



ปีการศึกษา 2533

ระบบควบคุมอุปกรณ์นำสายไฟฟ้า  
(Equipment Control By AC Line)

โดย

นายเหรียญชัย เญญาศุภวรรณ 32.6336 2-0

นายอภิรักษ์ ภาரசสาร 32.6338 2-0

อาจารย์ที่ปรึกษา

อาจารย์วิทยา ทิพย์สุวรรณพร



ปริญญาโทปีการศึกษา 2533

เรื่อง ระบบควบคุมอุปกรณ์ผ่านสายไฟฟ้า

Equipment control by AC line

ผู้จัดทำ

1. นายเพ็ญชัย เญจาคุภวรรณ 32.6336

2. นายอภิสิทธิ์ การสาร 32.6338

.....อาจารย์ที่ปรึกษา

( อาจารย์ วิทยา ทิพย์สุวรรณพร )

เลขหมั T 33041 44
เลขทะเบียน 027874
วัน, เดือน, ปี 12 ก.ค. 34

# ระบบควบคุมอุปกรณ์ผ่านสายไฟฟ้า

เหรียญชัย เณรจาตุรกรรม  
อภิชัย การสาร  
อ.วิทยา ทิพย์สุวรรณพร  
ปีการศึกษา 2593

## บทคัดย่อ

ระบบควบคุมอุปกรณ์ผ่านสายไฟฟ้านี้ เป็นโครงการที่สร้างขึ้นเพื่อเพิ่ม  
ความสะดวกในการควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ไม่ว่าจะในบ้าน หรือโรงงานอุตสาหกรรม  
โดยลักษณะของโครงการเป็นการประยุกต์ใช้ ระบบไมโครโปรเซสเซอร์ในการควบคุม  
ซึ่งสัญญาณควบคุมจะ ใช้การผสมสัญญาณควบคุมเข้ากับคลื่นพาห์ แล้วส่งเข้าไปในสาย  
ไฟฟ้าทางด้านตัวแม่ ทำการรับและแยกสัญญาณออกจากคลื่นพาห์ทางตัวลูก สัญญาณ  
ควบคุมที่ได้ก็จะส่งให้กับ ระบบไมโครโปรเซสเซอร์ที่ใช้ในการประมวลผลสัญญาณและ  
ส่งสัญญาณควบคุมอุปกรณ์ให้กับชุดควบคุมซึ่งมี 2 ลักษณะคือ ควบคุมกำลังไฟฟ้าได้ และ  
ควบคุมกำลังไฟฟ้าไม่ได้ (ปิดหรือเปิดเท่านั้น)

.....

# Equipment Control By AC Line

Rianchai Benjasupawan

Apiyuch Bharasarn

Vitthaya Thipsuwanporn

Academic Year 2533

## ABSTRACT

Equipment control by AC line is the project that more comfortable electric equipment's control in home or industry. Outline of this project is applied of microprocessor base control system. Control signal is modulated with carrier and sent it into the AC line at master section. The slave section receives the control signal and demodulated it from carrier. Microprocessor processes the signal and send it to the controller, that have 2 types of control, one is ON-OFF only and another can control equipment's power.

.....

## สารบัญ

บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีที่นำมาประยุกต์ใช้	3
2.1	เฟสล็อกคูล	3
2.2	การนำเฟสล็อกคูลมาประยุกต์ใช้กับการส่งข้อมูล	4
2.3	การมอดคูลเลทสัญญาณควบคุมกับวงจรกำเนิดความถี่ที่ควบคุมด้วยแรงดันไฟฟ้า	4
2.4	การดีมอดคูลเลทสัญญาณควบคุมโดยใช้เฟสล็อกคูล	5
2.5	ระบบสัญญาณโทรศัพท์แบบ DTMF	6
2.6	การเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก	7
2.7	การทริก(จุดชนวนทำงาน)ของไตรแอก	8
2.8	สัญญาณรบกวนความถี่วิทยุ	9
2.9	เทคนิคการใช้จุดตัดศูนย์	9
2.10	การควบคุมกำลังไฟแบบเฟสทริกเกอร์	11
2.11	วงจร Snubber	12
2.12	VDR อุปกรณ์ป้องกันแรงดันกระชอก	13
บทที่ 3	การคำนวณและการสร้าง	17
3.1	GUARD TIME ADJUSTMENT ของ 8870	17
3.2	โครงสร้างของระบบที่ใช้ในการทำงาน	19
3.3	การทำงานของภาคควบคุมอุปกรณ์ ON-OFF	25
3.4	การทำงานของภาค Digital to Analog convertor	26
3.5	การทำงานของภาคควบคุมมุมเฟส	27
3.6	ขั้นตอนการติดต่อระหว่าง MASTER และ SLAVE	32
บทที่ 4	การทดลองและผลการทดลอง	34
4.1	การทดลองวงจรรับส่งสัญญาณ DTMF	34
4.2	การทดลองรับสัญญาณที่ระยะต่างกัน	34
4.3	การทดลองภาคควบคุมมุมเฟส	36
บทที่ 5	บทวิจารณ์และสรุป	38

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก	39
ก. MCS 51 ไมโครคอนโทรลเลอร์แบบซีพียูเดี่ยว	40
ข. MT 8870 ไอซีถอดรหัสความถี่โทรทัศน์	46
ค. TCA 785 ไอซีควบคุมเฟส	51
ง. FLOW CHART ของ MASTER	54
จ. FLOW CHART ของ SLAVE	64
ฉ. โปรแกรมควบคุมส่วน MASTER	71
ช. โปรแกรมควบคุมส่วน SLAVE	92
กิตติกรรมประกาศ	105
หนังสืออ้างอิง	106



## บทที่ 1

### บทนำ

การใช้งานอุปกรณ์ไฟฟ้าภายในบ้านตามปกติ จะใช้การควบคุมกำลังงานที่ป้อนให้กับอุปกรณ์ด้วยสวิตช์ตัดต่อและสายไฟฟ้าซึ่งการใช้งาน จะต้องทำการควบคุมที่ตัวสวิตช์ซึ่งอยู่กับที่โดยตรง ดังนั้นจึงไม่สะดวกถ้าต้องการควบคุมอุปกรณ์ในขณะที่อยู่ห่างจากตัวสวิตช์ เช่น ในกรณีที่ต้องการควบคุมหลอดไฟฟ้าที่ชั้นล่าง ในขณะที่อยู่ที่ชั้น 3 อย่างนี้เป็นต้น

กรณีที่ต้องการควบคุมอุปกรณ์ที่อยู่ห่างออกไป จะต้องใช้ชุดอุปกรณ์ควบคุมระยะไกล (Remote Control) ซึ่งโดยทั่วไป จะใช้การควบคุมโดยส่งสัญญาณควบคุมไปกับตัวนำอื่น ๆ ที่สามารถพาสัญญาณควบคุมไปได้ไกล เช่น

- การส่งสัญญาณควบคุมโดยใช้แสงอินฟราเรด ซึ่งสามารถควบคุมอุปกรณ์ในระยะที่ไม่ห่างมากนักเพราะแสงอินฟราเรดจะสะท้อนวัตถุไม่ทะลุผ่านได้ การใช้งานก็เช่น การควบคุมเครื่องรับโทรทัศน์ เครื่องเล่นเทปบันทึกภาพดังที่เห็นใช้งานทั่วไปในปัจจุบัน

- การส่งสัญญาณควบคุมโดยใช้คลื่นวิทยุ สามารถทะลุวัตถุไปได้บ้าง ระยะการใช้งานขึ้นอยู่กับกำลังส่งของเครื่องส่ง แต่ปัญหาอยู่ที่กฎหมายควบคุมเกี่ยวกับความถี่ในเมืองไทยค่อนข้างจำกัดการใช้งาน และความถี่ส่วนเกินที่อาจเล็ดลอดออกไปรบกวนบ้านข้างเคียงที่อาจใช้คลื่นความถี่ที่เหมือนกัน

- การส่งสัญญาณควบคุม โดยการใช้สายไฟฟ้าภายในบ้าน (AC Line) เป็นตัวนำสัญญาณ สามารถทำการควบคุมได้ในทุก ๆ ตำแหน่งของตัวบ้านที่มีสายไฟไปถึง ลดการเดินสายเพื่อต่อพ่วงกับสวิตช์มากมาย ตัดปัญหาเรื่องการกวนกันกับบ้านข้างเคียง เนื่องจากสามารถปิดกั้นสัญญาณได้ด้วย Line noise filter ซึ่งจะทำให้การ coupling สัญญาณที่ไม่ใช่ 50 Hz ไม่ให้ออกไปจากตัวบ้าน และยังสามารถป้องกันสัญญาณรบกวนจากภายนอกไม่ให้เข้ามาภายในตัวบ้านด้วย ซึ่งโครงการนี้ก็ได้นำระบบนี้มาใช้

ลักษณะของโครงการจะมีชุดควบคุมจำนวน 2 ชุด โดยชุดแรกทำหน้าที่เป็นตัวแม่ ส่งสัญญาณควบคุมไปยังตัวลูก โดยทำการส่งผ่านสัญญาณควบคุมทางสายไฟฟ้า

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดที่สองทำหน้าที่เป็นตัวลูกรับสัญญาณที่ส่งผ่านจากตัวแม่มาทางสายไฟฟ้า แล้วแยกสัญญาณควบคุมออกเพื่อไปใช้ในการควบคุมต่อไป โดยชุดควบคุมตัวลูกนี้จะมีได้จำนวนหลายชุดในการนำไปใช้งาน สามารถทำผู้คิดได้ด้วยโปรแกรมที่ใช้กับชุดควบคุมตัวแม่ โดยตัวลูกจะสามารถเปลี่ยนการกำหนดหมายเลขประจำตำแหน่งของตนเองได้จากการเปลี่ยนคิวิตซ์กำหนดหมายเลขตำแหน่ง โดยไม่ต้องเปลี่ยนตัวโปรแกรมแต่อย่างใดทั้งสิ้น

### วัตถุประสงค์ของโครงการ

1. ศึกษากระบวนการส่งข้อมูล (สัญญาณควบคุม) โดยใช้การผสมข้อมูลกับความถี่พาห์ และการแยกสัญญาณข้อมูลออกจากความถี่พาห์
2. ศึกษากระบวนการความถี่โทรคัมภ์ (Dual Tone Multi Frequency : DTMF) และการใช้งาน IC เข็มรหัสและถอดรหัสสัญญาณ
3. ศึกษาการส่งข้อมูลสัญญาณควบคุมโดยใช้สายไฟสลั๊บ 220 โวลท์ภายในบ้านเป็นตัวนำในการส่งข้อมูล
4. ศึกษาวงจรป้องกันสัญญาณรบกวนภายในสายไฟสลั๊บ 220 โวลท์
5. ศึกษาการใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS 51
6. ศึกษาการทำงานของระบบไมโครโปรเซสเซอร์ในการควบคุมอุปกรณ์กำลังที่ใช้ไฟสลั๊บ 220 โวลท์
7. สร้างระบบควบคุมอุปกรณ์กำลังที่ใช้ไฟสลั๊บ 220 โวลท์โดยใช้การรับส่งสัญญาณควบคุมผ่านสายไฟสลั๊บ 220 โวลท์

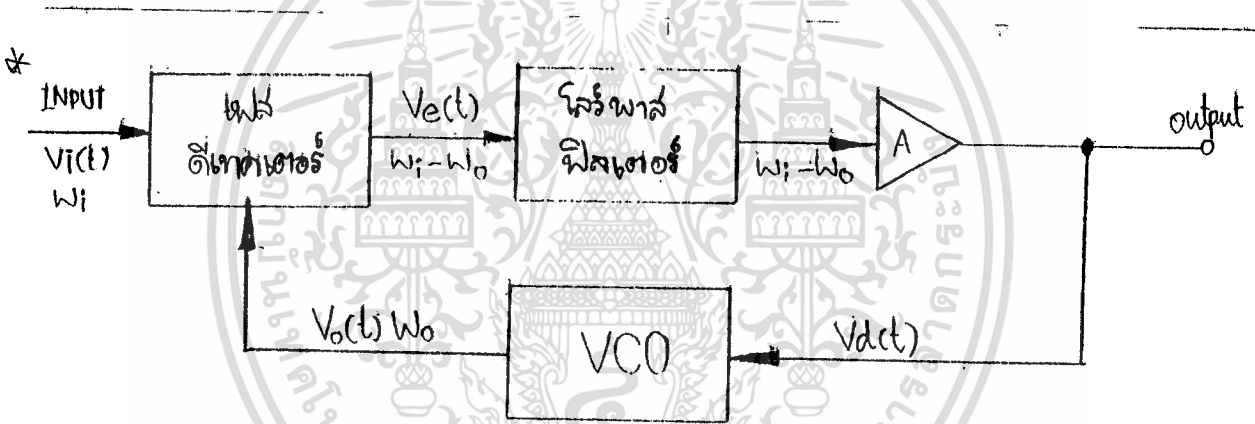
.....

ทฤษฎีที่นำมาประยุกต์ใช้

2.1 เฟสล็อกคัล (Phase Lock Loop : PLL)

ประกอบด้วย 3 ส่วนคือ

1. เฟสดีเทคเตอร์ (Phase Detector)
2. โลว์พาสฟิลเตอร์ (Low Pass Filter)
3. ความถี่ที่ควบคุมด้วยแรงดัน (Voltage Controlled Oscillator : VCO)



รูปที่ 2.1 Block Diagram ของวงจร Phase Lock Loop

$V_i(t)$  = สัญญาณอินพุต

$V_e(t)$  = โวลต์เตจคลาดเคลื่อน (Error voltage) ซึ่งเป็นผลคูณของ  $V_i(t)$  กับ  $V_o(t)$  แล้วทำให้เกิดผลบวกและผลต่างของความถี่ ( $\omega_i + \omega_o$ )

$V_d(t)$  = โวลต์เตจคลาดเคลื่อนที่มีความถี่  $\omega_i - \omega_o$  ป้อนให้ควบคุม VCO เพื่อเปลี่ยน ความถี่  $\omega_o$  ให้เข้าใกล้  $\omega_i$

การทำงานของวงจร

เมื่อยังไม่มีสัญญาณทางเข้า ป้อนให้กับระบบโวลต์เตจควบคุม  $V_d(t)$

แอกจะ เป็นศูนย์ สาร VCO จะทำงานตามความถี่การ  $\omega_o$  ที่ตั้งไว้ ไม่ ถ้ามีสัญญาณทางเข้าป้อนให้กับระบบ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเฟสคอมพาราเตอร์จะไปเปรียบเทียบเฟส และความถี่ของสัญญาณ VCO กับอินพุต แล้วสร้างโวลต์ที่แตกคลาดเคลื่อน  $V_e(t)$  แรงดันคลาดเคลื่อนนี้จะถูกกรองในโลว์พาสฟิลเตอร์ให้ความถี่ต่ำผ่านได้ จากนั้นจะขยาย แล้วป้อนให้กับ VCO ในลักษณะเช่นนี้ โวลต์ที่แตกควบคุม  $V_d(t)$  จะไปบังคับความถี่ของ VCO แปรไปในทิศทางที่ลดความแตกต่าง ระหว่างความถี่  $\omega_0$  กับความถี่  $\omega_i$  ลง จากนั้นด้วยคุณสมบัติของการป้อนกลับของเฟสล็อกคัลป์ ถ้าความถี่อินพุต  $\omega_i$  ใกล้เคียงกับ  $\omega_0$  ก็จะทำให้ VCO ซิงโครไนซ์ หรือล็อกกับสัญญาณอินพุตได้ เมื่ออยู่ในสภาวะล็อกอย่างนี้ความถี่ของ VCO จะเท่ากันกับ ความถี่ของสัญญาณอินพุต แต่ต่างเฟสกันเล็กน้อย เฟสที่ต่างกันนี้จะทำให้ความถี่ free running ของ VCO เลื่อนเข้าไปใกล้ค่าความถี่สัญญาณอินพุต  $\omega_i$  เพื่อรักษาสภาพการล็อกเอาไว้ให้ได้

**2.2 การนำเฟสล็อกคัลป์มาประยุกต์ใช้งานกับการส่งข้อมูล**

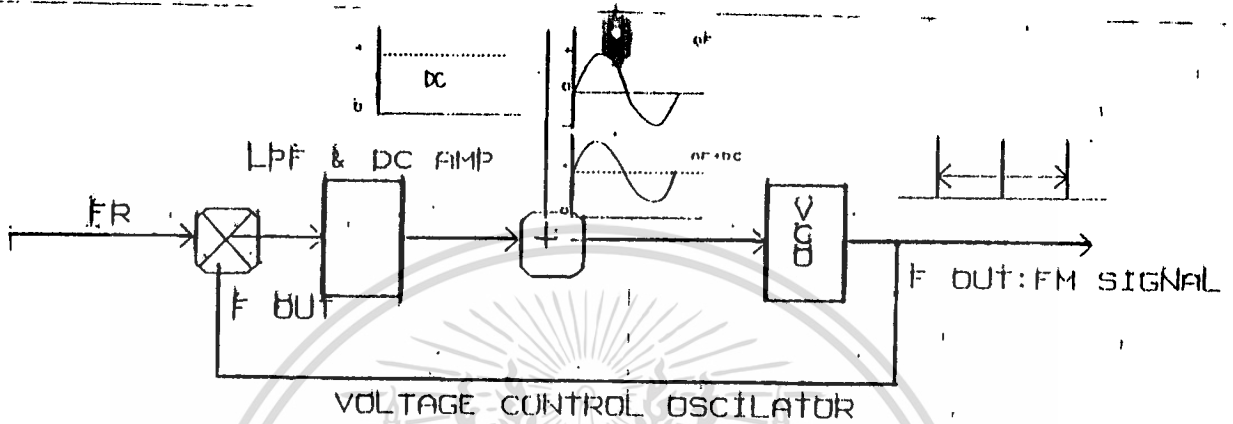
เฟสล็อกคัลป์ (Phase Lock Loop) คือระบบควบคุมแบบป้อนกลับ (Feedback Control) แบบหนึ่งที่สามารถเลือกความถี่ได้ด้วยการซิงโครไนส์ (Synchronize) กับความถี่ของสัญญาณอินพุตที่ต้องการเลือกและแทร็ค (track) ตามการเปลี่ยนแปลงของสัญญาณความถี่อินพุตมานั้น ๆ จากหลักการดังกล่าวสามารถนำเฟสล็อกคัลป์ไปประยุกต์ใช้งานต่าง ๆ มากมาย เช่น โทนีดีเทคเตอร์ (Tone Detector) การมอดคูลูเลชัน และดีมอดคูลูเลชันทางความถี่ (Frequency Modulation and Demodulation) เป็นต้น ซึ่งโครงงานนี้ได้นำเฟสล็อกคัลป์มาใช้ในการมอดคูลูเลชัน และดีมอดคูลูเลชันสัญญาณควบคุม โดยใช้ IC สำเร็จรูป ของ NS เบอร์ LM 565

**2.3 การมอดคูลูเลชันสัญญาณควบคุมกับวงจรกำเนิดความถี่ที่ควบคุมโดยแรงดันไฟฟ้า**

[Voltage Control Oscillator : VCO]

โดยทั่วไปวงจร VCO สามารถเปลี่ยนแปลงความถี่ตามระดับแรงดันไฟตรงที่ป้อนให้กับวงจร VCO ในขณะเดียวกันถ้านำสัญญาณเสียง (สัญญาณควบคุม) มารวมกับแรงดันไฟตรง จะทำให้สัญญาณเสียงถูกยกกระดบตามระดับแรงดันไฟตรง แล้วนำสัญญาณผสมระหว่างสัญญาณเสียงกับแรงดันไฟตรงป้อนให้วงจร VCO เพื่อควบคุมการ

ผลิตความถี่ของวงจร VCO ให้เปลี่ยนแปลงตามระดับแรงดันไฟตรงและสัญญาณเสียง จะเรียกหลักการนี้ว่าการมอดดูเลทสัญญาณแบบความถี่ (Frequency Modulation: FM) ดังแสดงในรูปที่ 2.2

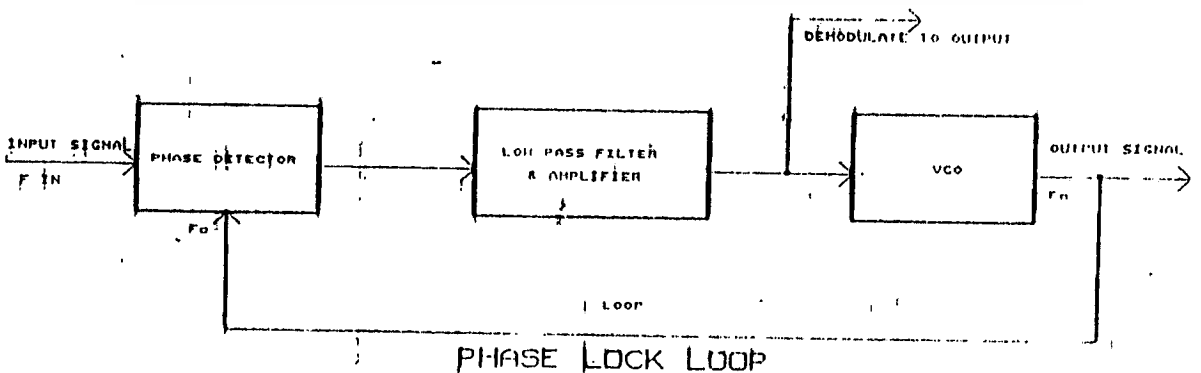


รูปที่ 2.2 การนำ PLL มาใช้ในการมอดดูเลท

จากรูปที่ 2.2 จะนำเอาเฟสล็อกคูลมาประยุกต์ใช้งานในการมอดดูเลทสัญญาณเสียงกับสัญญาณไฟตรง โดยนำสัญญาณความถี่ทางเอาท์พุทย้อนกลับมาเปรียบเทียบกับแรงดันไฟตรงไปผสมกับสัญญาณเสียง ป้อนให้วงจร VCO เพื่อผลิตสัญญาณมอดดูเลทต่อไป

### 2.4 การตีมอดดูเลทสัญญาณควบคุมโดยใช้เฟสล็อกคูล

การตีมอดดูเลทเป็นการแยกสัญญาณเสียง (หรือสัญญาณข้อมูล) ออกจากความถี่คลื่นพาห้ (Carrier) ที่ส่งมาจากเครื่องส่ง ซึ่งสามารถใช้หลักการของเฟสล็อกคูลมาช่วยทำการตีมอดดูเลทได้ดังรูป



รูปที่ 2.3 การตีมอดดูเลทด้วย PLL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของเฟสล็อกคัลป์ในการติมอดคูล์ทคือเมื่อมีสัญญาณมอดคูล์ท (Fin) เข้ามาที่วงจรเปรียบเทียบบเฟส ถ้าสัญญาณที่เข้ามามีเฟสไม่ตรงกับความถี่เอาท์พุทของวงจร VCO (F<sub>o</sub>)จากการป้อนกลับเข้ามาแล้ว จะทำให้เกิดสัญญาณความคลาดเคลื่อนออกไปทางเอาท์พุทของวงจรเปรียบเทียบบเฟส ขนาดของสัญญาณความคลาดเคลื่อนนี้ขึ้นอยู่กับความต่างเฟสของสัญญาณอินพุททั้งสองเปรียบเทียบกับและสัญญาณความคลาดเคลื่อนนี้จะนำไปขยายกำลังขึ้นโดยวงจรขยาย ซึ่งสัญญาณเอาท์พุทที่ออกจากวงจรขยายคือสัญญาณที่ติมอดคูล์ทแล้ว ส่วนที่เอาท์พุทของวงจรขยายนั้น จะนำไปป้อนให้กับวงจร VCO ด้วยเพื่อจะผลิตความถี่ออกมาเปรียบเทียบกับความถี่อินพุทต่อไป เมื่อสัญญาณความคลาดเคลื่อนได้รับการขยายจากวงจรขยาย ทำให้แรงดันที่ป้อนทางอินพุทของวงจร VCO มีการเปลี่ยนแปลงดังนั้นความถี่ที่ออกจากวงจร VCO ก็เกิดการเปลี่ยนแปลงด้วย และป้อนกลับไปเปรียบเทียบบเฟสกับสัญญาณอินพุทอีกครั้ง ซึ่งความถี่จากวงจร VCO จะพยายามล๊อคเข้ากับสัญญาณมอดคูล์ทอินพุท เนื่องจากสัญญาณอินพุทเป็นการเปลี่ยนแปลงตามความถี่ตามขนาดสัญญาณเสียงที่มอดคูล์ท ดังนั้นขนาดสัญญาณความคลาดเคลื่อนจะมีค่าเปลี่ยนแปลงตามความถี่สัญญาณความคลาดเคลื่อน นั่นก็คือสัญญาณเสียงที่มีการติมอดคูล์ทออกจากความถี่คลื่นนำแล้ว

## 2.5 ระบบสัญญาณโทรศัพท์แบบ DTMF

เครื่องโทรศัพท์ที่มีหน้าปัดเป็นแบบกดปุ่ม และใช้กรรมวิธีของDual Tone Multi frequency (DTMF) ในการส่งสัญญาณโทรศัพท์นั้น โดยทั่วไปหน้าปัดจะมี 12 ปุ่ม แบ่งเป็น 4 Rows และ 3 Columns และในเครื่องโทรศัพท์บางแบบอาจมีถึง 16 ปุ่ม โดยเพิ่ม Column ที่ 4 ขึ้นมาอีก

ความถี่ที่ใช้ในแต่ละ Row และ Column จะมีความถี่ต่างกัน ความถี่ของทั้ง 4 Rows เรียกว่าเป็นกลุ่มความถี่ต่ำ (Low Group Frequency) และความถี่ของทั้ง 3 หรือ 4 Columns เรียกว่าเป็นกลุ่มความถี่สูง (High Group Frequency) การกดปุ่มที่เลขหมายใดๆ จะทำให้วงจรอิเล็กทรอนิกส์ผลิตความถี่ออกมา 2 ความถี่ เช่น เมื่อกดเลข 5 ความถี่ที่ผลิตออกมา คือ 770 Hz และ 1336 Hz เป็นต้น

มาตรฐานของความถี่ที่ใช้และตำแหน่งของหมายเลขต่างๆ จะถูกจัดให้มีเอกสารนี้เป็นเอกสารทูลงรับไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะดังแสดงตามรูป สำหรับความผิดพลาดที่ยอมให้เกิดขึ้นได้จะเป็น 1.5 % สำหรับการผลิตความถี่ และ 2 % สำหรับการรับเลขหมาย

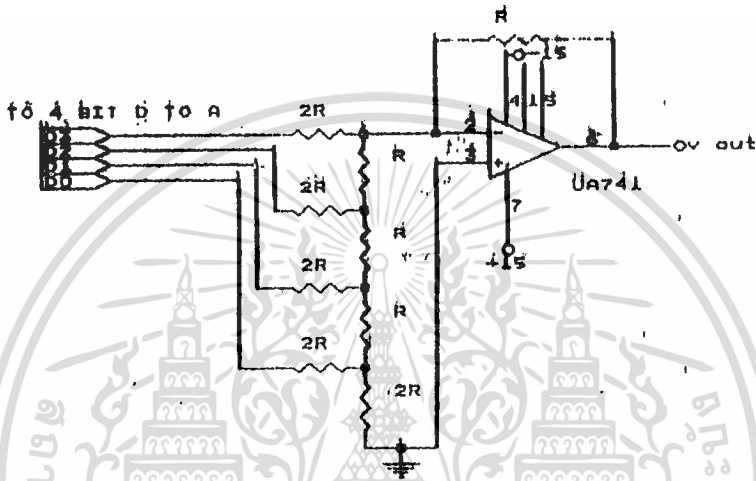
No.	F <sub>low</sub>	F <sub>high</sub>
1	697	1209
2	697	1336
3	697	1477
4	770	1209
5	770	1336
6	770	1477
7	852	1209
8	852	1336
9	852	1477
0	941	1336
*	941	1209
#	941	1477
A	697	1633
B	770	1633
C	852	1633
D	941	1633

ตารางที่ 2.1 ความถี่ที่ได้จากสัญญาณ DTMF

### 2.6 การเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก (Digital to Analog Conversion : DAC)

ในการทำงานระบบควบคุมที่ใช้ไมโครโปรเซสเซอร์ เนื่องจากการส่งสัญญาณของไมโครโปรเซสเซอร์จะใช้สัญญาณดิจิทัล ซึ่งไม่มีการแปรค่าของระดับแรงดันเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นในการใช้งานควบคุมที่ต้องใช้ระดับแรงดันในการควบคุม จึงจำเป็นจะต้องมีการ  
เปลี่ยนสัญญาณดิจิตอลที่ได้เป็นสัญญาณอนาล็อกซึ่งเป็นระดับแรงดันที่ต้องการ วงจรเปลี่ยน  
สัญญาณดิจิตอลเป็นอนาล็อกนิยมใช้วงจรที่เรียกว่า R-2R LADDER ซึ่งมีรูปแบบวงจรดังนี้



รูปที่ 2.4 วงจร R-2R LADDER DAC

ค่าแรงดันเอาต์พุตคำนวณได้จาก

$$V_{out} = [D3, /_2 + D2, /_4 + D1, /_8 + D0, /_{16}] * 5 \text{ volt}$$

โดย D3, D2, D1, D0 เป็น "1" IF HIGH

"0" IF LOW

## 2.7 การทริก(จุดชนวนทำงาน)ของไตรแอด

การทริกแบ่งเป็นสองแบบคือ

1. ชิงโครนัส จะกระตุ้นให้ทำงานที่จุดเดียวกันเสมอในทุก ๆ ครั้งคาบ  
เวลาของสัญญาณไฟสลับที่ให้โดยปรกติจะเป็นจุดหลังจากจุดตัดศูนย์เล็กน้อย เพื่อลดผล  
ของ RFI ให้น้อยที่สุด

2. อชิงโครนัสจะมีจุดเริ่มทำงานไม่แน่นอน ทำให้เกิดการกระชากของกระแส

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นเข้าใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ในการเริ่มต้นทำงานเพราะจุดที่เริ่มทำงานอาจไม่ใช่จุดตัดศูนย์ทำให้เกิด RFI ขึ้นมา

### 2.8 สัญญาณรบกวนความถี่วิทยุ

ในทุกๆครั้งที่ไตรแอกเริ่มทำงานนั้น กระแสที่จ่ายให้กับโหลดก็จะเพิ่มขึ้นอย่างรวดเร็วจนจากศูนย์ไปยังค่าที่กำหนดจากความต้านทานภายในของโหลด และแรงดันที่ให้อยู่ในช่วงเวลาสั้น ๆ ประมาณ  $2-3 \times 10^{-6}$  วินาที การเปลี่ยนแปลงอย่างทันทีทันใดนี้ จะส่งผลให้เกิดสัญญาณรบกวนความถี่วิทยุ หรือเรียกว่า RFI ขึ้น ค่า RFI ที่เกิดขึ้นนี้จะมีค่ามากที่สุด ในตำแหน่งที่มีการทริกที่เฟส 90 องศา และมีค่าต่ำสุดที่ 0 องศา และ 180 องศา ในทุก ๆ ครั้งคาบเวลาของสัญญาณไฟสลับที่ใช้

ในกรณีของวงจรหรือไฟนั้น อาจจะต้องใช้สายที่ต่อระหว่างวงจรกับโหลดยาวมาก ดังนั้น RFI จะมีผลมากเนื่องจากสามารถกระจายคลื่นสัญญาณรบกวนได้ตลอดความยาวของสายดังนั้น ในทางปฏิบัติควรจะมีวงจรรีไซแนสที่ทำหน้าที่ลดสัญญาณ RFI นี้ต่ออยู่ด้วย

### 2.9 เทคนิคการใช้จุดตัดศูนย์

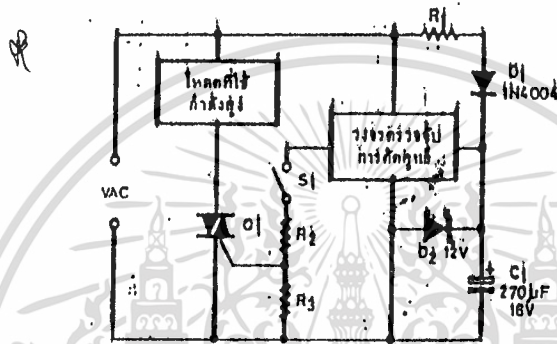
ในกรณีของโหลดที่ต้องการกำลังไฟสูง เช่น ตัวกำเนิดความร้อนที่ใช้ไฟฟ้าไต้ยใช้ไตรแอกควบคุมการทำงาน จำเป็นจะต้องใช้เทคนิคพิเศษในการลดค่าของ RFI นี้ ถ้ากรณีที่ใช้ไตรแอกทำหน้าที่เป็นสวิตช์เปิด/ปิดการทำงานเท่านั้น ผลของ RFI จะเกิดขึ้นทุก ๆ ครั้งที่มีการปิดวงจรสวิตช์ให้ระบบเริ่มทำงาน และผลนี้จะมีค่ามากที่สุด ถ้าในขณะที่เปิดระบบให้ทำงานในช่วงจังหวะที่มีเฟสเป็น 90 องศาพอดี RFI นี้จะลดลงได้โดยใช้เทคนิคของจุดตัดศูนย์เข้าช่วย ซึ่งจะต่อโดยตรงอยู่กับไฟสลับ วงจรตรวจจับการตัดศูนย์นั้นสามารถสร้างขึ้นได้โดยใช้ทรานซิสเตอร์หลาย ๆ ตัว โดยจะไม่บอกกล่าวในที่นี้ วงจรนี้จะทำหน้าที่เป็นตัวป้อนสัญญาณพัลส์เพื่อทริกขาเกตของไตรแอก ในขณะที่สัญญาณไฟสลับที่ให้มีการตัดศูนย์เกิดขึ้นทุกครั้ง ดังนั้นการทริกจะเกิดขึ้นในจังหวะที่เฟสของสัญญาณไฟสลับเป็น 0 องศา หรือ 180 องศา เท่านั้น จึงทำให้ RFI ที่เกิดขึ้นมีค่าน้อยที่สุด

เทคนิคการใช้จุดตัดศูนย์นี้ ใช้กับวงจรที่ต้องการปรับค่ากำลังไฟที่จะป้อนให้

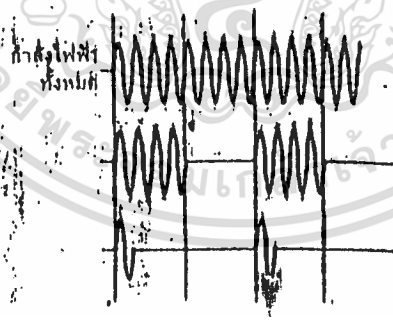
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับให้ลดได้ เพื่อลดค่าของ RFI ลงได้ เช่นกัน โดยเปลี่ยนสวิตช์  $S_1$  ในรูปที่ 2.5 ไปเป็นวงจรสร้างสัญญาณพัลส์ ที่สามารถปรับอัตราส่วนของช่วงสัญญาณที่มีลูกคลื่นกับช่วงเว้นว่างได้ ดังนั้น โหลดจะได้รับพัลส์เป็นช่วง ๆ ไป นั่นคืออาจจะได้รับ 4 ลูกคลื่น และเว้นไปอีก 4 ลูกคลื่นสลับกันเช่นนี้เรื่อย ๆ แต่ลูกคลื่นแต่ละลูกคลื่นที่ได้รับจะเต็มลูกคลื่นเสมอ



รูปที่ 2.5 การใช้ตัวตรวจจับตัดศูนย์เป็นตัวทริกไตรแอกเพื่อลดผลของ RFI ให้ต่ำที่สุด



รูปที่ 2.6 ตัวอย่างการแบ่งคาบเวลาการจ่ายพัลส์ให้สลับให้แก่โหลดเป็นช่วง ๆ โดยมีความยาวช่วงละ 8 ลูกคลื่น

ในรูปที่ 2.6 แสดงลักษณะของพัลส์ที่ป้อนให้แก่โหลดในวิธีนี้ โดยในที่นี้จะแบ่งคาบเวลาของการจ่ายพัลส์ทั้งหมดเป็นช่วง ๆ ที่เท่า ๆ กัน ในแต่ละช่วงจะมี 8 ลูกคลื่นถ้าไตรแอกถูกทริกให้นำกระแสเพียง 4 ลูกคลื่น นั่นก็หมายความว่าโหลดจะได้รับ

เอกสารนี้เป็นเอกสารของสำนักงานคณะกรรมการกฤษฎีกา กระทรวงยุติธรรม โดยผู้เขียนขอสงวนสิทธิ์ในการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังไฟเพียงครึ่งหนึ่งของไฟสลบทั้งหมด และถ้าถูกกำหนดให้ไตรแอนก้ากระแสเพียง  
ลูกคลื่นเดียว ส่วนอีก 7 ลูกคลื่นไม่นำกระแส ก็หมายความว่าโหลดจะได้รับกำลังไฟ  
เพียง 1 ใน 8 ส่วนเท่านั้น แต่ขอให้สังเกตว่าจังหวะของการทริกให้ไตรแอนก้าเริ่มทำ  
งานแต่ละครั้งจะเริ่มมีเฟส 0 องศาเท่านั้น ซึ่งจะเป็นการลดของ RFI ที่จะเกิดขึ้นได้  
ในขณะที่เดียวกันเราก็สามารถควบคุมกำลังไฟที่ป้อนให้แก่โหลดได้ตามต้องการ

### 2.10 การควบคุมกำลังไฟแบบเฟสทริกเกอร์

จากตัวอย่างการใช้งานไตรแอนก้าที่กล่าวมาตั้งแต่ต้นนี้เป็นการใช้งานในลักษณะ  
เป็นสวิตช์เปิด/ปิดการจ่ายไฟให้แก่โหลดดังกล่าว แต่เราสามารถใช้งานควบคุมกำลัง  
ไฟได้โดยใช้การทำงานแบบเฟสทริกเกอร์ กล่าวคือแทนที่จะทริกขาเกตด้วยสัญญาณไฟ  
ตรงนั้นตรง ๆ ก็ทริกโดยมีการหน่วงของเฟสด้วยวงจรอีกส่วนหนึ่ง



รูปที่ 2.7 การเปลี่ยนแปลงค่าของกำลังไฟที่ป้อนให้แก่โหลด โดยกำหนดได้จากตำแหน่งเวลาของการทริกให้กับไตรแอนก้า

การหน่วงเฟสมีผลคือ ถ้าไตรแอนก้าทริกที่ตำแหน่งเฟส 10 องศาหลังจากที่  
ทุก ๆ ครึ่งรูปคลื่นเริ่มเข้ามา กำลังไฟเกือบทั้งหมดก็จะถูกป้อนให้แก่โหลด แต่ถ้าท  
การทริกที่ตำแหน่งเฟส 90 องศา หลังจากทุก ๆ ครึ่งคลื่นเริ่มเข้ามา จะทำให้กำลังไฟ  
ที่ป้อนให้แก่โหลดนั้น ลดลงเหลือเพียงครึ่งหนึ่งของกำลังทั้งหมด และถ้าไปทริกที่ตำแหน่ง  
เฟส 170 องศา หลังจากทีทุก ๆ ครึ่งรูปคลื่นเข้ามาแล้ว จะมีเพียงกำลังไฟส่วนน้อย

เอกสารนี้เป็นเอกสารทสวงนเวลาสำหรับกรใช้งานเพื่อศรคศึกษาเท่านั้น ไม่นอยู่ให้เผยแพร่หรืออณนหาในการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

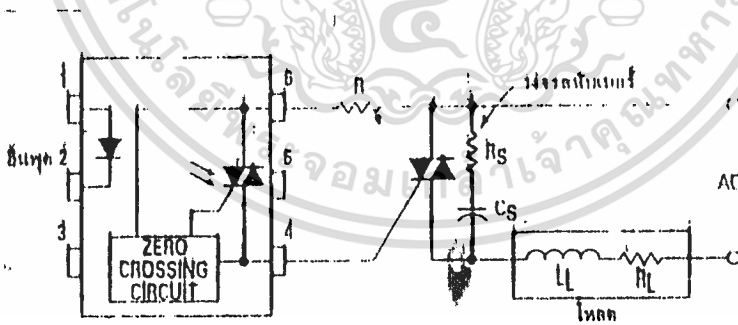
เท่านั้นที่ป้อนให้แก่โหลด ขอให้ดูรูปประกอบจะเข้าใจดีขึ้น

มีหลาย ๆ วิธีที่สามารถควบคุมตำแหน่งเฟสที่ต้องการทริก แต่ที่นิยมกันมากมีอยู่ 3 วิธี คือ

1. การใช้ UJT สร้างสัญญาณทริก
2. ไอซีที่สร้างขึ้นเพื่อทำงานนี้โดยเฉพาะ
3. ไดแอก (DIac) ร่วมกับวงจร ตัวต้านทาน และตัวเก็บประจุ

### 2.11 วงจร Snubber

ในการใช้งานวงจรควบคุมอุปกรณ์ไฟฟ้าด้วยไทรสเตอร์การทำงานกับโหลดที่มีค่าความเหนี่ยวนำมากจนทำให้เกิดผลทางไฟฟ้า เช่น มีการสปาร์คอย่างแรงเมื่อเปิดเปิดเบรกเกอร์ให้ลุดหรือเกิดไฟตกวูบหนึ่งเมื่อเปิดเครื่องซึ่งเป็นข้อสังเกตอย่างหนึ่งได้จากสายตาซึ่งผลของความเหนี่ยวนำ อาจทำให้อุปกรณ์ไทรสเตอร์ทำงานได้เองโดยไม่ต้องมีการทริกเกิด เมื่อเป็นเช่นนี้จึงทำให้เกิดผลเสียในการควบคุมเราจะแก้ไขปัญหานี้ได้ด้วยการใส่วงจร RC เพื่อลดผลค่าความเหนี่ยวนำของโหลดซึ่งเรียกว่าวงจร snubber เป็นวงจร RC ต่ออนุกรมแล้วขนานไปกับขั้วหลักของไทรสเตอร์ ดังรูปที่ 2.8



รูปที่ 2.8 แสดงการใช้งานวงจร Snubber

การคำนวณหาค่า RC ที่จะใช้ในวงจรมีสูตรการคำนวณดังนี้

$$f = \{dv/dt\} / \{2rV_p\}$$

$$C_s = 1 / \{(2rf)^2 L\}$$

$$R_s = \sqrt{L/C_s}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่า  $L$  ของโหลดจะได้จากการวัดหรือจากป้ายบอกคุณสมบัติ

(name plate)

$dv/dt$  ดูจากตารางของไทรสเตอร์ (V/ $\mu$ S)

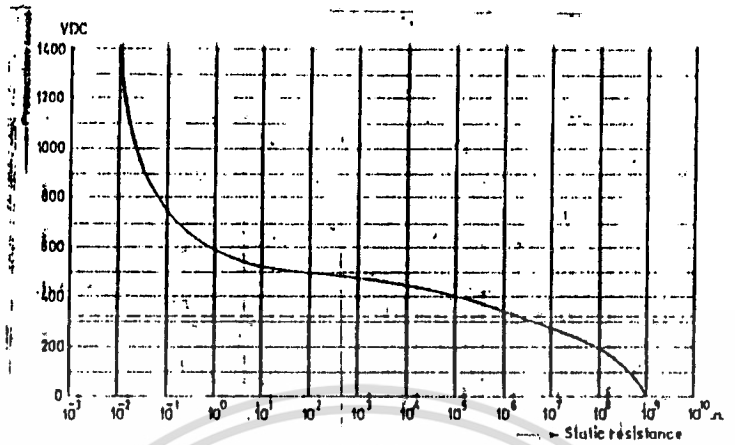
$V_p$  คือ ค่าแรงดันสูงสุด =  $V_{rms} * \sqrt{2}$

$C_s$  ในวงจร Snubber จะจำกัดค่า  $dv/dt$  จากภายนอกไม่ให้เกินค่า  $dv/dt$  ของไทรสเตอร์ส่วน  $R_s$  เป็นตัวจำกัดกระแสกระชากจาก  $C_s$  และไทรสเตอร์ นำกระแส

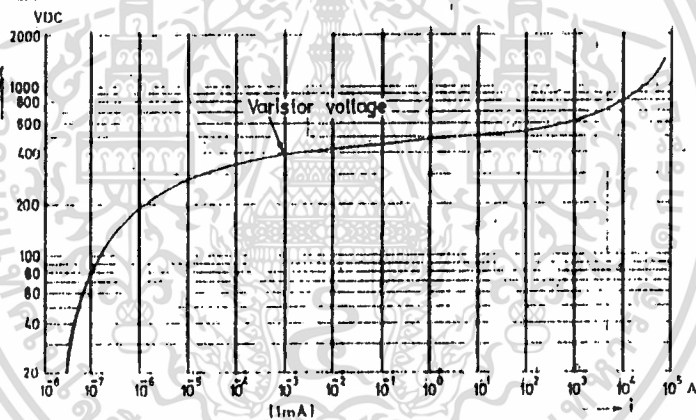
## 2.12 VDR อุปกรณ์ป้องกันแรงดันกระชาก

VDR (Voltage Dependent Resistor) หรือเรียกอีกอย่างว่า MOV (Metal Oxide Varistor) แต่มักจะเรียกกันง่าย ๆ ว่า "วาริสเตอร์" เป็นอุปกรณ์ทางไฟฟ้าที่มีคุณสมบัติพิเศษแตกต่างจากตัวอื่น ๆ คือมันจะทำหน้าที่เป็นตัวต้านทานที่แปรค่าได้ตามแรงดันที่ตกคร่อมตัวมัน โดยจะมีค่าความต้านทานสูงมาก ๆ (อาจถึง 100 Mohm) เมื่อแรงดันที่ตกคร่อมตัวมันต่ำกว่า rating voltage แต่เมื่อแรงดันเพิ่มขึ้นจนเกินกว่า rating voltage แล้ว ค่าความต้านทานจะลดต่ำลงมาก ๆ อย่างรวดเร็ว (อาจเหลือเพียง 0.1 ohm เมื่อแรงดันสูงกว่า rating voltage 2 เท่า) ซึ่งจะทำให้มีกระแสจำนวนมาก (ในช่วง 100 ถึง 10,000 A) ไหลผ่านตัวมันเปรียบเสมือนว่าตัวมันทำหน้าที่ลัดวงจรได้เลยก็ว่าได้

เพื่อให้เห็นการทำงานที่ชัดเจนยิ่งขึ้น จึงแสดงกราฟแสดงความสัมพันธ์ของแรงดันกับความต้านทานของ VDR และความสัมพันธ์ของแรงดันกับกระแสไฟฟ้าไว้ดังรูปที่ 2.9 และ 2.10 หรือถ้าจะสรุป VDR ก็คือซีเนอร์ไดโอดไฟสลับนั่นเอง



รูปที่ 2.9 ตัวอย่างแสดงความสัมพันธ์ของแรงดันและความต้านทานของ VDR ที่มี rating voltage = 250 Vrms



รูปที่ 2.10 ตัวอย่างแสดงความสัมพันธ์ของแรงดันและกระแสในตัว VDR ที่มี rating voltage = 250 Vrms และแสดงค่า varistor voltage ให้เห็นด้วย

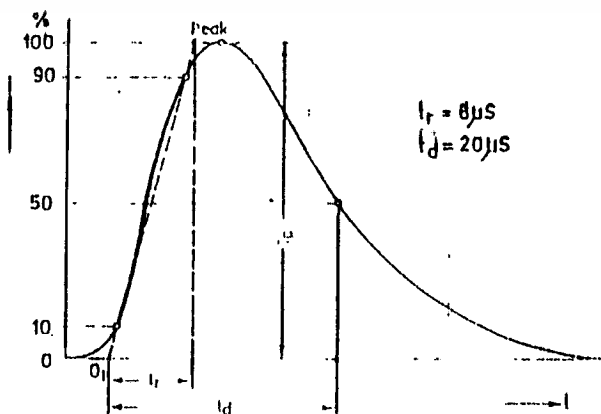
คุณสมบัติที่สำคัญอีกข้อหนึ่งของ VDR ซึ่งเป็นตัวกำหนดความสามารถในการรับพลังงาน จะเรียกว่า energy absorption rating โดยจะมีผลกับปริมาณกระแสไฟฟ้า และช่วงเวลาที่มันจะทนได้โดยไม่พังไปเสียก่อน และค่าพลังงานที่จะทนรับได้นี้มีหน่วยเป็นจูล (หรือ W.sec) ขึ้นอยู่กับขนาดของตัว VDR ด้วย ถ้าขนาดใหญ่มากจะสามารถรับกระแสได้สูง หรือสามารถทนกับภาวะการลัดวงจรภายในตัวมันในช่วงเวลานาน ๆ ได้

เนื่องจากในขั้นตอนการผลิตตัว VDR นี้ ผู้ผลิตไม่สามารถผลิตให้ VDR มีค่าเอกสารนี้เป็นเอกสารทสงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rating voltage เท่ากันทุกตัวได้ เขาจึงใช้วิธีจัดกลุ่มค่าที่ใกล้เคียงกันเอาไว้ด้วยกัน แล้วกำหนดเป็นค่า rating voltage เอาไว้ค่าหนึ่ง ซึ่งจะเป็นค่าในช่วงกลาง ๆ ของแต่ละกลุ่ม ตัวอย่าง เช่น VDR ที่ใช้ในโครงงานนี้เป็นของยี่ห้อ ZENAMIC ที่หาซื้อได้ในประเทศ โดยเลือกใช้เบอร์ Z10L391 ซึ่งได้กำหนดค่า varistor voltage เอาไว้ที่ 351-429 VDC

ค่า varistor voltage คือค่าแรงดันเมื่อป้อนแรงดันไฟตรงเข้าตัว VDR แล้ว เกิดกระแสไหลผ่านตัวมัน 1 mA และถ้าแรงดันเพิ่มขึ้นกว่านี้อีกเพียงเล็กน้อย จะเกิดกระแสไหลผ่านตัวมันเพิ่มขึ้นอย่างรวดเร็ว ดังนั้นจึงมักจะใช้ค่า varistor voltage เป็นตัวกำหนดการใช้งานหรือ rating voltage จากตัวอย่างเบอร์ Z10L391 จึงได้ค่าแรงดันป้องกันวงจรสำหรับโพลลับในช่วง  $351 / \sqrt{2} - 429 / \sqrt{2}$  หรือ 248-303 Vrms แต่เนื่องจากการใช้งานจริง ๆ จำเป็นต้องมีการเผื่อเอาไว้ ด้วยคือลดลงเล็กน้อยเพื่อความปลอดภัยของตัวเอง และยังรักษาอุปกรณ์ให้ปลอดภัยได้ ผู้ผลิตจึงกำหนดค่า rating voltage เป็น 235 Vrms เมื่อเรานำมาใช้กับไฟบ้านขนาด 220 V จึงสามารถใช้ได้อย่างสบาย โดยต้องยอมให้มีกระแสสูญเสียไหลผ่านตัวมันน้อยกว่า 1 mA

คุณสมบัติที่น่าสนใจอีกข้อหนึ่งนั่นคือ current rating ซึ่งเป็นตัวกำหนด กระแสสูงสุดที่ผ่านตัวมันได้ในสภาวะของเงื่อนไขที่กำหนด ก่อนที่ตัวมันจะเสียหาย โดยทั่วไปจะกำหนดเงื่อนไขเป็น  $8 * 20 \mu\text{sec}$  หรือ  $8 / 20 \mu\text{sec}$  ความหมายของเงื่อนไขตัวนี้มาจากการป้อนรูปคลื่นให้เกิดกระแส surge ดังรูปที่ 2.11



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าในรูปแบบใด ๆ อีก การป้อนรูปคลื่นเพื่อหา current rating ตามที่กำหนดของ VDR ใช้

รูปคลื่นของกระแสที่ได้ในรูปที่ 2.11 จะได้ค่า  $t_r$  และ  $t_d$  โดย  $t_r$  เกิดจากการวัดค่ากระแสที่ 10 % และ 90 % ของกระแสสูงสุด แล้วลากเส้นไปหาจุดสูงสุด ( $I_p$ ) พร้อมกับลากเส้นตั้งฉากลงมา สำหรับ  $t_d$  วัดจากจุดเริ่มต้นของ  $t_r$  ไปยังจุดที่กระแสลดลงมาเหลือ 50% ดังนั้นเงื่อนไขที่กำหนัดไว้  $8 \times 20 \mu\text{sec}$  หรือ  $8/20 \mu\text{sec}$  ก็คือ  $t_r$  กับ  $t_d$  นั่นเอง โดยที่  $t_r$  คือ  $8 \mu\text{sec}$  และ  $t_d$  คือ  $20 \mu\text{sec}$

คุณสมบัติ current rating หรือกระแสสูงสุดเมื่อ  $t_r = 8 \mu\text{sec}$  และ  $t_d = 20 \mu\text{sec}$  ของเบอร์ Z10L391 ที่ใช้ในโครงงานนี้มีค่าสูงถึง 1,250 A และ current rating นี้จะลดลงเหลือประมาณ 15 A และถ้าต้องการให้ VDR ตัวนี้ทนกับสภาวะเช่นนี้ได้ 100 ครั้ง ค่ากระแสสูงสุดที่ไหลผ่านตัวมันจะลดลงจาก 15 A เหลือเพียง 3.5 A เท่านั้น

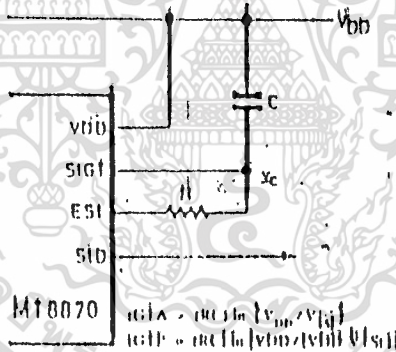
เราสามารถนำเอา VDR หลาย ๆ ตัวมาต่อขนานกัน เพื่อเพิ่มความสามารถในการรับพลังงานและกระแส surge ได้แต่เนื่องจากค่า varistor voltage ของแต่ละเบอร์มีโอกาสแตกต่างกันสูงมากถึง 10 % ดังนั้นอัตราส่วนในการแบ่งกระแส surge ไปยัง VDR แต่ละตัวจึงแตกต่างกันมาก เช่นตัวที่มี varistor voltage ต่ำที่สุด อาจต้องรับกระแส surge สูงกว่าตัวอื่นเป็น 100 เท่าก็ได้ถ้าอยู่ในสภาวะเลวร้ายจริง ๆ

แต่ถ้าใช้จะเป็นผลเสียไปเลยก็เดียวเมื่อเทียบกับการเลือกใช้แบบตัวใหญ่ตัวเดียวกับตัวเล็กหลาย ๆ ตัวแล้ว ในเรื่องของราคาอาจประหยัดกว่ามาก และช่วงเวลาในการป้องกันกระแส surge จะยาวนานกว่า เพราะถ้ากระแสมากเกินไป VDR จะทำงานต่อเนื่องกันไป จากตัวที่มีค่า varistor voltage ต่ำสุดไปหาตัวที่มีค่าสูงสุด หรือถ้าจะพังก็จะพังเรียงตัวกันไปด้วย ปลอดภัยกว่าการใช้ตัวใหญ่ตัวเดียวแล้วพังไปเลย

การคำนวณและการสร้าง

3.1 GUARD TIME ADJUSTMENT ของ MT 8870

สำหรับคำว่าการ์ดไทม์ (GuardTime) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่ตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณนั้นถูกต้องหรือพูดได้ว่าเวลาที่เรที่ตั้งไว้โดยRC ก็คือการ์ดไทม์นั่นเอง เมื่อสัญญาณความถี่เข้ามานานเท่ากับหรือมากกว่าเวลาที่ตั้งไว้ จึงจะสามารถแปลงเป็นตัวเลขได้ถ้าสัญญาณความถี่เข้ามาสั้นกว่า ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาและคำนวณเวลาได้จากรูปที่ 3.1



รูปที่ 3.1 วงจรตรวจสอบสัญญาณ ( STEERING ) อย่างง่าย

วงจร STEERING ง่าย ๆ แสดงดังรูปที่ 3.1 ซึ่งเป็นการประยุกต์ใช้งานค่าของอุปกรณ์ จะถูกเลือกตามสูตรต่อไปนี้

$$T_{REC} = TDP + TGTP$$

$$T_{ID} = TDA + TGTA$$

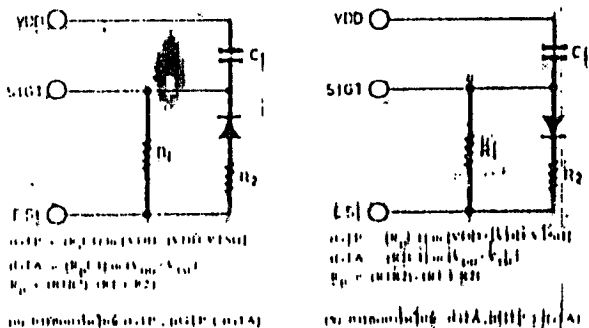
โดย  $T_{REC}$  - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

$T_{ID}$  - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปเผยแพร่หรือใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $T_{DP}$  - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง
- $T_{DA}$  - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง
- $T_{GTP}$  - การ์ดไทม์ของการปรากฏความถี่ DTMF ที่ถูกต้อง
- $T_{GTA}$  - การ์ดไทม์ของการหายไปของความถี่ DTMF

ค่าของ  $T_{DP}$  เป็นค่า PARAMETER ของตัวไอซีเอง (ดูได้จากตาราง SPECIFICATION) และค่า  $T_{REC}$  เป็นค่าช่วงเวลาดำสุด ที่ตัวไอซีจะทำการ decode คู่ TONE ได้ค่าของ  $C = 0.1 \mu F$  นี้ เป็นค่าที่ใช้อยู่ในการใช้งานทั่ว ๆ ไป การออกแบบวงจร STEERING มีให้เลือกแตกต่างกันหลายแบบ ทั้งนี้ขึ้นอยู่กับลักษณะสัญญาณ TONE ที่ตัว IC จะทำการ DECODE โดยจะสามารถเลือกช่วงเวลา GUARD TIME ได้อย่างอิสระ สำหรับช่วงสัญญาณที่ปรากฏ ( $T_{GTP}$ ) และช่วงสัญญาณหาย ( $T_{GTA}$ ) ซึ่งค่าทั้ง 2 นี้ เป็น spec สำคัญของการทำงานของ IC ซึ่งจะทำให้เกิดการ DECODE หรือไม่ต่อช่วงสัญญาณที่เข้ามา การปรับ GUARD TIME ผู้ออกแบบจะเป็นคนกำหนด PARAMETER ต่าง ๆ ตามสภาวะของสัญญาณตัวอย่าง เช่น สัญญาณที่เข้ามา มีสัญญาณอื่นปนเข้ามา (TALK OFF) การออกแบบให้  $T_{REC}$  มาก ๆ จะป้องกันการ TALKOFF ได้ เพราะช่วงเวลา  $T_{REC}$  มาก ๆ จะลดโอกาสที่จะเป็นไปได้อย่าง TONE แผลกปลอมที่เกิดจากสัญญาณเสียงพูด ในทางกลับกัน ถ้าออกแบบ  $T_{REC}$  สั้น ๆ กับ  $T_{DO}$  (inter digit pause) นาน ๆ จะเหมาะสมสำหรับสัญญาณที่มี noise กวนอย่างมาก ซึ่งในการออกแบบใช้งานก็ใช้  $T_{REC}$  สั้น ๆ สูตรในการคำนวณแสดงในรูปที่ 3.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดๆทั้งสิ้น **รูปที่ 3.2** สูตรวงจรตรวจจุดค้นสัญญาณที่ใช้งานจริง

จากสูตร การลัด GUARD TIME ;  $T_{GTF} < T_{GTA}$

โดย  $T_{GTF} = (R_i C_i) \ln [ VDD - VTST ]$

กำหนด  $R_i = 220K$  ;  $VDD = 5 VOLT$  ;  $V_{TST}$  ใน SPEC กำหนด = 2.2 VOLT

$$\begin{aligned} T_{GTF} &= (220 * 10^3 * 0.1 * 10^{-6}) \ln [ (5 / (5 - 2.2)) ] \\ &= 0.022 * \ln 1.7857 \\ &= 12.8 \text{ mS} \end{aligned}$$

$T_{GTA} = (R_p C_i) \ln [ VDD - VTST ]$

$$R_p = R1 // R2$$

กำหนด  $R2 = 220 K$

$$R_p = 220 // 220$$

$$= 110 K$$

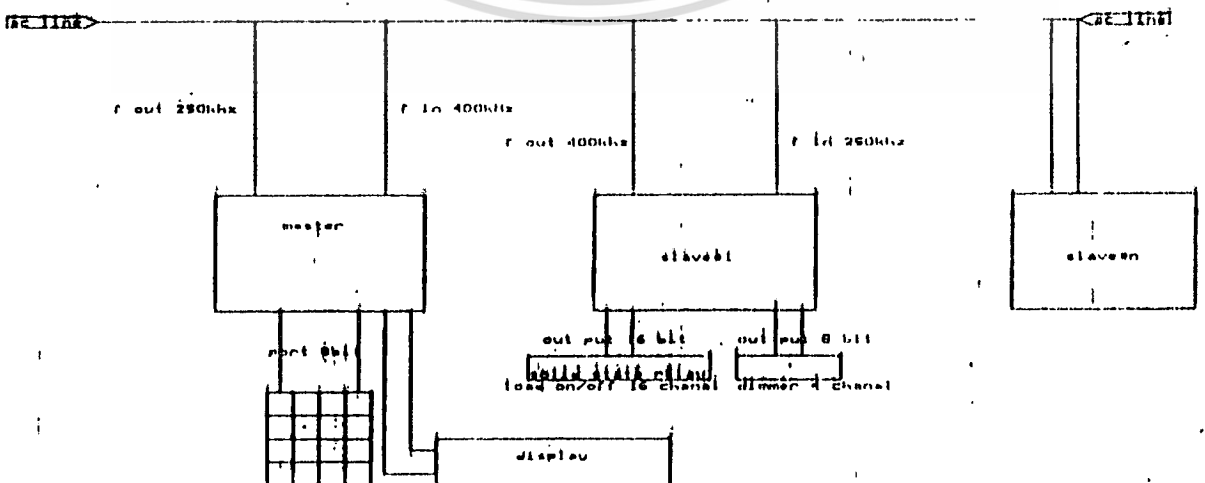
$T_{GTA} = (110 * 10^3 * 0.1 * 10^{-6}) \ln (5 / 2.2)$

$$= 0.011 * 0.8209$$

$$= 9.0299 \text{ mS}$$

### 3.2 โครงสร้างของระบบที่ใช้ในการทำงาน

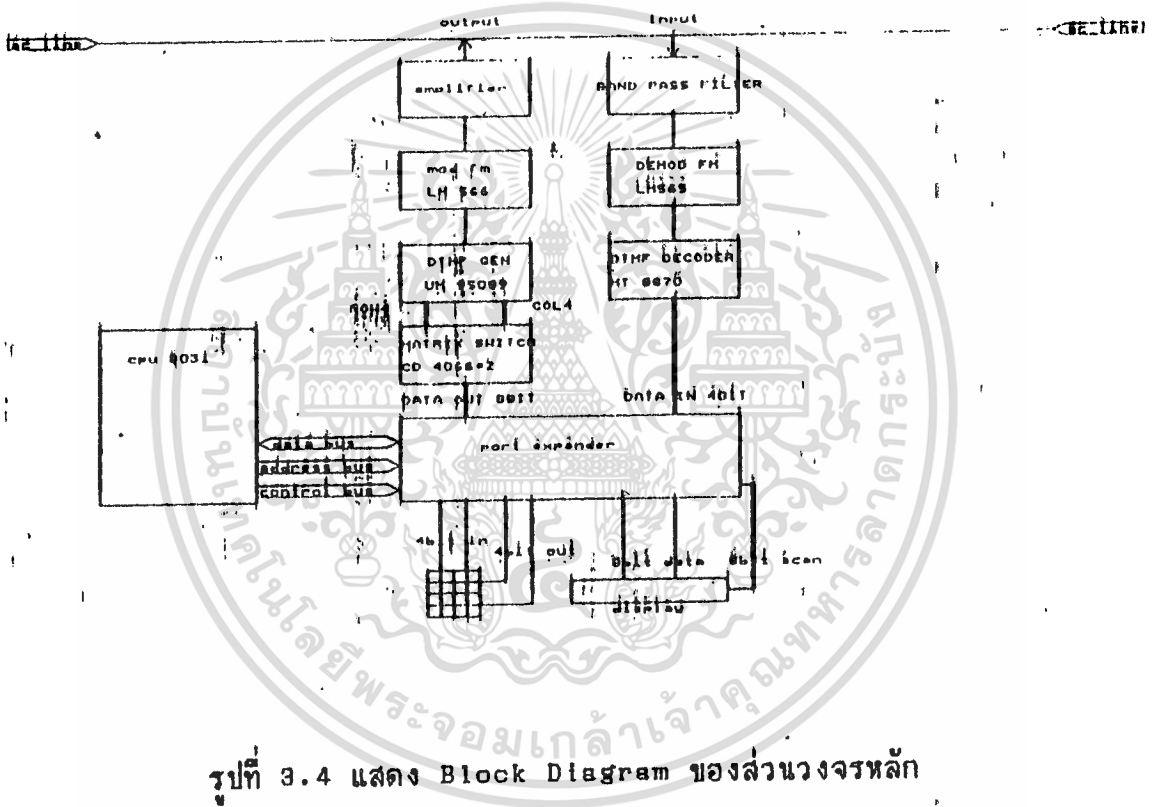
การทำงานโดยรวมของโครงการนี้ประกอบด้วยรายละเอียดต่าง ๆ ดังแสดงในรูปต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 3.3 แสดง block diagram โดยรวมของโครงการ นี้ที่มีการนำไปใช้

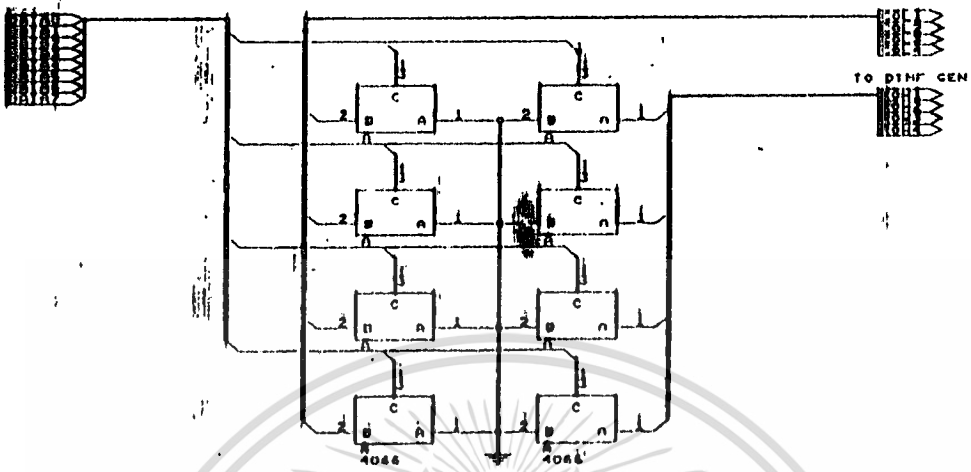
วงจรจะประกอบขึ้นด้วยส่วนใหญ่ ๆ 2 ส่วนคือ

1. ส่วนวงจรหลักที่ใช้ในการควบคุม (Master) ประกอบด้วย ส่วนส่งสัญญาณ ควบคุม คีย์บอร์ด และภาคแสดงผล
2. ส่วนวงจรรย่อยที่ใช้ในการควบคุม (slave) ประกอบด้วยส่วนรับข้อมูล ส่วนประมวลผล ส่วนควบคุมอุปกรณ์ทั้งสองแบบ

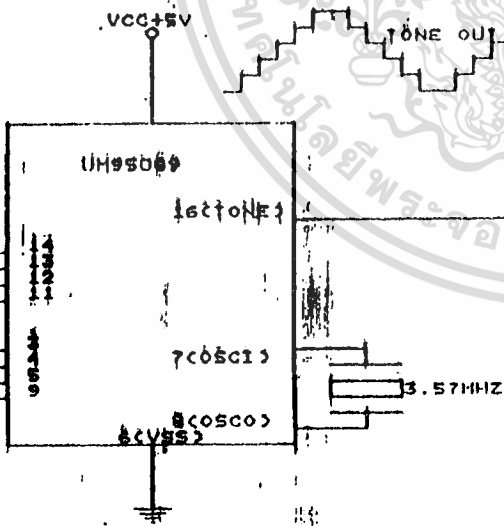


รูปที่ 3.4 แสดง Block Diagram ของส่วนวงจรหลัก

การทำงานของส่วนวงจรหลัก ประกอบด้วยไมโครโปรเซสเซอร์ที่ใช้ในการควบคุมส่วนแสดงผล คีย์บอร์ด และการส่งสัญญาณควบคุมผ่าน MATRIX SWITCH ที่ใช้ในการสร้างสัญญาณโดย DTMF Generator UM 95089 ที่จะสร้างสัญญาณ DTMF เข้าสู่ภาค Frequency Modulation กับสัญญาณคลื่นพาห์โดยใช้ LM566 สัญญาณที่ได้จะถูกนำไปขยายด้วยวงจรขยายก่อนป้อนเข้าสู่สายไฟฟ้าต่อไป



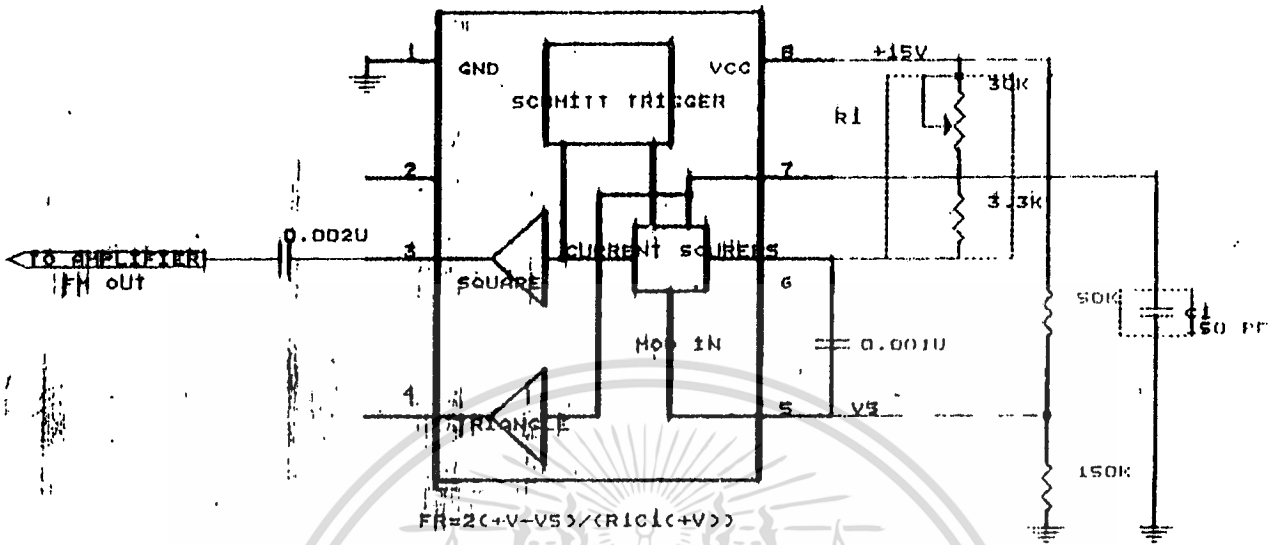
รูปที่ 3.5 วงจรสวิตช์อิเล็กทรอนิกส์ประกอบการสร้างสัญญาณ DTMF



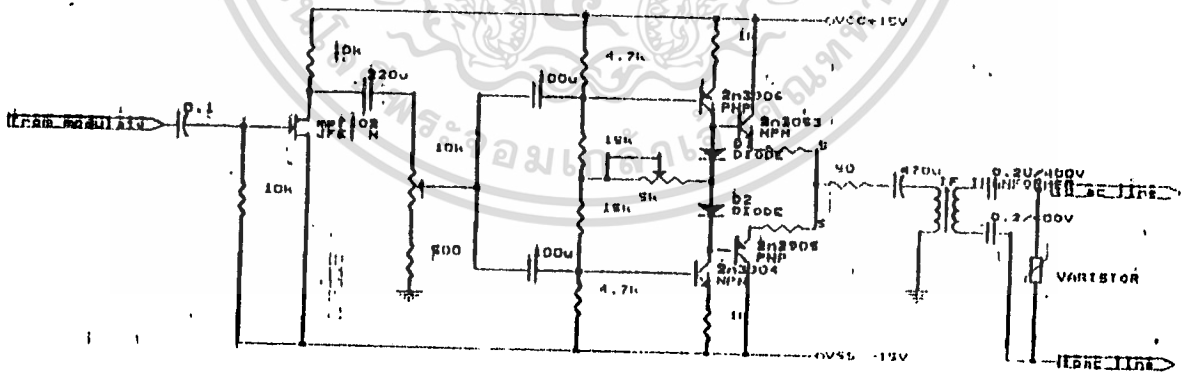
Active INPUT	OUTPUT FREQUENCY
R1	697
R2	770
R3	852
R4	941
C1	1209
C2	1336
C3	1477
C4	1633

รูปที่ 3.6 แสดงวงจรสร้างสัญญาณ DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

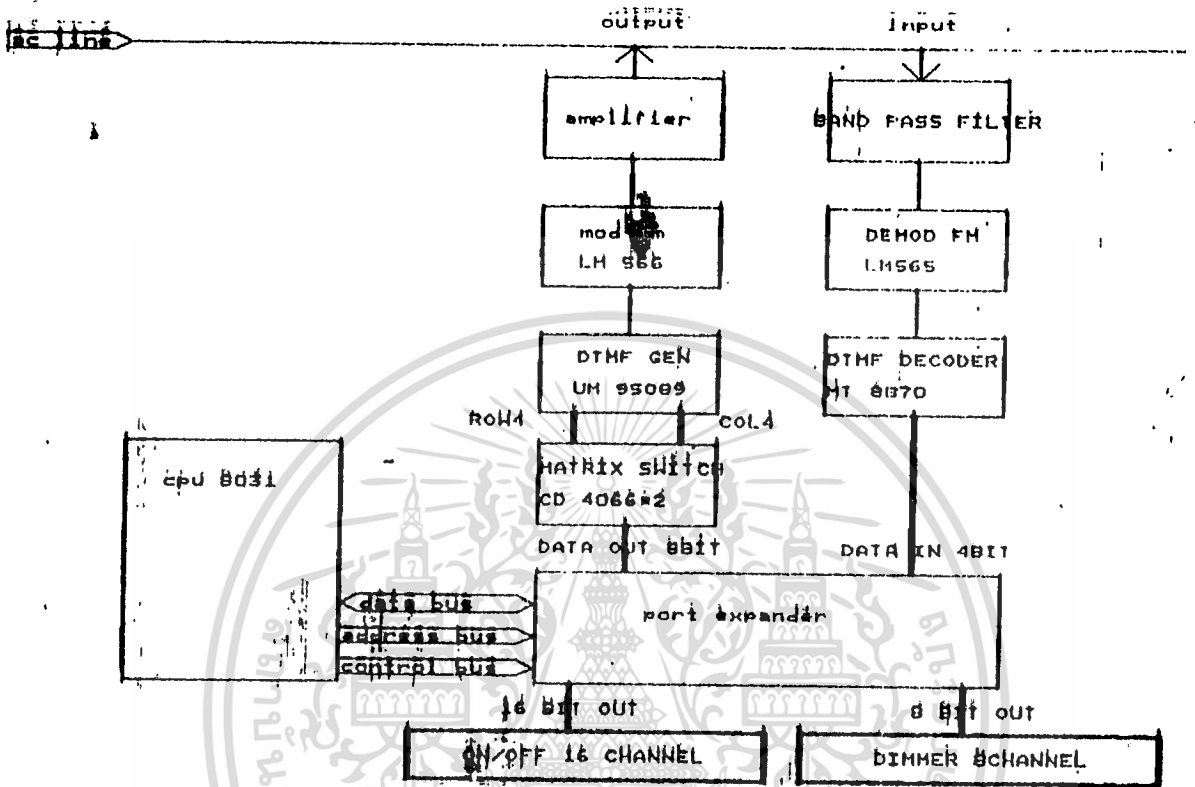


รูปที่ 3.7 แสดงวงจร Modulator สัญญาณความคม



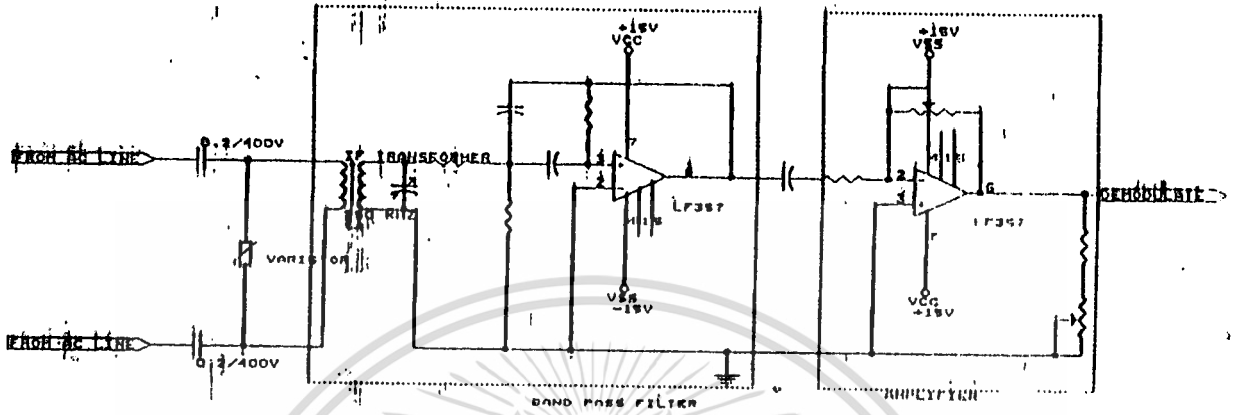
รูปที่ 3.8 ภาคขยายสัญญาณเพื่อป้อนเข้าสู่ AC Line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

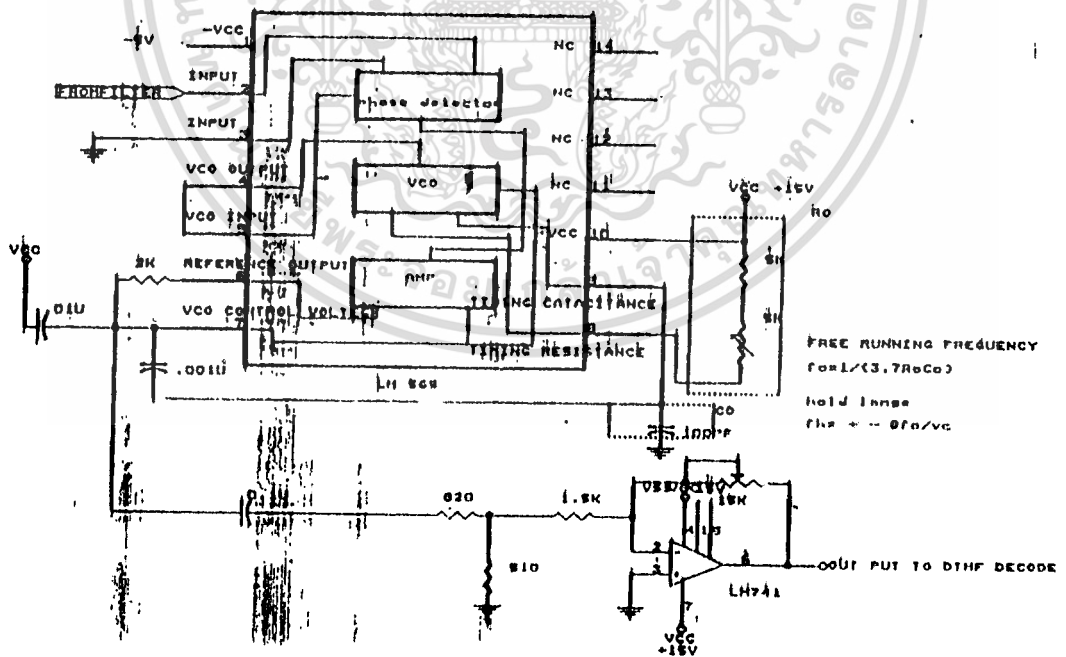


รูปที่ 3.9 แสดง Block Diagram ของวงจรย่อย

การทำงานของส่วนวงจรย่อยจะทำการรับข้อมูลที่มาจกสายไฟฟ้า ผ่านวงจร Band Pass Filter เพื่อเลือกเฉพาะสัญญาณที่ต้องการกับความถี่เท่าที่สัญญาณที่ได้จะถูกนำมาผ่านการติมอดคูล์ชั่นที่ใช้ไอซี LM 565 เพื่อแยกเอาเฉพาะสัญญาณควบคุมที่เป็นสัญญาณDTMF ไปผ่านวงจร DTMF Decoder แปลงเป็นสัญญาณดิจิตอลขนาด 4 บิตที่จะถูกนำไปประมวลผลด้วยระบบไมโครโปรเซสเซอร์ หลังจากนั้นไมโครโปรเซสเซอร์ จะทำการส่งสัญญาณไปควบคุมการทำงานของภาคควบคุมอุปกรณ์ต่อไป

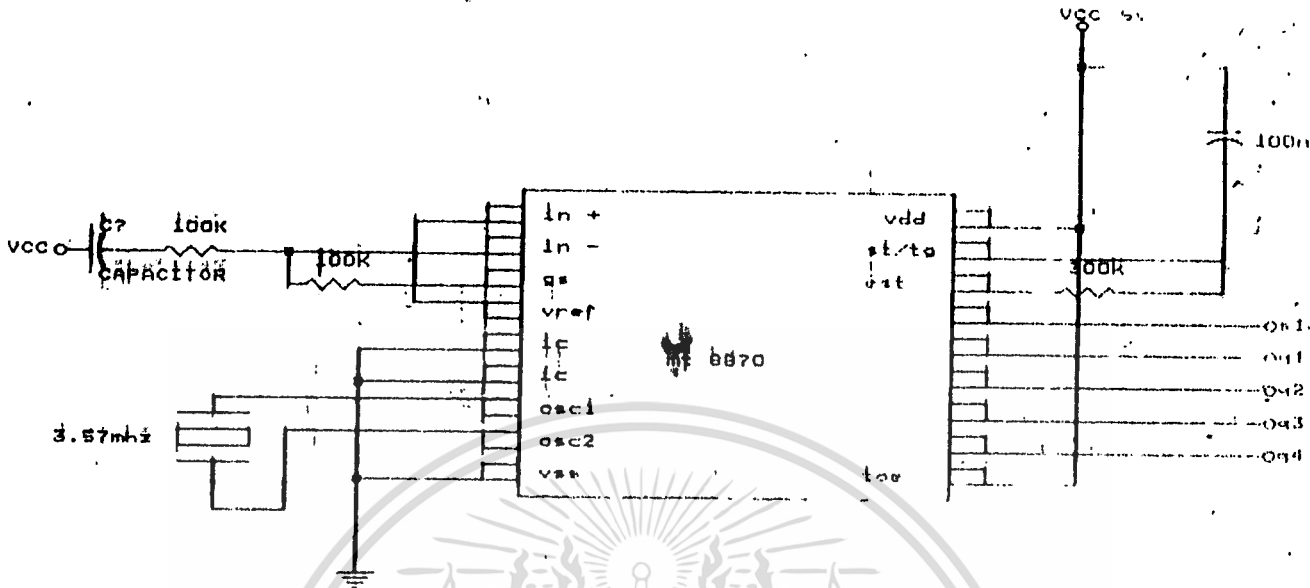


รูปที่ 3.10 วงจร Band Pass Filter

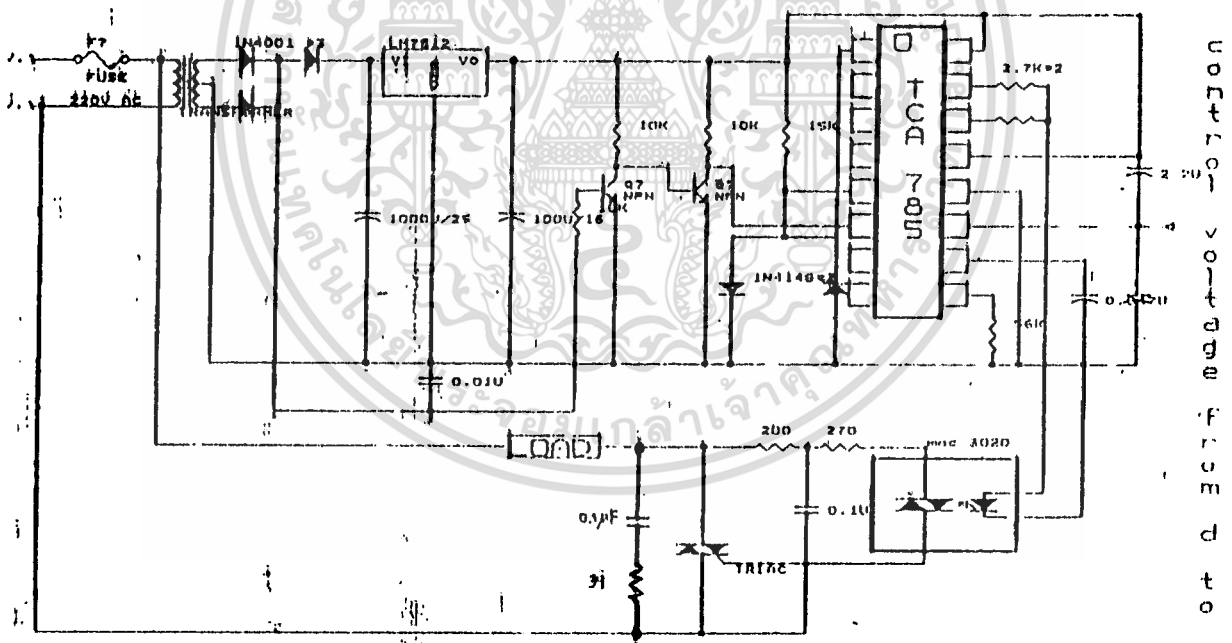


รูปที่ 3.11 วงจร Demodulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 วงจรส่วน DTMF Decoder



รูปที่ 3.13 วงจรควบคุมหลอด ON/OFF

### 3.3 การทำงานของภาคควบคุมอุปกรณ์ ON-OFF

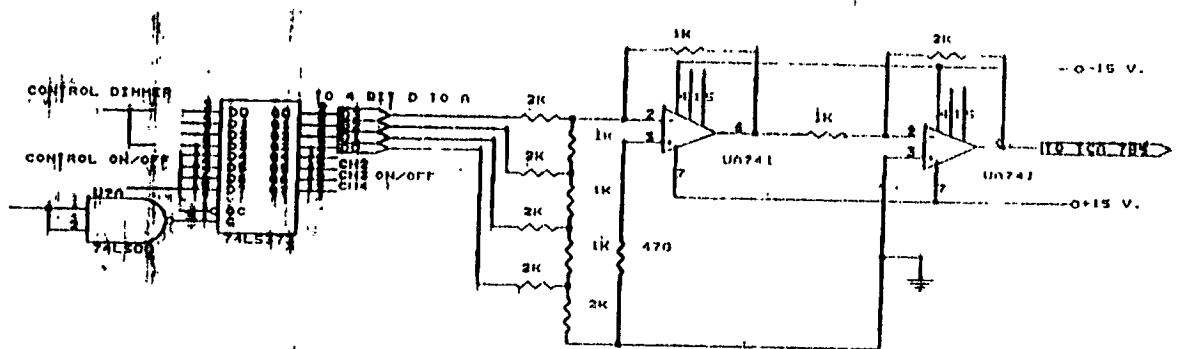
จากวงจรเราสามารถปรับระดับสัญญาณลอจิกป้อนเข้าที่ขั้ววงจรให้ทำงานทางขา base ของทรานซิสเตอร์ โดยใช้ตัวเชื่อมโยงทางแสง (opto-coupler) แบบ LED กับไดรแอด ด้าน LED มีตัวต้านทาน R1 ทำหน้าที่ จำกัดกระแสที่ไหลผ่านไม่ให้เกินค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูงสุดที่ตัวเชื่อมโยงทางแสงจะทนได้คือ 50 mA D1 เป็นตัวป้องกันการเสียหายของตัวเชื่อมโยงทางแสงจากการป้อนแรงควบคุมกลับซ้ำ ในตัวเชื่อมโยงทางแสงตัวนี้มีวงจรตรวจจับแรงดันศูนย์ในตัวเอง เมื่อเราป้อนแรงดันควบคุมเข้าไป R1 จะควบคุมให้กระแสไหลผ่าน LED ภายใน IC1 ให้มีค่าประมาณ 3-40 mA LED ที่อยู่ใน IC1 ทำให้ไฟโตะไตรแอดภายในนำกระแส แต่ไฟโตะไตรแอดจะนำกระแสเมื่อแรงดันในสาย (รูปคลื่น sine) มีค่าเริ่มจากศูนย์เพิ่มขึ้นทางบวกหรือทางลบ

เมื่อไฟโตะไตรแอดนำกระแสก็จะไปทริกขา gate ของไตรแอด Q1 ทำให้ Q1 นำกระแสทำให้หลอดทำงาน จากการที่มีการตรวจจับแรงดันศูนย์ทำให้ Q1 ทำงานนำกระแสที่จุดนี้ทุกครั้ง จากการได้รับการทริกจากไฟโตะไตรแอด ดังนั้นวงจรจะตัดและต่อวงจรที่จุดศูนย์ของแรงดันในสายเสมอ R2 และ R3 เป็นตัวแบ่งกระแสที่ไหลผ่าน gate ของไตรแอดไม่ให้สูงเกินไป C1 เป็นตัวป้องกันการทำงานผิดพลาดจากสัญญาณรบกวนภายนอก R4 และ C2 เป็นตัวป้องกัน transient จากการเพิ่มแรงดันของหลอดอย่างรวดเร็ว ซึ่งจะทำให้ไตรแอดเสียได้ นอกจากนี้ C2 ยังเป็นตัวลดสัญญาณรบกวนและแรงดันกระชากในสายไฟด้วย

### 3.4 การทำงานภาค Digital to Analog

สัญญาณดิจิตอลขนาด 4 บิต จะถูกล่วงมาจากเอาต์พุตพอร์ทของระบบไมโครโปรเซสเซอร์ ผ่านวงจร R-2R LADDER ออกเป็นสัญญาณแอนะล็อกขนาดระหว่าง 0 ถึง -5 โวลต์ ซึ่งจะถูกนำไปขยายด้วยวงจร Inverting Amplifier เกนขยาย 2 เท่า ทำให้เอาต์พุตของวงจรเป็นระดับ 0 ถึง 10 โวลต์ซึ่งจะถูกนำไปป้อนให้กับ TCA 785 เพื่อใช้ในการควบคุมกำลังงานของอุปกรณ์ต่อไป

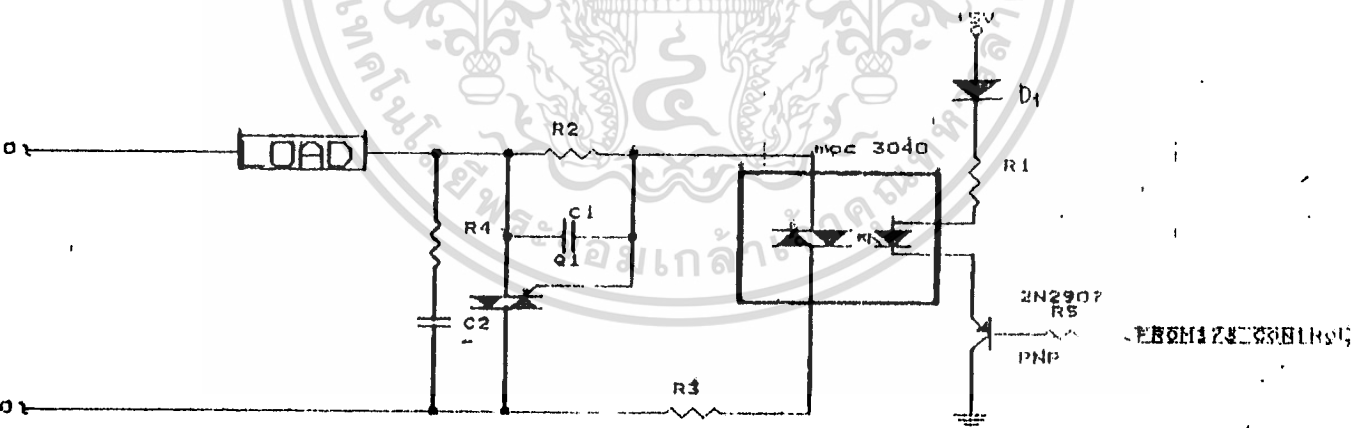


DIGITAL TO ANALOG

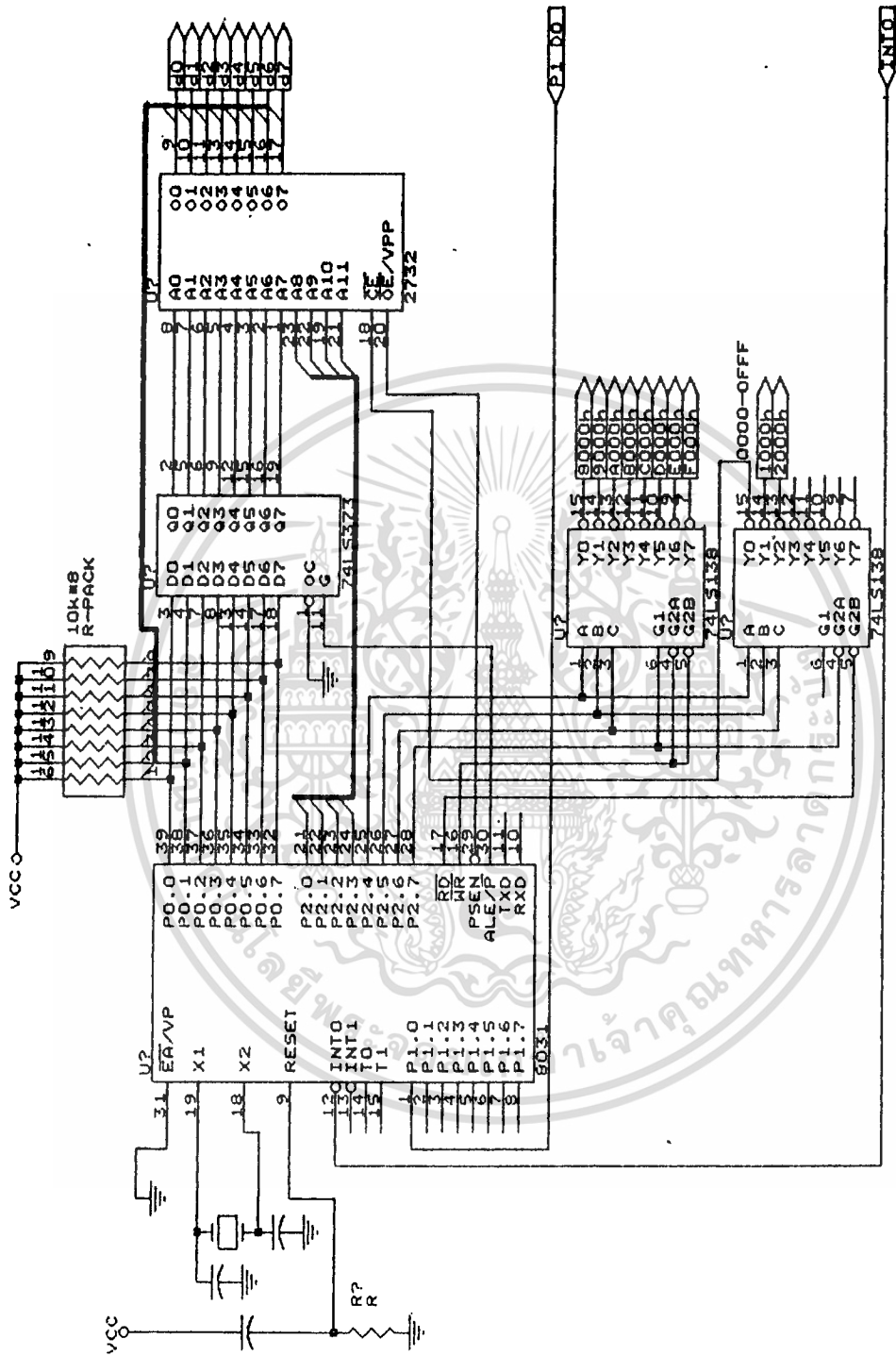
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.14 วงจร Digital TO Analog  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุตบแต่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การทำงานภาคควบคุมมัมเฟล

สัญญาณรูปไซน์จะถูกป้อนผ่าน R15k เข้าขา 5 ของ TCA 785 เพื่อใช้เป็นสัญญาณซิงค์ในการสร้างสัญญาณฟันเลื่อยเพื่อควบคุมเฟสที่ได้จากค่า R 56k ที่ขา 9 กำหนดกระแสผ่านตัวเก็บประจุ C 0.047  $\mu$ F ที่ขา 10 ซึ่งจะนำไปเปรียบเทียบกับแรงดันอินพุตควบคุมที่เข้ามาทางขา 11 ที่รีจิสเตอร์ 2SC458 สองตัวต่อกันเป็นวงจรรีนาเวอร์ทเพื่อใช้ในการ disable ขา inhibit สัญญาณเอาต์พุตทางขา 14; 15 จะถูกนำไปควบคุมไฟโตไดโอดภายใน OPTO เบอร์ MOC 3020 ที่มีเอาต์พุตเป็นไดโอด เพื่อนำไปควบคุมการทำงานของไทรแอกต่อไป R 200, 270 โอห์มจะควบคุมกระแสที่ไหลผ่านขาเกตของไทรแอกไม่ให้มากเกินไป R 39 โอห์มและ C 0.1  $\mu$ F ต่อเป็นวงจรป้องกันแรงดันทรานเซียนท์เนื่องจากโหลด (Snubber) ไม่ให้ทำอันตรายกับไทรแอก

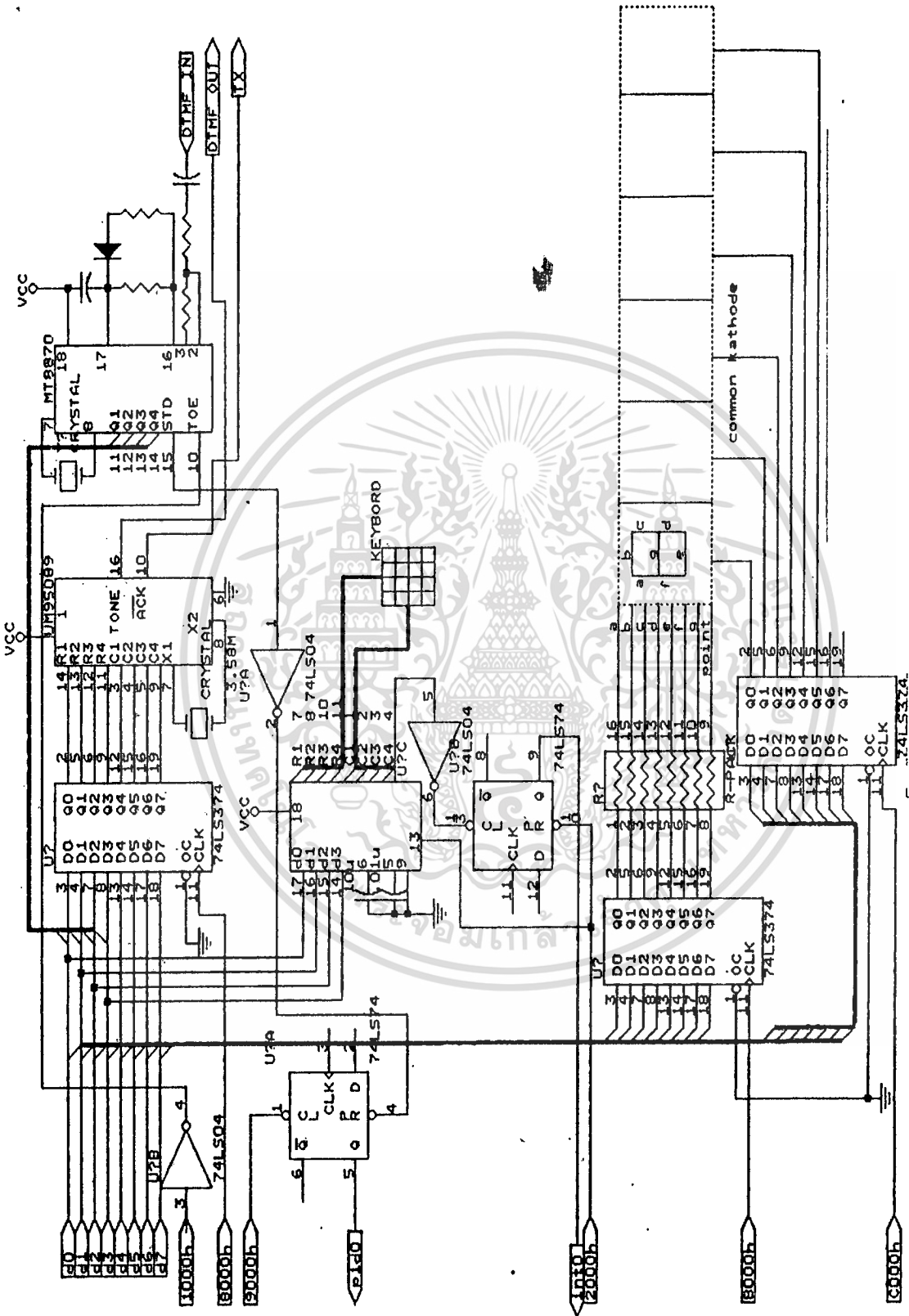


รูปที่ 3.15 ภาคควบคุมกำลังไฟด้วยมัมเฟล



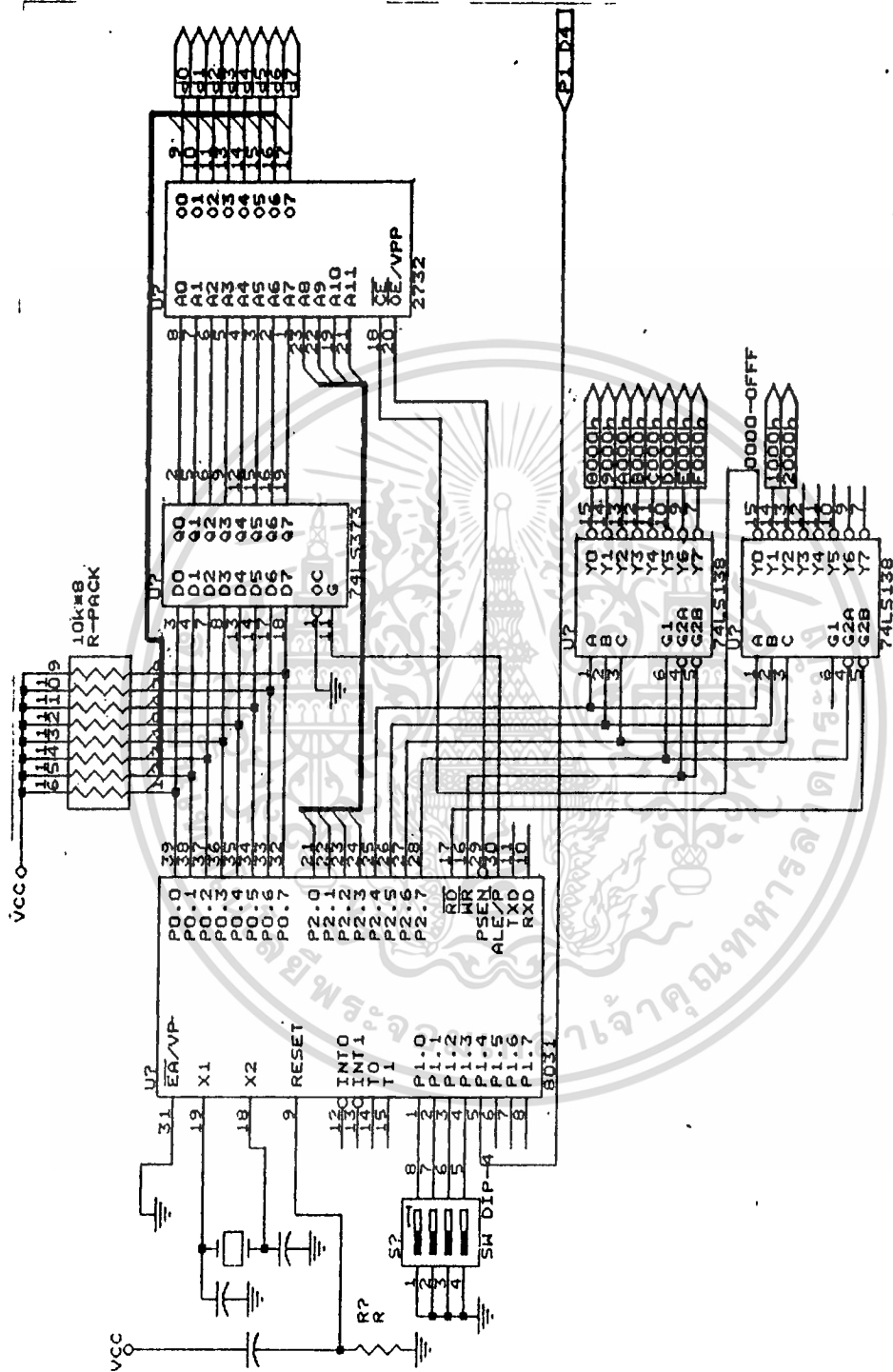
รูปที่ 3.16 วงจร CPU ของส่วนวงจรหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 วงจรประกอบของส่วนวงจรหลัก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



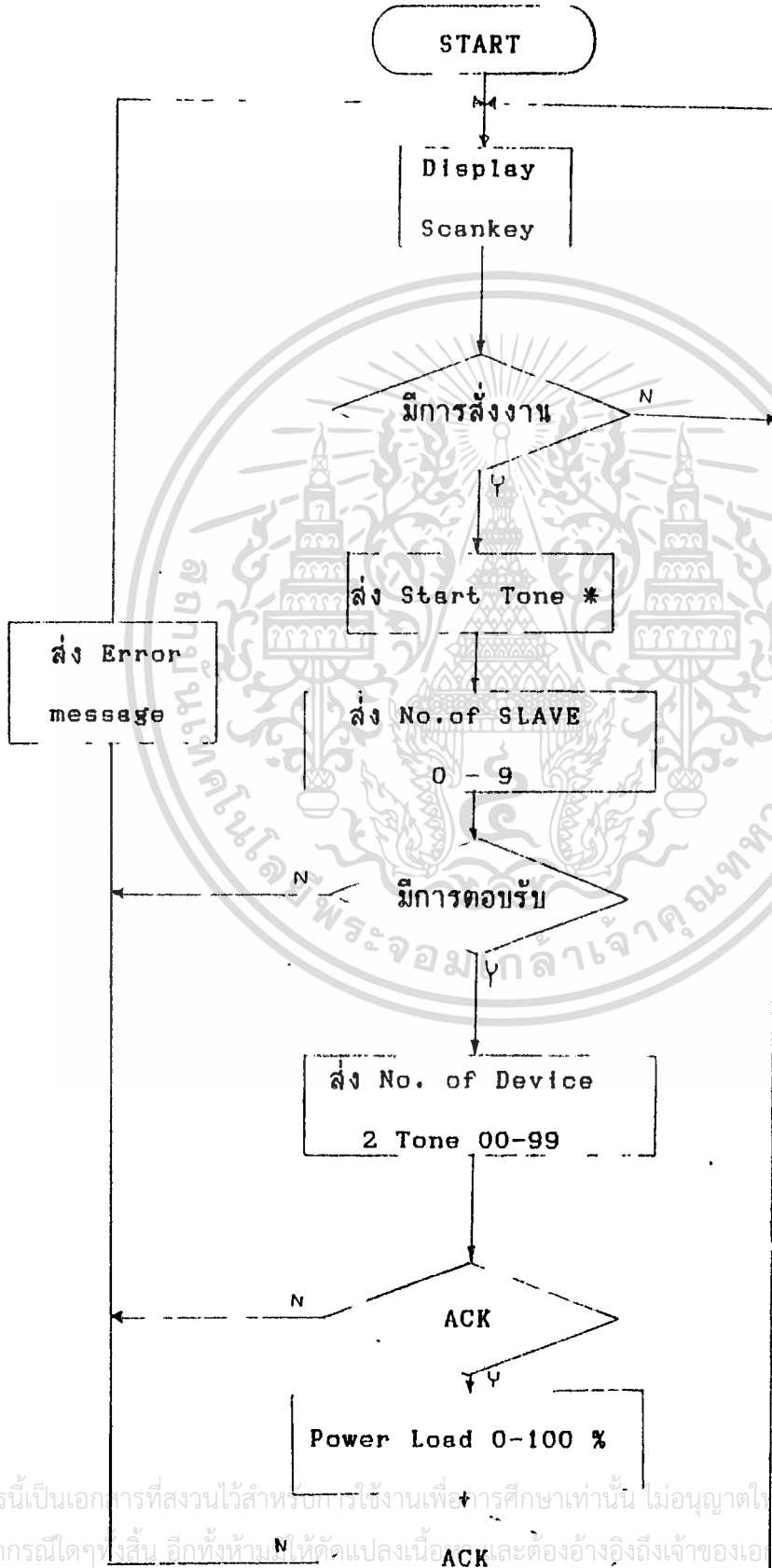
รูปที่ 3.18 วงจร CPU ของส่วนวงจรย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



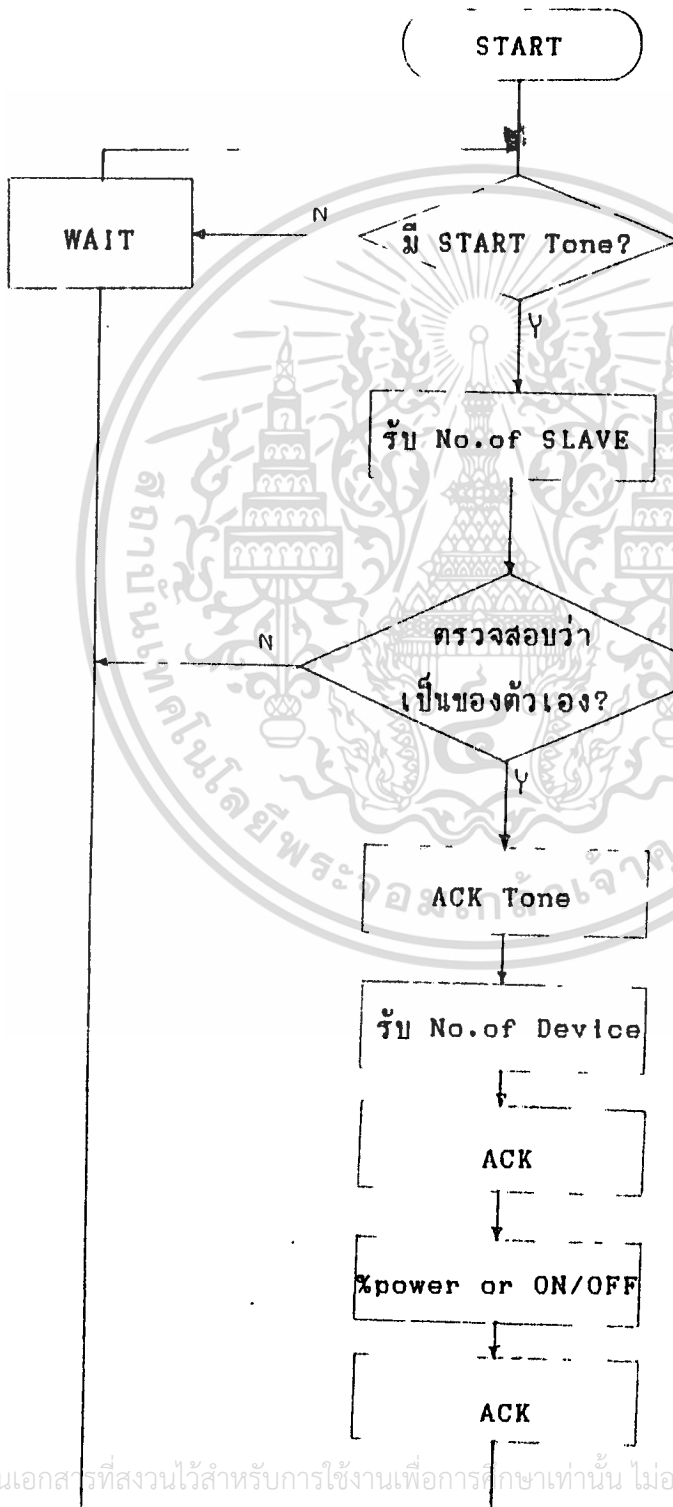
### 3.6 ขั้นตอนการติดต่อระหว่าง MASTER และ SLAVE

-MASTER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม N ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-SLAVE



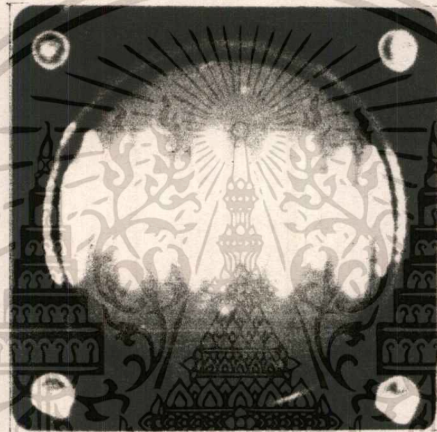
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองวงจรรับส่งสัญญาณ DTMF

เป็นการทดลองเพื่อเปรียบเทียบสัญญาณ DTMF ที่ได้จากทางภาคส่ง และสัญญาณที่ทางด้านรับสามารถรับได้



รูปที่ 4.1 สัญญาณที่ได้จากทางด้านส่ง

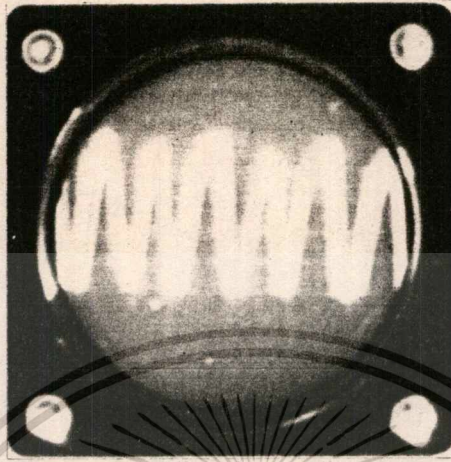


รูปที่ 4.2 สัญญาณที่ได้ทางด้านรับ

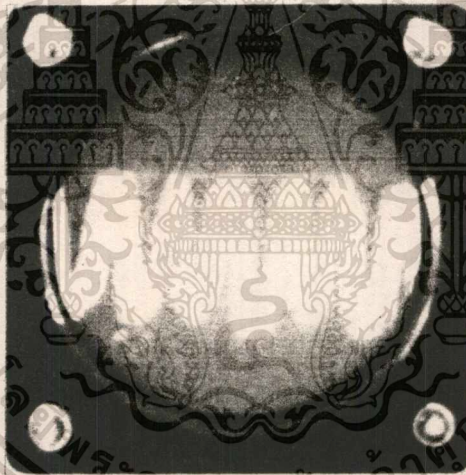
4.2 การทดลองรับสัญญาณที่ระยะต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ทางเพื่อการศึกษาเท่านั้น สัญญาณที่ส่งไปใช้ประโยชน์ด้านการค้า  
เป็นการทดลองเปรียบเทียบรูปร่างของสัญญาณที่ได้ เมื่อมีการเปลี่ยนแปลง  
เมื่อกฎมิต่างกัน อีกทงห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

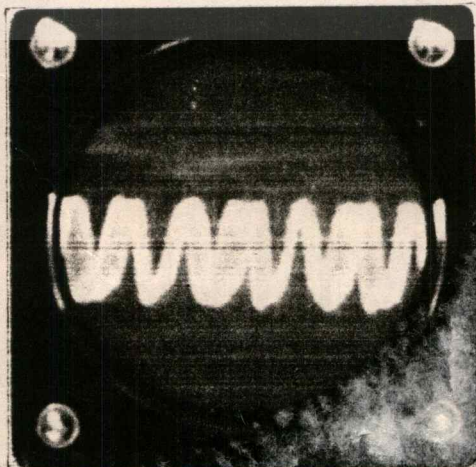
ระยะทางในการรับส่งสัญญาณ



รูปที่ 4.3 สัญญาณที่ได้จากระยะทาง 1 เมตร

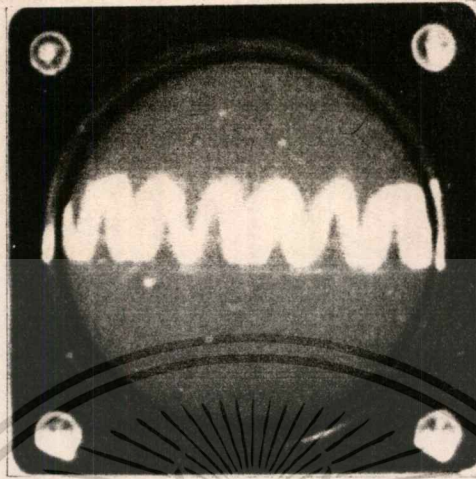


รูปที่ 4.4 สัญญาณที่ได้จากระยะทาง 10 เมตร



รูปที่ 4.5 สัญญาณที่ได้จากระยะทาง 50 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งรูปที่ 4.5 สัญญาณที่ได้จากระยะทาง 50 เมตร สารทศครั้งที่มีการนำไปใช้



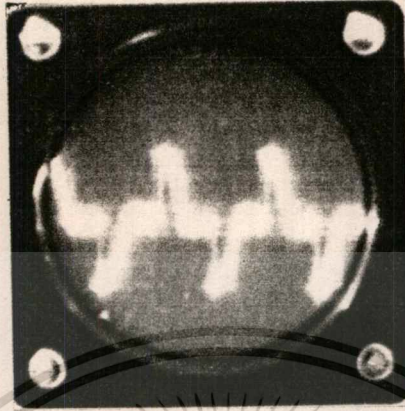
รูปที่ 4.6 สัญญาณที่ได้จากระยะทาง 100 เมตร

#### 4.3 การทดลองภาคความคมมเฟส

เป็นการทดลองวัดรูปสัญญาณที่ตกคร่อมโหนดที่ได้จากการเปลี่ยนแปลงค่าสัญญาณควบคุมดิจิทัลค่าต่าง ๆ กัน



รูปที่ 4.7 สัญญาณที่ตกคร่อมโหนดเมื่อสัญญาณควบคุม 00H



รูปที่ 4.8 สัญญาณที่ตกคร่อมโหลดเมื่อสัญญาณควบคุม 08H



รูปที่ 4.9 สัญญาณที่ตกคร่อมโหลดเมื่อสัญญาณควบคุม 0FH

.....

บทวิจารณ์และสรุป

การดำเนินการของโครงการที่ได้จัดทำไปส่วนใหญ่ยังอยู่ในขั้นศึกษาและทดลอง  
วงจรที่ทำการออกแบบขึ้นมาแยกเป็นภาค ๆ การทำงานของวงจรที่ใช้โดยรวมยังไม่ให้  
ผลอยู่ในระดับที่สมบูรณ์ โดยยังมีจุดบกพร่องที่ต้องนำไปพัฒนาต่อไปดังนี้

๑. ความแรงของสัญญาณยังไม่สามารถที่จะทำการส่งข้อมูลไปได้ไกลเพียงพอ  
ตามที่ต้องการ ซึ่งต้องพัฒนาต่อไป

๒. ปัญหาสัญญาณรบกวนที่เกิดขึ้นในสายไฟหลายรูปแบบมีผลต่อการทำงานของ  
วงจรอย่างมาก เช่น สัญญาณรบกวนความถี่สูงและทรานเซียนจากโหลดประเภทตัว  
เหนี่ยวนำ ปัญหาจากแรงดันกระชอกของโหลดและอื่น ๆ สัญญาณรบกวนอันเกิดจากการ  
เดินสายภายในเครื่องควบคุมรบกวนการทำงานของระบบไมโครโปรเซสเซอร์

๓. ฟังก์ชันที่ใช้งานยังมีน้อยเกินไป ทำให้ความสดวกในการใช้งานมีไม่สูง  
เพียงพอในระดับที่ควรเป็น

ปัญหาต่าง ๆ ที่เกิดขึ้นต้องมีการพัฒนาระบบต่อไป จนกว่าผลการทำงานของ  
วงจรจะอยู่ในขั้นที่น่าพอใจ โดยมีแนวทางแก้ไขดังนี้

๑. ถ้าต้องการส่งสัญญาณไปในสายที่มีความยาวมาก ๆ ต้องเพิ่มชุดทวน  
สัญญาณ (Repeater) เพื่อเพิ่มระยะทางในการส่ง

๒. สร้างวงจรป้องกันสัญญาณรบกวนหลายรูปแบบ เพื่อแก้ปัญหาเพราะรูปแบบ  
ของสัญญาณรบกวนในสายไฟฟ้ามีมากมาย

๓. พัฒนาโปรแกรมต่อไปเพื่อเพิ่มฟังก์ชันการทำงาน

.....



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก. MCS-51 ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว

ไอซีตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (ไม่ต้องต่อกับอุปกรณ์ภายนอกก็ทำงานได้) มีความสะดวกในการใช้งาน และเขียนโปรแกรมควบคุมด้วยภาษาเบสิกได้โดยไม่ต้องศึกษาการทำงานของวงจรเหมือนภาษาแอสเซมบลี หรือจะใช้ภาษาแอสเซมบลีก็ได้

เดิมซี๒๓ ๕1 ไมโครคอนโทรลเลอร์ประกอบด้วยคุณสมบัติ

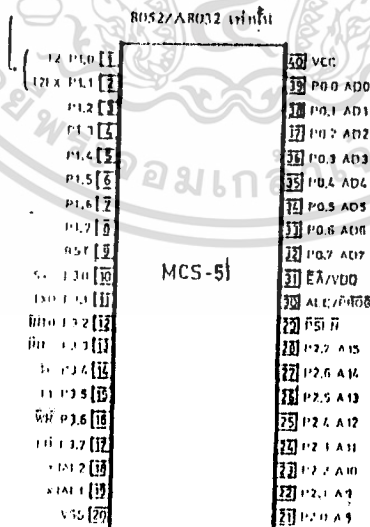
- |   |   |
|---|---|
| - ซีพียู ๘ บิต ที่ควบคุมได้ง่าย   | - ภาาเขต 4 หรือ ๘ กิโลไบต์ (ซีพียู ๘751 และ 8752)   |
| - เครื่องการวางนตถจกครั้งละ 1 บิตได้  | - มีแอดเดรสของหน่วยความจำสำหรับเก็บโปรแกรม จำนวนทั้งหมด ๘4 กิโลไบต์                                     |
| - สายอินพุตและเอาต์พุตจำนวน 3๒ เส้นให้เลือกแอดเดรสแยกส่งหากจากกันได้  | - มีแอดเดรสของหน่วยความจำสำหรับเก็บข้อมูล จำนวนทั้งหมด ๘4 กิโลไบต์                                      |
| - มีเวร็มร่วจวไรภายในเขต 1๒8 ไบต์หรือ 25๖ ไบต์  | - มีคำสั่งทั้งหมดมี 111 คำสั่ง (๘๘ คำขง)  |
| - มงรตั้งเวลา/วงจรในใภเขต 2, 3 หรือ 16 บิต  | - ทำงานด้วยเลขฐานสิบ และฐานสิบหก  |
| - การหนลเป็น UART (Universal Synchronous Asynchronous Receiver Transmitter) ที่รับส่งข้อมูลอนุกรมได้สองทิศทาง | - ตัวแปลภาษาเบสิกมีแอด ๘ กิโลไบต์ (8025 AH-BASIC)   |
| - อินเตอร์รัพท์ แบ่งลกภเป็น 2 ระดับจาก 5 หรือ ๘ แล่ง  | - ควบคุมอิทรณภายในใภตัวควบคุมเบสิก (8052 AH-BASIC)  |
| - มีสัญญาณนาฬิกาภายในตัว  | - มีคำสั่งเฉพาะของภาษาเบสิกที่ใช้สำหรับอินพุตและเอาต์พุต วงจรนำ และอินเตอร์เฟลแบบอนุกรม (8052 AH-BASIC) |
| - มีหน้ ขความจำอ่าหรับเก็บข้อมูลภายใน   |   |

MCS-51 เป็นไอซีที่ออกแบบมาสนองความต้องการของผู้ใช้คือมีสายอินพุตและเอาต์พุตภายในตัวเอง พอร์ทของเอาต์พุตและอินพุตบัฟเฟอร์อินเตอร์เฟล และสายควบคุมอื่น ๆ ที่ใช้แยกข้อมูลกับแอดเดรส และยังมีชุดคำสั่งเพิ่มขึ้นเป็นพิเศษเพื่อใช้จัดการข้อมูล และวงจรตั้งเวลา/วงจรนับด้วย

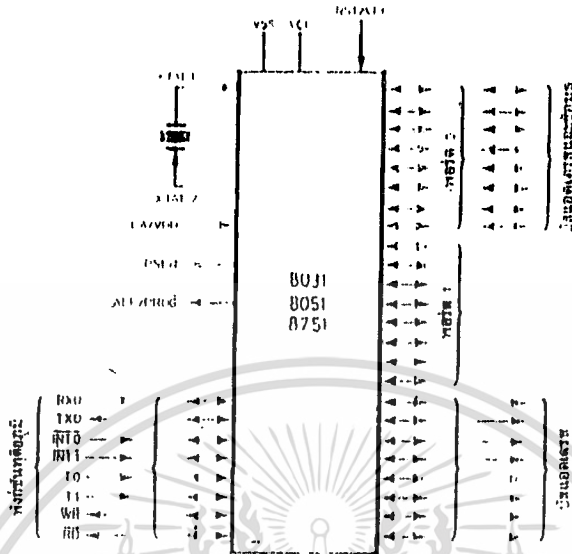
เลขที่	หน่วยความจำภายใน		สี/เวลา/วินาที	ลักษณะพอร์ต ภายนอก
	โปรแกรม	ข้อมูล		
8052A11	8K x 8 ROM	256 x 8 RAM	1 x 16 วินาที	6
8051A11	4K x 8 ROM	128 x 8 RAM	2 x 16 วินาที	5
8051	4K x 8 ROM	128 x 8 RAM	1 x 16 วินาที	5
8032A11	ไม่มี	256 x 8 RAM	1 x 16 วินาที	6
8031A11	ไม่มี	128 x 8 RAM	2 x 16 วินาที	5
-8011	ไม่มี	128 x 8 RAM	2 x 16 วินาที	5
875111	4K x 8 EPROM	128 x 8 RAM	2 x 16 วินาที	5
873111-12	4K x 8 EPROM	128 x 8 RAM	2 x 16 วินาที	5

ตารางที่ ก1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

จากตารางที่ ก1 เป็นรายการของตระกูลเอ็มซีเอส 51 ซึ่งแสดงถึงจำนวนของหน่วยความจำ วงจรตั้งเวลา/วงจรรนับ และระดับของการอินเทอร์รัพท์

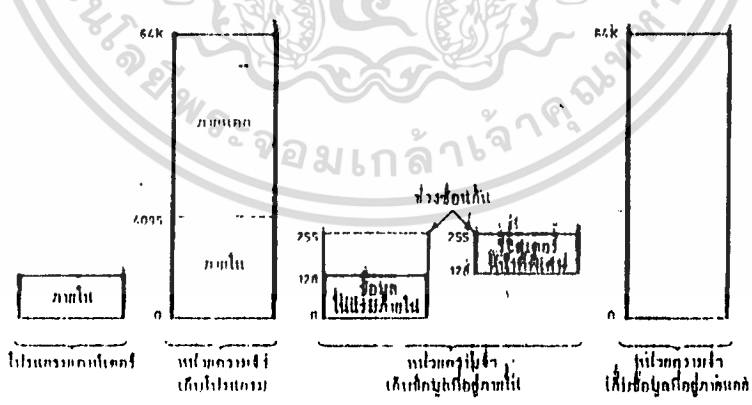


เอกสารนี้เป็นเอกสารที่รูปที่ ก1 แสดงการจัดวางขาต่างๆ ของเอ็มซีเอส 51 ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก2 แสดงหน้าที่ของพอร์ตเมื่อคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก

จากรูปที่ ก1 และ ก2 นั้นแสดงจำนวนขาและหน้าที่การทำงานของ 8051 ซึ่งจะเหมือนกันหมดในตระกูลเอ็มทีเอส 51



- INTERNAL - หน่วยความจำภายใน
- EXTERNAL - หน่วยความจำภายนอก
- PROGRAM COUNTER - วงจรนับคำสั่งโปรแกรม
- PROGRAM MEMORY - หน่วยความจำโปรแกรม
- INTERNAL DATA RAM - หน่วยความจำข้อมูลภายใน
- SPECIAL FUNCTION REGISTERS - ฟิลด์คำสั่งพิเศษ
- INTERNAL DATA MEMORY - หน่วยความจำข้อมูลภายใน
- EXTERNAL DATA MEMORY - หน่วยความจำข้อมูลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ ก3 โครงสร้างภายในหน่วยความจำของเอ็มทีเอส 51

### หน่วยความจำของ MCS 51

จากรูปที่ ก3 เป็นหน่วยความจำภายในตัวของ MCS-51 หน่วยความจำนี้แบ่งได้เป็น 2 กลุ่ม คือ หน่วยความจำสำหรับเก็บโปรแกรม และหน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำแรมมีแอดเดรสที่ต่ำกว่า 4 หรือ 6 กิโลไบต์บรรจุอยู่ในรอม ส่วนเอ็มซีเอส 51 ที่ไม่มีรอมภายในจะให้หน่วยความจำภายนอกซึ่งอาจเป็นรอมแรม หรืออีพรอมแทนก็ได้ เอ็มซีเอส 51 จะอ่านหน่วยความจำสำหรับเก็บโปรแกรมเข้ามาเป็นภาษาเครื่องตามลำดับ ส่วนหน่วยความจำสำหรับเก็บข้อมูลจะใช้เป็นที่เก็บตัวแปร, การคำนวณผลลัพท์ทันที, การจัดการกับข้อมูลขนาด 16 บิต (word), ตารางที่ใช้ในการค้นหาค่าต่าง ๆ และทำหน้าที่อื่นที่คล้ายกัน

หน่วยความจำสำหรับเก็บข้อมูลใช้ร่วมกับหน่วยความจำภายนอกได้ถึง 640 กิโลไบต์ ซึ่งเลือกใช้รอม หรือแรมก็ได้ และยังมีรีจิสเตอร์พิเศษที่ใช้หน่วยความจำภายนอกของแรมได้ 128 หรือ 256 กิโลไบต์

### รีจิสเตอร์ภายในเอ็มซีเอส 51

เอ็มซีเอสมีรีจิสเตอร์อำนวยความสะดวกในการทำงานต่าง ๆ ประกอบด้วย แอคคิวมูลเตอร์ รีจิสเตอร์สถานะ สแตกพอยน์เตอร์ ดาต้าพอยน์เตอร์ (2\*8 บิต หรือ 1\*16 บิต) พอร์ตหมายเลขศูนย์ถึงพอร์ตหมายเลขสาม รีจิสเตอร์แบบคู่ที่ใช้ในการรับและส่งข้อมูลขนานอนุกรม รีจิสเตอร์ 16 บิต ที่เป็นวงจรตั้งเวลาและวงจรรับรีจิสเตอร์ ซึ่งจองไว้สำหรับใช้กับตัวนับที่ 3 รีจิสเตอร์คำสั่ง สำหรับหน้าที่พิเศษ ( เช่นการอินเตอร์รัพท์ RTC : Real Time Clock) และอินพุท เอาท์พุทแบบอนุกรม

### สายต่าง ๆ ของบัสและพอร์ต

จะเห็นโครงสร้างภายในของเอ็มซีเอส 51 ในรูปที่ ก3 เราจะสมมติให้มีบัสสองทิศทาง 4 เส้น และพอร์ตขนาด 8 บิต ตามทฤษฎี แต่ที่จริงนั้นจะใช้ได้ก็ต่อเมื่อมีการใช้หน่วยความจำภายในตัว (รอมหรือแรม)

เมื่อไม่ใช้หน่วยความจำภายใน พอร์ต 0 และพอร์ต 2 จะถูกใช้เป็นบัสของข้อมูลและแอดเดรส ดังนั้นพอร์ต 2 พอร์ตยังคงใช้งานเป็นอินพุทและเอาต์พุทพอร์ต ไม่สามารถใส่ไดรฟ์อื่น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ทำหน้าที่เป็นสายสัญญาณแอดเดรส  $A_7 \dots A_0$  ส่วนพอร์ท 0 ทำหน้าที่เป็นสายสัญญาณแอดเดรส  $A_7 \dots A_0$  ออกจากบิตข้อมูล  $D_7 \dots D_0$

เอาต์พุตของขา RD และ WR มาจากสายเอาต์พุตของพอร์ท 3 โดยโปรแกรมภายในใช้สัญญาณ RD และ WR เพื่อการเขียนและอ่านข้อมูลกับข้อมูลของหน่วยความจำภายนอก

### ขา $\overline{PSEN}$ ของ MCS 51

ขา  $\overline{PSEN}$  เป็นขารับสัญญาณสำหรับเปิดให้มีการอ่านหน่วยความจำภายนอก ถ้าสิ่งเกิดทุกๆ รอบคำสั่งระหว่างการทำงานด้วยโปรแกรมในรอมหรือในอีพรอม จะเห็นว่าสัญญาณ  $\overline{PSEN}$  ทำงานถึงสองครั้งเหมือนสัญญาณ ALE เพราะว่ามี การอ่านข้อมูลจำนวน 2 ไบท์ ในแต่ละรอบคำสั่ง

ขา  $\overline{PSEN}$  นี้ไม่ทำงานเมื่อมีโปรแกรมในหน่วยความจำภายใน และถ้าหากหน่วยความจำภายนอกไม่มีข้อมูลบรรจุอยู่  $\overline{PSEN}$  ก็จะไม่ทำงานด้วยเช่นกัน

ส่วน 8052 AH BASIC ไม่ใช้สัญญาณ  $\overline{PSEN}$  เลย เพราะใช้รอมภายในตัว ซึ่งมีหน้าที่เป็นตัวแปลภาษาเบสิก

### ขา $\overline{EA}$ ของ MCS 51

เป็นขาอินพุตที่ใช้ร่วมกับแอดเดรสภายนอก โดยจะมีค่าลอจิก "0" เมื่อโปรแกรมเมอร์อ่านคำสั่งจากหน่วยความจำภายนอก (ปกติโปรแกรมเมอร์จะอ่านหน่วยความจำภายในน้อยกว่าอ่านจากหน่วยความจำภายนอก) ขา  $\overline{EA}$  ยังมีหน้าที่เป็นอินพุตสำหรับป้อนแรงดัน 21 โวลต์ เพื่อโปรแกรมให้กับอีพรอม (สำหรับกรณีที่ใช้ 8751 หรือ 8752)

### อินเตอร์รัทท์และชุดคำสั่ง

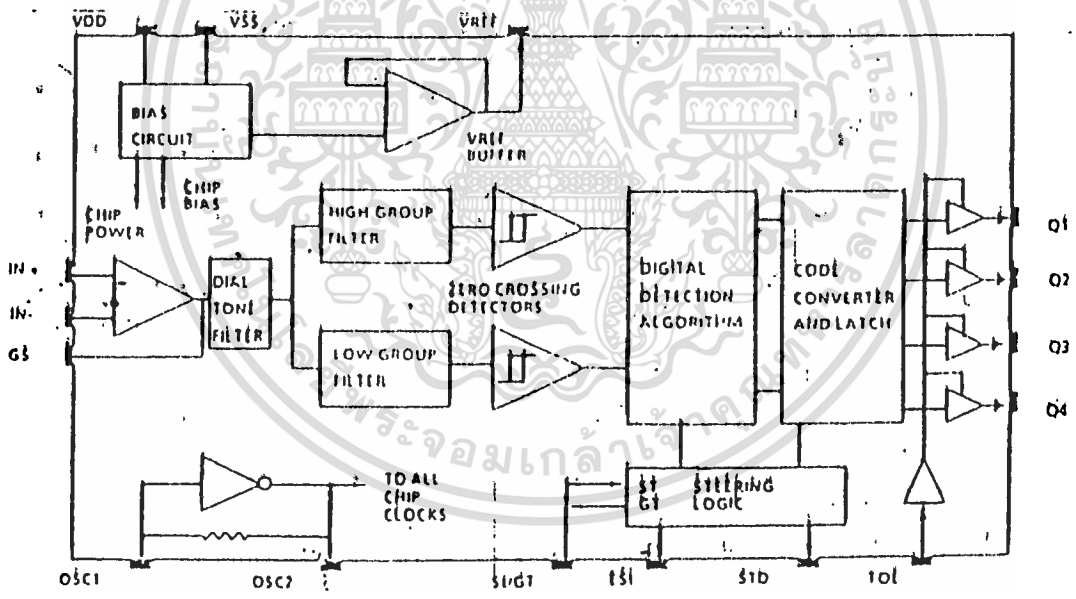
8051 รับการอินเตอร์รัทท์ได้ 5 แหล่ง ส่วน 8052 รับอินเตอร์รัทท์จากอุปกรณ์อื่นๆได้ถึง 6 แหล่ง คือ ขา INTO และ ขา INT1 (กำหนดให้ใช้ระดับพัลส์หรือขอบขาขึ้น ขาลงได้) วงจรนับและวงจรตั้งเวลาที่ 0 และ 1 (สำหรับ 8052 เพิ่ม)  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้งานเพื่อการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### ข. MT 8870 ไอซีถอดรหัสความถี่โทรศัพท์

เป็นไอซีที่มีการทำงานแปลงสัญญาณความถี่โทรศัพท์ชนิดกดปุ่ม (ชนิด Tone หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอลขนาด 4 บิต โดยใช้งานร่วมกับ X'TAL 3.579 MHz เท่านั้น ความหมายของคำว่า ถอดรหัสความถี่โทรศัพท์ หมายถึง การแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด TONE หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอล ซึ่งไอซี MT 8870 ใช้แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 บิต

ในยุคก่อน การออกแบบวงจรถอดรหัสความถี่ของโทรศัพท์ มักใช้ไอซีจำนวน เฟลลือกลุ่ม ซึ่งสร้างปัญหาสารพัด ไม่ว่าเรื่องของความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจร ขนาดของวงจรที่ใหญ่ เพราะต้องใช้ไอซีจำนวนมาก



รูปที่ ข1 BLOCK ภายในของ MT 8870

#### คุณสมบัติ MT8870

- เป็นตัวรับและถอดรหัสความถี่ (DTMF receiver)
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดใหม่ (Guard Time) ได้

เอกสารนี้เป็นเอกสารลิขสิทธิ์หรือทรัพย์สินทางปัญญาของผู้อื่น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

flow	high	NU	TOE	04	03	02	01
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	1336	0	H	1	0	1	0
941	1209	*	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

รูปที่ ๒2 แสดงค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ

**ภาคกรองสัญญาณความถี่**

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่สูงและความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิดคาปาซิเตอร์ฟิลเตอร์ (six-order switched capacitor band pass filter) ซึ่งความถี่ที่แยกออกได้ 2 ช่วงคือช่วงความถี่สูงและความถี่ต่ำ

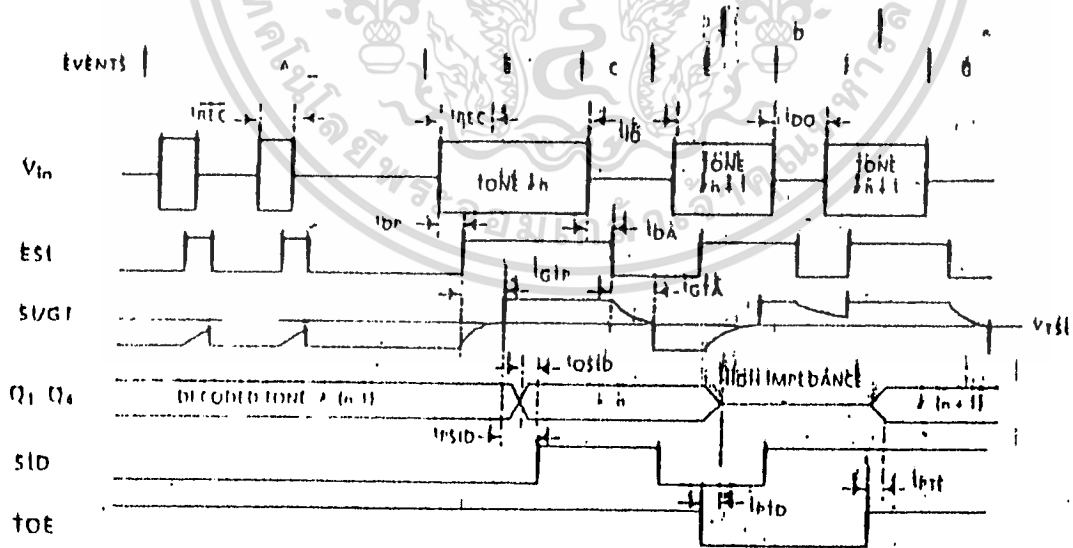
**ภาคถอดรหัส**

ความถี่ DTMF ที่ถูกรองเรียบร้อยแล้ว จะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิทัล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา Est (early steering) ก็จะแอดทิฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่าง ๆ นั้น แสดงดังรูปที่ ๒2

### ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นเวลาพอสมควรมิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น " HIGH " นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา เมื่อขา Est เป็น " HIGH " ทำให้  $V_c$  สูงขึ้นตัวเก็บประจุ C จะคายประจุ ทำให้แรงดัน  $V_c$  สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานขอให้คุณดูจากแผนภูมิเวลาหรือไทม์มิงไดอะแกรม (timing diagram) ดังรูป จะเข้าใจได้ง่ายกว่า

### ขั้นตอนการทำงาน



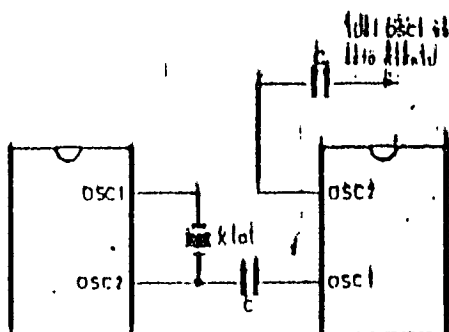
## อธิบายขั้นตอนการทำงาน

- A - ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาท์พุทไม่เปลี่ยน
- B - ความถี่ #  $n$  ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกลอทรหัส และแลทช์ไว้ที่เอาท์พุท
- C - จบความถี่ #  $n$  ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่จนกว่าจะได้รับ ความถี่ที่ถูกต้องใหม่
- D - เอาท์พุทเปลี่ยนเป็นไฮอิมพีแดนซ์
- E - ความถี่ #  $n+1$  ถูกตรวจพบ คาบเวลาถูกต้องความถี่ถูกลอทรหัสและ แลทช์ไว้
- F - ความถี่ #  $n+1$  หายไป ช่วงห่างไม่ถูกต้องเอาท์พุทยังคงแลทช์อยู่
- G - จบความถี่ #  $n+1$  ช่วงห่างถูกต้อง เอาท์พุทยังคงแลทช์อยู่จนถึง ความถี่ใหม่ที่ถูกต้อง

วงจร Steering จะทำงานตรงกันข้าม เพื่อให้ได้ช่วงหยุดระหว่างสัญญาณ INTER DIGIT ถูกต้อง ดังนั้นเป็นการดีราวกับว่า เป็นการตัดสัญญาณที่สั้นเกินไป เพื่อเป็นการพิจารณาว่าสัญญาณถูกต้อง ตัว IC MT 8870 นี้จะมีช่วงผิดพลาดในการขาดหาย ของสัญญาณ (drop out) สั้นด้วย เพื่อที่จะพิจารณาช่วงหยุดที่ถูกต้องของสัญญาณ ประโยชน์อันนี้รวมกับความสามารถในการเลือก STEERING TIME CONSTANT ภายนอก จะทำให้ผู้ออกแบบเลือกคุณสมบัติในการใช้งานได้มากมายหลายแบบ

### ภาคกำเนิดความถี่

ในภาคนี้ ภายใน IC จะมีวงจรวจรเวลาอยู่ใน เพียงแต่ต่อ X'TAL ขนาด 3.579 MHz ก็สามารภใช้งานได้ทันที การต่อวงจรถ่ายกำเนิดความถี่ แสดงดังรูป



C = 30pF    Xtal = 5.5614 Hz

### รูปที่ ข4 การต่อวงจรผลิตความถี่

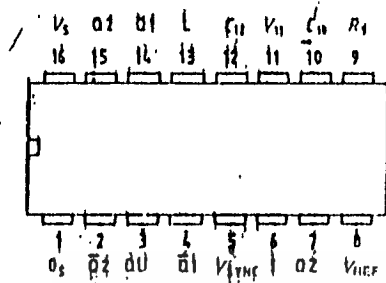
#### ค. TCA 785 ไอซีควบคุมเฟส (Phase Control IC)

TCA 785 เป็นไอซีที่ใช้ในการควบคุมการทำงานของไทรสเตอร์ ไตรแอค และทรานซิสเตอร์ พัลส์ทริกสามารถควบคุมมุมเฟสได้ตั้งแต่ 0 ถึง 180 องศา ใช้งานทั่วไปในวงจรคอนเวอเตอร์ AC Controller ควบคุมกระแสสามเฟส

#### คุณสมบัติของ TCA 785

- ควบคุมจุดผ่านศูนย์ได้แน่นอน
- ใช้งานได้กว้างขวาง
- ใช้เป็นวงจรสวิทช์ที่จุดศูนย์
- LSL Compatible
- ใช้งานกับไฟสามเฟสได้ (3 ไอซี)
- กระแสเอาต์พุต 250 mA
- ช่วงกระแสแรมป์กว้าง
- ช่วงอุณหภูมิการทำงานกว้าง

Pin configuration top view



Pin description

Pin	Symbol	Function
1	$V_5$	Ground
2	$Q_2$	Output 2 Inverted
3	$Q_U$	Output U
4	$Q_1$	Output 1 Inverted
5	$V_{SYNC}$	Synchronous voltage
6	I	Inhibit
7	$Q_Z$	Output Z
8	$V_{HLF}$	Stabilized voltage
9	$R_9$	Ramp resistance
10	$C_{10}$	Ramp capacitance
11	$V_{11}$	Control voltage
12	$C_{12}$	Pulse extension
13	L	Long pulse
14	$Q_1$	Output 1
15	$Q_2$	Output 2
16	$V_5$	Supply voltage

รูปที่ ค1 แสดงรายละเอียดของขา TCA 785

การทำงานของ TCA 785

สัญญาณเชิงค้จะถูกส่งจาก Line Voltage ผ่านค่าความต้านทานสูง ๆ เข้าเพื่อคำนวณหาจุดตัดค้ศูนย์ (voltage  $V_5$ ) และนำไปสร้างสัญญาณแรมป์ ซึ่งเกิดจากการประจุ  $C_{10}$  ด้วยกระแสคงที่ที่ค้คำนวณจากค่า  $R_9$  นำไปเปรียบเทียบกับเพื่อสร้างสัญญาณทริกเกอร์  $V_{11}$  (มุมทริก W) ที่ขึ้นกับแรงดัน  $V_{11}$  ซึ่ง W จะเลื่อนได้ระหว่างมุม 0 ถึง 180 องศา

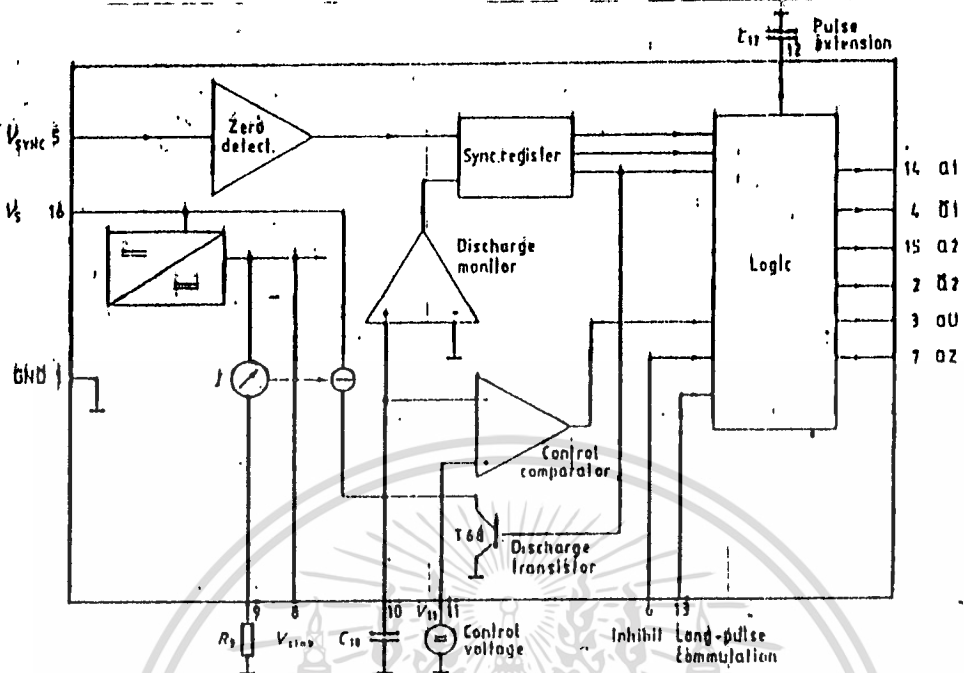
ทุก ๆ ครึ่งค้ล้นจะมีพัลส์ขวกประมาณ 30  $\mu s$  ออกที่ขา  $Q_1$  และ  $Q_2$  ซึ่งปรับความกว้างพัลส์จนถึง 180 องศาได้โดยกำหนดค่า  $C_{12}$  ถ้าขา 12 ต่อลงกราวด์ พัลส์จะเลื่อนระหว่าง W กับ 180 องศา  $Q_1, Q_2$  จะตรงข้ามกับ  $Q_1, Q_2$

สัญญาณ W + 180 องศา ใช้สำหรับควบคุมลอจิกภายนอกทางขา 3

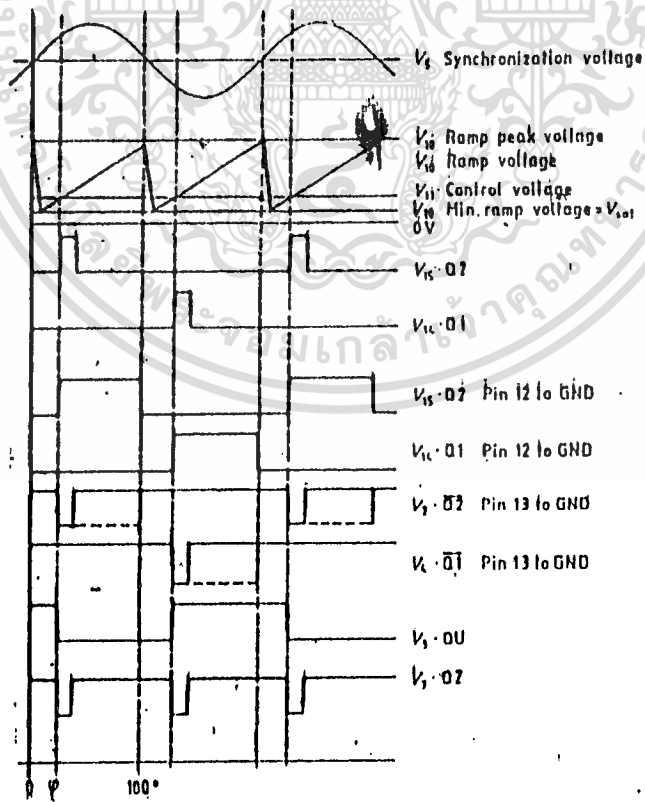
สัญญาณ  $Q_Z$  (pin 7) เป็นสัญญาณ  $Q_1$  NOR  $Q_2$

inhibit input ใช้สำหรับ Disable  $Q_1, Q_2, Q_1, Q_2$

ขา 13 ใช้ขยายเอาท์พุท  $Q_1, Q_2$  ให้เต็มค้ล้น ( 180 - W )



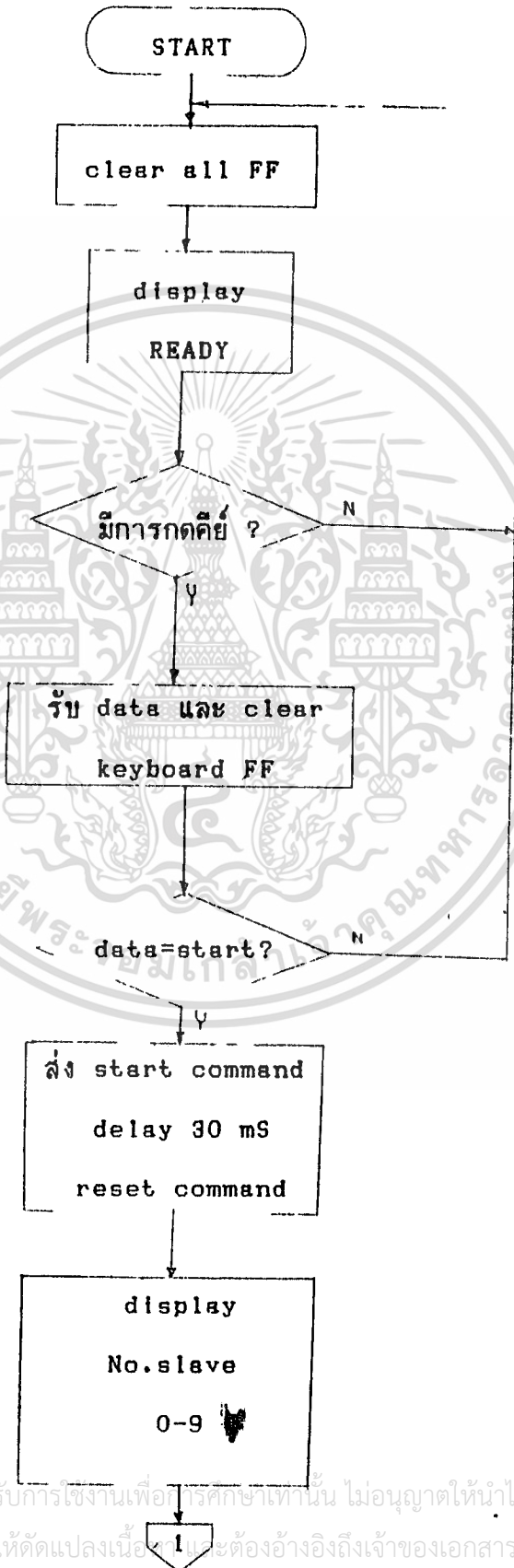
รูปที่ ค2 แสดง Block Diagram ของ TCA 785



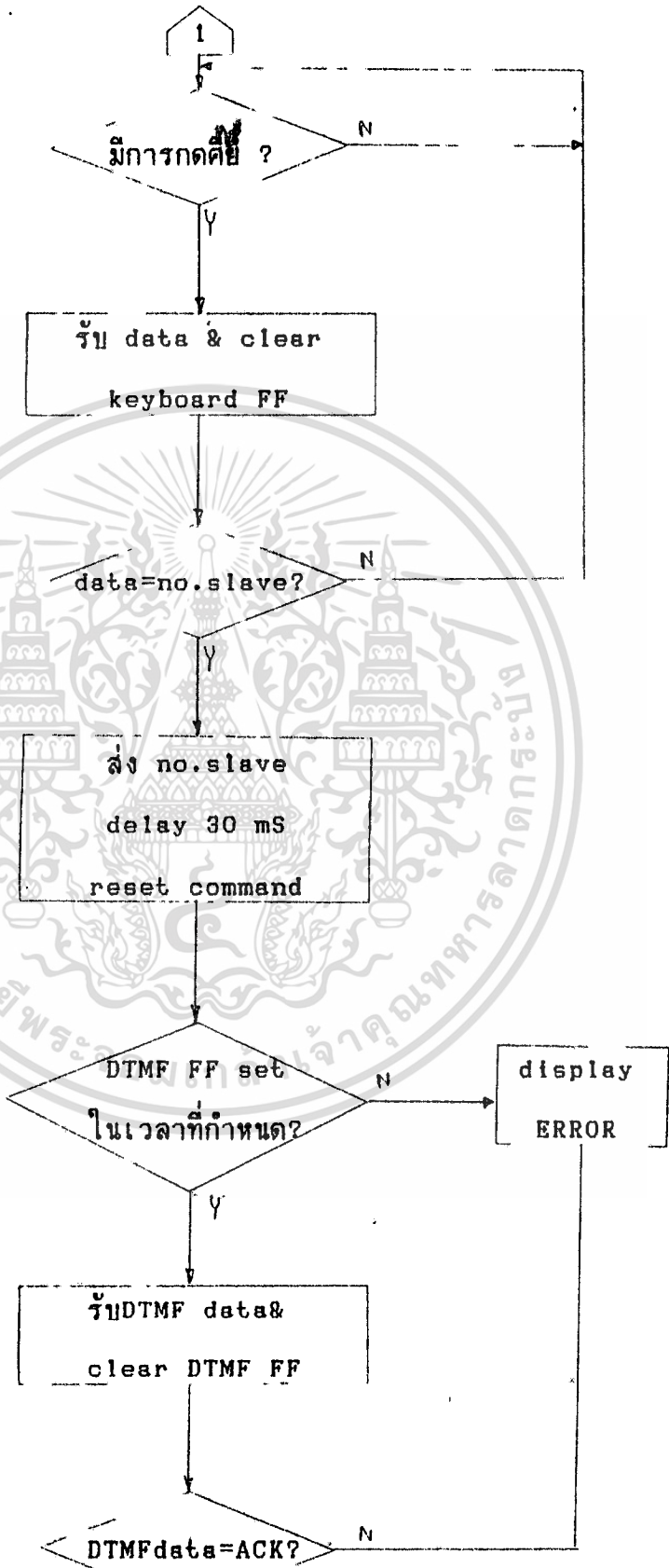
รูปที่ ค3 แสดง Timing Diagram ของ TCA 785

เอกสารนี้เป็นเอกสารที่...  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

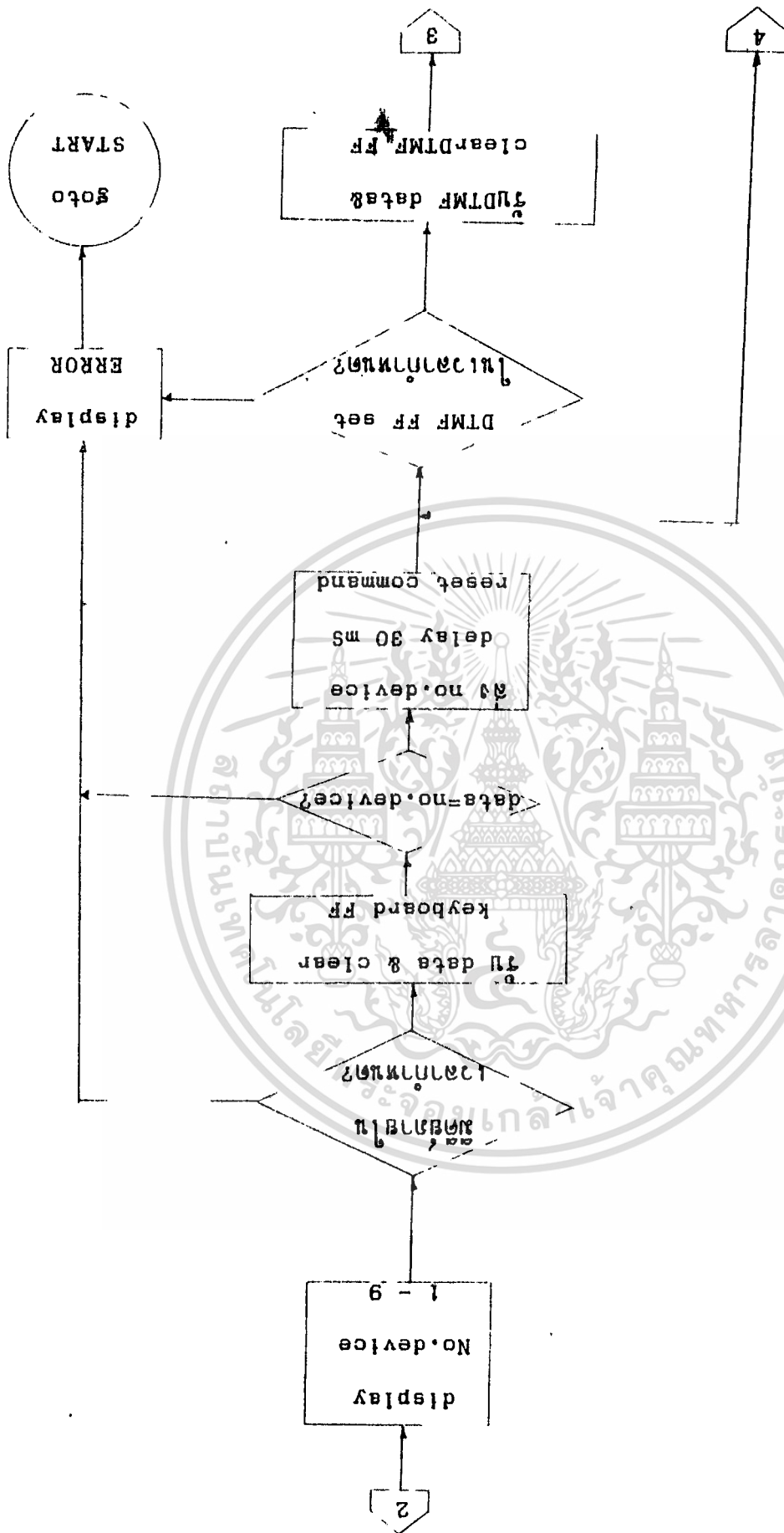
ง. FLOW CHART ของส่วน MASTER



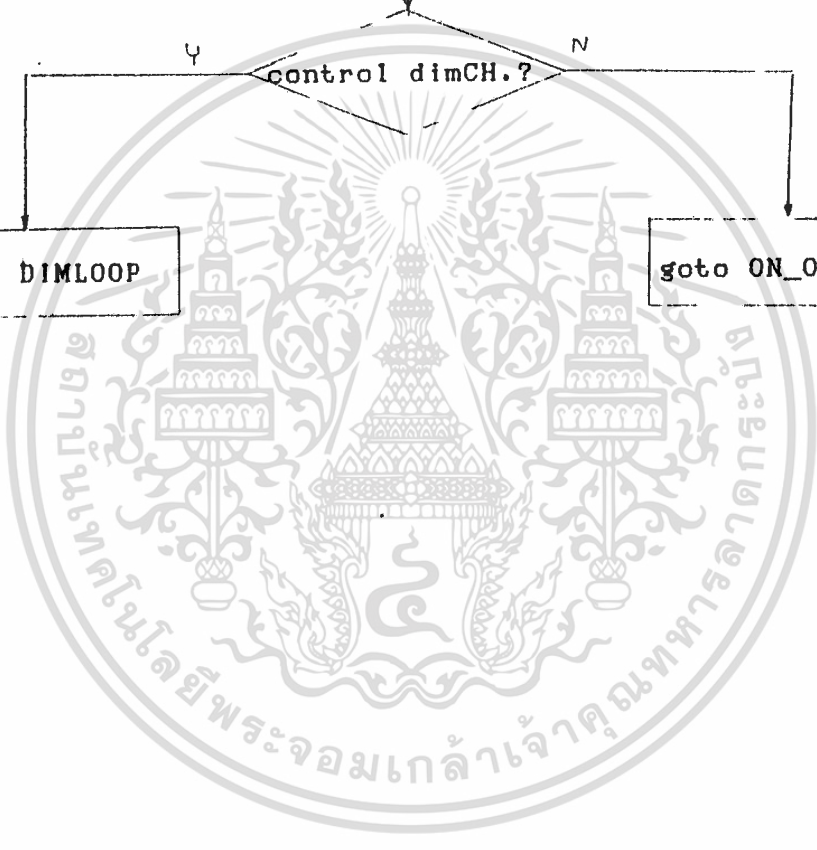
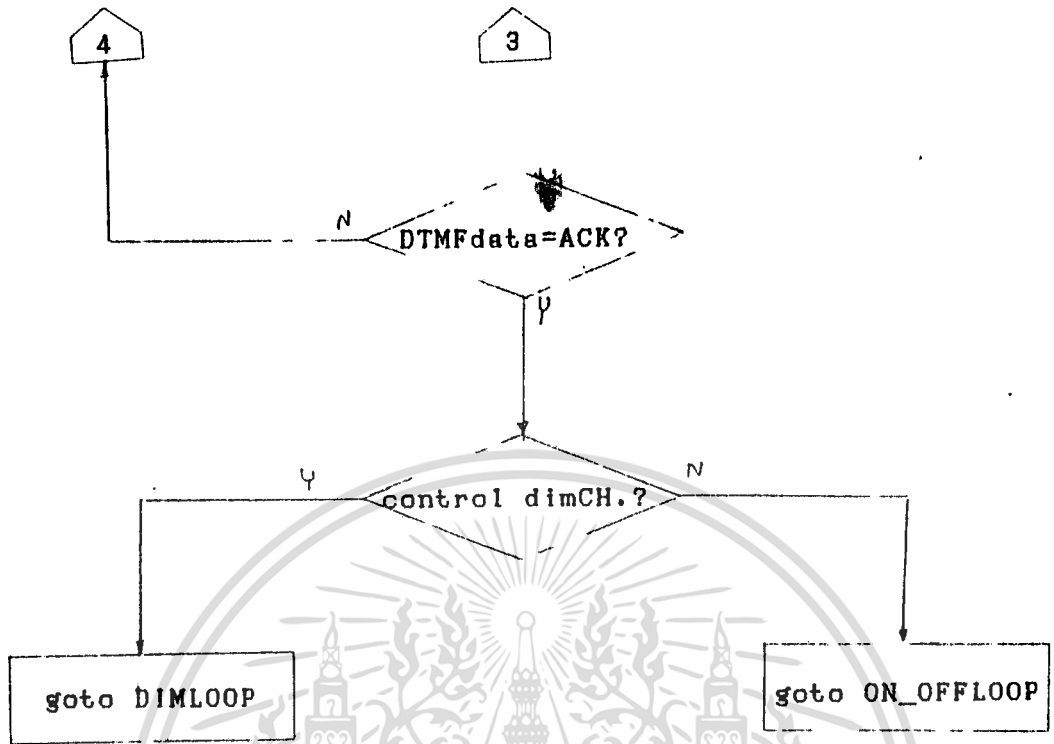
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



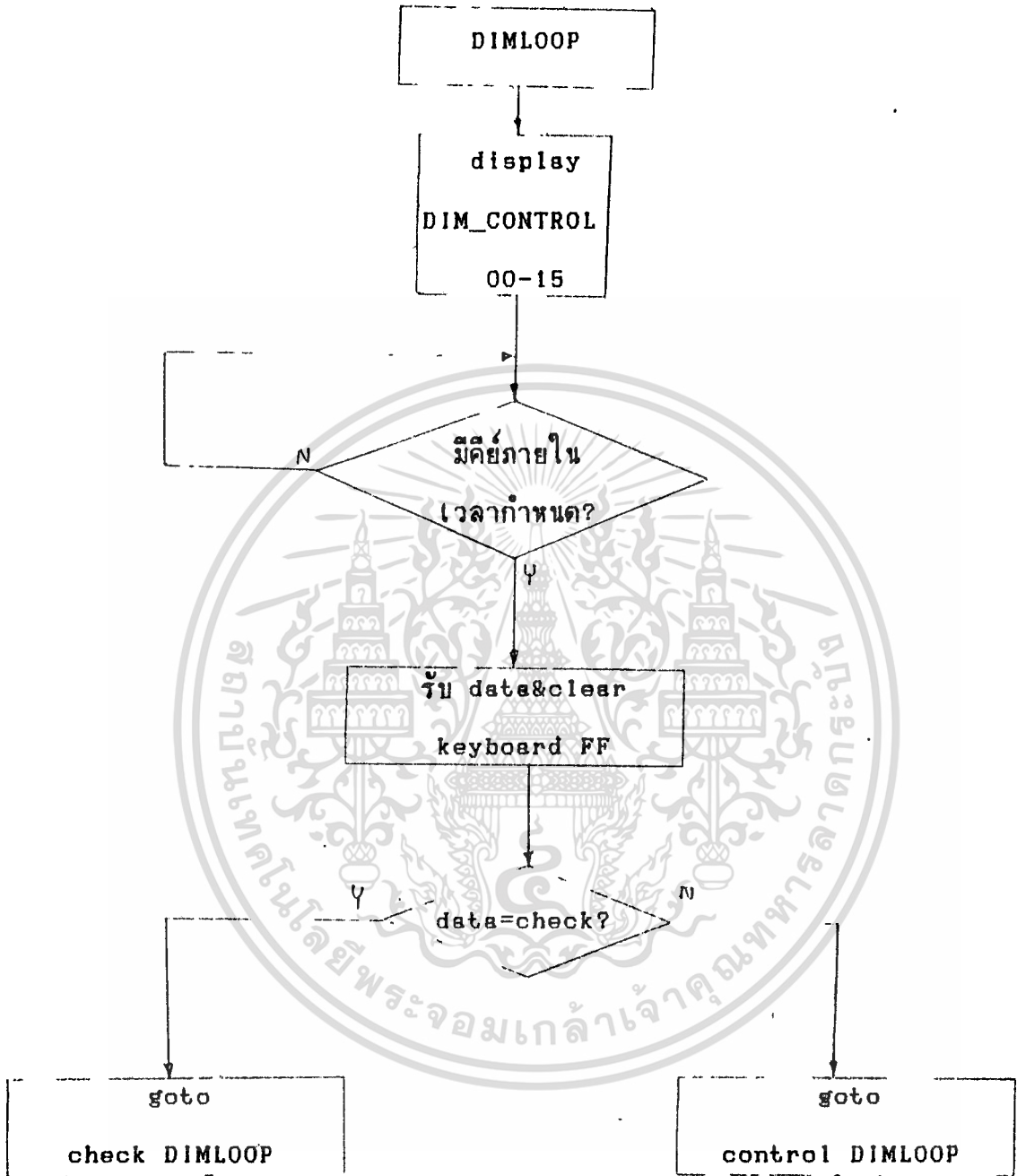
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



check DIMLOOP

ส่ง check key  
delay 30 mS  
reset command

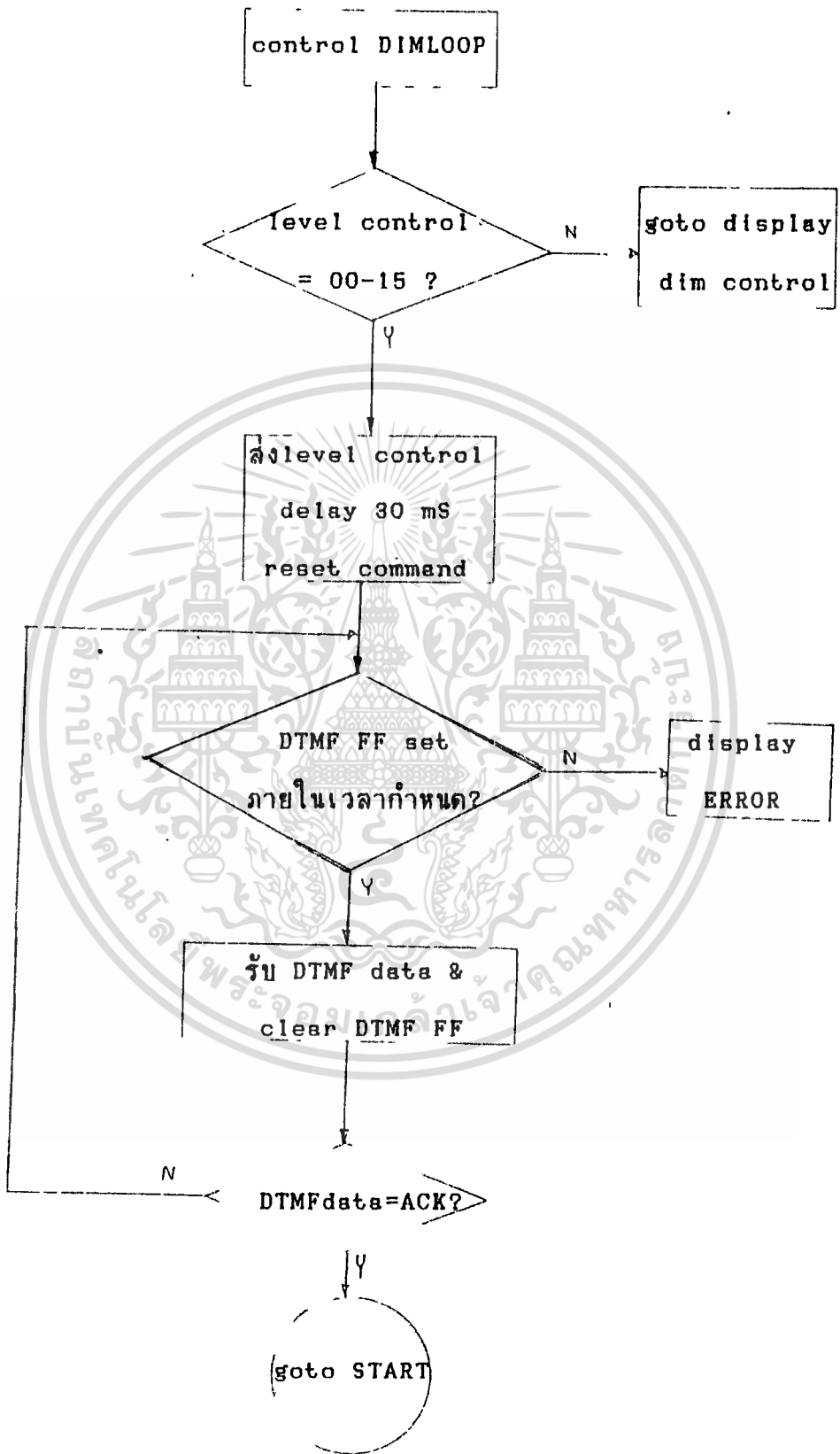
มี data DTMF  
ในเวลากำหนด

display ERROR  
& goto START

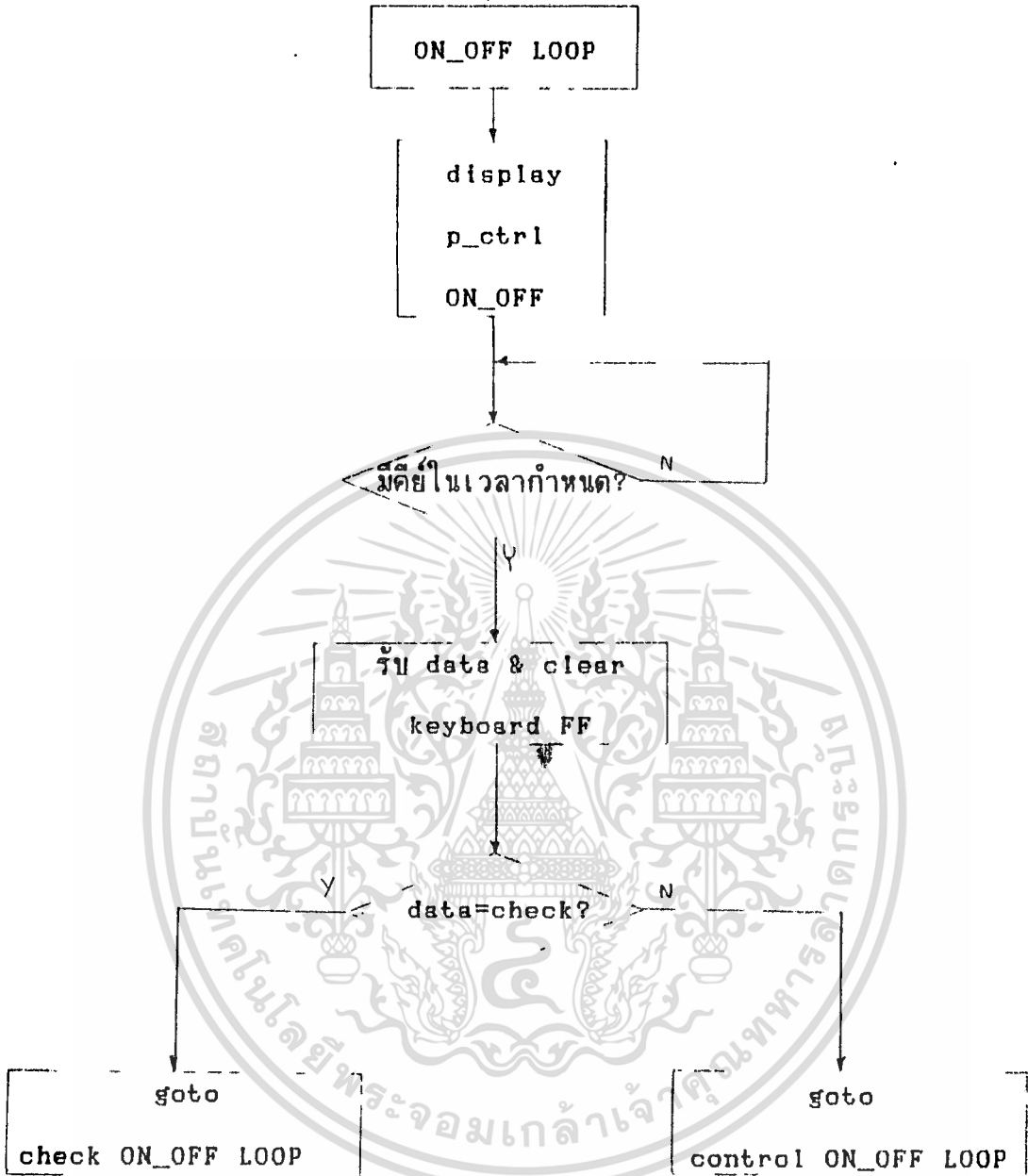
รับ DTMF data &  
clear DTMF FF

display ระดับของ device  
ที่ต้องการจาก DTMF data

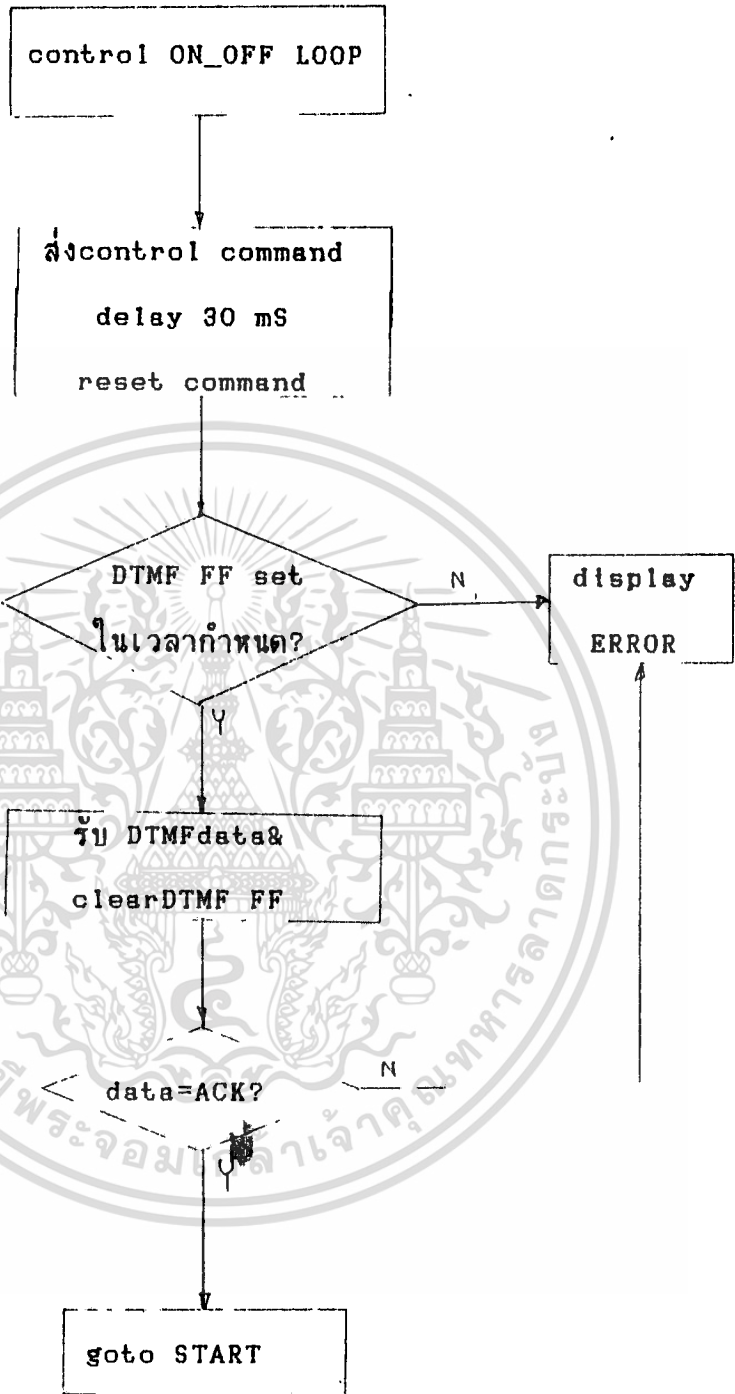
goto START

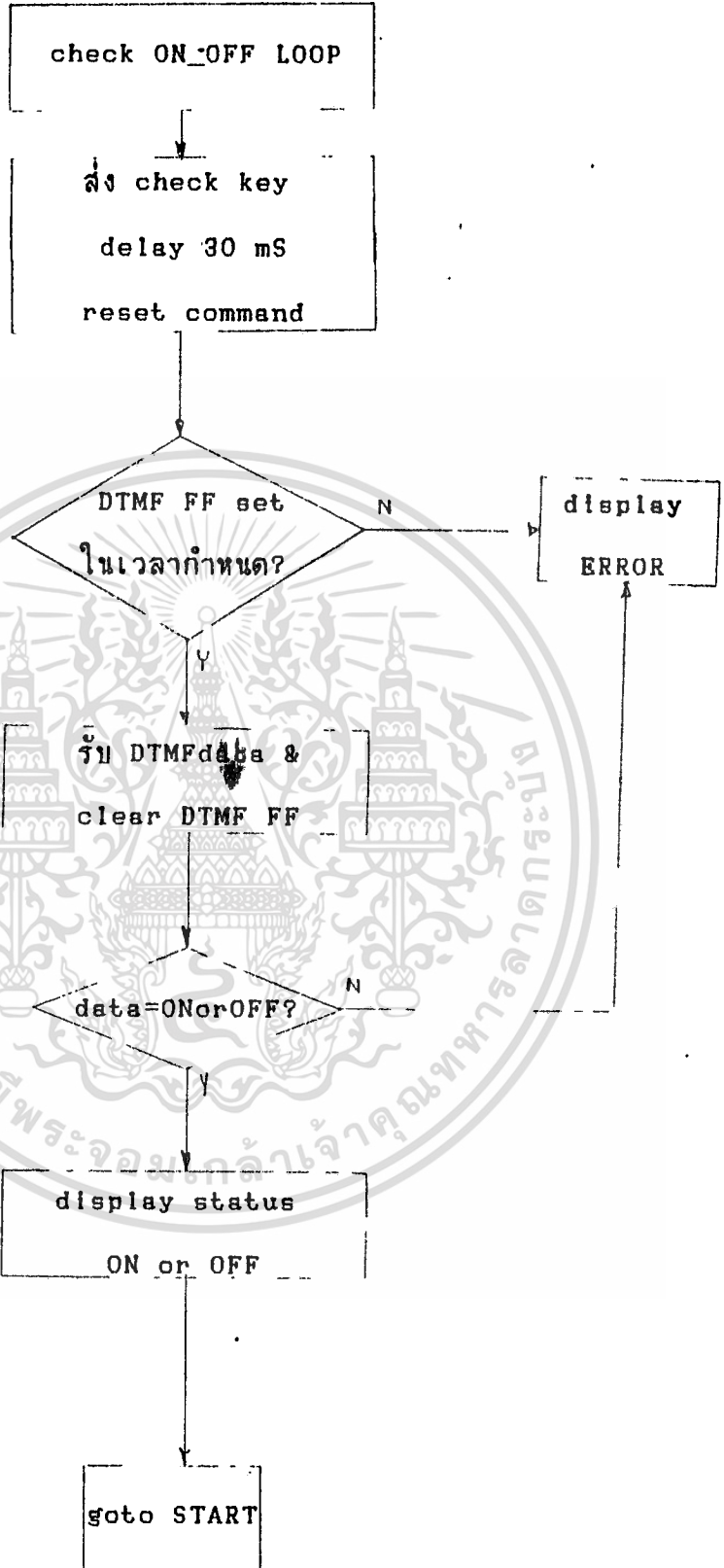


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



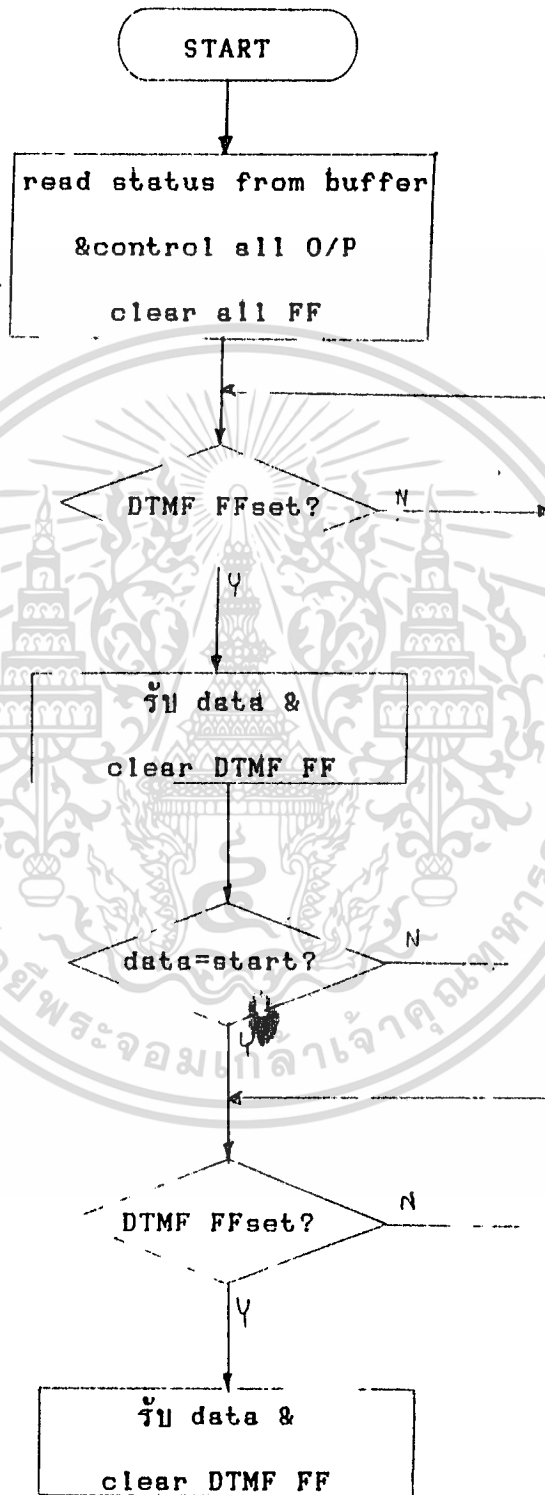
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

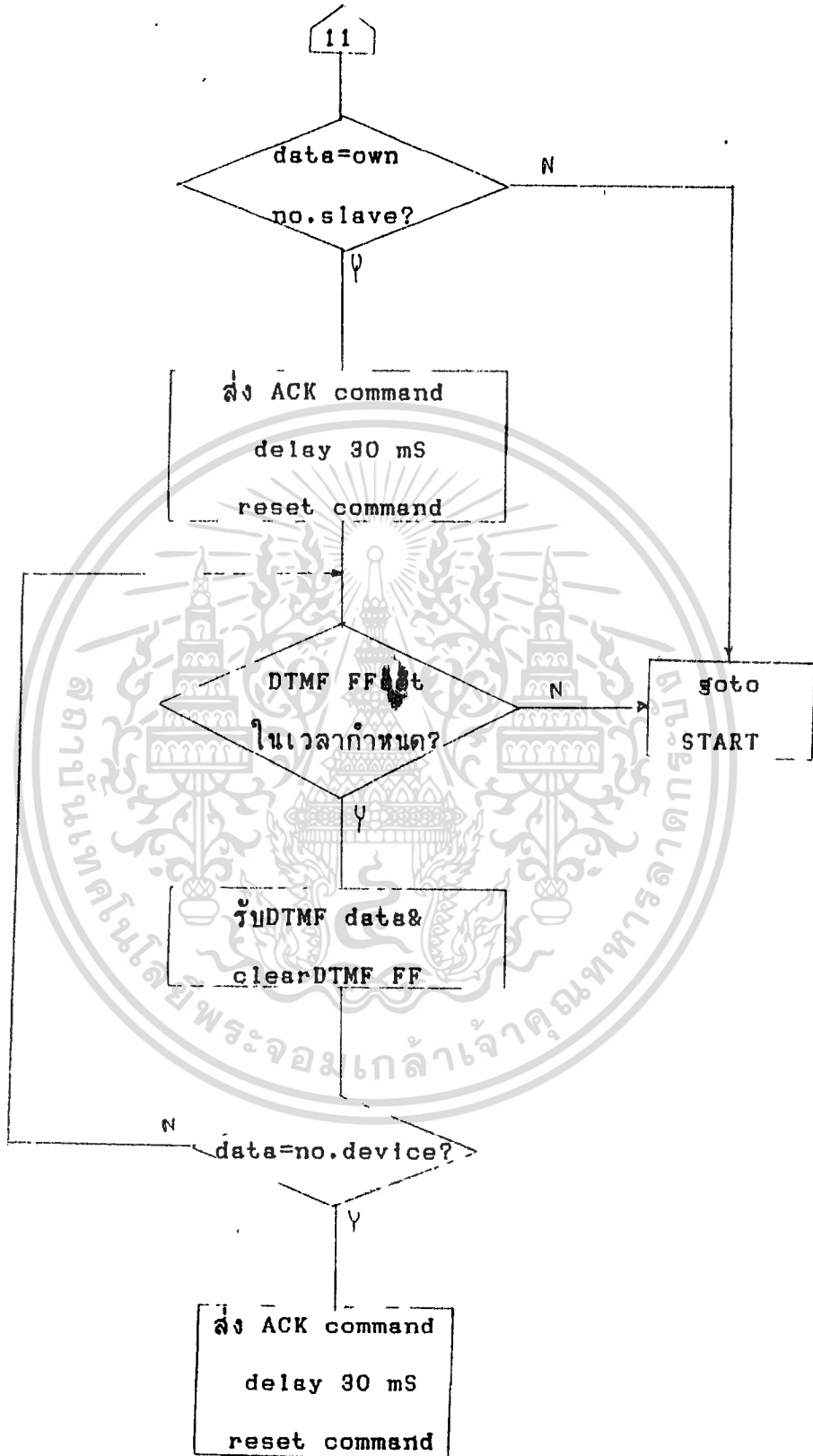


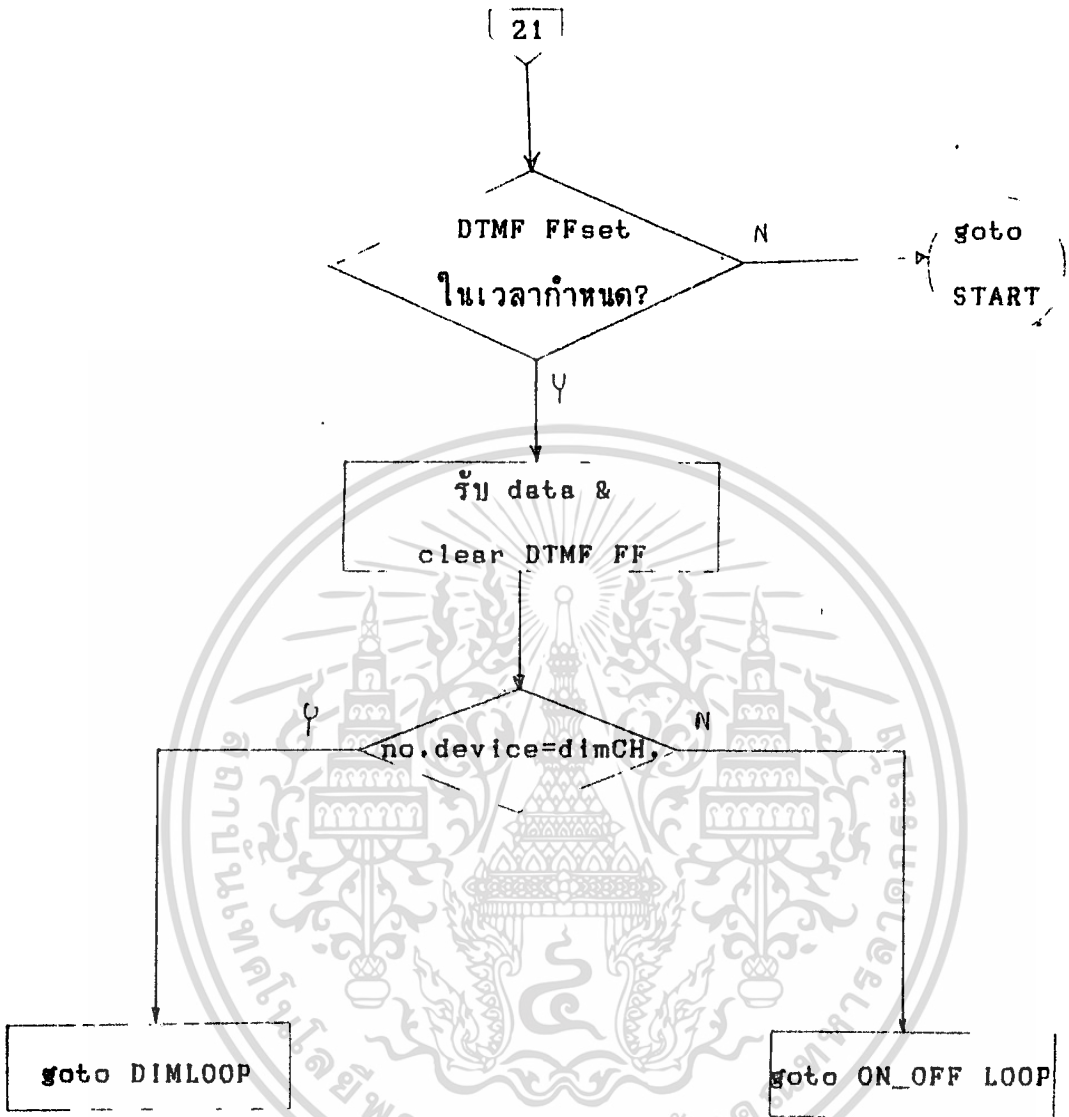


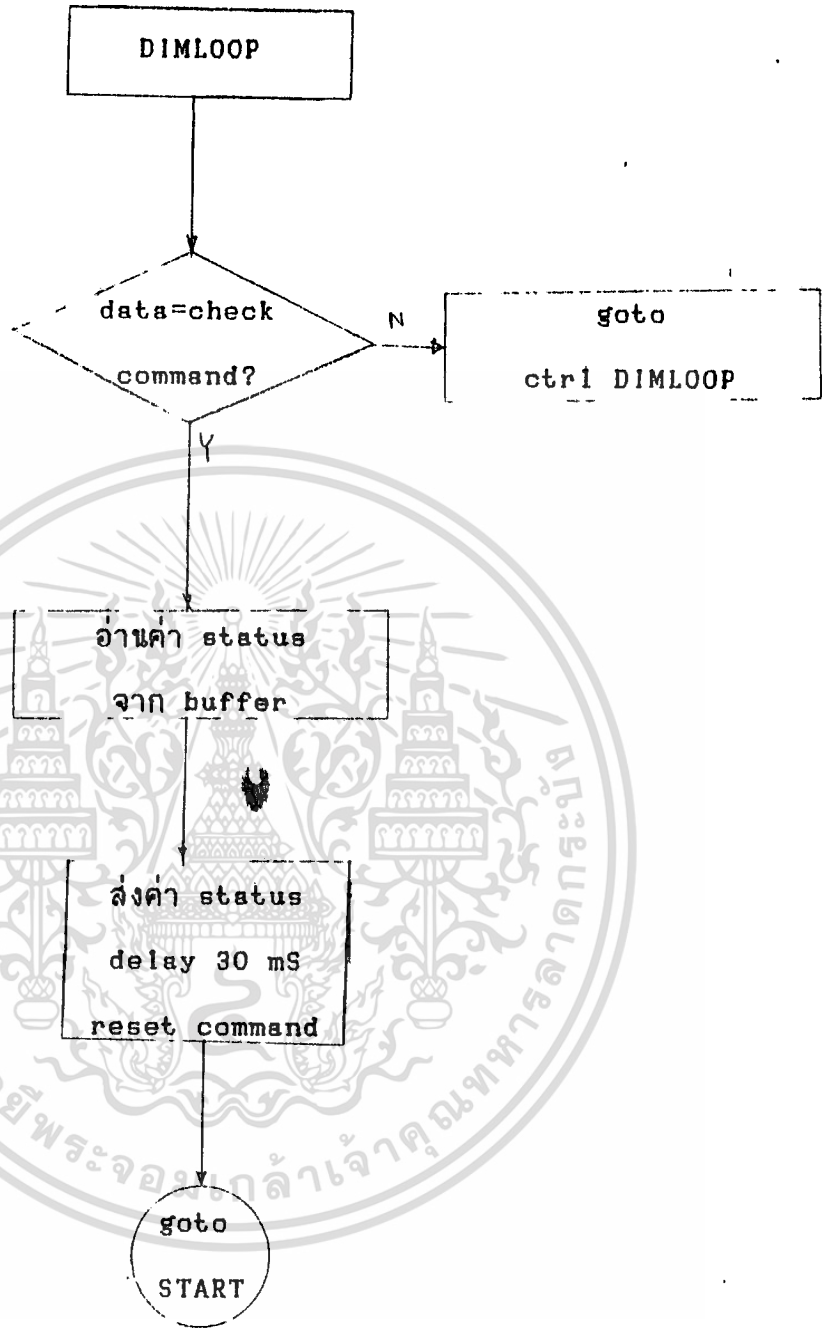
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จ. FLOW CHART ของส่วน SLAVE

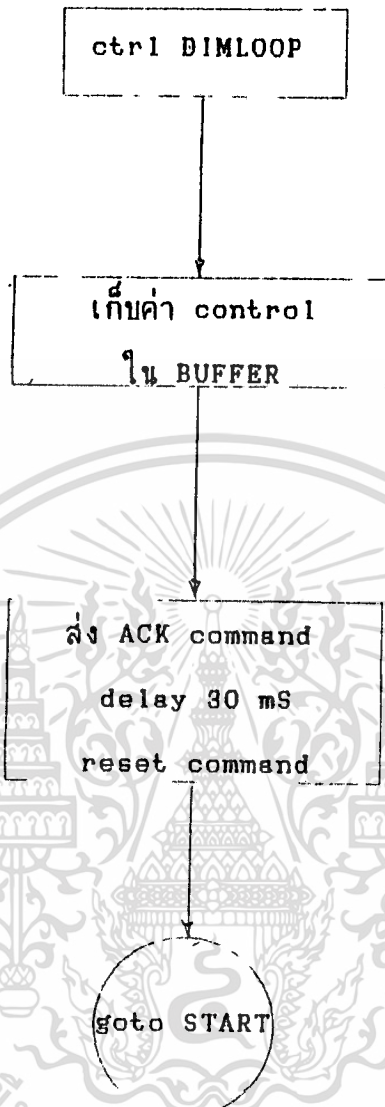


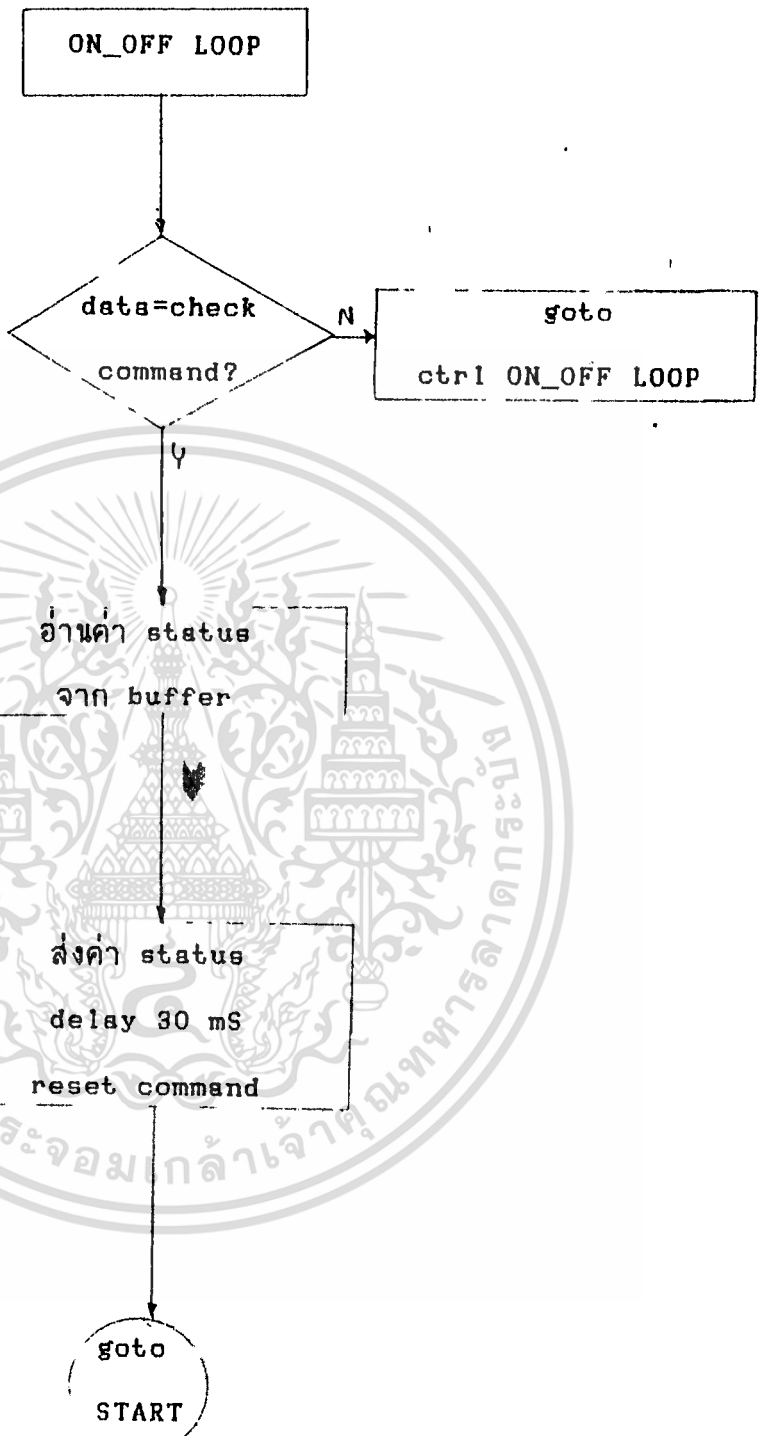




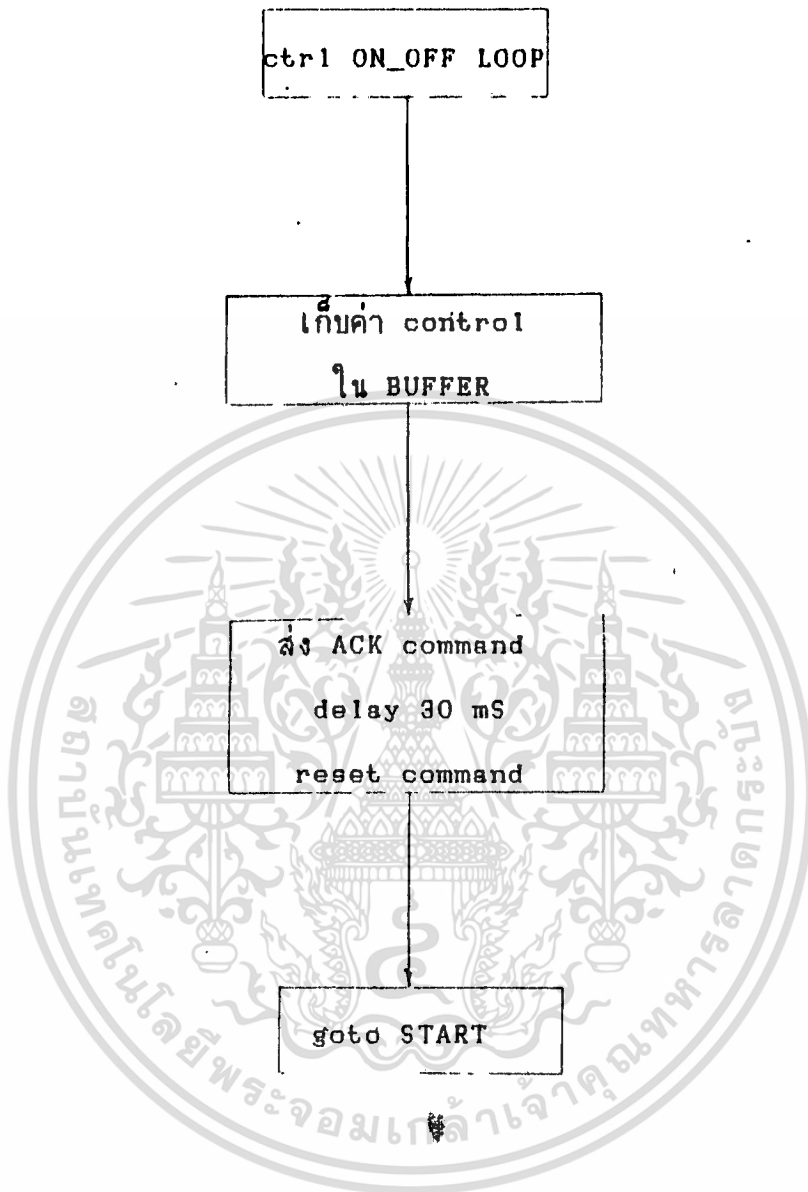


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จ. โปรแกรมควบคุมของส่วน MASTER

READY	EQU	0F00H
NO_SLAV	EQU	0F06H
NO_9	EQU	0F0CH
ERROR	EQU	0F12H
NO_DEVI	EQU	0F18H
N1_9	EQU	0F1EH
DIM_CTL	EQU	0F24H
NOO_15	EQU	0F2AH
P_CTRL	EQU	0F30H
ON_OFF	EQU	0F36H
NOOR1	EQU	0F3CH
DTMF_PTT	EQU	0FE0H
Dis_PTT	EQU	0FDOH
ORG		0F00H
DB		44H, 5EH, 7DH, 67H, 0EBH, 80H ; READY..
DB		45H, 0C7H, 4BH, 0EH, 7DH, 07H ; No. SLAV
DB		00H, 00H, 3FH, 40H, 7BH, 00H ; 0-9
DB		5EH, 44H, 44H, 47H, 0C4H, 80H ; ERROR..
DB		45H, 0C7H, 67H, 5EH, 07H, 0CH ; No. DEVI
DB		00H, 00H, 21H, 40H, 7BH, 00H ; 1-9
DB		67H, 0CH, 0D5H, 1EH, 4EH, 0EH ; DIM. CTL
DB		00H, 3FH, 3FH, 40H, 21H, 5BH ; 00-15
DB		7CH, 40H, 1EH, 4EH, 44H, 0EH ; P_CTRL
DB		47H, 5CH, 5CH, 40H, 47H, 45H ; OFF-ON
DB		00H, 3FH, 47H, 44H, 21H, 00H ; Oor1

```
ORG OFE0H ; DTMF PATTERN
DB 77H, OEEH, ODEH, OBEH, OEDH, ODDH
DB OBDH, OEBH, ODBH, OBBH, OD7H, OE7H
DB OB7H, 7EH, 7DH, 7BH, OD7H, OE7H
DB OB7H, 7EH, 7DH, 7BH
```

```
ORG OFD0H ; DISPLAY PATTERN
DB 3FH, 21H, 76H, 73H, 69H, 5BH
DB 5FH, 31H, 7FH, 7BH
```

```
ORG 0000H
SJMP START01
```

```
ORG 0040H
START01: MOV SP, #2FH ; SET NEW STACK
CLR A
ORL A, #0FFH
MOV DPTR, #9000H
MOVX @DPTR, A ; CLEAR 8870 FF
MOV DPTR, #0E000H
MOVX @DPTR, A ; CLEAR KEYBOARD FF
START: MOV DPTR, #READY
LCALL DISPLAY ; DISPLAY READY
START1: LCALL DISP1
JB P1.1, BIT1
SJMP START1 ; KEY PRESS
```

```
DISPLAY:  MOV    R4,#06H                ; SET TOTAL BUFFER
          MOV    RO,#10H                ; SET DISP BUFFER

LOAD_DATA: CLR    A
          MOVC   A,@A+DPTR
          MOV    @RO,A                  ; LOAD DISP BUFFER
          INC    DPTR                   ; INC ADDRESS LOAD
          INC    RO                     ; INC ADD. BUFFER
          DJNZ   R4,LOAD_DATA

DISP1:    MOV    R4,#06H                ; SET TOTAL DISPLAY
          MOV    RO,#10H
          MOV    B,#10111111B

DISPO1:   MOV    DPTR,#0B000H
          MOV    A,@RO
          MOVX   @DPTR,A                ; OUT DATA DISPLAY
          MOV    DPTR,#0C000H
          MOV    A,B
          RR     A                       ; ROTATE DIGIT
          MOVX   @DPTR,A                ; SELECT DIGIT
          MOV    B,A
          MOV    R6,#235

DLP1:     DJNZ   R6,DLP1
          ORL    A,#0FFH
          MOVX   @DPTR,A                ; CLEAR DIGIT
          INC    RO
          DJNZ   R4,DISPO1
          RET
```

```
BIT1:    LCALL BEEP

MOV      DPTR,#0E000H          ; READ DATA KEY AND
MOVX     A,@DPTR              ; CLEAR KEYBOARD FF
ANL      A,#0FH
CJNE     A,#0EH,START        ; CHECK IS START ?
MOV      DPTR,#8000H
MOV      A,#7DH              ; SEND START "E"
MOVX     @DPTR,A
MOV      R6,#80
H1:      MOV      R7,#255      ; ##### D
H2:      NOP                  ; ##### E
          DJNZ     R7,H2       ; ##### L
          DJNZ     R6,H1       ; ##### AY 50ms
MOV      A,#0FFH             ; RESET COMMAND
MOVX     @DPTR,A
SLAVE1:  MOV      DPTR,#NO_SLAV
          LCALL   DISPLAY
MOV      R1,#03
N_SLAVE1: MOV     R2,#25
N_SLAVE:  LCALL   DISP1        ; DISPLAY No.SLAVE
          DJNZ   R2,N_SLAVE
          DJNZ   R1,N_SLAVE1
MOV      DPTR,#0E000H        ; PROTECT REPEAT KEY
MOVX     @DPTR,A
MOV      DPTR,#NO_9
          LCALL   DISPLAY        ; DISPLAY 0-9
MOV      R1,#15
```

```
N_091: MOV R2,#25
N_09: LCALL DISP1
      JB P1.1,LOOP2 ; CHECK KEY PRESS
      DJNZ R2,N_09 ; GOTO LOOP2
      DJNZ R1,N_091
      AJMP DISERROR

LOOP2: LCALL BEEP
      MOV DPTR,#0E000H ; IN&CLEAR KEYBOARD
      MOVX A,@DPTR
      ANL A,#0FH
      MOV RO,A
      CLR C
      CJNE A,#0FH,S1
      LJMP SLAVE1
S1: CJNE A,#0EH,S2
      LJMP SLAVE1
S2: CJNE A,#0DH,S3
      LJMP SLAVE1
S3: CJNE A,#0CH,S4
      LJMP SLAVE1
S4: CJNE A,#0BH,S5
      LJMP SLAVE1
S5: CJNE A,#0AH,SLAVE ; NOT 0-9 GOTO
      LJMP SLAVE1 ; KEY AGAIN
SLAVE: MOV DPTR,#DTMF_PTT ; 0 - 9 ONLY
      MOV A,RO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV C  A,@A+DPTR
MOV   DPTR,#8000H
MOVX  @DPTR,A           ; SEND DTMF DATA
MOV   R6,#80
G1:   MOV   R7,#255     ;##### D
G2:   NOP              ;##### E
      DJNZ  R7,G2      ;##### L
      DJNZ  R6,G1      ;##### A Y 50 mS
      MOV   A,#0FFH
      MOVX  @DPTR,A    ; RESET COMMAND
      MOV   R1,#10
DEC1: NOP
      JB    P1.0,READ1
      NOP              ; CHECK 8870 FF
      MOV   R5,#80     ; DELAY 50mS*10
LOOPEE: MOV  R6,#255
LPEE:  NOP
      DJNZ  R6,LPEE
      DJNZ  R5,LOOPEE  ; DELAY 50mS *10
      DJNZ  R1,DEC1
DISERROR: MOV  DPTR,#ERROR ; IF NO ACK
          LCALL DISPLAY ; DISPLAY ERROR
          MOV   R1,#08
DISERR:  MOV   R2,#25
DISERR1: LCALL DISPI
          DJNZ  R2,DISERR1
```

```
MOV DPTR,#0E000H
MOVX A,@DPTR
AJMP START
```

```
READI: MOV DPTR,#0D000H ; INPUT DATA
MOVX A,@DPTR ; FROM 8870
ANL A,#0FH
MOV R2,A
ORL A,#0FFH
MOV DPTR,#9000H
MOVX @DPTR,A ; CLEAR 8870 FF
MOV A,R2
CJNE A,#0AH,DEC1 ; CHECK DATA=ACK ?
DEVI: MOV DPTR,#NO_DEVI
LCALL DISPLAY
MOV R1,#05
DSP_DEV: MOV R2,#25
DISP_DEV: LCALL DISP1
DJNZ R2,DISP_DEV
DJNZ R1,DSP_DEV
MOV DPTR,#0E000H ; PROTECT REPEAT KEY
MOVX @DPTR,A
MOV DPTR,#N1_9 ; DISPLAY 1-9
LCALL DISPLAY
MOV R1,#15
DN1_91: MOV R2,#25
DN1_9: LCALL DISP1
```

```
CLR    C
JB     P1.1,LOOP3      ; CHECK KEY PRESS ?
DJNZ   R2,DN1_9
DJNZ   R1,DN1_91
AJMP   DISERROR
```

```
LOOP3:  LCALL BEEP
        MOV  DPTR,#0E000H
        MOVX A,@DPTR      ; IN&CLEAR KEYBOARD
        ANL  A,#0FH       ; SELECT LOW BYTE
        MOV  RO,A
        MOV  Z3H,A        ; PUT IN CH_BUFFER
        CJNE A,#00H,LO
        AJMP DEV1
LO:     CJNE A,#0FH,L1
        AJMP DEV1
L1:     CJNE A,#0EH,L2
        AJMP DEV1
L2:     CJNE A,#0DH,L3
        AJMP DEV1
L3:     CJNE A,#0CH,L4
        AJMP DEV1
L4:     CJNE A,#0BH,L5
        AJMP DEV1
L5:     CJNE A,#0AH,DEVC
        AJMP DEV1
```

```
DEVC:      NOP                                ; ONLY 1-9 TO DEVC
           MOV  A,23H                          ; 23:BUFFER ADDRESS
           MOV  DPTR,#DTMF_PTT
           MOVC A,@A+DPTR
           MOV  DPTR,#8000H
           MOVX @DPTR,A                        ; SEND NO. CH.
           MOV  R6,#80
F1:        MOV  R7,#255                        ; ##### D
F2:        NOP                                ; ##### E
           DJNZ R7,F2                          ; ##### L
           DJNZ R6,F1                          ; ##### AY 50mS
           MOV  A,#OFFH
           MOVX @DPTR,A                        ; RESET COMMAND
           MOV  DPTR,#9000H                    ; CLS 8870
           MOVX @DPTR,A                        ; BEFORE ACK
           MOV  R1,#10
DEC2:      NOP
           JB   P1.0,READ2                      ; CHECK 8870 FF SET?
           NOP
           MOV  R5,#80                          ; DELAY 50mS *10
LOOPDD:    MOV  R6,#255                        ; WAIT FOR ACK 0.5S
LPDD:      NOP
           DJNZ R6,LPDD
           DJNZ R5,LOOPDD                      ; DELAY 50mS*10
           DJNZ R1,DEC2
           AJMP DISERROR                       ; IF NOT SET GOTO DISPLAY_ERROR
```

```
READ2:  MOV    DPTR,#0D000H          ; READ DATA 8870
        MOVX   A,@DPTR
        ANL   A,#0FH
        MOV   R2,A
        ORL   A,#0FFH
        MOV   DPTR,#9000H
        MOVX  @DPTR,A                ; CLEAR 8870 FF
        MOV   A,R2
        CJNE  A,#0AH,DEC2             ; DATA <> ACK GOTO
        CLR   C                       ; READ AGAIN
        MOV   A,23H                   ; MOVE DATA FROM BUFFE
        SUBB  A,#01H
        JZ    DIM2                    ; IF CH 01 GOTO DIM2
        CLR   C
        MOV   A,23H
        SUBB  A,#02H
        JZ    DIM2                    ; IF CH 02 GOTO DIM2
        AJMP  ONLOOP1                 ; 03 - 09 GOTO ONLOOP1

DIM2:   MOV   DPTR,#DIM_CTL
        LCALL DISPLAY
        MOV   R1,#02

DIS_DIM: MOV   R2,#50

DIS_DIMC: LCALL DISP1
        DJNZ  R2,DIS_DIMC
        DJNZ  R1,DIS_DIM
        MOV   DPTR,#0E000H
```

```
MOVX @DPTR,A ; PROTECT REPEAT KEY
MOV DPTR,#NOO_15
LCALL DISPLAY
MOV R1,#25
DIS_151: MOV R2,#25
DIS_15: LCALL DISP1
JB P1.1,LOOP5 ;KEY PRESS GOTO LOOP5
DJNZ R2,DIS_15
DJNZ R1,DIS_151
AJMP DISERROR
JDIM1: AJMP DIM1 ; TRANSIT JUMP TO DIM1
LOOP5: LCALL BEEP
MOV DPTR,#0E000H
MOVX A,@DPTR ; IN&CLEAR KEYBOARD
ANL A,#0FH ; SELECT 4 BIT LOW
MOV R3,A
CJNE A,#0CH,JDIM1 ; NOT CHECK GOTO DIM1
MOV A,#0B7H ; ASK STATUS
MOV DPTR,#8000H
MOVX @DPTR,A ; SEND CHECK KEY
MOV R6,#80
E1: MOV R7,#255 ; ##### D
E2: NOP ; ##### E
DJNZ R7,E2 ; ##### L
DJNZ R6,E1 ; ##### AY 50mS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    A,#OFFH
MOVX   @DPTR,A                ; RESET COMMAND
MOV    DPTR,#9000H
MOVX   @DPTR,A                ; CLS8870 BEFORE ACK
MOV    R2,#10
DECS:  NOP
        JB    P1.0,READ3      ; 8870 FF SET ?
        NOP
        MOV   R5,#80          ; DELAY 50mS *10
LOOPCC: MOV   R6,#255
LPCC:  DJNZ  R6,LPCC
        DJNZ  R5,LOOPCC      ; DELAY 50mS *10
        DJNZ  R2,DECS
        AJMP  DISERROR      ; NOT SET GOTO
        ; DISPLAY ERROR
READ3: MOV   DPTR,#0D000H
        MOVX  A,@DPTR        ; INPUT 8870 DATA
        ANL  A,#0FH
        MOV  R2,A
        ORL  A,#OFFH
        MOV  DPTR,#9000H
        MOVX @DPTR,A        ; CLEAR 8870 FF
        MOV  A,R2
        MOV  B,#10
        DIV  AB
        MOV  DPTR,#DIS_PTT
```

```
MOV 12H,A ; PUT UPBYTE IN DISPLAY BUFFER .
XCH A,B
MOV DPTR,#DIS_PTT
MOVC A,@A+DPTR
MOV 13H,A ; PUT LOW_BYTE IN BUFFER
MOV 10H,#00H
MOV 11H,#00H
MOV 14H,#00H
MOV 15H,#00H
MOV R1,#08
LPDISP1: MOV R2,#25
LPDISP: LCALL DISP1
DJNZ R2,LPDISP
DJNZ R1,LPDISP1
AJMP START
DIM1: LCALL BEEP
MOV DPTR,#OE000H
MOVX @DPTR,A ;PROTECT REPEAT KEY
MOV DPTR,#NOO_15
LCALL DISPLAY .
MOV R2,#10
LOPDIM01: MOV R1,#25
LOPDIM1: JB P1.1,LPDIM1 ; CHECK KEY PRESS(2)
LCALL DISP1
DJNZ R1,LOPDIM1
DJNZ R2,LOPDIM01
AJMP DISERROR
```

```
LPDIM1:    LCALL BEEP
           MOV    DPTR,#0E000H
           MOVX   A,@DPTR                ; IN&CLEAR KEYBOARD
           ANL   A,#0FH                  ; SELECT 4 BIT LOW
           MOV   R1,A
           MOV   A,R3
           SWAP  A
           ADD  A,R1
           CLR   C
           SUBB  A,#16H
           JC    UNDER3                  ; DIM LEVEL>15 GOTO
           AJMP  DIM2                      ; READ KEY AGAIN
UNDER3:    MOV   A,R3                      ; R3 IS 0 OR 1 ONLY
           SWAP  A
           ADD  A,R1
           MOV   DPTR,#DTMF_PTT
           MOVC  A,@A+DPTR
           MOV   DPTR,#8000H
           MOVX  @DPTR,A                  ; SEND DIM.LEVEL
           MOV   R6,#80
D1:        MOV   R7,#255                  ; ##### D
D2:        NOP                            ; ##### E
           DJNZ  R7,D2                    ; ##### L
           DJNZ  R6,D1                    ; ##### AY 50mS
           MOV   A,#0FFH
           MOVX  @DPTR,A                  ; RESET COMMAND
```

```
MOV R2,#10
DEC5: NOP
      JB P1.0,READ5 ; CHECK 8870 FF SET?
MOV R5,#80
LOOPHH: MOV R6,#255
LPHH:  NOP
      DJNZ R6,LPHH ; DELAY 50ms * 10
      DJNZ R5,LOOPHH
      DJNZ R2,DEC5 ; WAIT FOR ACK 0.5 S
      AJMP DISERROR ; IF NOT SET GOTO DISPLAY_ERR
READ5: MOV DPTR,#0D000H
      MOVX A,@DPTR ; INPUT 8870 DATA
      ANL A,#0FH ; SELECT 4BIT LOW
      MOV R3,A ; KEEP A IN R3
      ORL A,#0FFH
      MOV DPTR,#9000H
      MOVX @DPTR,A ; CLEAR 8870 FF
      MOV A,R3
      CJNE A,#0AH,DEC5 ; DATA <> ACK GOTO
      AJMP START ; READ AGAIN
ONLOOP1: MOV DPTR,#P_CTRL ; DISPLAY P_CONTROL
        LCALL DISPLAY
        MOV R2,#02
DP_PCTL1: MOV R1,#25
DP_PCTL: LCALL DISP1
```

```
DJNZ R1,DP_PCTL ; SET DISPLAY TIME
DJNZ R2,DP_PCTL1
MOV DPTR,#ON_OFF ; DISPLAY ON_OFF
LCALL DISPLAY
MOV R2,#02
DP_ONF1: MOV R1,#25
DP_ONF: LCALL DISP1
DJNZ R1,DP_ONF ; SET DISPLAY TIME
DJNZ R2,DP_ONF1
MOV DPTR,#OE000H
MOVX @DPTR,A ; PROTECT REPEAT KEY
MOV DPTR,#NOOR1 ; DISPLAY 0 OR 1
LCALL DISPLAY
MOV R2,#15
DP_OOR11: MOV R1,#25
DP_OOR1: LCALL DISP1
JB P1.1,LOOP12 ; JUMP TO LOOP12 IF KEY PRESS
DJNZ R1,DP_OOR1
DJNZ R2,DP_OOR11
AJMP DISERROR ; NO KEY PRESS DISPLAY ERROR

LOOP12: LCALL BEEP
MOV DPTR,#OE000H
MOVX A,@DPTR ; IN&CLEAR KEYBOARD
ANL A,#0FH ; SELECT 4 BIT LOW
MOV RO,A
CJNE A,#0CH,ONLOOP2 ; NOT CHECK KEY GOTO ONLOOP2
```

```
MOV    A,#0B7H
MOV    DPTR,#8000H
MOVX   @DPTR,A                ; SEND CHECK KEY
MOV    R6,#80
B1:    MOV    R7,#255          ; ##### D
B2:    NOP                    ; ##### E
        DJNZ  R7,B2          ; ##### L
        DJNZ  R6,B1          ; ##### AY 50mS
MOV    A,#0FFH
MOVX   @DPTR,A                ; RESET COMMAND
MOV    DPTR,#9000H
MOVX   @DPTR,A                ; CLEAR 8870 FF
MOV    R1,#10
DEC6:  NOP
        JB   P1.0,READ6      ; 8870 FF SET ?
        NOP
MOV    R5,#80                  ; DELAY 50 mS
LOOP11: MOV    R6,#255
LP11:  NOP                    ; WAIT FOR ACK 0.5S
        DJNZ  R6,LP11
        DJNZ  R5,LOOP11      ; DELAY 50mS
        DJNZ  R1,DEC6
MISS:  AJMP  DISERROR        ; IF NO ACK GOTO
        ; DISPLAY ERROR
READ6: MOV    DPTR,#0D000H
        MOVX  A,@DPTR        ; INPUT 8870 DATA
        ANL  A,#0FH          ; SELECT 4 BIT LOW
```

```
MOV R2,A
ORL A,#OFFH
MOV DPTR,#9000H
MOVX @DPTR,A ; CLEAR 8870 FF
MOV A,R2
CLR C
SUBB A,#00H
JNZ NOTZERO ; IF=0,LOAD PTTN OFF
MOV 10H,#00H
MOV 11H,#00H
MOV 15H,#00H
MOV 12H,#47H ; PATTERN "0"
MOV 13H,#5CH ; PATTERN "F"
MOV 14H,#5CH ; PATTERN "F"
MOV R1,#08
DISP_OFF1: MOV R5,#25
DISP_OFF: LCALL DISP1
DJNZ R5,DISP_OFF
DJNZ R1,DISP_OFF1
AJMP START
JONLOOP2: AJMP ONLOOP2 ; TRANSIT JUMP ONLOOP2
NOTZERO: MOV A,R2
CLR C
SUBB A,#01H
JNZ MISS ; IF=1,LOAD PTTN ON
MOV 10H,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    11H,#00H
MOV    15H,#00H
MOV    12H,#47H           ; PATTERN "O"
MOV    13H,#45H           ; PATTERN "N"
MOV    14H,#00H
MOV    R1,#08
DISP_ON1: MOV    R5,#25
DISP_ON:  LCALL DISP1
         DJNZ   R5,DISP_ON
         DJNZ   R1,DISP_ON1
         AJMP  START
ONLOOP2: MOV    A,RO
         CLR   C
         SUBB  A,#00H
         JNZ   NOTZERO2
         MOV   A,#77H           ; LOAD PATTERN
         AJMP  SEND1           ; SEND "O"
NOTZERO2: MOV   A,RO
         CLR   C
         SUBB  A,#01H
         JNZ   CONLOOP1       ; IF NOT OOR1 GOTO OOR1 AGAIN
         MOV   A,#0EEH        ; LOAD PATTERN SEND "1"
SEND1:  MOV   DPTR,#8000H
         MOVX  @DPTR,A         ; SEND ON OR OFF
         MOV   R6,#80
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
A1:      MOV    R7,#255                ; ##### D
A2:      NOP                          ; ##### E
          DJNZ  R7,A2                 ; ##### L
          DJNZ  R6,A1                 ; ##### AY 50mS
          MOV   A,#OFFH
          MOVX  @DPTR,A                ; RESET COMMAND
          MOV   DPTR,#9000H
          MOVX  @DPTR,A                ; CLS 8870 FF BEFORE ACK
          MOV   R2,#10
DEC9:    NOP
          JB    P1.0,FINAL             ; 8870 FF SET ?
          NOP
          MOV   R5,#80                 ; DELAY 50mS
FINII:   MOV   R6,#255
FINI:    NOP                          ; WAIT FOR ACK FINAL 0.5 S
          DJNZ  R6,FINI
          DJNZ  R5,FINII               ; DELAY 50mS
          DJNZ  R2,DEC9
JDISERROR: AJMP DISERROR              ; GOTO DISPLAY ERROR

FINAL:   LCALL BEEP
          MOV   DPTR,#0D000H
          MOVX  A,@DPTR                ; INPUT 8870 DATA
          ANL  A,#0FH
          MOV   DPTR,#9000H
          MOV   R1,A                   ; CLEAR 8870 FF
          MOVX  @DPTR,A
```

```
MOV    A,R1
CJNE  A,#0AH,DEC9    CHECK <> ACK GOTO IN AGAIN
LJMP  START
```

```
CONLOOP1:  AJMP  ONLOOP1    ; TRANSIT JUMP DISPLAY OOR1 AGAIN
```

```
BEEP:     MOV    R5,#10
BP1:     MOV    R7,#0FFH    ; BEEP 250 MICROSEC.
BEEPLoop: MOV    DPTR,#0A000H ; ADDRESS BEEP PORT
          MOV    R6,#05    ; SET FREQ.1 KHz
BEEP1:   MOVX   @DPTR,A    ; BEEP ON
          DJNZ  R6,BEEP1
          MOV    R6,#05
BEEPO:   NOP     ; BEEP OFF
          NOP
          DJNZ  R6,BEEPO
          DJNZ  R7,BEEPLoop
          DJNZ  R5,BP1
          RET
END
```

ซ. โปรแกรมควบคุมของส่วน SLAVE

```
ON_OFF.0    BIT    00H
ON_OFF.1    BIT    01H
ON_OFF.2    BIT    02H
ON_OFF.3    BIT    03H
ON_OFF.4    BIT    04H
ON_OFF.5    BIT    05H
ON_OFF.6    BIT    06H
ON_OFF      EQU    20H
DIM1        EQU    21H
DIM2        EQU    22H
DTMF_PTT    EQU    OFE0H

ORG         OFE0H
DB          77H,0EEH,ODEH,OBEH,OEDH,ODDH
DB          OBDH,OEBH,ODBH,OB BH,OD7H,OE7H
DB          OB7H,7EH,7DH,7BH

ORG         0000H
SJMP       START

ORG         0040H
START:     MOV    SP,#2FH                ; NEW STACK POINTER
START01:   ORL    A,#0FFH
           MOV    DPTR,#9000H
           MOVX  @DPTR,A                ; CLEAR 8870 FF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    A,20H                ; LOAD DATA ON-OFF
MOV    DPTR,#0E000H
MOVX   @DPTR,A              ; CONTROL ON-OFF
MOV    A,21H                ; LOAD DATA DIM1
MOV    DPTR,#0A000H
MOVX   @DPTR,A              ; CONTROL DIM1
MOV    A,22H                ; LOAD DATA DIM2
MOV    DPTR,#0B000H
MOVX   @DPTR,A              ; CONTROL DIM2
START1: NOP
JNB    P1.4,START1          ; 8870 FF SET ?
MOV    R6,#80
L100:  MOV    R7,#255        ; | DELAY BEFOR READ
LP100: NOP                  ; | & CLEAR 8870
DJNZ   R7,LP100             ; | 50mS
DJNZ   R6,L100              ; | -----
MOV    DPTR,#0C000H
MOVX   A,@DPTR              ; INPUT DATA
ANL    A,#0FH                ; 4 BIT LOW
MOV    R2,A
ORL    A,#0FFH
CLR1:  MOV    DPTR,#9000H
MOVX   @DPTR,A              ; CLEAR 8870 FF
MOV    A,R2
CJNE   A,#0EH,START1        ; IF DATA<>START GOTO START1
MOV    R5,#15
SLP101: MOV    R3,#255        ; #
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SLP1:    MOV    R4,#255                ; #
SLP1:    JB     P1.4,START2           ; 8870 FF SET ?
        DJNZ   R4,SLP11              ; WAIT FOR DEVICE KEY
        DJNZ   R3,SLP1
        DJNZ   R5,SLP101
        MOVX   @DPTR,A
        AJMP   START01               ; RUN TIME OUT GOTO WAIT KEY START
START2:   MOV    R6,#80                ; |-----
L300:    MOV    R7,#255                ; | DELAY BEFOR READ
LP300:    NOP                          ; | & CLEAR 8870
        DJNZ   R7,LP300              ; | 50mS KEY NUM_SLAVE
        DJNZ   R6,L300                ; |-----
        MOV    DPTR,#0C000H
        MOVX   A,@DPTR                ; INPUT 8870 DATA
        ANL    A,#0FH
        MOV    R1,A
        ORL    A,#0FFH
        MOV    DPTR,#9000H
        MOVX   @DPTR,A                ; CLEAR 8870 FF
        MOV    A,P1
        ANL    A,#0FH
        MOV    1FH,R1
        CJNE   A,1FH,START1           ; CHECK OWN NO.SLAVE
        MOV    A,#0D7H
        MOV    DPTR,#8000H
        MOVX   @DPTR,A                ; IF YES SEND ACK
        MOV    R6,#80
```

```
J1:      MOV    R7,#255                ; ##### D
J2:      NOP                          ; ##### E
          DJNZ  R7,J2                  ; ##### L
          DJNZ  R6,J1                  ; ##### AY 50mS
          MOV   A,#OFFH
          MOV   DPTR,#8000H
          MOVX  @DPTR,A                ; RESET COMMAND
          MOV   R5,#15                 ; #####
BLP12:   MOV   R3,#255                 ; #####
BLP1:    MOV   R4,#255                 ; #####
BLP11:   JB    P1.4,START3            ; 8870 FF SET ?
          DJNZ  R4,BLP11               ; WAIT 5s
          DJNZ  R3,BLP1                ; #####
          DJNZ  R5,BLP12               ; #####
          AJMP  START01                ; TIME OUT TO START1

START3:  MOV   R6,#80                  ; |-----|
L400:    MOV   R7,#255                 ; | DELAY BEFOR READ
LP400:   NOP                          ; | & CLEAR 8870
          DJNZ  R7,LP400                ; | 50mS KEY No.DEVI
          DJNZ  R6,L400                 ; |-----|
          MOV   DPTR,#0C000H
          MOVX  A,@DPTR                 ; INPUT 8870 DATA
          ANL  A,#OFH
          MOV   R1,A
          MOV   23H,A
          ORL  A,#OFFH
```

```
MOV DPTR,#9000H
MOVX @DPTR,A ; CLEAR 8870 FF
MOV DPTR,#8000H
MOV A,#0D7H
MOVX @DPTR,A
MOV R6,#80
Z1: MOV R7,#255 ;##### D
Z2: NOP ;##### E
DJNZ R7,Z2 ;##### L
DJNZ R6,Z1 ;##### AY 50mS
MOV A,#OFFH
MOVX @DPTR,A ; RESET COMMAND
MOV A,23H
CLR C
SUBB A,#02H
JZ JDIMLOOP ; DIM CH. TO DIMLOOP
MOV A,23H
CLR C
SUBB A,#01H
JZ JDIMLOOP ; DIM CH. TO DIMLOOP
MOV R5,#15 ; #
SLP701: MOV R3,#255 ; #
SLP7: MOV R4,#255 ; #
SLP77: JB P1.4,START5 ; 8870 FF SET ?
DJNZ R4,SLP77 ; WAIT FOR DEVICE KEY
DJNZ R3,SLP7 ; KEY 3-9 OR STATUS
DJNZ R5,SLP701
```

```

                AJMP  START01                ; TIME OUT TO START01
START5:         MOV   R6,#80                ; |-----
L500:          MOV   R7,#255                ; | DELAY BEFOR READ
LP500:         NOP                          ; | & CLEAR 8870
                DJNZ  R7,LP500              ; | 50mS KEY ON-OFF.
                DJNZ  R6,L500              ; |-----
                MOV   DPTR,#0C000H
                MOVX  A,@DPTR                ; INPUT DATA
                ANL   A,#0FH
                MOV   RO,A
                ORL   A,#0FFH
                MOV   DPTR,#9000H
                MOVX  @DPTR,A                ; CLEAR 8870 FF
                MOV   A,RO
                CJNE  A,#0CH,CTRL3          ; DATA<>CHECK TO CTRL3
                MOV   A,23H
                CLR   C
                SUBB  A,#03H                ; COMPARE CH.3
                JNZ   STATE4
                MOV   C,00H                ; LOAD STATUS 3
                AJMP  CSTATE
JDIMLOOP:      AJMP  DIMLOOP
STATE4:        MOV   A,23H                ; LOAD CH.
                CLR   C
                SUBB  A,#04H                ; COMPARE CH.4
                JNZ   STATES
```

```
MOV C,01H ; LOAD STATUS 4
AJMP CSTATE ; GOTO SEND STATUS
STATE5: MOV A,23H ; LOAD CH.
CLR C
SUBB A,#05H ; COMPARE CH.5
JNZ STATE6
MOV C,02H ; LOAD STATUS 5
AJMP CSTATE ; GOTO SEND STATUS
STATE6: MOV A,23H ; LOAD CH.
CLR C
SUBB A,#06H ; COMPARE CH.6
JNZ STATE7
MOV C,03H ; LOAD STATUS 6
AJMP CSTATE ; GOTO SEND STATUS
STATE7: MOV A,23H ; LOAD CH.
CLR C
SUBB A,#07H ; COMPARE CH.7
JNZ STATE8
MOV C,04H ; LOAD STATUS 7
AJMP CSTATE ; GOTO SEND STATUS
STATE8: MOV A,23H ; LOAD CH.
CLR C
SUBB A,#08H ; COMPARE CH.8
JNZ STATE9
MOV C,05H ; LOAD STATUS 8
AJMP CSTATE ; GOTO SEND STATUS
STATE9: MOV A,23H ; LOAD CH.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

```
CLR    C
SUBB   A,#09H           ; COMPARE CH.9
JNZ    JERROR
MOV    C,06H           ; LOAD STATUS 9
AJMP   CSTATE         ; GOTO SEND STATUS
JERROR: AJMP  START01
CSTATE: JC     SEND_ON
        AJMP  SEND_OFF
SEND_ON: MOV   A,#0EEH       ; LOAD DATA "1"
STATE:  NOP
        MOV   DPTR,#8000H
        MOVX  @DPTR,A        ; SEND STATUS
        MOV   R7,#80         ; |-----
LOOPCC: MOV   R6,#255        ; |
LPCC:   NOP                ; | TONE DELAY
        DJNZ  R6,LPCC        ; | 50 mS
        DJNZ  R7,LOOPCC      ; |-----
        MOV   A,#0FFH
        MOVX  @DPTR,A        ; RESET COMMAND
        AJMP  START01
SEND_OFF: MOV  A,#77H        ; LOAD DATA "0"
        AJMP  STATE         ; SEND STATUS
CTRL3:  MOV   DPTR,#8000H    ; SEND ACK
        MOV   A,#0D7H
        MOVX  @DPTR,A
        MOV   R6,#80
W1:     MOV   R7,#255        ;##### D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
W2:      NOP                      ; ##### E
          DJNZ R7,W2              ; ##### L
          DJNZ R6,W1              ; ##### AY 50mS
          MOV A,#OFFH
          MOVX @DPTR,A            ; RESET COMMAND
          MOV A,23H                ; LOAD DATA CH.
          CJNE A,#03H,CTRL4       ; CHECK No. CH.3
          MOV A,RO
          JNZ SET3                 ; CHECK ON OR OFF
          CLR ON_OFF.0            ; OFF CH.3
          AJMP START01
SET3:    SETB ON_OFF.0            ; ON CH.3
          AJMP START01
CTRL4:   MOV A,23H                ; LOAD DATA CH.
          CJNE A,#04H,CTRL5       ; CHECK No. CH.4
          MOV A,RO
          JNZ SET4                 ; CHECK ON OR OFF
          CLR ON_OFF.1            ; OFF CH.4
          AJMP START01
SET4:    SETB ON_OFF.1            ; ON CH.4
          AJMP START01
CTRL5:   MOV A,23H                ; LOAD DATA CH.
          CJNE A,#05H,CTRL6       ; CHECK No. CH.5
          MOV A,RO
          JNZ SET5                 ; CHECK ON OR OFF
          CLR ON_OFF.2            ; OFF CH.5
          AJMP START01
```

```
SET5:    SETB  ON_OFF.2                ; ON CH.5
         AJMP  START01

CTRL6:   MOV   A,23H                  ; LOAD DATA CH.
         CJNE  A,#06H,CTRL7          ; CHECK No. CH.6
         MOV   A,RO
         JNZ   SET6                  ; CHECK ON OR OFF
         CLR   ON_OFF.3              ; OFF CH.6
         AJMP  START01

SET6:    SETB  ON_OFF.3                ; ON CH.6
         AJMP  START01

CTRL7:   MOV   A,23H                  ; LOAD DATA CH.
         CJNE  A,#07H,CTRL8          ; CHECK No. CH.7
         MOV   A,RO
         JNZ   SET7                  ; CHECK ON OR OFF
         CLR   ON_OFF.4              ; OFF CH.7
         AJMP  START01

SET7:    SETB  ON_OFF.4                ; ON CH.7
         AJMP  START01

CTRL8:   MOV   A,23H                  ; LOAD DATA CH.
         CJNE  A,#08H,CTRL9          ; CHECK No. CH.8
         MOV   A,RO
         JNZ   SET8                  ; CHECK ON OR OFF
         CLR   ON_OFF.5              ; OFF CH.8
         AJMP  START01

SET8:    SETB  ON_OFF.5                ; ON CH.8
         AJMP  START01

CTRL9:   MOV   A,23H                  ; LOAD DATA CH.
```

```
CJNE  A,#09H,ERROR1      ; CHECK No. CH.9
MOV   A,R0
JNZ   SET9                ; CHECK ON OR OFF
CLR   ON_OFF.6            ; OFF CH.9
ERROR1:  AJMP  START01
SET9:   SETB  ON_OFF.6     ; ON CH.9
        AJMP  START01
DIMLOOP: MOV   R5,#15      ; #
SLP801: MOV   R3,#255      ; #
SLP8:   MOV   R4,#255      ; #
SLP88:  JB    P1.4,DLPO11  ; 8870 FF SET ?
        DJNZ  R4,SLP88     ; WAIT DEVICE KEY 5s
        DJNZ  R3,SLP8
        DJNZ  R5,SLP801
        AJMP  START01     ; TIME OUT TO START01
DLPO11: MOV   R6,#80      ; |-----
L600:   MOV   R7,#255     ; | DELAY BEFOR READ
LP600:  NOP                    ; | & CLEAR 8870
        DJNZ  R7,LP600    ; | 50ms KEY DIM
        DJNZ  R6,L600     ; |-----
        MOV   DPTR,#0C000H
        MOVX  A,@DPTR     ; INPUT 8870 DATA
        ANL  A,#0FH
        MOV  RO,A
        ORL  A,#0FFH
        MOV  DPTR,#9000H
        MOVX @DPTR,A     ; CLEAR 8870 FF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    A,RO
CJNE  A,#0CH,DIM01      ; IF<>CHECK TO DIM01
MOV    A,23H
CLR    C
SUBB   A,#01H           ; CHECK CH. = 01 ?
JNZ    CDIM2
MOV    A,21H
CPL    A
ANL    A,#0FH
AJMP   STATUS01
CDIM2: MOV    A,23H
CJNE  A,#02H,ERROR     ; CHECK CH. = 02 ?
MOV    A,22H
CPL    A
ANL    A,#0FH
STATUS01: MOV    DPTR,#DTMF_PTT
        MOVC   A,@A+DPTR
        MOV    DPTR,#8000H
        MOVX   @DPTR,A      ; SEND DIM LEVEL
        MOV    R6,#80      ; |-----
K600:  MOV    R7,#255      ; | DELAY SEND TONE
KP600: NOP                ; |
        DJNZ   R7,KP600    ; | 50ms
        DJNZ   R6,K600    ; |-----
        MOV    A,#0FFH
        MOVX   @DPTR,A      ; RESET COMMAND
ERROR: AJMP   START01
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DIM01:    MOV    DPTR,#8000H
          MOV    A,#0D7H
          MOV    R3,#80
LPACK11:  MOV    R4,#255
LPACK1:   MOVX   @DPTR,A
          DJNZ   R4,LPACK1
          DJNZ   R3,LPACK11
          MOV    DPTR,#8000H
          MOV    A,#0FFH
          MOVX   @DPTR,A
ELPO11:   MOV    A,23H
          CJNE   A,#01H,DIM_CH2 ; IF CH.1
          MOV    A,RO
          CPL    A
          ANL    A,#0FH
          MOV    RO,A
          MOV    21H,RO ; LOAD DIM. LEVEL
          AJMP   START01
DIM_CH2:  MOV    A,29H
          CJNE   A,#02H,ERROR ; IF CH.2
          MOV    A,RO
          CPL    A
          ANL    A,#0FH
          MOV    RO,A
          MOV    22H,RO ; LOAD DIM. LEVEL
          AJMP   START01
```

END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กิตติกรรมประกาศ

ระบบควบคุมอุปกรณ์ผ่านสายไฟฟ้านี้แล้วเสร็จลงได้ก็ด้วยความร่วมมือ และสนับสนุนจากหลาย ๆบุคคลไม่ว่าคณาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ทุกท่าน อาจารย์วิทยา ทิพย์สุวรรณพร อาจารย์ที่ปรึกษาที่ให้คำปรึกษา แนะนำทางการแก้ปัญหาที่เกิดขึ้นกับโครงการนี้ เพื่อน ๆ ทุกคนที่ให้ยืมสถานที่ทำโครงการและ เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาโครงการนี้ คณาจารย์แผนกอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือทุกท่านซึ่งเอื้ออาทรต่อลูกศิษย์เสมอมา ไม่เว้นแม้คราวนี้ ซึ่งให้ยืมใช้เครื่องคอมพิวเตอร์และสถานที่ในการทำโครงการนี้ ผู้จัดทำขอขอบพระคุณทุกท่านทั้งเอ่ยถึงและมิได้เอ่ยถึงในความเอื้อเอื้อเพื่อทำให้โครงการนี้สำเร็จลงได้ด้วยดี



หนังสืออ้างอิง

1. สมยศ โลหะวิทยวิกรานต์และสมเกียรติ นุลศิริ, "ปลั๊กไฟไร้สัญญาณรบกวน" วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 86, หน้า 163-171, 2531
2. นลพตุง ผดุงกุล, "ไตรแอดและเอสซีอาร์ ตอนที่ 1", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 91, หน้า 246-254, 2532
3. ประเสริฐ โรจน์สุวิวัฒน์, "ลำโพงไอไฟไม่ต้องเดินสาย", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 86, หน้า 148-155, 2531
4. กนก เจนจิระพงศ์เวช, "สัญญาณรบกวนกับระบบจ่ายไฟ", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 76, หน้า 155-164, 2530
5. ยืน ภู่วรรณ, "จะลดปัญหาสัญญาณรบกวนในระบบไมโครโปรเซสเซอร์ได้อย่างไร", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 72, หน้า 164-175, 2529
6. สมยศ ภูงามแปลง, "MCS-51 ไมโครคอนโทรลเลอร์แบบทิพเดียว", วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์, ฉบับที่ 93, หน้า 264-269, 2532
7. คู่มือไอซีไมโครโปรเซสเซอร์ MCS 51 Microprocessor Data Book
8. Embedded Control Applications handbook, INTEL
9. The TTL Data Book. Volume 2, TEXAS INSTRUMENT, 1985

.....

## การใช้งานระบบควบคุมอุปกรณ์ผ่านสายไฟฟ้า

1. เมื่อเปิดเครื่องหน้าปัดจะแสดงคำว่า " rEAdy.. " เมื่อต้องการสั่งงานอุปกรณ์ไฟฟ้าจะต้องเริ่มต้นด้วยการกดปุ่ม st
2. เครื่องจะแสดงคำว่า " no.sLav " แล้วตามด้วย " 0-9 " ต้องกดหมายเลขประจำตัวของส่วนวงจรย่อยให้ตรงตามที่ตั้งด้วยคิฟสวิทช์
3. ถ้าไม่กดหมายเลข 0-9 เครื่องจะแสดงตามข้อ 2. ใหม่
4. เมื่อกดหมายเลขประจำวงจรย่อยถูกเครื่องจะแสดงคำว่า " no.devI " และตามด้วย " 1-9 " กดหมายเลขช่องที่ต้องการควบคุม ถ้าเครื่องไม่แสดงตามข้างต้นจะแสดงคำว่า " Error.. " ประมาณ 5 วินาทีแล้วกลับไปแสดงคำว่า " rEAdy.. " แสดงว่าไม่มีวงจรย่อยตามหมายเลขที่ต้องการ
5. เมื่อกดหมายเลขช่องแล้วจะมี 2 กรณีคือ
  - 5.1 ถ้าหมายเลขช่องเป็นช่อง 1 และ 2 เครื่องจะแสดงคำว่า " dim.CtL " ตามด้วย " 00-15-" ให้กดระดับกำลังไฟฟ้าที่ต้องการควบคุมได้ 16 ระดับ ถ้าถูกต้องเครื่องจะกลับไปสถานะเริ่มต้นตามข้อ 1. หรือถ้าต้องการตรวจสอบสถานะให้กดปุ่ม C เครื่องจะแสดงสถานะของอุปกรณ์ช่องนั้น
  - 5.2 ถ้าหมายเลขช่องเป็นช่อง 3 ถึงช่อง 9 เครื่องจะแสดงคำว่า " P\_CtLrL " ตามด้วย " OFF-on " และ " 0or1 " กด 1 ถ้าต้องการเปิดอุปกรณ์ 0 ถ้าต้องการปิดอุปกรณ์ถ้าถูกต้องเครื่องจะกลับไปอยู่ในสถานะตามข้อ 1. หรือถ้าต้องการตรวจสอบสถานะของอุปกรณ์กดปุ่ม C เครื่องจะแสดงสถานะของช่องนั้นออกมา

### สรุปการใช้งาน

1. กดปุ่มเริ่มสั่งงาน st
2. กดหมายเลข SLave ให้ตรงตามที่ต้องการ
3. กดหมายเลขช่องที่ต้องการ
4. กดปิดเปิด(0or1) หรือระดับกำลัง(00-15) หรือตรวจสอบ(C)