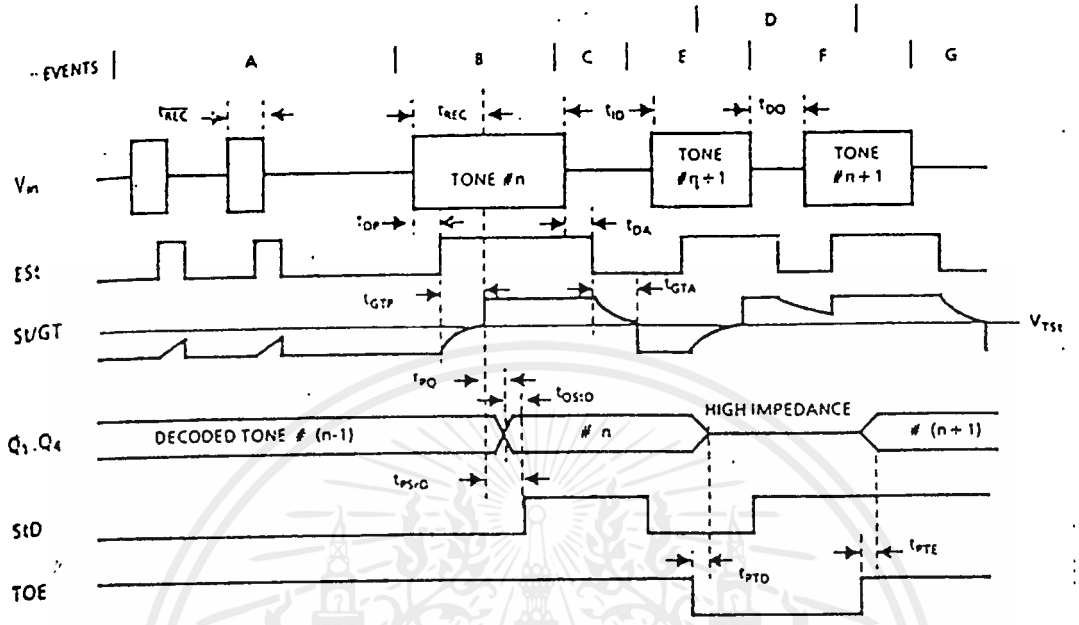
1. ภาคกรองความถี่ (filter section)
2. ภาคถอดรหัส (decoder section)
3. ภาคตรวจสอบสัญญาณ (steering circuit)
4. ภาคขยายสัญญาณความแตกต่าง (diferential input)
5. ภาคกำเนิดความถี่ (oscillator)

รายละเอียด และหน้าที่การทำงานของแต่ละขา

- IN+ นอนอินเวอร์ติ้งออปแอมป์อินพุต (Non inverting)
- IN- อินเวอร์ติ้งออปแอมป์อินพุท (Inverting)
- GS เป็นขาที่ Feedback จากเอาท์พุท ให้ผู้ใช้ได้เชื่อมตัวความต้านทานเพื่อล็อกค่า Gain ได้ตามต้องการ
- V_{REF} ค่าความต่างศักย์เปรียบเทียบที่เอาท์พุท ปกติมีค่า Vdd/2
- INH ถ้าให้ขานี้เป็น high จะไม่มีการถอดรหัส A.B.C.D ที่ปรากฏอยู่บนแป้นโทรศัพท์
- PWDN ถ้าให้ขานี้เป็น high จะหยุดการทำงานของไอซีเบอร์นี้
- OSC1 อินพุทของสัญญาณนาฬิกา
- OSC2 เอาท์พุทของสัญญาณนาฬิกา โดยใช้กับผลึก (Crystal) ที่มีความถี่ 3.5795 Mhz ต่อเข้าระหว่าง osc1 และ osc2
- Std Delay Steening Output ให้ค่าลอจิกเป็น 1 เมื่อมีข้อมูลระดับแรงดัน
- EST/ Early Steering Output จะให้ค่าลอกจิก 1 เมื่อข้อมูลถูกต้องและ ถ้าเป็น 0 จะไม่สัญญาณผ่านมา
- ST/GT Steering input/time output (bi-direction) ทำหน้าที่ส่งสัญญาณควบคุมวงจร RC ภายนอกเพื่อควบคุมไทม์
- Vdd ไฟเลี้ยงบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Vdd ไฟเลี้ยงบวก



รูปที่ 3.8 ไทม์มิ่งไดอะแกรมของไอซีเบอร์ MT8870

ขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาท์พุทไม่เปลี่ยน

B - ความถี่ #n ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่ จนถึงความถี่ใหม่ที่ถูกต้อง

C - จบความถี่ #n ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่ จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D - เอาท์พุทเปลี่ยนเป็น ไฮอิมพีแดนซ์

E - ความถี่ #n+1ถูกตรวจพบคาบความถูกต้องความถี่ถูกถอดรหัสและแลตช์ไว้

F - ความถี่ #n+1 หายไป ช่วงห่างไม่ถูกต้อง เอาท์พุทยังคงแลทช์อยู่

G - จบความถี่ #n+1 ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่ จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

Vin - สัญญาณความถี่ DTMF ที่เข้ามา

Est - Early Steening Output ใช้แสดงความถี่ที่ถูกต้อง

St/Gt - Steering input/Guard time output สำหรับต่อกับ RC ภายนอก

o1-o4 - เอาท์พุท BCD ขนาด 4 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

StD - Delayed Steering output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (input) ใช้ควบคุม o1-o4 ให้เป็นไฮอิมพีแดนซ์

t_{REC} -คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสิ้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

t_{D} -เวลาสิ้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

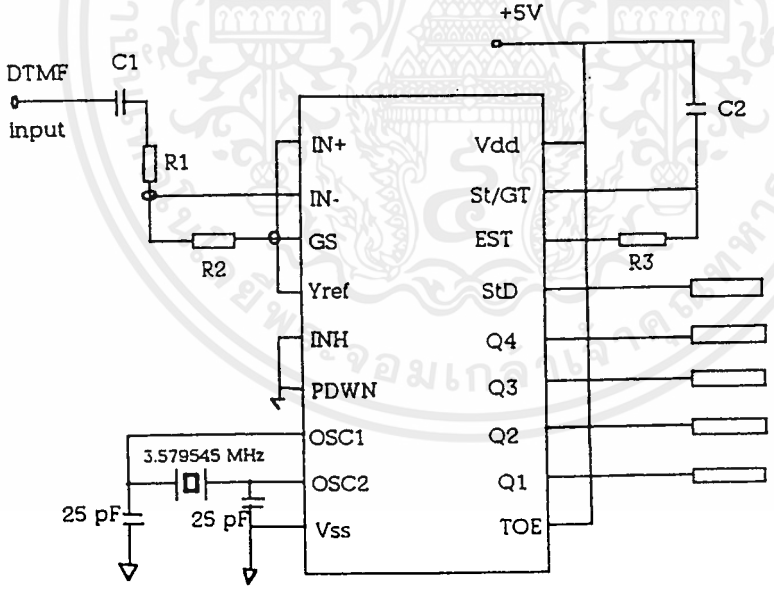
t_{Do} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหมายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF

t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DTMF



รูปที่ 3.9 วงจรถอดรหัสสัญญาณ DTMF เป็นเลขฐานสองขนาด 4 บิทที่ใช้ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F_{LOW}	F_{HIGH}	No	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1209	0	H	1	0	1	0
941	1336	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	Any	L	Z	Z	Z	Z

ตาราง 3.1 แสดงความถี่ค่าต่างๆที่ถูก Decode ออกมาได้

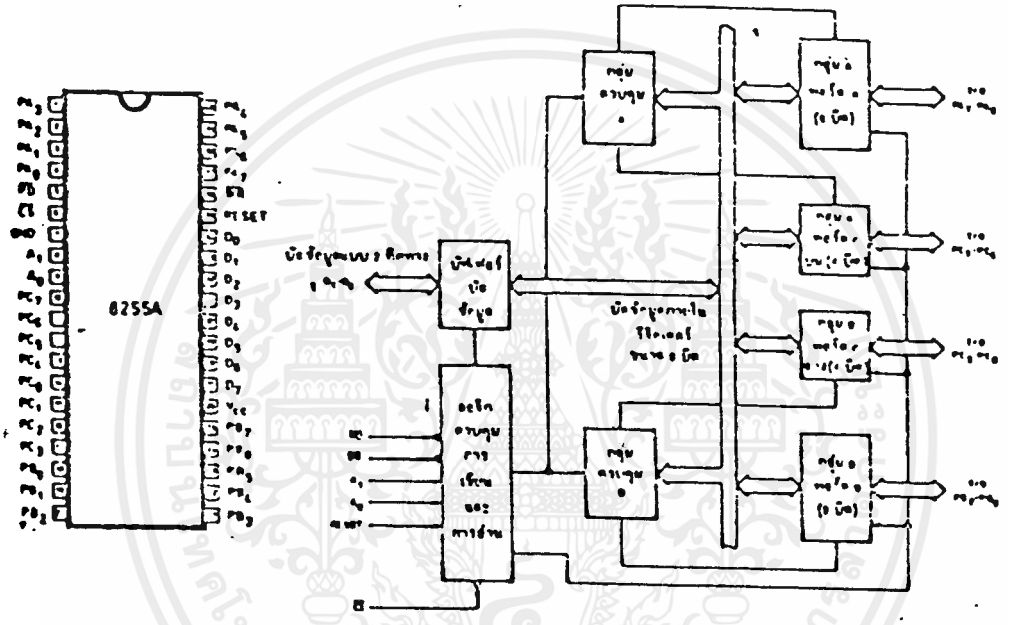
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038079

บทที่ 4

หลักการทำงานและการออกแบบ Interface Card โดยใช้ IC 8255

การใช้งาน 8255

8255 เป็นไอซีที่มี 40 ขา ได้รับการออกแบบมาให้มีสัญญาณเชื่อมต่อกับ 8088 การเรียกพอร์ทของ 8255 จะเรียกพอร์ทต่าง ๆ ว่า พอร์ท A พอร์ท B และ พอร์ท C โดยในพอร์ท C จะแยกเป็น 2 ส่วน คือ พอร์ทล่างจำนวน 4 บิต และพอร์ทบนอีก 4 พอร์ท พอร์ททุกพอร์ทเป็นทั้งอินพุทและเอาท์พุท



รูปที่ 4.1 แสดงขาของ 8255 และ บล๊อกไดอะแกรมภายใน

จากรูปเป็นแผนผังภายในของไอซีและการจัดวางของขาไอซี 8255 การทำงานของวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงาน โดยไมโครโปรเซสเซอร์จะส่งคำสั่งมาโปรแกรมการทำงาน หรือกำหนดรูปแบบของพอร์ทให้เป็นอินพุทเอาท์พุทได้

4.1 ขาต่าง ๆ ของ 8255

ขาทั้ง 40 ขา ของ 8255 จะประกอบด้วย

- DO-D7 เป็นขาที่ข้อมูลทั้งอินพุทและเอาท์พุท ต้องผ่านเข้าออก จะต่อเข้ากับบัสของไมโครโปรเซสเซอร์ เพื่อให้สามารถอ่านเขียนข้อมูลจากพอร์ทเหล่านี้ได้
- CS ขานี้เป็นขาสัญญาณอินพุทที่จะรับสัญญาณภายนอกเพื่อเลือกชิป 8255 เพื่อให้ไมโครโปรเซสเซอร์ สามารถเขียนหรืออ่านข้อมูลจากพอร์ทได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RD เป็นสัญญาณอินพุทที่ต้องส่งมาจากซีพียู เมื่อสัญญาณที่ขานี้เป็น 0 และสัญญาณที่ CS เป็น "0" ด้วย 8255 จะให้ซีพียูอ่านข้อมูลจากบัสในขณะที่พอร์ทเป็นอินพุท
- WR เป็นสัญญาณการเขียน จะแอดทีปเพื่อสัญญาณ WR และสัญญาณ CS เป็น "0" สัญญาณนี้จะมาจากซีพียูเมื่อต้องการเขียนข้อมูลลงพอร์ทที่กำหนด
- A0-A1 ลอจิกของสัญญาณทั้งสองจะถอดรหัสออกเป็น 4 รหัส เพื่อกำหนดรีจิสเตอร์ภายในที่เชื่อมต่อกับพอร์ทอินพุทเอาท์พุทของ 8255

-RESETเป็นสัญญาณที่ส่งมาจากภายนอกเข้ามาทำการรีเซ็ท8255 เพื่อเคลียสถานะต่าง ๆ ของ 8255

-PAO-PA7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255 ที่ชื่อพอร์ท A การเลือกพอร์ทที่เลือกโดยสัญญาณ แอดเดรส A0-A1

-PBO-PB7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255 ที่ชื่อพอร์ท B การเลือกพอร์ทที่เลือกสัญญาณแอดเดรส A0-A1

-PCO-PC7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255 ที่ชื่อพอร์ท B การเลือกพอร์ทที่เลือกสัญญาณแอดเดรส A0-A1

<u>รีจิสเตอร์ภายใน 8255</u>

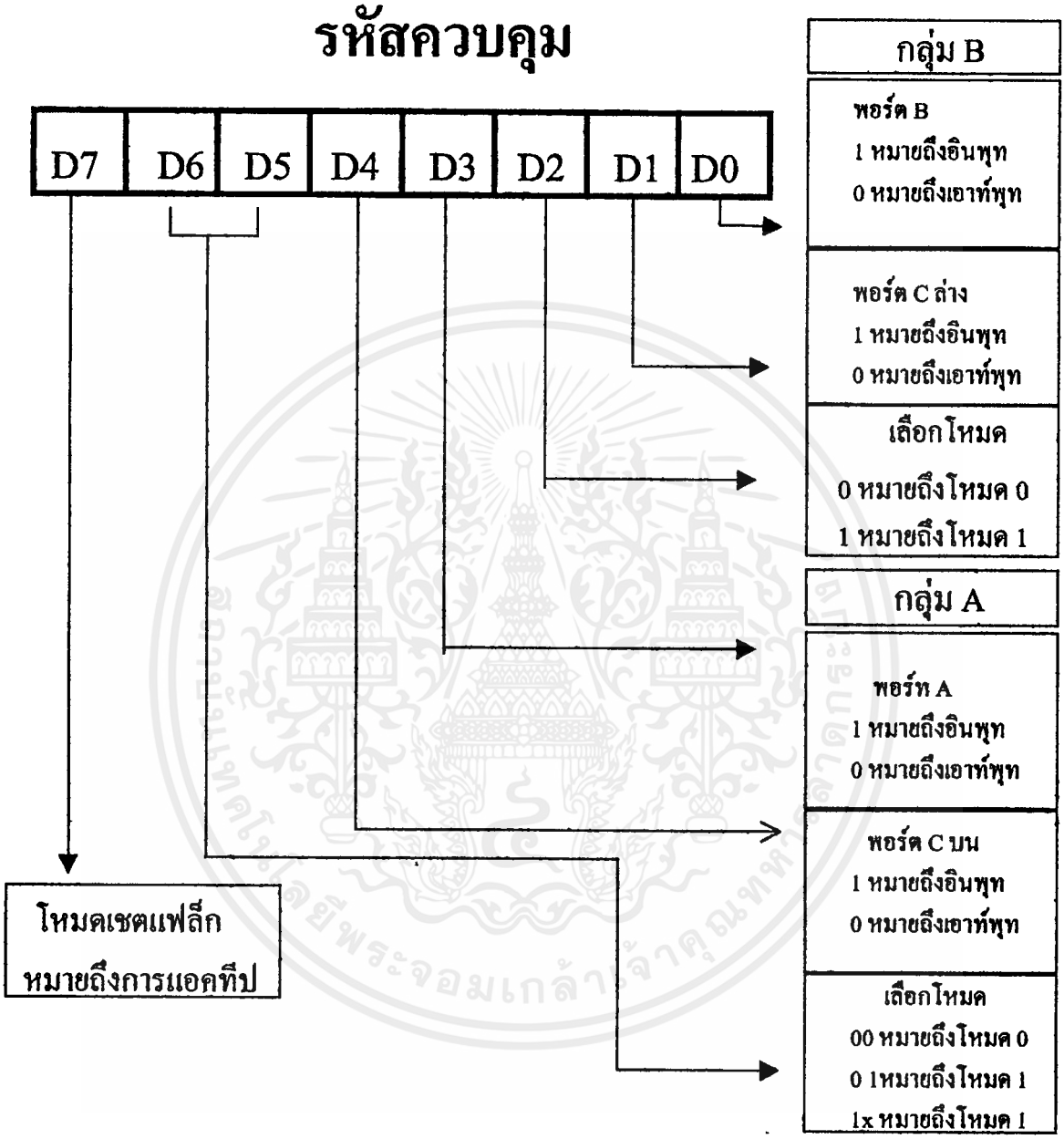
การใช้งาน 8255 จะต้องส่งรหัสควบคุมเข้าไปยังพอร์ทข้อมูลควบคุมเพื่อควบคุมการทำงานของ 8255 โดยใช้สัญญาณควบคุมพอร์ทหมายเลข 13H การควบคุมการทำงานของ 8255 มีหลายโหมด แต่ละโหมดจะแตกต่างกันออกไป การโปรแกรมให้ 8255 ทำงานนั้นจะทำได้ 3 โหมด คือ โหมด 0 โหมด 1 และ โหมด 2

$\overline{RD}$	$\overline{WR}$	$\overline{A1}$	$\overline{A0}$	ความหมาย
1	0	0	0	เขียนพอร์ต A ซึ่งเป็นข้อมูล
0	1	0	0	อ่านพอร์ต A ซึ่งเป็นข้อมูล
1	0	0	1	เขียนพอร์ต B ซึ่งเป็นข้อมูล
0	1	0	1	อ่านพอร์ต B ซึ่งเป็นข้อมูล
1	0	1	0	เขียนพอร์ต C ซึ่งเป็นข้อมูล
0	1	1	0	อ่านพอร์ต C ซึ่งเป็นข้อมูล
1	0	1	1	เขียนข้อมูล ซึ่งเป็นรหัสควบคุม
0	1	1	1	อ่านเข้ามา ซึ่งไม่มีความหมายใด

ตารางที่ 4.1 สัญญาณการควบคุมการทำงานของ IC 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดโหมดการทำงาน จะต้องส่งข้อมูลคำสั่งเข้าไปโปรแกรมในพอร์ทควบคุมของ8255 ซึ่งในที่นี้ใช้พอร์ทหมายเลข 13 แต่ละบิทของข้อมูลที่ส่งไปจะมีความหมายในตัวเอง ลักษณะความหมายของแต่ละบิทในรหัสควบคุมแสดงได้ดังรูป



รูปที่ 4.2 ความหมายของบิทต่างๆ ที่ใช้ในรหัสควบคุม

การโปรแกรม 8255 คือการให้ค่ารหัสบิทต่าง ๆ เข้าไปในรหัสควบคุมแล้วส่งไปยังรีจิสเตอร์ของพอร์ทควบคุม

บิต D7 เป็นบิตที่แสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น 1 หมายถึงรหัสควบคุมนี้มีผลต่อการเปลี่ยนแปลงการเซทโหมดต่าง ๆ ของ 8255

บิต D6 และ D5 เป็นการเลือกโหมดของพอร์ท A ซึ่งมี 3 โหมด คือ โหมด0 โหมด1 และโหมด 3

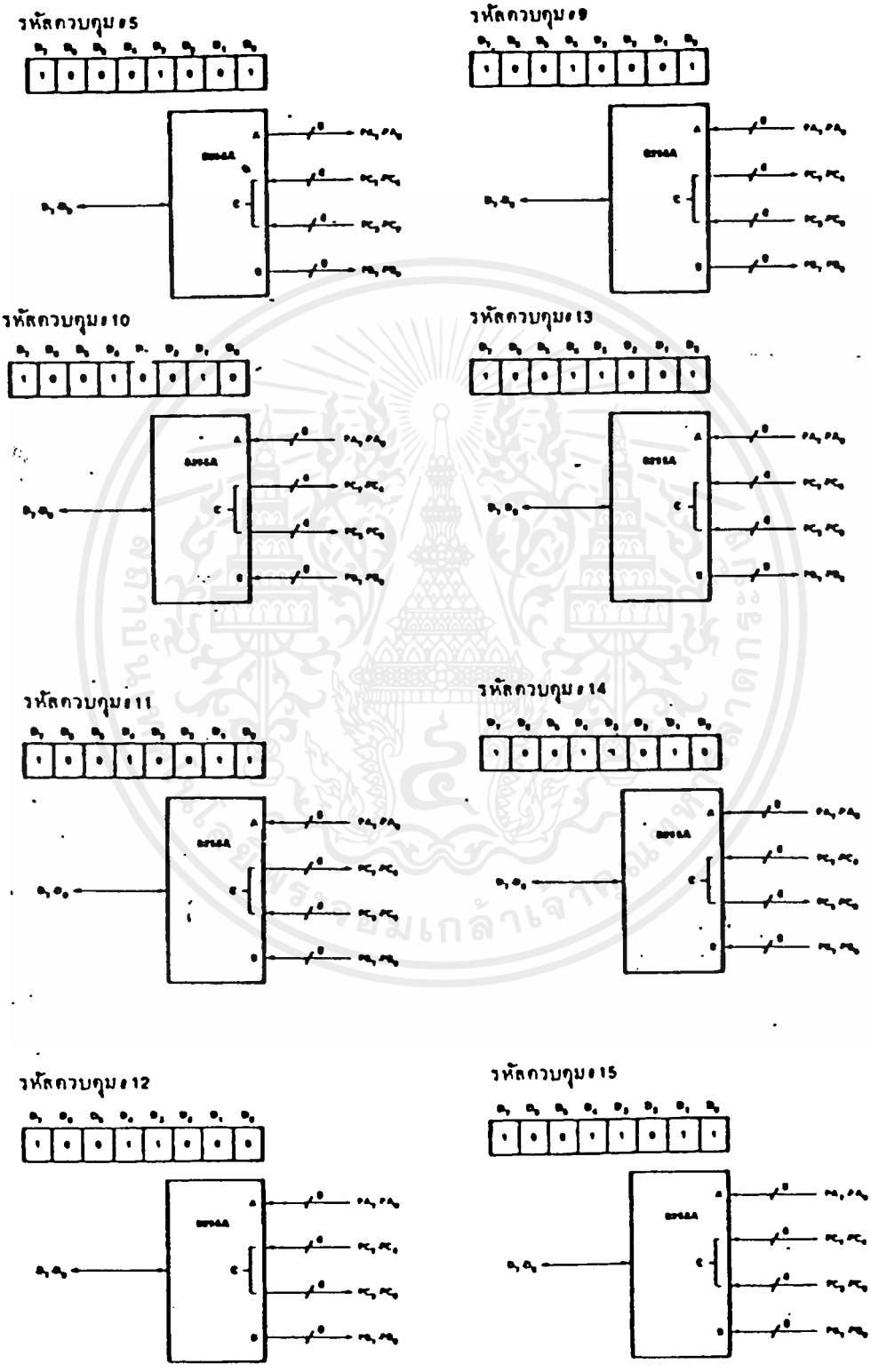
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต D4 ค่าเป็น 0 หมายถึงการกำหนดพอร์ท A เป็นเอาท์พุท

ค่าเป็น 1 หมายถึง การกำหนดพอร์ท A เป็นอินพุท

บิต D3 เป็นบิตที่บอกถึงการเซ็ทของพอร์ท C บน ถ้าเป็น 0 จะเป็นอาท์พุต

บิต D2 เป็นบิตที่บอกถึงการเซ็ทโหมด



รูปที่ 4.3 แสดงการทำงานของพอร์ทต่างๆของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับ SLOT มาตรฐานของ IBM PC

SAO- SA9 (อินพุท/เอาท์พุท) เป็นแอดเดรสของระบบที่ใช้ติดต่อกับหน่วยความจำและอุปกรณ์อินพุท/เอาท์พุท สายสัญญาณนี้จะต่อกับหน่วยความจำได้ 1 MB แต่ถ้าต้องการเชื่อมขยายแอดเดรสจะต้องใช้สายแอดเดรส LA17 LA23 การใช้สัญญาณ SAO SA19 จะต้องแอดทีฟขณะที่สัญญาณ BALE เป็น 1 และจะแลตช์ไปใช้ในขณะเปลี่ยนจาก 1 ไป 0 สัญญาณ BALE เป็นสัญญาณที่มาจากไมโครโปรเซสเซอร์ หรือ ดีเอ็มเอคอนโทรลเลอร์

CLK (เอาท์พุท) เป็นสัญญาณนาฬิกาของระบบ ในกรณีที่ไอบีเอ็มเอที จะส่งสัญญาณนี้เป็นสัญญาณขนาด 6 MHZ โดยมีช่วงเวลาประมาณ 157 นาโนวินาที สัญญาณเป็นรูปสี่เหลี่ยมมี DUTY CYCLE 50 เปอร์เซ็นต์ สัญญาณนี้มีจุดมุ่งหมายเพื่อใช้ในการซิงโครนัสระบบ มิได้มีจุดมุ่งหมายสำหรับใช้เป็นฐานเวลา

RESET DRV (เอาท์พุท) สัญญาณนี้ใช้สำหรับรีเซตระบบในขณะเปิดเครื่อง หรือขณะที่แหล่งจ่ายไฟที่ขาด หรือไฟตก สัญญาณนี้จะแอดทีฟเมื่อลอจิกเป็น 1

SDO SD7 (อินพุท/เอาท์พุท) เป็นสัญญาณข้อมูลขนาด 8 บิท ที่ใช้ติดต่อกับหน่วยความจำ ไมโครโปรเซสเซอร์ และอุปกรณ์อินพุท/เอาท์พุท บิต DO เป็นบิตที่มีนัยสำคัญน้อยที่สุด

BALE (เอาท์พุท) เป็นสัญญาณที่ใช้สำหรับการแลตช์แอดเดรสของระบบ สัญญาณนี้จะมาจาก 80288 ตัวควบคุมบัสสัญญาณที่จะใช้แลตช์แอดเดรสเมื่อเปลี่ยนจาก 1 กับ 0 และสัญญาณนี้จะได้จะรับการทำให้เป็น 1 ขณะที่กำลังทำ ดีเอ็มเอ

I/0 CHK (อินพุท) สัญญาณตรวจสอบของอินพุท/เอาท์พุท เพื่อบอกข้อมูลกับระบบ เช่นเดียวกับการตรวจสอบพาริตี้ ดังนั้นถ้าบนอินพุทมีข้อผิดพลาด สัญญาณนี้จะแอดทีเพื่อให้ส่งสัญญาณตอนในลักษณะ PARITYERROR

I/O CHRDY (อินพุท) สัญญาณนี้จะได้รับการทำให้เป็น 0 ด้วยหน่วยความจำหรืออุปกรณ์อินพุท เอาท์พุท การใช้สัญญาณนี้ก็เพื่อให้อุปกรณ์อินพุท/เอาท์พุทที่ช้าจะได้ติดต่อกับระบบด้วยการส่งสัญญาณมายัง ซีพียู เพื่อซิงโครนัสระบบได้

IRO3 IRO7 (อินพุท) เป็นสัญญาณอินเตอร์รัพต์

IOR (อินพุท/เอาท์พุท) สัญญาณอินพุท เอาท์พุทเป็นสัญญาณที่ส่งมาจาก ซีพียู การควบคุมสัญญาณนี้มาจาก80266 และดีเอ็มเอ คอนโทรลเลอร์ สัญญาณนี้แอดทีฟ 0

SMEMR (เอาต์พุท/เอาพุท) สัญญาณนี้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำ SMEMR ใช้สำหรับติดต่อกับหน่วยความจำในส่วน 1 MB แรก หรือ ถอดรหัสมาจากแอดเดรสส่วนล่าง ส่วน SMEMR นี้แอดทีฟกับหน่วยความจำได้หมด 16 MB

DROO DRO3 (อินพุท) สัญญาณการของดีเอ็นเอแชนแนล 0-3 โดยสัญญาณนี้จะมาจากอุปกรณ์อินพุท/เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AEN (เอาท์พุท) อีนาเบิลแอดเดรส เป็นสัญญาณที่ใช้สำหรับการแยกบัสแอดเดรสในการทำดีเอ็มเอ เมื่อสัญญาณนี้แอดทีฟ แสดงว่าดีเอ็มเอ คอนโทรลเลอร์ควบคุมการทำงานทางแอดเดรสแทนซีพียู

REFRESH (อินพุท/เอาท์พุท) เป็นสัญญาณที่ใช้ในการแสดงสัญญาณรีเฟรชไซเคิล สัญญาณนี้ส่งมาจากไมโครโปรเซสเซอร์ผ่านทางช่องอินพุท/เอาท์พุท

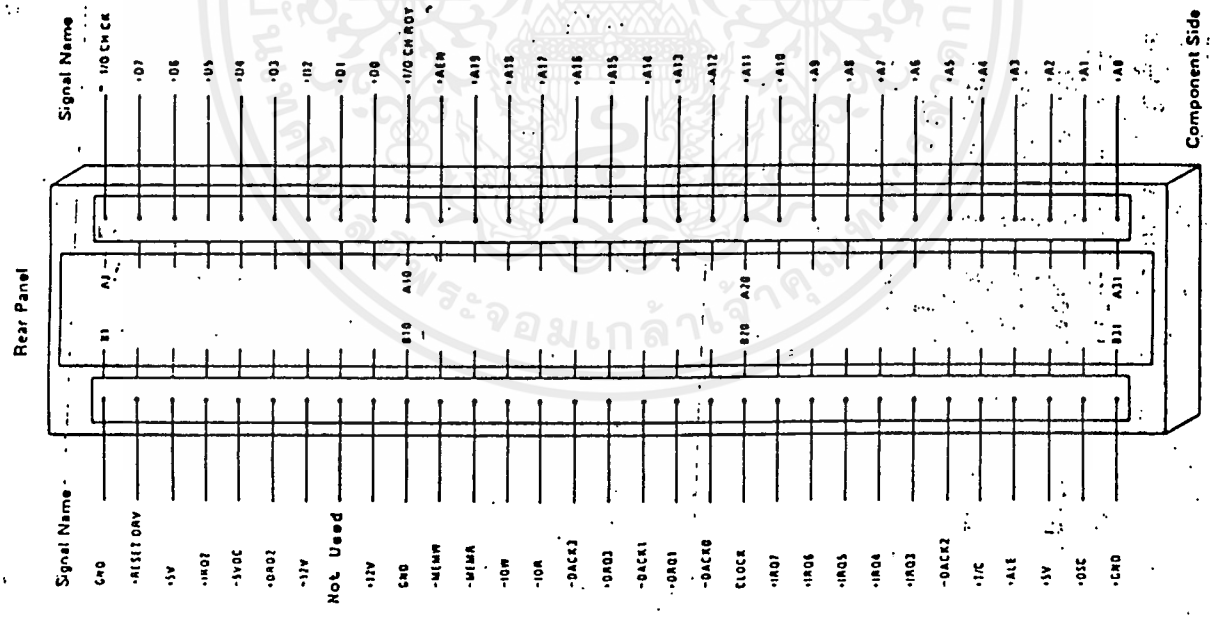
T/C (เอาท์พุท) สัญญาณ TERMINAL COUNT เป็นสัญญาณพัลซ์เมื่อดีเอ็มเอนับจำนวนมาครบตามที่กำหนด

MEM CS16 (อินพุท) สัญญาณนี้ส่งมาบอกเมนบอร์ด ถ้าหากการถ่ายเทข้อมูลต้องการสถานะรอ

IO CS16 (อินพุท) สัญญาณนี้เป็นตัวส่งมาบอกเมนบอร์ดว่าอินพุท/เอาท์พุท ต้องการสถานะรอ

OSC (เอาท์พุท) สัญญาณนาฬิกา 70 นาโนวินาที หรือ ประมาณ 1431818 MHz

OWS (อินพุท) เป็นสัญญาณที่บอกซีพียูว่าการทำงานหนึ่งรอบของบัสไม่จำเป็นต้องแทรกสถานะรอ



รูปที่ 4.4 แสดงขาต่างๆของสล๊อต IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

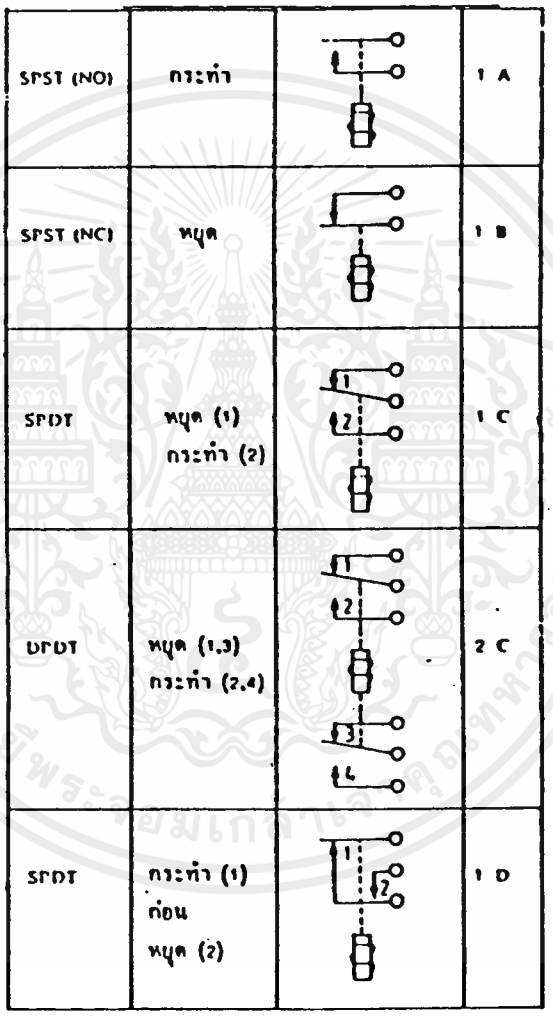
ตารางที่ 4.2 แสดงแอดเดรสพอร์ทของ IBM PC

Number Port (Hex)	Type of Equipment
001-01F	DMA CONTROLLER 1,8237A-5
020-03F	INTERRUPT CONTROLLER หมายเลข 1,8259A ดัมหลัก
040-05F	TIMMER 8254-2
060-06F	8042 KEYBOARD
070-07F	CLOCK,NMI-CMOS RAM
080-09F	DMA PAGE REGISTER
0A0-0BF	INTERRUPT CONTROLLER 2,8259A
0C0-0DF	DMA CONTROLLER 28237A-5
0D0	CLEAR MATH PROCESSOR
0F1	RESET MATH PROCESSOR
0F8-0FF	MATH PROCESSOR
1F0-1F8	HARD DRI
200-207	IO GAME
208-277	NOT USE
278-27F	PRNTER PORT NO 2
280-2F7	NOT USE
2F8-2FF	SERAL PORT NO 2
300-31F	PROTOTYPE CARD
320-35F	NOT USE
360-36F	SPARE
370-377	NOT USE
378-37F	PRNTER PORT NO 1
380-38F	SDLC ,BSYNC 2
390-3AF	NOT USE
3B0-3BF	MONOCHROM AND PRNTER
3C0-3CF	SPARE
3D0-3DF	COLOUR DISLAY MONITOR
3F0-3F7	CONTROL DISKET
3F8-3FF	SERAL PORT NO 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน รีเลย์

ในการใช้งานรีเลย์ จะต้องเริ่มพิจาณาจากขนาดรูปร่าง จำนวนหน้าสัมผัส อัตรางานกำลังและราคา โดยแบบหน้าสัมผัสของรีเลย์ จะดูจากรูปร่างของขั้ว โพลที่ใช้ตัวย่อ P และแขน โทรล ตัวย่อ T ถ้าหากมีเพียงชุดเดียวจะใช้ตัวย่อ S ถ้าเป็นคู่จะใช้ D เช่น SPST SPDT DPDT ขดลวดของรีเลย์สามารถใช้ขับหน้าสัมผัสน้อยชุดหรือหลายชุดก็ได้ ซึ่งจะมีลักษณะการใช้งานดังรูป



รูปที่ 4.5 แสดงการจัดตัวรีเลย์ระบุเป็นรูปแบบ A-D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของรีเลย์

รีเลย์ที่ใช้อยู่โดยทั่วไปมีให้เลือกใช้ได้หลายแบบ สามารถดูได้จากตาราง

ชนิด	ขนาด (นิ้ว)	อัตราหน้าสัมผัส	ค่าแรงดันของขดลวด	เวลาสวิตชิ่ง
ใช้งานทั่วไป (แบบปลั๊กอิน)	0.9-1.5W 0.9-1.4D 1.2-2.0H	3-10A 28 V_{DC} 120-240 V_{ac}	6, 12, 24, 120V 1.2 W DC 2 VA ac	15-30 ms
ใช้งานทั่วไป (ขนาดเล็ก)	0.6-1.2W 0.4-0.75D 0.4-0.7H	1-5 VA 28 V_{DC} 120 หรือ 240 V_{ac}	5, 6, 12 24, 48 V_{DC} 0.5-1 W	5-10 ms
หรีดรีเลย์	0.9-1.2W 0.3-0.8D 0.3-0.5H ใช้กับรูปแบบ DIP ได้ด้วย	0.5-2A 5-50W 28-250 V_{DC} และ ac	5, 6, 12, 24 V_{DC} 50-400 mW (ใช้ได้กับแรงดันต่ำ ถึง 1V)	0.2-1 ms
เฮอร์เมติก TO-5	0.6-1W 0.3-0.5D 0.3-0.9H	0.5-5A 28 V_{DC} 115 V_{ac}	4-32 V_{DC} 120 mW	5 ms
เฮอร์เมติก และมีการซีลด์	0.6-1W 0.3-0.5D 0.3-0.9H	0.5-2A 28 V_{DC} 115 V_{ac}	5-115 V_{DC} 100-400 mW	5 ms
RF รีเลย์	0.8-1W 0.4-1D 0.4-1H	10-25 W RF hot To 150W RF dry	5-50 V_{DC} 250 mW	5-10 ms
กำลัง	ขนาดทั่วไป 2-4 นิ้ว	10-40A 120, 240 V_{ac} และสูงกว่า	6, 12, 24, 120 208, 240V 2-5 W DC 6-20 VA ac	15-50 ms

ตารางที่ 4.3 คุณสมบัติของรีเลย์ที่ใช้ทั่วไป

1 รีเลย์แบบปลั๊กอิน มีขาในแบบซ็อกเก็ตออกแบบมาให้ใช้สำหรับงานกระแสปานกลางมีลักษณะเป็นซ็อกเก็ต

2 หรีดรีเลย์ มีทั้งแบบเปลือยและผนึกซึ่งคุณสมบัติคือ ทำงานได้เร็วมาก ประมาณ 50 ไมโครเซค โดยที่มีเลย์ทั่วไปจะมีความเร็วประมาณ 5-30 ไมโครเซคเท่านั้น แต่ข้อเสียของรีเลย์แบบนี้คือ ทนแรงดันและกระแสได้ต่ำ

3 รีเลย์แบบกำลัง (POWERSWITCHING RELAY) เป็นรีเลย์ที่มีขนาดใหญ่และมีโครงสร้างเป็นแบบเปิด ใช้สำหรับการสวิตช์ทุกระดับกำลังงาน โดยปกติมักใช้ในวงจรควบคุมต่าง รีเลย์ชนิดนี้ทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

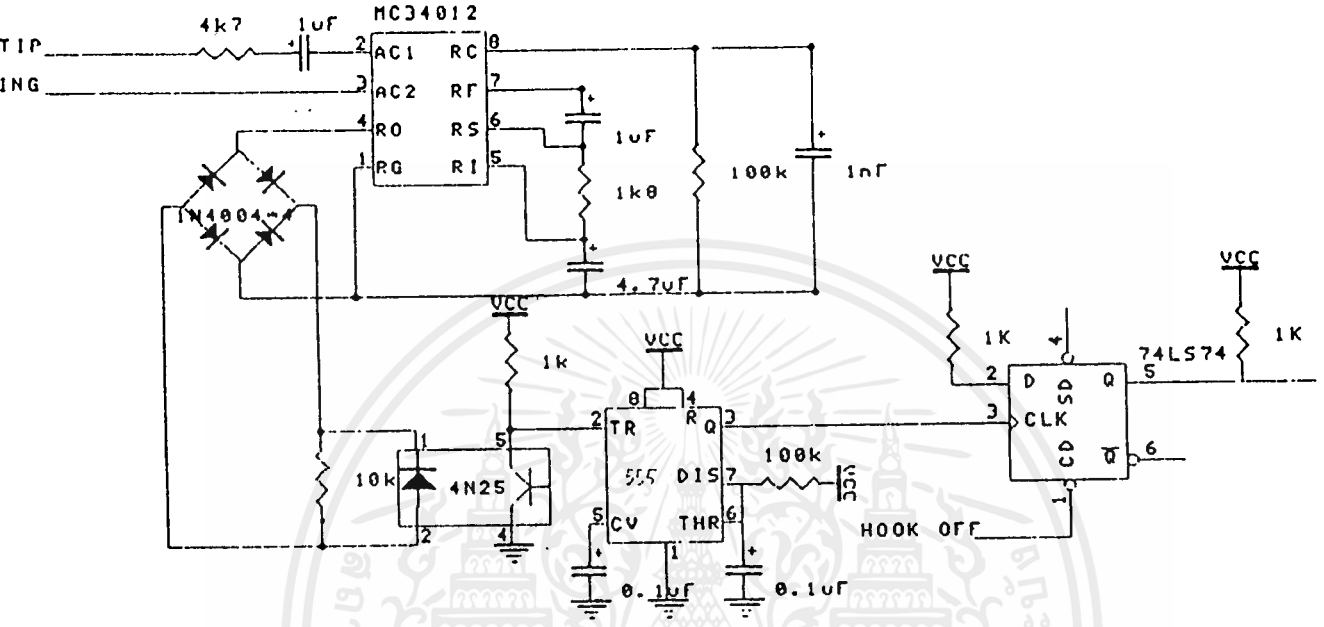
กระแสไฟสูงสุดถึง 30 A ขนาดโดยทั่วไป 2-3 นิ้ว กำลังงานที่สูญเสียไปประมาณ 2 วัตต์และใช้กับแรงดัน 5-10 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและวิจารณ์ผลการทดลอง

วงจรตรวจจับสัญญาณโทรศัพท์



ารทดสอบ

เมื่อทำการประกอบวงจรแล้วทำการป้อนไฟเลี้ยงให้กับวงจร จากนั้นจึงทำการทดสอบส่วนวงจร IC 555 โดยทำการป้อนสัญญาณ "0" ที่ขา 2 ของ #IC 555 ปรากฏว่า Relay ไม่มีการทำงาน แต่เมื่อป้อนสัญญาณ "1" ที่ขา 2 ของ #IC 555 ปรากฏว่า Relay มีการทำงาน

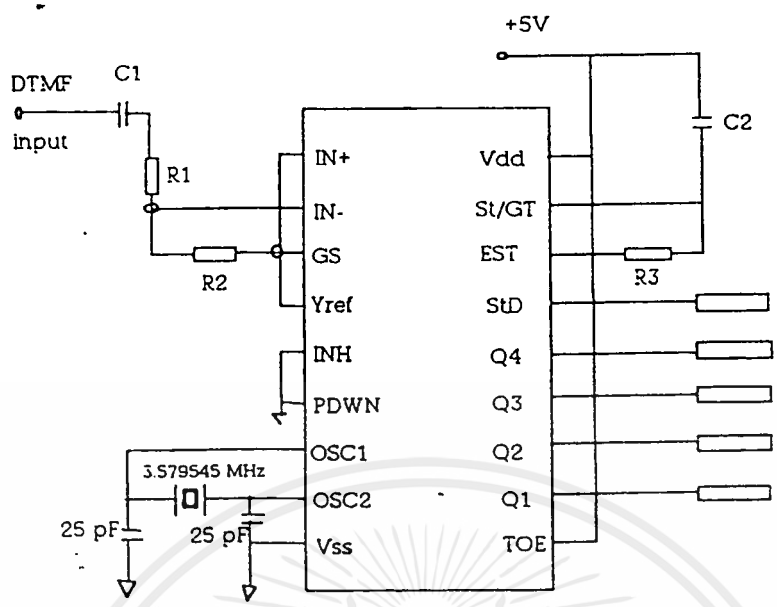
จากนั้นจึงทำการต่อสัญญาณโทรศัพท์เข้าทาง Ring-Tip หรือ ขา 2-3 ของ #IC MC34012 แล้วทำการ Dial จากภายนอก ผลปรากฏว่าเมื่อสัญญาณเรียกดัง 1 ครั้ง Relay ก็มีการทำงาน ซึ่งแสดงว่าการทำงานของวงจรส่วนนี้ถูกต้อง

ข้อเสีย

- วงจรนี้ไม่สามารถกำหนดจำนวนครั้งในการเรียกได้ คือจะทำงานทันทีที่ได้รับสัญญาณเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถอดรหัสสัญญาณ DTMF



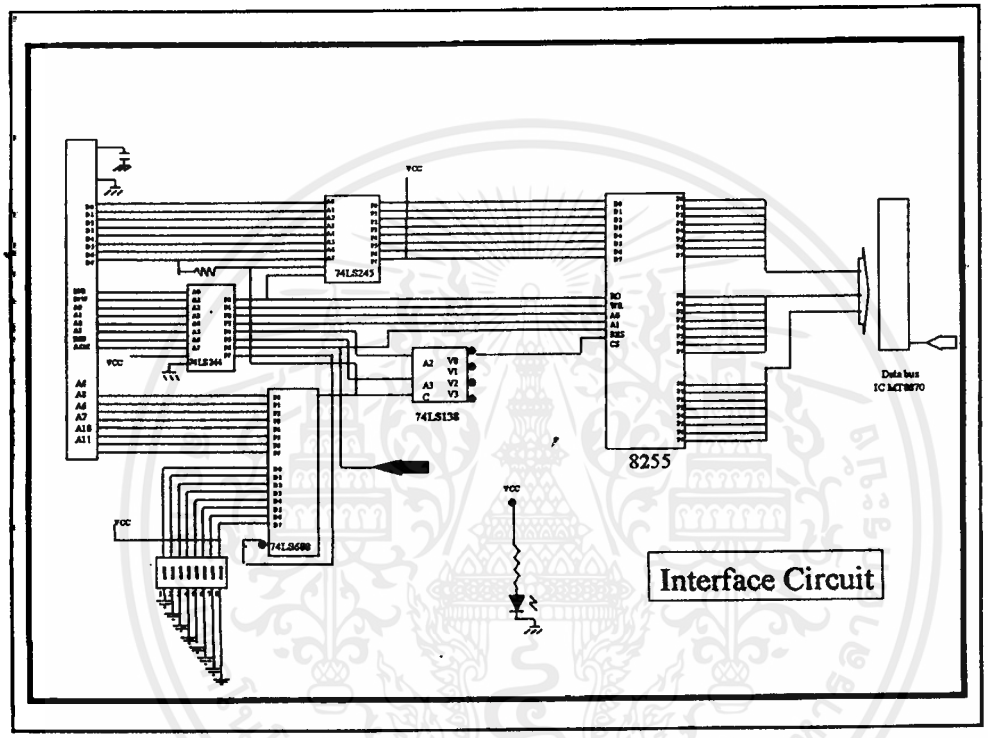
การทดสอบ

เมื่อทำการประกอบวงจรแล้วทำการป้อนไฟเลี้ยงให้กับวงจร จากนั้นจึงนำขาอินพุทของวงจรต่อเข้ากับสัญญาณโทรศัพท์ จากนั้นจึงทำการ Dial จากภายนอกเมื่อ Relay ทำงาน ซึ่งแสดงว่าวงจรต่อเข้ากับสัญญาณโทรศัพท์เรียบร้อยแล้วจึงทำการ กดปุ่มตัวเลขแล้วทำการวัดสัญญาณที่ขา Q1,Q2,Q3,Q4 การวัดทำโดยการต่อ LED ที่ขาทั้ง 4 ซึ่งได้ข้อมูลดังตาราง

ปุ่ม / ขาสัญญาณ	Q4	Q3	Q2	Q1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
*	-	-	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Interface Card



การทดสอบ

การทดสอบจะทำการทดสอบ #IC แต่ละตัวก่อนว่ามีการทำงานตามที่ต้องการหรือไม่

- ทดสอบ IC 74LS688 เมื่อทำการป้อนสัญญาณเข้าที่ขา P0-P7 และ Q0-Q7 จากนั้นสังเกตุที่ LED ที่ต่อที่ขา เอาท์พุท ปรากฏว่า ถ้ามีการป้อนข้อมูลที่ตรงกันทั้งหมด LED จะดับ แต่ถ้าการป้อนสัญญาณแตกต่างกันไม่ว่าขาใด LED จะติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทดสอบ IC 74LS139 เมื่อป้อนไฟเลี้ยงเรียบร้อยและทำการป้อนสัญญาณ "0" ให้กับขา CS ซึ่งเป็นขาที่ทำให้ 74LS139 ทำงาน จากนั้นทำการป้อนสัญญาณให้ ขา2-ขา3 ซึ่งเป็นขาอินพุท แล้วสังเกตุ LED ที่ต่ออยู่กับ เอาท์พุท ซึ่งได้ค่าดังตาราง

อินพุท		เอาท์พุท			
A	B	Y0	Y1	Y2	Y3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

เมื่อทำการทดสอบ IC แต่ละตัวแล้วจึงทำการประกอบวงจรทั้งหมดแล้วประกอบเข้ากับคอมพิวเตอร์แล้วทำการทดสอบโดยใช้คำสั่งส่งข้อมูลไปยัง 8255

ขั้นการทดลอง

- ส่งข้อมูลที่ใช้ควบคุมการทำงาน 8255 (Control Data)
- ส่งข้อมูลเอาท์พุทไปที่พอร์ท A ซึ่งมีการส่งข้อมูลดังตาราง

ข้อมูลที่ส่ง	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
0xff	1	1	1	1	1	1	1	1
0xf0	1	1	1	1	0	0	0	0
0xf1	1	1	1	1	0	0	0	1
0xf2	1	1	1	1	0	0	1	0
0xf3	1	1	1	1	0	0	1	1
0xf4	1	1	1	1	0	1	0	0
0xf5	1	1	1	1	0	1	0	1
0xf6	1	1	1	1	0	1	1	0
0x0f	0	0	0	0	1	1	1	1
0x00	0	0	0	0	0	0	0	0
0x99	1	0	0	1	1	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รับข้อมูลจากพอร์ท B ซึ่งมีการป้อนข้อมูลโดยการป้อนสัญญาณข้าที่ขา PB0-PB7ดังตาราง

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	ข้อมูลที่รับได้
1	1	1	1	1	1	1	1	0xff
1	1	1	1	0	0	0	0	0xf0
1	1	1	1	0	0	0	1	0xf1
1	1	1	1	0	0	1	0	0xf2
1	1	1	1	0	0	1	1	0xf3
1	1	1	1	0	1	0	0	0xf4
1	1	1	1	0	1	0	1	0xf5
1	1	1	1	0	1	1	0	0xf6
0	0	0	0	1	1	1	1	0x0f
0	0	0	0	0	0	0	0	0x00
1	0	0	1	1	0	0	1	0x99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Motorola, **"Motor's Telecommunications Device Data"** Motorola Inc, 1984

2. The nical Information Center. "Opto Electronic Data" Motorola 224p. 1989

3. The nical Informantion Center. " Lincar IC Data Book" National 860p 1987

4. ดำริ สุภาพ **"ผ่ารีเลย์"**, วารสารเซมิคอนดักเตอร์อิเลคทรอนิกส์, ฉบับที่ 106, 2533, หน้า 89-96.

5. ชาญวิทย์ ยี่สาร, ชำนาญ ทัตตานนท์ และ ถาวร เสนชัย, **"เครื่องรับโทรศัพท์ที่ทำงานได้หลายหน้าที่"**, ปริญญานิพนธ์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2531.

6. ปรีชา สุรัชภากรณ์ และ อนุชาติ บุญมาก, "On Hook Dialing Hold", ปริญญานิพนธ์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง ปีการศึกษา 2539.

7. **เปรมจิตร วิสุทธิสิระ, " วงจรแปลงดิจิตอลเป็นอะนาลอก"** วารสารเซมิคอนดักเตอร์อิเลค ทรอนิกส์, ฉบับที่ 102, 2533, หน้า 272-279

8. ธานินทร์ ถาวรศาสนวงศ์ และทินกร ตุ๊ก, การ์ดอินเตอร์เฟสไอบีเอ็ม / พีซี, ฟิสิกส์เซ็นเตอร์ การพิมพ์, 256 หน้า

9. ยืน ภู่วรรณ **"ทฤษฎีการประยุกต์ไมโครโปรเซสเซอร์"** ซีเอ็ดยูเคชั่น 238 หน้า 2536

10. สุทธินันท์ พรศิริกุล, **"ลีกอีกนิดกับโทรศัพท์"** วารสารเซมิคอนดัดเตอร์อิเล็กทรอนิกส์, ฉบับที่ 120 - 127, บริษัท ซีเอ็ดยูเคชั่น จำกัด, กันยายน 2535 เมษายน 2536

11.อุทัย จึงภักดี, **"อุปกรณ์กำเนิดเสียงเปียโซอิเล็กตริก"** วารสารเซมิคอนดักเตอร์ อิเล็กทรอนิกส์, ฉบับที่ 144, บริษัท ซีเอ็ดยูเคชั่น จำกัด, พฤศจิกายน 2537.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

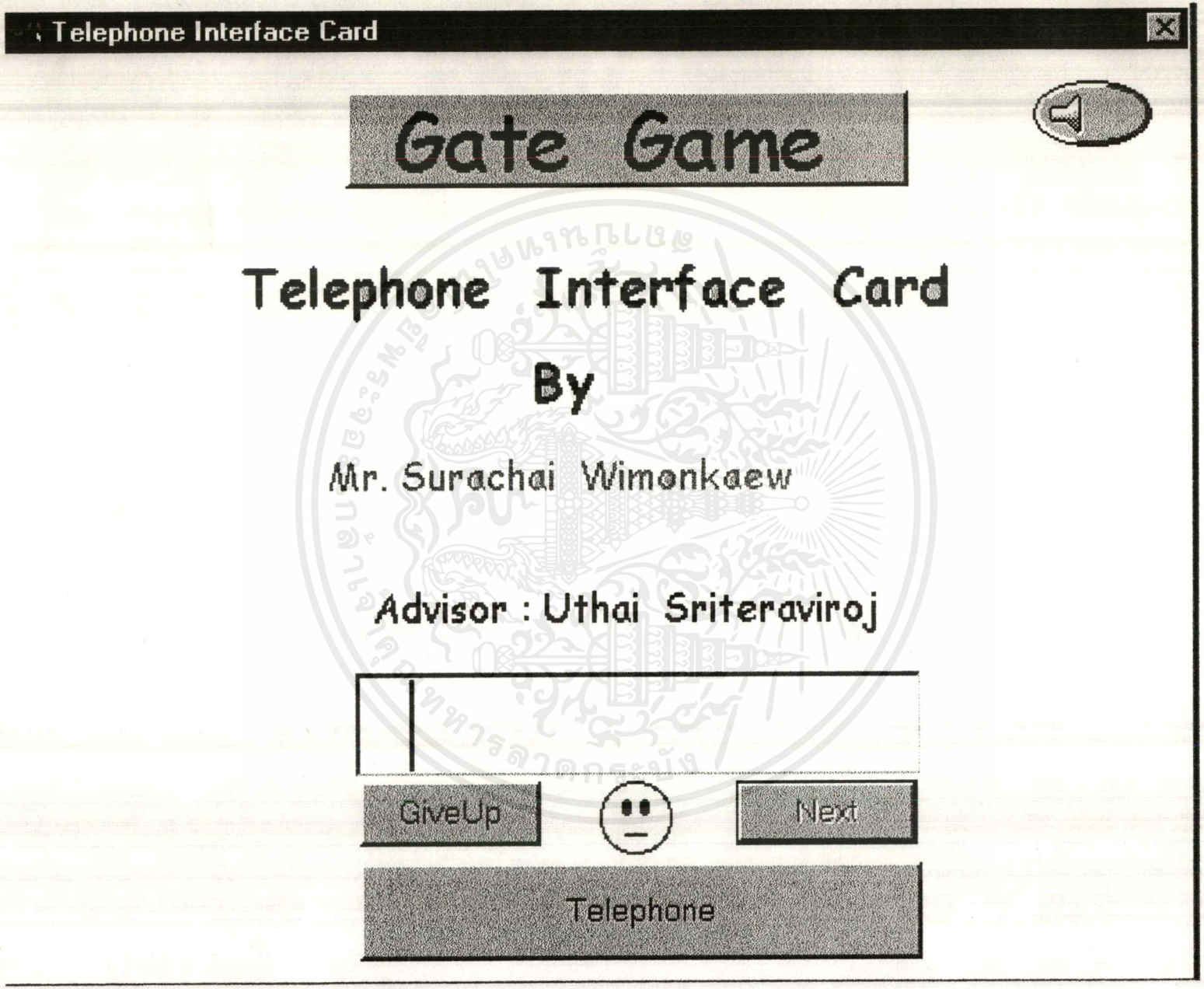
กิตติกรรมประกาศ

โครงงานนี้สามารถสำเร็จลุล่วงได้ด้วยดี ก็เนื่องจากอาจารย์ในภาควิชา ได้ให้การสนับสนุนด้วยดี และบุคคลที่สำคัญที่สุดคือ อาจารย์ อุทัย ศรีธีระวิโรจน์ อาจารย์ที่ปรึกษา ที่ให้การสนับสนุนและเป็นผู้ที่ให้คำชี้แนะในการทำงานอย่างดียิ่ง จึงเป็นแรงผลักดันให้ผู้จัดทำออกแบบและพัฒนาโครงงานชิ้นนี้ จนสุดความสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Telephone Interface Card
Gate Game
Telephone Interface Card
By
Mr. Surachai Wimonkaew
Advisor : Uthai Sriteraviroj
GiveUp
Next
Telephone

```
#include "stdafx.h"
#include <stdio.h>
#include <wchar.h>
#include <conio.h>
#include "speakn.h"
#include <mmsystem.h>

/////////////////////////////////////////////////////////////////////////////
// Sound helpers

static void PlaySound(LPCTSTR lpszSound)
{
    HRSRC hRes; // resource handle to wave file
    HGLOBAL hData;
    BOOL bOk = FALSE;
    if ((hRes = ::FindResource(AfxGetResourceHandle(), lpszSound,
      _T("sound"))) != NULL &&
      (hData = ::LoadResource(AfxGetResourceHandle(), hRes)) != NULL)
    {
        // found the resource, play it
        bOk = sndPlaySound((LPCTSTR)::LockResource(hData),
            SND_MEMORY|SND_SYNC|SND_NODEFAULT);
        FreeResource(hData);
    }
    if (!bOk)
    {
        static BOOL bReported = FALSE;
        if (!bReported)
        {
            AfxMessageBox(IDS_CANNOT_PLAY_SOUND);
            bReported = TRUE;       // once please
        }
    }
}

inline static void PlaySound(UINT nIDS)
    { PlaySound(MAKEINTRESOURCE(nIDS)); }

/////////////////////////////////////////////////////////////////////////////
// CSpeakNDlg

CSpeakNDlg::CSpeakNDlg(BOOL bNoPen)
    : CDialog(bNoPen ? IDD_NOPENDIALOG : CSpeakNDlg::IDD)
{
    //{{AFX_DATA_INIT(CSpeakNDlg)
        // NOTE: the ClassWizard will add member initialization here
    //}}AFX_DATA_INIT

    m_bNoPen = bNoPen;
    m_lpszNextQuestion = NULL;
    m_bNoAnswerCheck = FALSE;
}

BEGIN_MESSAGE_MAP(CSpeakNDlg, CDialog)
    //{{AFX_MSG_MAP(CSpeakNDlg)
    ON_COMMAND(IDC_REPLAY_SOUND, OnReplaySound)
    ON_EN_CHANGE(IDC_INPUT_EDIT, OnUpdateStatus)
    ON_COMMAND(IDC_GIVE_UP, OnGiveUp)
    ON_COMMAND(IDC_PICTURE, OnReplaySound)
    ON_BN_CLICKED(IDC_GETCARD, OnGetcard)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

BOOL CSpeakNDlg::LoadLesson(LPCTSTR lpLessonName)
{
    // load lesson from resource
    HRSRC hRes; // resource handle to lesson data
    HGLOBAL hData;
    if ((hRes = ::FindResource(AfxGetResourceHandle(), lpLessonName,
      _T("lesson"))) == NULL ||
      (hData = ::LoadResource(AfxGetResourceHandle(), hRes)) == NULL)
        return FALSE;
    m_lpszNextQuestion = (LPCSTR)::LockResource(hData);
    return TRUE;
}

void CSpeakNDlg::DoDataExchange(CDataExchange* pDX)
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CSpeakNDlg)
    DDX_Control(pDX, IDOK, m_buttonNext);
    //}}AFX_DATA_MAP
}


BOOL CSpeakNDlg::OnInitDialog()
{
    UpdateData(FALSE); // call DoDataExchange to initialize m_buttonNext
    m_buttonNext.EnableWindow(FALSE);

    ASSERT(m_targetWord.IsEmpty());     // not started yet

    // set the font of the prompt text to something bigger
    LOGFONT logfont;
    memset(&logfont, 0, sizeof(logfont));
    logfont.lfHeight = 40;
    logfont.lfWeight = FW_BOLD;
    CString szFont;
    szFont.LoadString(IDS_FONT_NAME);           // TrueType font
    lstrcpy(logfont.lfFaceName, szFont);
    VERIFY(m_biggerFont.CreateFontIndirect(&logfont));
    PromptText().SetFont(&m_biggerFont);
    InputEdit().SetFont(&m_biggerFont);

    // load the bitmaps for bitmap buttons
    VERIFY(m_replayButton.AutoLoad(IDC_REPLAY_SOUND, this));
    InputEdit().ShowWindow(FALSE);      // start with input disabled

    // load initial picture
    VERIFY(m_pictureButton.SubclassDlgItem(IDC_PICTURE, this));
    VERIFY(m_pictureButton.LoadBitmaps(_T("intro"), NULL, NULL));

    // Make the dialog visible, and update
    ShowWindow(TRUE);       // SHOW_OPENWINDOW
    UpdateWindow();

    PlaySound(IDSOUND_WELCOME);
    AdvanceLesson();
    return FALSE;       // focus already set
}

void CSpeakNDlg::OnReplaySound()
{
    InputEdit().SetFocus();
    PlaySound(m_targetRes);
}

void CSpeakNDlg::OnOK()
{
    // check results
    CString result;
    InputEdit().GetWindowText(result);
    if (result != m_targetWord)
    {
        PlaySound(IDSOUND_INCORRECT);
        AfxMessageBox(IDS_TRY_AGAIN);
        return;
    }
    PlaySound(IDSOUND_CORRECT);
    AdvanceLesson();
}

void CSpeakNDlg::OnGiveUp()
{
    PlaySound(IDSOUND_GIVEUP);
    SetAnswerText(m_targetWord);        // show answer
    OnReplaySound();
    AdvanceLesson();
}

void CSpeakNDlg::SetAnswerText(LPCTSTR lpsz)
{
    // setting the window text for an edit control will cause EN_CHANGE
    //  control notifications, so we lock them out while setting the
    //  text programmatically
```


ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ASSERT(!m_bNoAnswerCheck);
    m_bNoAnswerCheck = TRUE;
    InputEdit().SetWindowText(lpsz);
    m_bNoAnswerCheck = FALSE;
}

////////////////////////////////////////////////////////////////////////////
// Advancing to the next lesson

void CSpeakNDlg::AdvanceLesson()
{
    if (*m_lpszNextQuestion == '\0')
    {
        // out of questions
        PlaySound(IDSOUND_GOODBYE);
        EndDialog(IDOK);
        return;
    }

// The lesson resource consists of several pairs of words (that the user guesses)
// and bitmap resource names.  The words and resource names are stored in the
// resource as ANSI text.  This text must be converted to UNICODE.

#ifdef _UNICODE
    TCHAR szT[20];
    int nLen;
    nLen = strlen(m_lpszNextQuestion);
    mbstowcs(szT, m_lpszNextQuestion, nLen);
    szT[nLen] = 0;
    m_targetWord = szT;
    m_lpszNextQuestion += nLen + 1;

    nLen = strlen(m_lpszNextQuestion);
    mbstowcs(szT, m_lpszNextQuestion, nLen);
    szT[nLen] = 0;
    m_targetRes = szT;
    m_lpszNextQuestion += nLen + 1;
#else
    m_targetWord = m_lpszNextQuestion;
    m_lpszNextQuestion += m_targetWord.GetLength() + 1;

    m_targetRes = m_lpszNextQuestion;
    m_lpszNextQuestion += m_targetRes.GetLength() + 1;
#endif

    m_targetWord.MakeUpper();
    m_targetRes.MakeUpper();

    PlaySound(IDSOUND_QUESTION);

    // draw the picture (bitmap with the same name as the target)
    if (!m_pictureButton.LoadBitmaps(m_targetRes))
    {
        AfxMessageBox(IDS_PICTURE_UNAVAILABLE);
        VERIFY(m_pictureButton.LoadBitmaps(_T("intro"), NULL, NULL));
            // go back to the initial bitmap
    }
    m_pictureButton.Invalidate(TRUE);

    SetAnswerText(_T(""));
    InputEdit().ShowWindow(TRUE);

    OnUpdateStatus();               // set appropriate face
    OnReplaySound();                // ask question

    if (*m_lpszNextQuestion == '\0')
    {
        CString strDone;
        strDone.LoadString(IDS_DONE);
        m_buttonNext.SetWindowText(strDone);
    }
}

////////////////////////////////////////////////////////////////////////////
// Happy face status indicator

void CSpeakNDlg::OnUpdateStatus()
{
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CString result;
    InputEdit(),GetWindowText(result);

    UINT nIDI = IDI_FACE_NEUTRAL;        // default
    if (result == m_targetWord)
    {
        m_buttonNext.EnableWindow();
        nIDI = IDI_FACE_HAPPIER;         // exact match
    }
    else
    {
        m_buttonNext.EnableWindow(FALSE);
        if (result.IsEmpty())
            nIDI = IDI_FACE_NEUTRAL;         // not started yet
        else
        {
            nIDI = (_tcsncmp(m_targetWord, result, _tcslen(result)) == 0?
                IDI_FACE_HAPPY : IDI_FACE_SAD);
        }
    }

    HICON hNew = ::LoadIcon(AfxGetResourceHandle(), MAKEINTRESOURCE(nIDI));
    ASSERT(hNew != NULL);
    ::DestroyIcon(StatusFace().SetIcon(hNew));
    UpdateWindow();                  // draw everything now

    if (m_bNoAnswerCheck)
        return;     // don't update

    if (nIDI == IDI_FACE_HAPPIER)
    {
        // exact match - automatic advance
        OnReplaySound();
        PlaySound(IDSOUND_CORRECT);
    }
}

////////////////////////////////////////////////////////////////////////////
// CSpeakNApp

BOOL CSpeakNApp::InitInstance()
{
    Enable3dControls();

    BOOL bNoPen = TRUE;  // no pen-aware controls

    // Creates a simple dialog and do it
    CSpeakNDlg mainDlg(bNoPen);
    if (!mainDlg.LoadLesson(_T("SAMPLE1")))
        return FALSE;
    m_pMainWnd = &mainDlg;
    mainDlg.DoModal();

    // that's all, quit app
    ::PostQuitMessage(0);
    return TRUE;
}

CSpeakNApp NEAR theApp;

////////////////////////////////////////////////////////////////////////////

void CSpeakNDlg::OnGetcard()
{
    int data ;
    _outpd(0x303,0x90);
    CDialog about(IDD_Answer);
    about.DoModal();
    data = 30 ;
    printf("\tData Input : %d\n", data);
    do
    {
        data=_inpd(0x301);
        printf("\tData Input : %d\n", data);
    }
    while (data == 0xff);
    printf("\tData Input : %d\n", data);
```

ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//////////////////////////////////////////////////////////////////////////////
// Lesson data - included by SPEAKN.RC

#ifdef APSTUDIO_INVOKED
#error : Do not edit this file with App Studio - edit SPEAKN.RC instead
#endif

SAMPLE1 lesson PRELOAD
BEGIN
    // list of target words
    "AND\0AND\0",
    "OR\0OR\0",
    "NOT\0NOT\0",
    "NAND\0NAND\0",
    0 // end of data
END

; AND
AND                 bitmap      res\AND.dib
AND                 sound       res\Q_AND.WAV

; OR
OR                  bitmap      res\OR.dib
OR                  sound       res\Q_OR.wav

; NOT
NOT                 bitmap      res\NOT.dib
NOT                 sound       res\Q_NOT.wav

; NAND
NAND                bitmap      res\NAND.dib
NAND                sound       res\Q_NAND.wav

//////////////////////////////////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CString result;
    InputEdit().GetWindowText(result);

    UINT nIDI = IDI_FACE_NEUTRAL;        // default
    if (result == m_targetWord)
    {
        m_buttonNext.EnableWindow();
        nIDI = IDI_FACE_HAPPIER;         // exact match
    }
    else
    {
        m_buttonNext.EnableWindow(FALSE);
        if (result.IsEmpty())
            nIDI = IDI_FACE_NEUTRAL;         // not started yet
        else
        {
            nIDI = (_tcsncmp(m_targetWord, result, _tcslen(result)) == 0?
                IDI_FACE_HAPPY : IDI_FACE_SAD);
        }
    }

    HICON hNew = ::LoadIcon(AfxGetResourceHandle(), MAKEINTRESOURCE(nIDI));
    ASSERT(hNew != NULL);
    ::DestroyIcon(StatusFace().SetIcon(hNew));
    UpdateWindow();                      // draw everything now

    if (m_bNoAnswerCheck)
        return;     // don't update

    if (nIDI == IDI_FACE_HAPPIER)
    {
        // exact match - automatic advance
        OnReplaySound();
        PlaySound(IDSOUND_CORRECT);
    }
}

////////////////////////////////////////////////////////////////////////////
// CSpeakNApp

BOOL CSpeakNApp::InitInstance()
{
    Enable3dControls();

    BOOL bNoPen = TRUE;  // no pen-aware controls

    // Creates a simple dialog and do it
    CSpeakNDlg mainDlg(bNoPen);
    if (!mainDlg.LoadLesson(_T("SAMPLE1")))
        return FALSE;
    m_pMainWnd = &mainDlg;
    mainDlg.DoModal();

    // that's all, quit app
    ::PostQuitMessage(0);
    return TRUE;
}

CSpeakNApp NEAR theApp;

////////////////////////////////////////////////////////////////////////////

void CSpeakNDlg::OnGetcard()
{
    int data ;
    _outpd(0x303,0x90);
    CDialog about(IDD_Answer);
    about.DoModal();
    data = 30 ;
    printf("\tData Input : %d\n", data);
    do
    {
        data=_inpd(0x301);
        printf("\tData Input : %d\n", data);
    }
    while (data == 0xff);
    printf("\tData Input : %d\n", data);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//////////////////////////////////////////////////////////////////////////////
// Sound data - included by SPEAKN.RC

#ifdef APSTUDIO_INVOKED
#error : Do not edit this file with App Studio - edit SPEAKN.RC instead
#endif

IDSOUND_WELCOME     sound       res\welcome.wav
IDSOUND_QUESTION    sound       res\question.wav
IDSOUND_CORRECT     sound       res\correct.wav
IDSOUND_INCORRECT   sound       res\incorrec.wav
IDSOUND_GIVEUP      sound       res\giveup.wav
IDSOUND_GOODBYE     sound       res\goodbye.wav
```

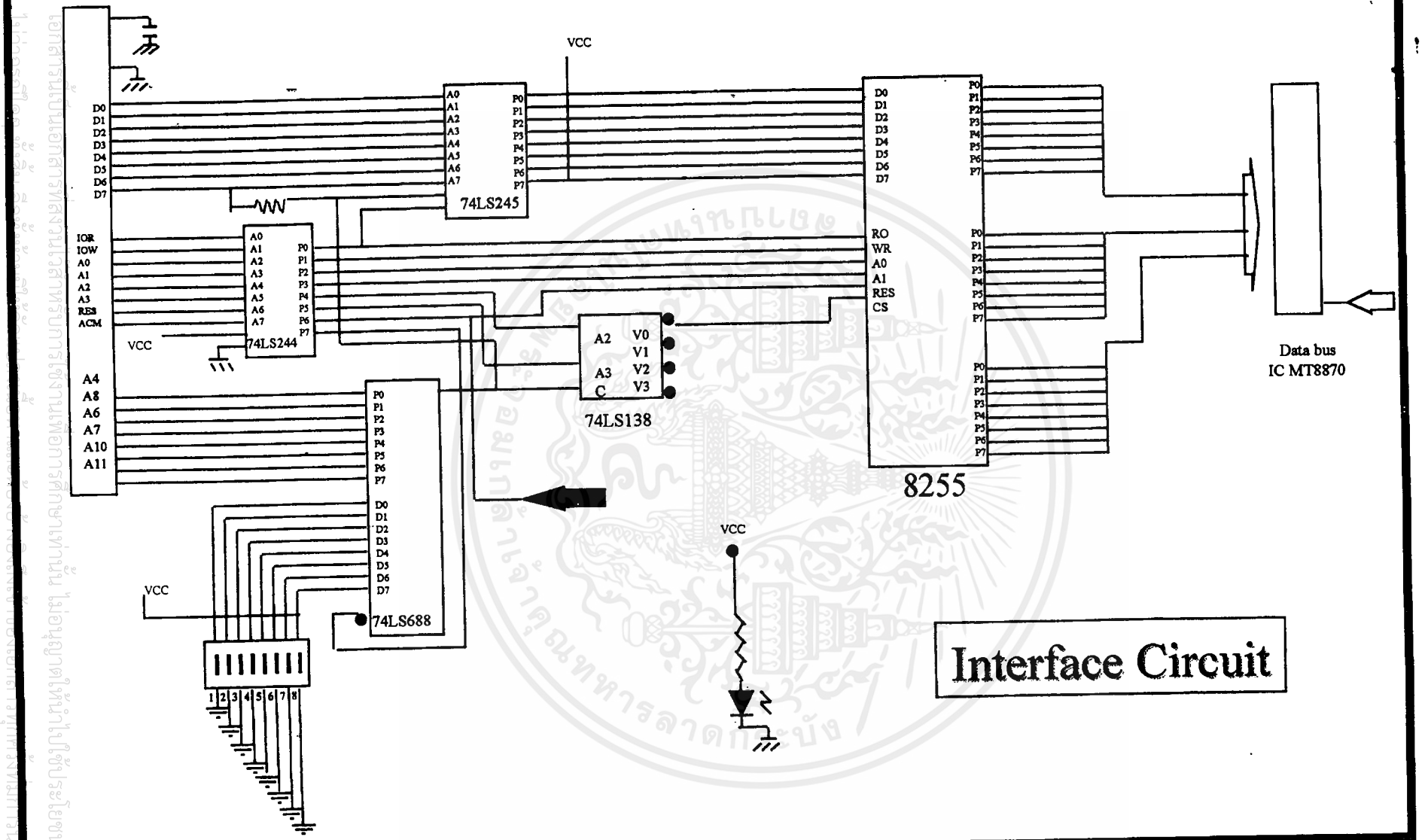
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

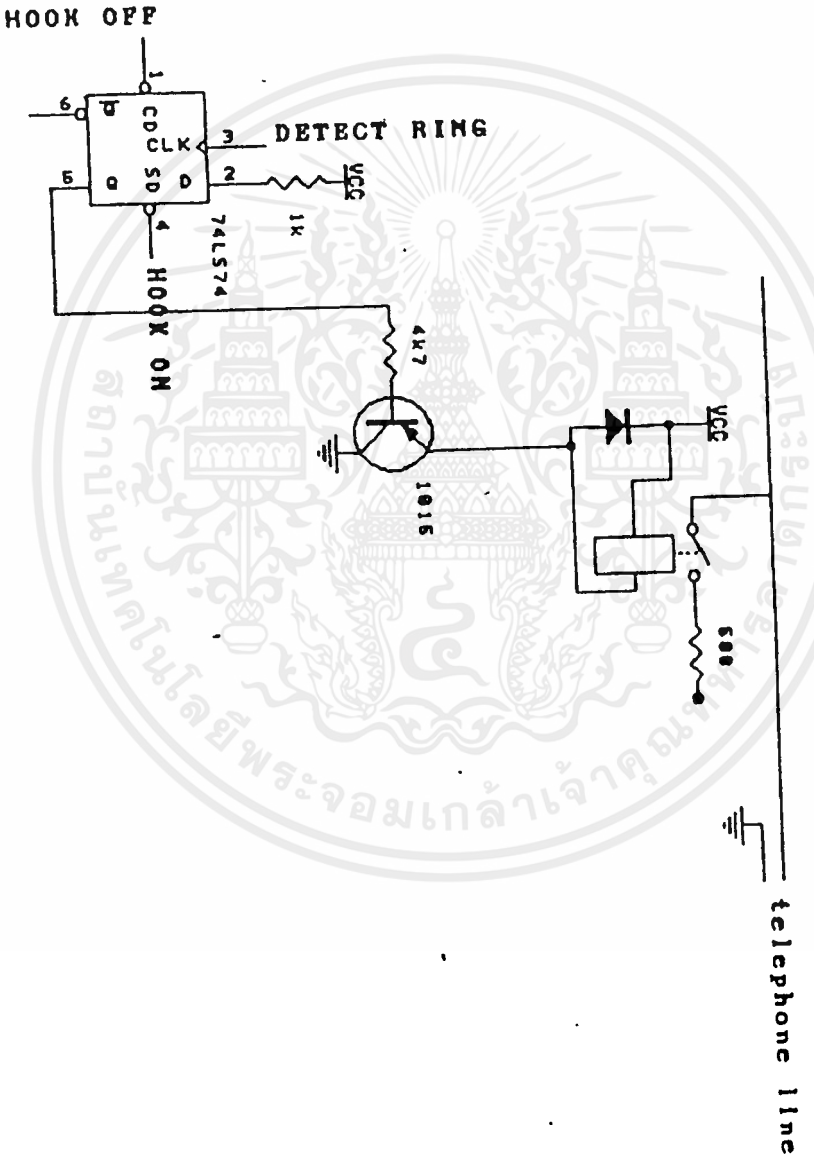
```
// speakn.h : Declares the class interfaces for the SpeakN application.
//
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (C) 1992-1995 Microsoft Corporation
// All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and related
// electronic documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

#include "resource.h"       // resource IDs

/////////////////////////////////////////////////////////////////////////////
// CSpeakNDlg:
//   The main user interface to this application is just one big dialog

class CSpeakNDlg : public CDialog
{
// Constructors
public:
    CSpeakNDlg(BOOL bNoPen);
    BOOL LoadLesson(LPCTSTR lpLessonName);

    //{{AFX_DATA(CSpeakNDlg)
    enum { IDD = IDD_PENDIALOG };
    CButton m_buttonNext;
    //}}AFX_DATA

// Operations
    void        AdvanceLesson();

// Attributes (the current question/lesson)
    // current question (from resource)
    CString         m_targetWord;           // target word (upper case)
    CString         m_targetRes;

    // resource containing a list of target words and associated bitmap names, in ANSI
    LPCSTR          m_lpszNextQuestion; // empty string => done

// Implementation
protected:
    void DoDataExchange(CDataExchange* pDX);

    BOOL m_bNoPen;      // TRUE => no PenWindows (run less functional)
    BOOL m_bNoAnswerCheck;  // TRUE => disable test for answer

    // special controls
    CFont           m_biggerFont;
    CBitmapButton   m_pictureButton; // main bitmap done as a special button
    CBitmapButton   m_replayButton; // bitmap button for replay sound

    // normal Windows controls
    CStatic&    StatusFace()    // face which smiles or frowns
                    { return *(CStatic*)GetDlgItem(IDC_STATUS_FACE); }
    CStatic&    PromptText()    // prompt area
                    { return *(CStatic*)GetDlgItem(IDC_PROMPT_TEXT); }

    void        SetAnswerText(LPCTSTR lpsz);
    CEdit&      InputEdit()     // user input for answer text
                                //  (a normal Edit or a Pen Boxed Edit)
                    { return *(CEdit*)GetDlgItem(IDC_INPUT_EDIT); }

    // message handlers
    //{{AFX_MSG(CSpeakNDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnReplaySound();
    afx_msg void OnUpdateStatus();
    virtual void OnOK();        // Guess
    virtual void OnGiveUp();
    afx_msg void OnGetcard();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////
// CSpeakNApp:
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

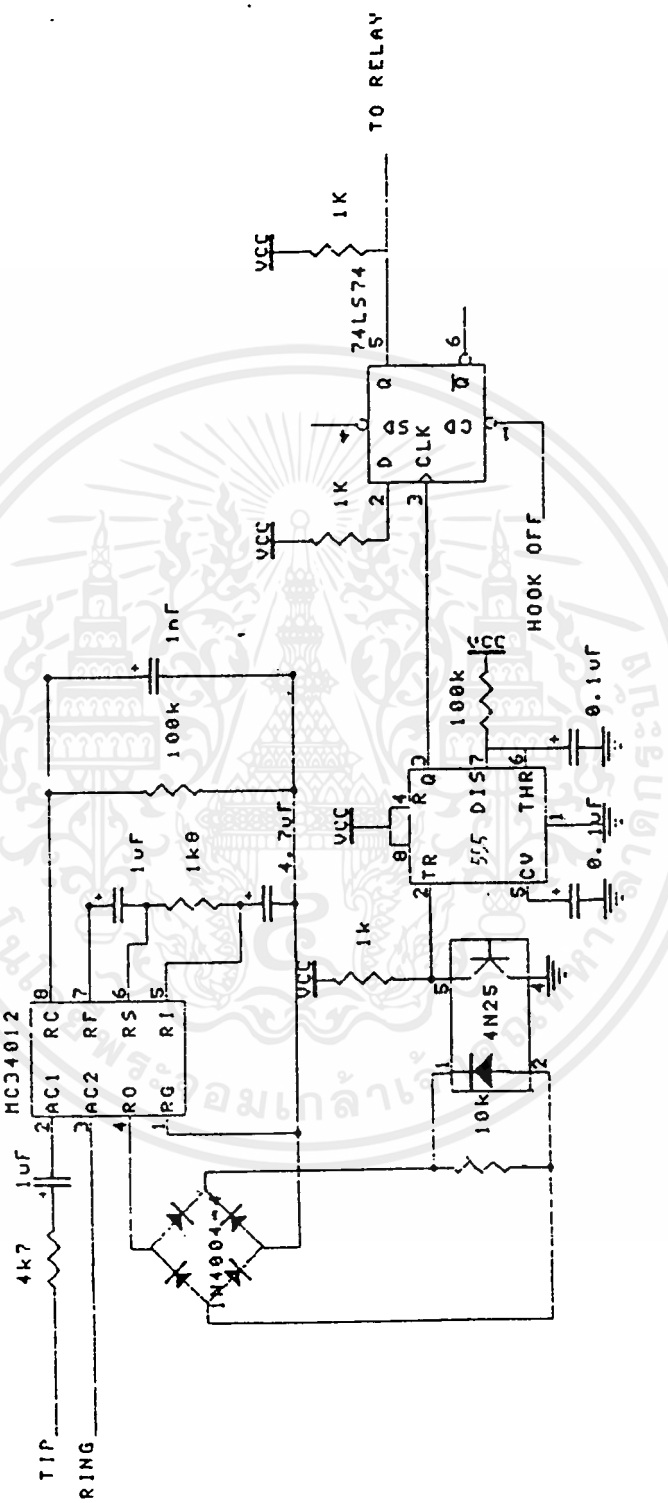
```
class CSpeakNApp : public CWinApp
{
public:
    BOOL InitInstance();
};

/////////////////////////////////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
74LS245
74LS244
74LS138
74LS688
8255
Data bus
IC MT8870
Interface Circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงวงจรรวมที่ใช้งานในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้