

**อากาศยาน 4 ใบพัด
4 ROTOR AIRCRAFT**



โดย

นาย ชีราวุธ

ระย้า

นาย ตึก

ไชยคำ

นาย สมเจตน์

บุญเนียม

21
2551
2150

เลขานุ

เลขทะเบียน **83087**

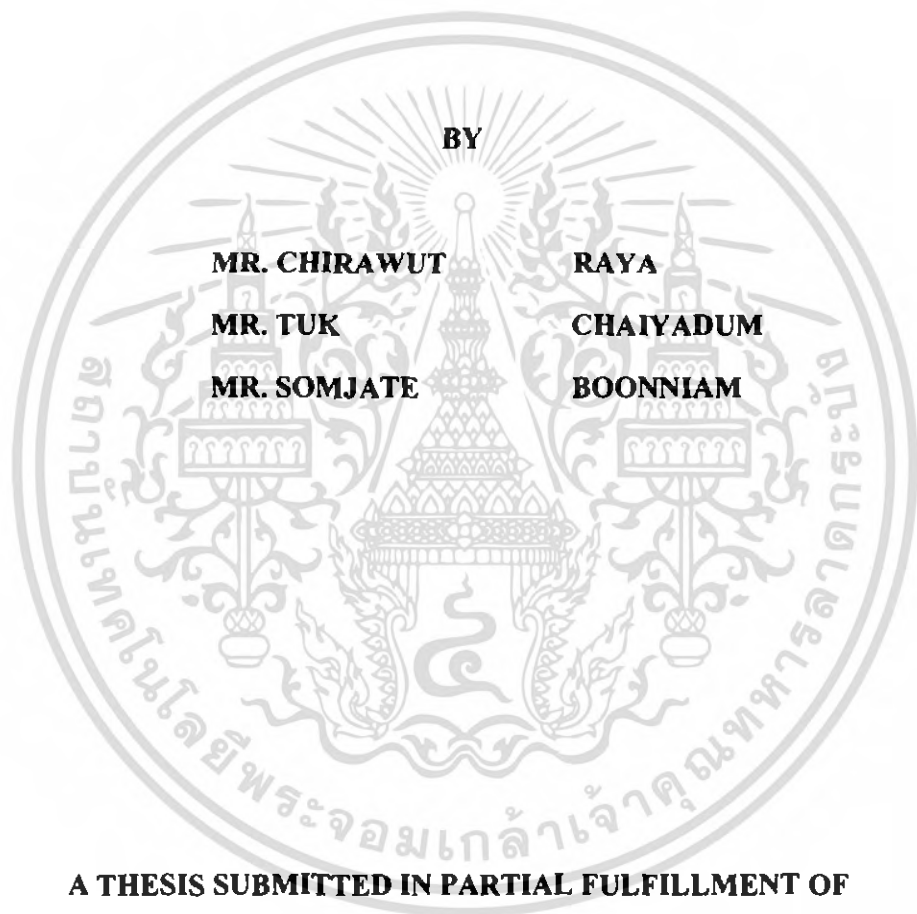
วัน,เดือน,ปี **- 5 ส.ค. 2551**

b. **11964078**
i.

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 ROTOR AIRCRAFT



BY

MR. CHIRAWUT

RAYA

MR. TUK

CHAIYADUM

MR. SOMJATE

BOONNIAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาบัตร อากาศยาน 4 ใบพัด
ชื่อนักศึกษา นาย ชีราวุธ รัชชา รหัสประจำตัว 48015574
นาย ศุภ ไซยคำ รหัสประจำตัว 48015577
นาย สมเจตน์ บุญนิยม รหัสประจำตัว 48015596

อาจารย์ที่ปรึกษา ผศ สมภพ แก้วมีชัย
ระดับการศึกษา ปริญญาตรีวิศวกรรมศาสตร์ สาขาวิศวกรรมสารสนเทศ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2550

ปริญญาบัตรฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง

(ผศ. สมภพ แก้วมีชัย)

อาจารย์ผู้ควบคุมปริญญาบัตร

หัวข้อปริญญานิพนธ์	อากาศยาน 4 ใบพัด		
ชื่อนักศึกษา	นาย ชีราวุธ	ระช้ำ	รหัสประจำตัว 48015574
	นาย ดูก	ไชยคำ	รหัสประจำตัว 48015577
	นาย สมเจตน์	บุญเนียม	รหัสประจำตัว 48015596
อาจารย์ที่ปรึกษา	ผศ. สมภพ แก้วมีชัย		
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตร์ สาขาวิศวกรรมสารสนเทศ		
ภาควิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2550		

บทคัดย่อ

ปริญญานิพนธ์นี้เสนอการออกแบบอากาศยานแบบ 4 ใบพัด ซึ่งสามารถลอยนิ่งอยู่บนอากาศและสามารถบินไปในทิศทางที่เราต้องการได้ โดยการควบคุมผ่านอุปกรณ์ควบคุมไมโครคอนโทรลเลอร์และยังสามารถรับน้ำหนักของอุปกรณ์เสริมได้

โดยปริญญานิพนธ์นี้แบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนฮาร์ดแวร์ ซึ่งประกอบด้วย ไอซีไมโครคอนโทรลเลอร์และวงจรควบคุมมอเตอร์ ส่วนซอฟต์แวร์ เป็นส่วนที่ใช้สำหรับควบคุมการทำงานของไมโครคอนโทรลเลอร์ 8051 ซึ่งเขียนด้วยโปรแกรมภาษาซี

Thesis Title	4 Rotor Aircraft		
Student	Mr. Chirawut	Raya	ID. 48015574
	Mr. Tuk	Chaidum	ID. 48015577
	Mr Somjate	Boonniam	ID. 48015596
Advisor	Asst. Prof. Sompoph Kaewmechar		
Graduate Level	Bachelor Degree of Information Engineering		
Department	Information Engineering		
Academic Year	2007		

ABSTRACT

In this thesis presents a designing of aircraft like 4 propeller , which can float still stay on the air and can fly go to in the direction that us wants by control equipment microcontroller and can take the weight of the equipment application.

The thesis is separated into two parts, hardware and software. The hardware consist of IC microcontroller and electronic speed control. The software for 8051 microcontroller used the c language.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดีในทุกๆ ด้าน ก็ด้วยความร่วมมือและการช่วยเหลือเป็นอย่างดีจากท่านอาจารย์และเจ้าหน้าที่ทุกๆ ท่านในภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยเฉพาะอย่างยิ่งจากท่านผู้ช่วยศาสตราจารย์ สมภพ แก้วมีชัย ซึ่งท่านเป็นอาจารย์ที่ปรึกษาทางด้านโครงงานนี้ของผู้จัดทำ ที่ได้ให้คำปรึกษาและคำแนะนำต่างๆ ที่มีประโยชน์อย่างยิ่งต่อการทำโครงงานนี้ให้เป็นผลสำเร็จลุล่วงไปได้ด้วยดีตลอดมา และต้องขอขอบพระคุณทุกๆ ท่านที่ได้ให้ความช่วยเหลือเป็นอย่างมาก โดยทางผู้จัดทำหวังว่าคงจะได้รับความอนุเคราะห์จากทุกๆ ท่านอีกในโอกาสต่อไป

สุดท้ายนี้ต้องขอกราบขอบพระคุณ คุณพ่อและคุณแม่ที่ได้ให้การสนับสนุนทุนทรัพย์ ทางด้านการศึกษาและกำลังใจด้วยดีตลอดมา

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญภาพ	ง
สารบัญตาราง	ญ
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการศึกษา	1
1.3 ขอบเขตของการทำโครงงานปริญญานิพนธ์	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงานของโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการ	
2.1 การทำงานของไบพัต	3
2.1.1 ข้อมูลทั่วไป	3
2.1.2 ความหมายของค่าที่ควรทราบ	4
2.1.3 แรงและความถี่ที่กระทำบนไบพัตขณะทำการบิน	8
2.2 ไมโครคอนโทรลเลอร์ MCS-51	10
2.2.1 ไมโครคอมพิวเตอร์	10
2.2.2 ไมโครคอนโทรลเลอร์	11
2.2.3 เหตุผลที่ต้องเลือกไอซีไมโครคอนโทรลเลอร์ MCS-51	12
2.2.4 ความหมายของไมโครคอนโทรลเลอร์	13
2.2.5 โครงสร้างของไมโครคอนโทรลเลอร์	13
2.2.6 การจัดขาต่างๆ ของ MCS-51	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.2.7 โครงสร้างของพอร์ตอินพุตและเอาต์พุต (I/O Port Structure)	18
2.2.8 โครงสร้างหน่วยความจำ	20
2.2.9 หน่วยความจำภายนอก	22
2.2.10 การติดต่อกับหน่วยความจำโปรแกรมภายนอก	23
2.2.11 การติดต่อกับหน่วยความจำข้อมูลภายนอก	25
2.2.12 การใช้งานพอร์ตสื่อสารอนุกรม	28
2.2.13 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51	29
2.2.14 ข้อมูลและคุณลักษณะสำคัญของไอซี AT89C4051	35
2.3 มอเตอร์กระแสตรง	38
2.3.1 การขับและกลับทิศทางของมอเตอร์กระแสตรง	38
2.3.2 การควบคุมความเร็วของมอเตอร์กระแสตรง	40
2.3.3 การควบคุมความเร็วมอเตอร์ขั้นพื้นฐาน	40
2.4 ทรานซิสเตอร์สนามไฟฟ้า (FET)	44
2.4.1 ชนิดของมอสเฟต	44
2.4.2 ดีมอสเฟต (D-MOSFET)	45
2.4.3 อีมอสเฟต (E-MOSFET)	49
2.5 การมอดูเลตสัญญาณดิจิทัล (Digital Modulation)	55
2.5.1 Amplitude Shift Keying (ASK)	56
2.5.2 Frequency Shift Keying (FSK)	58
2.5.3 Phase Shift Keying (PSK)	60
บทที่ 3 การออกแบบโครงสร้าง	62
3.1 โครงสร้างทางฮาร์ดแวร์	62
3.1.1 ส่วนของชุดควบคุม	63
3.1.2 ส่วนของโครงสร้าง	66
3.2 โครงสร้างทางซอฟต์แวร์	68

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการทดลอง	69
4.1 การทดลองวงจรขั้วมอเตอร์	69
4.2 ผลการทดลอง	70
บทที่ 5 สรุปผลและปัญหาการทำโครงการ	82
5.1 สรุปผลการทำโครงการ	82
5.2 ปัญหาที่พบและการแก้ไข	82
5.3 ข้อเสนอแนะ	83

บรรณานุกรม

ภาคผนวก ก

ภาคผนวก ข



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูป	หน้า
บทที่ 2	
รูปที่ 2.1 การเคลื่อนที่ของแรง Thrust	3
รูปที่ 2.2 แสดงความสัมพันธ์ระหว่างแรง Thrust และอากาศ	3
รูปที่ 2.3 ส่วนต่างๆ ของใบพัด	4
รูปที่ 2.4 รูปร่างของใบพัด	4
รูปที่ 2.5 Thrust Face	5
รูปที่ 2.6 Blade Angle	5
รูปที่ 2.7 Blade Element	6
รูปที่ 2.8 แรงลมที่กระทบกับใบพัด	7
รูปที่ 2.9 Blade Path	7
รูปที่ 2.10 Geometric Pitch และ Effective Pitch	8
รูปที่ 2.11 แรงที่กระทำต่อใบพัดขณะทำงาน	9
รูปที่ 2.12 แสดงโครงสร้างภายในของ MCS-51	15
รูปที่ 2.13 แสดงขนาดต่างๆของ 8051	16
รูปที่ 2.14 ขาของ MCS-51 ที่ใช้ต่อกับ XTAL	18
รูปที่ 2.15 โครงสร้างพอร์ตทั้ง 4 ของ MCS-51	19
รูปที่ 2.16 การต่อพอร์ทเข้ากับระบบบัสภายในของ MCS-51	20
รูปที่ 2.17 การจัดหน่วยความจำของ MCS-51	20
รูปที่ 2.18 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต	22
รูปที่ 2.19 ไคอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล	23
รูปที่ 2.20 ไคอะแกรมการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก	24
รูปที่ 2.21 การต่อ MCS - 51 กับหน่วยความจำโปรแกรมภายนอก	25
รูปที่ 2.22 ไคอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอก	26
รูปที่ 2.23 การต่อหน่วยความจำโปรแกรมกับ MCS-51	27
รูปที่ 2.24 การรีเซต MCS-51	27

สารบัญภาพ(ต่อ)

รูป	หน้า
รูปที่ 2.25 ผังการทำงานโหมด 0	31
รูปที่ 2.26 ผังการทำงานโหมด 1	32
รูปที่ 2.27 ผังการทำงานโหมด 2	33
รูปที่ 2.28 ผังการทำงานโหมด 3	34
รูปที่ 2.29 แสดงขาต่างๆ ของไอซี AT89C4051	35
รูปที่ 2.31 แสดงโครงสร้างภายในของ AT89C4051	37
รูปที่ 2.32 แสดงไคอะแกรมของสัญญาณ Brown-Out Detection	37
รูปที่ 2.33 แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง	38
รูปที่ 2.34 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์	39
รูปที่ 2.35 แสดงการใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน	39
รูปที่ 2.36 แสดงการใช้ทรานซิสเตอร์เป็นวงจรถับและ กำหนดทิศทางของมอเตอร์กระแสตรง	40
รูปที่ 2.37 วงจรควบคุมความเร็วของมอเตอร์กระแสตรงแบบใช้ตัวต้านทานอนุกรม	41
รูปที่ 2.38 กราฟแสดงคุณสมบัติการควบคุมด้วยตัวต้านทานที่ปรับค่าได้	41
รูปที่ 2.39 การควบคุมความเร็วโดยเปลี่ยนค่าแรงดัน	42
รูปที่ 2.40 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycle ที่ต่างๆ กัน	43
รูปที่ 2.41 แสดงความกว้างของพัลส์ขนาดต่างๆ และค่า Duty Cycle ของช่วงพัลส์ ที่มีความถี่คงที่	44
รูปที่ 2.42 วงจรสมมูลย์ของมอสเฟต	45
รูปที่ 2.43 โครงสร้างภายในของคิมอสเฟตแบบแซนแนล n	45
รูปที่ 2.44 การทำงานของคิมอสเฟตแบบแซนแนล n	46
รูปที่ 2.45 กราฟแสดงคุณสมบัติของคิมอสเฟตแบบแซนแนล n	47
รูปที่ 2.46 โครงสร้างและกราฟแสดงคุณสมบัติของคิมอสเฟตแบบแซนแนล p	48
รูปที่ 2.47 แสดงสัญลักษณ์คิมอสเฟตแบบแซนแนล n และ p	48
รูปที่ 2.48 โครงสร้างเบื้องต้นของคิมอสเฟตแบบแซนแนล n	49

สารบัญภาพ(ต่อ)

รูป	หน้า
รูปที่ 2.49 แสดงการทำงานของอิมอสเฟท	50
รูปที่ 2.50 กราฟแสดงคุณสมบัติของอิมอสเฟท	51
รูปที่ 2.51 กราฟแสดงคุณสมบัติของ	53
รูปที่ 2.52 แสดงโครงสร้างของอิมอสเฟทแบบแซนแนล p	53
รูปที่ 2.53 แสดงสัญลักษณ์ของอิมอสเฟทแบบแซนแนล p และแซนแนล n	54
รูปที่ 2.54 ลักษณะของสัญญาณ ASK – FSK– PSK	56
รูปที่ 2.55 การผสมสัญญาณของข้อมูล 01010 ด้วยวิธีการ ASK ในกรณีนี้ Bit Rate = Baud Rate	57
รูปที่ 2.56 Spectrum ของสัญญาณ ASK ที่มีความถี่ของสัญญาณพาหะ f_c และอัตราเร็วของข้อมูล Baud N_{baud}	58
รูปที่ 2.57 การผสมสัญญาณของข้อมูล 01101 ด้วยวิธีการ FSK ในกรณีนี้ Bit Rate = Baud Rate	58
รูปที่ 2.58 Spectrum ของสัญญาณ FSK ที่มีความถี่ของสัญญาณพาหะ f_{c1} และ f_{c0} และอัตราเร็วของข้อมูล Baud N_{baud}	59
รูปที่ 2.59 Constellation Diagram ของการผสมสัญญาณแบบ PSK แบบ Binary PSK ซึ่งใช้เฟส 2 เฟส แทนข้อมูล 0 และ 1	60
รูปที่ 2.60 Constellation Diagram ของการผสมสัญญาณแบบ PSK แบบ Dibit (ซ้าย) และ Tribit (ขวา) ตามลำดับ	60
รูปที่ 2.61 การผสมสัญญาณของข้อมูล 0110101100 ด้วยวิธีการ PSK ในกรณีนี้ Bit Rate มีค่าเป็น 2 เท่าของ Baud Rate	61

สารบัญภาพ(ต่อ)

รูป	หน้า
บทที่ 3	
รูปที่ 3.1 บล็อกโคอะแกรมของอากาศยานแบบ 4 ใบพัด	62
รูปที่ 3.2 แสดงวงจรส่วนภาคส่ง	63
รูปที่ 3.3 แสดงวงจรส่วนภาครับ	64
รูปที่ 3.4 วงจรควบคุมความเร็วมอเตอร์	65
รูปที่ 3.5 แสดงโครงสร้างส่วนใบพัด	66
รูปที่ 3.6 โครงสร้างค้ำข้างของอากาศยาน	67
รูปที่ 3.7 แสดงโครงสร้างทั้งหมดของอากาศยาน	67
รูปที่ 3.8 Flow Chart แสดงสถานะการทำงานของโปรแกรมควบคุม	68
บทที่ 4	
รูปที่ 4.1 ที่ค่า Input 0 %	72
รูปที่ 4.2 ที่ค่า Output 0 %	72
รูปที่ 4.3 ที่ค่า Input 10 %	72
รูปที่ 4.4 ที่ค่า Output 10 %	72
รูปที่ 4.5 ที่ค่า Input 20 %	72
รูปที่ 4.6 ที่ค่า Output 20 %	72
รูปที่ 4.7 ที่ค่า Input 25 %	73
รูปที่ 4.8 ที่ค่า Output 25 %	73
รูปที่ 4.9 ที่ค่า Input 30 %	73
รูปที่ 4.10 ที่ค่า Output 30 %	73
รูปที่ 4.11 ที่ค่า Input 35 %	73
รูปที่ 4.12 ที่ค่า Output 35 %	73
รูปที่ 4.13 ที่ค่า Input 40 %	74
รูปที่ 4.14 ที่ค่า Output 40 %	74

สารบัญภาพ(ต่อ)

รูป	หน้า
รูปที่ 4.14 ที่ค่า Input 45 %	74
รูปที่ 4.15 ที่ค่า Output 45 %	74
รูปที่ 4.16 ที่ค่า Input 50 %	74
รูปที่ 4.17 ที่ค่า Output 50 %	74
รูปที่ 4.19 ที่ค่า Input 55 %	75
รูปที่ 4.21 ที่ค่า Output 55 %	75
รูปที่ 4.19 ที่ค่า Input 60 %	75
รูปที่ 4.22 ที่ค่า Output 60 %	75
รูปที่ 4.23 ที่ค่า Input 65 %	75
รูปที่ 4.24 ที่ค่า Output 65 %	75
รูปที่ 4.25 ที่ค่า Input 70 %	76
รูปที่ 4.26 ที่ค่า Output 70 %	76
รูปที่ 4.27 ที่ค่า Input 75 %	76
รูปที่ 4.28 ที่ค่า Output 75 %	76
รูปที่ 4.29 ที่ค่า Input 80 %	76
รูปที่ 4.30 ที่ค่า Output 80 %	76
รูปที่ 4.31 ที่ค่า Input 85 %	77
รูปที่ 4.32 ที่ค่า Output 85 %	77
รูปที่ 4.33 ที่ค่า Input 90 %	77
รูปที่ 4.34 ที่ค่า Output 90 %	77
รูปที่ 4.35 ที่ค่า Input 95 %	77
รูปที่ 4.36 ที่ค่า Output 95 %	77
รูปที่ 4.37 ที่ค่า Input 100 %	78
รูปที่ 4.38 ที่ค่า Output 100 %	78
รูปที่ 4.39 แสดงการต่ออุปกรณ์ต่างๆ ลงบอร์ดทางด้านส่ง	78
รูปที่ 4.40 แสดงการต่อแผงควบคุมไอซี TRW 2 4 G ทางด้านรับ	79

สารบัญภาพ(ต่อ)

รูป	หน้า
รูปที่ 4.41 แสดงการต่อแผงควบคุมไอจี TRW 2.4 G ทางด้านส่ง	79
รูปที่ 4.42 แสดงการประกอบแผงควบคุมมอเตอร์ทางด้านรับ	80
รูปที่ 4.43 แสดงการประกอบ โครงสร้างของอากาศยาน	80
รูปที่ 4.44 แสดงการต่ออุปกรณ์ต่างๆ เข้ากับโครงสร้างของอากาศยาน	81
บทที่ 5	
รูปที่ 5.1 แสดงโครงสร้างของอากาศยาน เมื่อมีการเปลี่ยนแปลงความยาวของแกนใบพัดทั้ง 4 ด้าน	83



สารบัญตาราง

	ตาราง	หน้า
บทที่ 2		
	ตารางที่ 2.1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ต่างๆ	14
	ตารางที่ 2.2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3	17
	ตารางที่ 2.3 โหมดการทำงานของพอร์ตสื่อสารอนุกรม	28
	ตารางที่ 2.4 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3	36
บทที่ 4		
	ตารางที่ 4.1 แสดงค่าระดับสัญญาณเข้าเพื่อเช็คค่า ESC	70
	ตารางที่ 4.2 แสดงค่าระดับสัญญาณที่เขียนขึ้นมาจาก MCS51	71



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากทุกวันนี้การสำรวจพื้นที่ที่ยากต่อการเข้าถึงนั้นทำได้ยาก และมีปัญหาอุปสรรคอีกทั้งถ้าจะใช้เครื่องบินจริงนั้นก็จะต้องเสียค่าใช้จ่ายมากมายมหาศาล ซึ่งในบางครั้งเราแค่ต้องการสำรวจพื้นที่ที่บางส่วนของเท่านั้น เช่น พื้นที่บริเวณบ้าน พื้นที่บริเวณบริษัท หรือหน่วยงาน และการสำรวจพื้นที่ ที่มีอันตรายต่ออันตรายในการเข้าไปสำรวจของมนุษย์ เช่น การสำรวจรอยร้าวตามผนังอาคารที่สูงๆ และ ที่มีสารเคมีที่ฟุ้งกระจายในอากาศ

จากปัญหาที่กล่าวมาแล้วนั้นจึงทำให้เกิดแนวความคิดที่ว่า ถ้าเรามีอุปกรณ์หรือเครื่องมือที่ใช้ในการสำรวจในมุมสูง ก็จะทำให้ประหยัดงบประมาณในการสำรวจพื้นที่และมีความปลอดภัยในการสำรวจหารอยแตกร้าวของพื้นผิวผนังของอาคารสูงๆ ได้ จึงก่อให้เกิดแนวความคิดในการสร้างอากาศยาน 4 ใบพัด (4 ROTOR AIRCRAFT)

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาและเรียนรู้การควบคุมการทำงานของไมโครคอนโทรลเลอร์
ในตระกูล MCS-51 ในการติดต่อกับอุปกรณ์ต่างๆ

1.2.2 เพื่อศึกษาและเรียนรู้การทำงานของการบินแบบ 4 ใบพัด

1.2.3 เพื่อศึกษาและเรียนรู้การทำงานของการบินรับ-ส่งของการควบคุมระยะไกล
(Remote Control)

1.2.4 เพื่อนำไปประยุกต์ใช้กับการสำรวจในพื้นที่ที่ยากจะเข้าถึง

1.3 ขอบเขตของการทำโครงการปริญญาโท

1.3.1 อากาศยาน สามารถขึ้น-ลง ทางแนวคิ่งได้ และสามารถเคลื่อนที่ไปทางซ้าย-ขวาได้

1.3.2 อากาศยาน สามารถบังคับการทำงานได้ในระยะไกลประมาณ 30 เมตร

1.3.3. อากาศยาน สามารถทำงานได้ประมาณ 5 นาที

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 เพื่อเป็นการสร้างเครื่องมือที่ช่วยเหลือผู้ประสบภัยจากภัยพิบัติทางธรรมชาติ
- 1.4.2 เพื่อเป็นการสร้างแนวคิดในการประยุกต์และพัฒนาหุ่นยนต์สำรวจ
- 1.4.3 ได้ศึกษาการใช้โปรแกรมคอมพิวเตอร์ในการควบคุมและสั่งงาน

1.5 ขั้นตอนการดำเนินงานของโครงการ

- 1.5.1 สร้างแนวความคิดโครงการ
- 1.5.2 รวบรวมข้อมูลที่จะนำมาใช้ในโครงการ
- 1.5.3 ออกแบบวงจรและเขียนโปรแกรมการทำงาน
- 1.5.4 ทดสอบการทำงาน
- 1.5.5 ออกแบบและสร้างแผ่นวงจรพิมพ์
- 1.5.6 ทำการลงอุปกรณ์
- 1.5.7 ทดสอบการทำงาน ถ้ายังใช้งานไม่ได้ต้องทำการแก้ไขข้อบกพร่อง
- 1.5.8 ทำเอกสาร หลังจากทดสอบการทำงานแล้ว

บทที่ 2

ทฤษฎีและหลักการ

2.1 การทำงานของใบพัด

2.1.1 ข้อมูลทั่วไป

Thrust เป็นแรงที่ใช้ขับเคลื่อน อากาศยานไปในอากาศ Thrust เป็นระบบที่สร้างขึ้นเพื่อใช้ผลักดันอากาศยาน ซึ่งมีอยู่หลายแบบและหลายวิธีการในการสร้างระบบผลักดันนี้ขึ้นมา ถึงแม้ว่ามันจะอาศัย โดยใช้ กฎของ Newton ข้อที่สาม ใบพัดก็เป็นหนึ่งในระบบขับเคลื่อนอากาศยาน จุดมุ่งหมายของ ใบพัดก็คือการขับเคลื่อนอากาศยาน ให้เคลื่อนที่ไปในอากาศ ใบพัดประกอบใบ คิวใบ หรือกลีบ ตั้งแต่สองกลีบ หรือสองใบขึ้นไป ต่อกันด้วยที่ศูนย์กลาง ซึ่งเรียกว่า Hub และ Hub นี้ทำหน้าที่ยึด ใบพัดแต่ละกลีบ หรือแต่ละใบ เข้ากับ Shaft ของเครื่องยนต์

รูปที่ 2.1 การเคลื่อนที่ของแรง Thrust

ใบพัด สร้างขึ้นให้มีลักษณะหรือรูปร่าง เป็น Airfoil คล้ายกับลักษณะของปีกเครื่องบิน เมื่อ ใบพัดหมุนโดยการหมุนของเครื่องยนต์ ใบพัดก็จะสร้าง แรง ยก ไปทางด้านหน้าของเครื่องบิน และ แรงยกส่วนนี้เราเรียกว่า Thrust ที่จะทำให้เครื่องบินเคลื่อนที่ไปข้างหน้า อากาศยานส่วนใหญ่ มี ใบพัดแบบที่ใช้ดึง เครื่องบิน ผ่านไปในอากาศ ใบพัดประเภทนี้เรียกว่า ใบพัดแบบ Tractor อากาศยานบางเครื่อง ใช้ใบพัดแบบผลัก ให้เครื่องบินเคลื่อนที่ไปในอากาศ เรียกใบพัด ประเภทนี้ว่า Pusher



รูปที่ 2.2 แสดงความสัมพันธ์ระหว่างแรง Thrust และอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ความหมายของคำที่ควรทราบ

Leading Edge ก็คือส่วน แรกของใบพัดที่ หมุนตัดกับอากาศ เมื่อใบพัดตัดอากาศ อากาศก็จะ ไหลผ่าน บนด้านหน้าของใบพัด และส่วนที่เป็นส่วน โกงของใบพัด

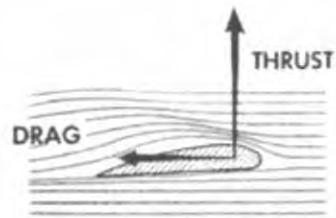


Blade Face ก็คือส่วนล่างของ ใบพัด หรือ ส่วนล่างของ Airfoil แต่เราอาจจะเรียกว่าเป็น ด้านหลังของใบพัด

รูปที่ 2.4 รูปร่างของใบพัด

Thrust Face ก็คือส่วนที่มีความโค้งพื้นผิว ของใบพัด หรือบางที่เราเรียกว่าด้านหน้าของ ใบพัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



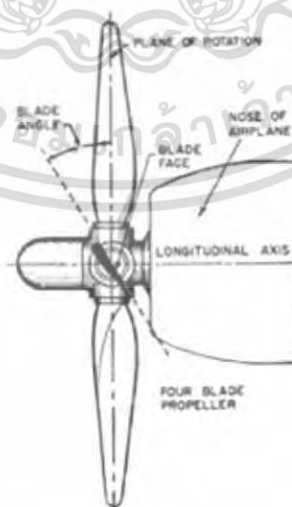
รูปที่ 2.5 Thrust Face

Blade Shank (Root) คือส่วนของกลีบใบพัดที่อยู่ส่วนหัว หรือ ส่วนที่อยู่ติดกับ ส่วนตรงกลาง (hub)

Blade Tip คือส่วนปลายสุดของใบพัด

Plane of Rotation คือระนาบจินตนาการของการหมุนของใบพัดที่ตั้งฉากกับ แกนของเครื่องยนต์ ระนาบนี้จะเป็นระนาบวงกลม ตามที่ใบพัดหมุน

Blade Angle คือมุมที่เกิดจาก ส่วนของด้านหลังของใบพัด หรือ Blade Face กับ ระนาบการหมุนของใบพัด มุมที่เกิดขึ้นตลอดระยะ ความยาวของใบพัด ตั้งแต่โคนถึงปลายใบพัด จะไม่เท่ากัน เหตุผลในการที่มุมตลอดใบพัด มีค่าต่างกัน เพราะว่า ระยะความยาวของใบพัด จากแกนศูนย์กลางการหมุนไม่เท่ากัน ทำให้ความเร็วของใบพัดแต่ละส่วนมีความเร็วไม่เท่ากัน ซึ่งมีผลกับแรงที่เกิดขึ้น ดังนั้นการออกแบบจึง ต้องการให้ทุกๆส่วนของใบพัด มี มุม Angle of Attack ของส่วนของตัวเองที่มี ประสิทธิภาพที่สุด ที่สามารถสร้างแรง Thrust ตามความเร็วรอบที่ ออกแบบมา



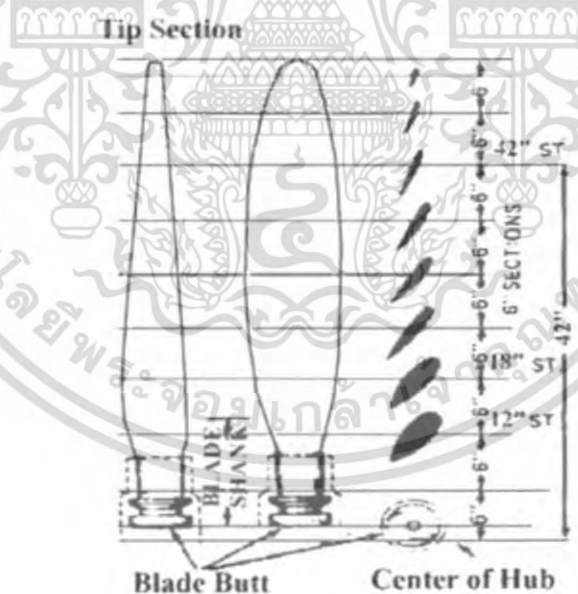
รูปที่ 2.6 Blade Angle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Blade Element คือส่วนย่อยๆของความยาวของใบพัด เหมือนนำส่วนย่อยๆนี้มาต่อกัน ขึ้นมาเป็นใบพัดหรือ Blade Airfoil ส่วนย่อยๆนี้ วางอยู่ในตำแหน่ง ที่ทำมุมกับระนาบการหมุนที่ต่างกัน

เหตุผลในการวางส่วนต่างๆที่มุมต่างกัน เพราะว่า ส่วนย่อยๆต่างๆของใบพัดนั้นมีความเร็วในการหมุนที่ต่างกัน ส่วนของใบพัดที่อยู่ด้านในติดกับศูนย์กลางจะมีความเร็วที่ช้ากว่าส่วนที่อยู่ไกลออกไปที่ส่วนปลายของใบพัด ถ้าหากว่าทุกส่วนตลอดความยาวของใบพัดมีมุมเท่ากันหมด ทิศทางของลมที่กระทบกับใบพัดก็จะไม่กระทบกับใบพัดที่มุม Angle of Attack ที่เท่ากัน นี่เป็นเพราะ ความเร็วของใบพัดตลอดระยะความยาวจะไม่เท่ากัน

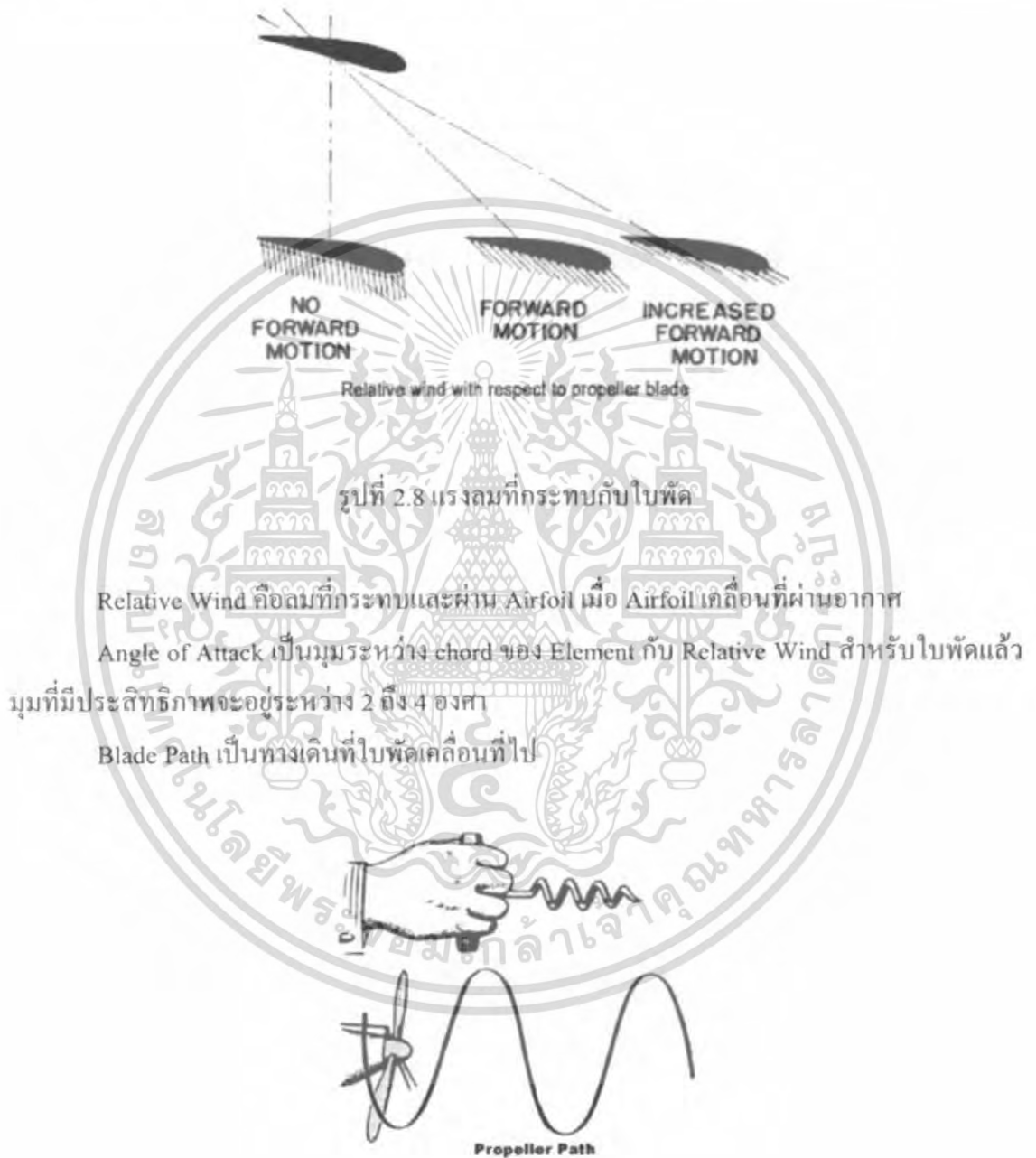
ใบพัดจะมีลักษณะบิดเป็นเกลียวเล็กน้อย (เนื่องจากการมีมุมที่ต่างกันในแต่ละส่วนของใบพัด)ในตัวของมัน ด้วยเหตุผลที่สำคัญเมื่อใบพัดหมุนไปรอบๆ แต่ละส่วนของใบพัดหมุนด้วยความเร็วที่ต่างกัน การที่ใบพัดบิดเล็กน้อยนั้นหมายความว่าแต่ละส่วนที่เคลื่อนที่ไปข้างหน้าด้วยความเร็วเท่ากัน จึงทำให้ไม่เกิดแรงที่ส่วนใดส่วนหนึ่งของใบพัดมากกว่ากันซึ่งจะทำให้เกิดอาการงอได้ (Bending)



รูปที่ 2.7 Blade Element

Thrust ที่เกิดจากใบพัดที่ติดอยู่กับ Shaft ของเครื่องยนต์ขณะที่ใบพัดหมุน ขณะทำการบิน แต่ละส่วนเคลื่อนไหวพร้อมกับการเคลื่อนที่ไปข้างหน้าของเครื่องบิน และการหมุนรอบของใบพัด ส่วนที่หมุนช้าก็จะมีมุม Angle of Attack ที่มากกว่าในการสร้าง Thrust ดังนั้นรูปร่าง (Cross) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาดเห็นไปใช้ประโยชน์ด้านการศึกษาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

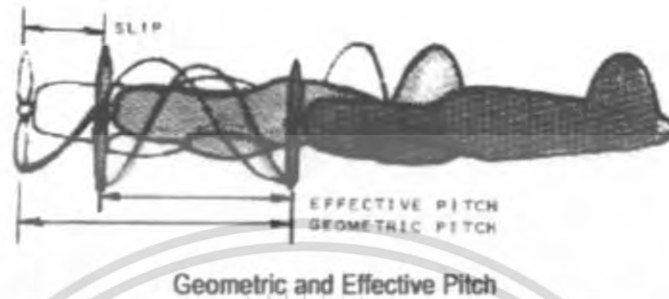
Section)ของใบพัดจำเป็นต้องเปลี่ยนแปลงจากโคนใบพัดไปจนถึงปลายของใบพัด และการเปลี่ยนรูปร่างเช่นนี้ทำให้ใบพัดมีลักษณะบิด(Twist)ของใบพัด



รูปที่ 2.9 Blade Path

Pitch อ่างถึงระยะทางที่เป็นเกลียวเหมือนเกลียวของสกรูที่เคลื่อนที่ไปข้างหน้าหนึ่งรอบ ซึ่งก็เหมือนใบพัดเคลื่อนที่ไปข้างหน้าเมื่อหมุนอยู่ในอากาศ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Geometric Pitch เป็นระยะทางในทางทฤษฎีที่ใบพัดควรจะเคลื่อนที่ไปข้างหน้าเมื่อใบพัดหมุนไปหนึ่งรอบ



รูปที่ 2.10 Geometric Pitch และ Effective Pitch

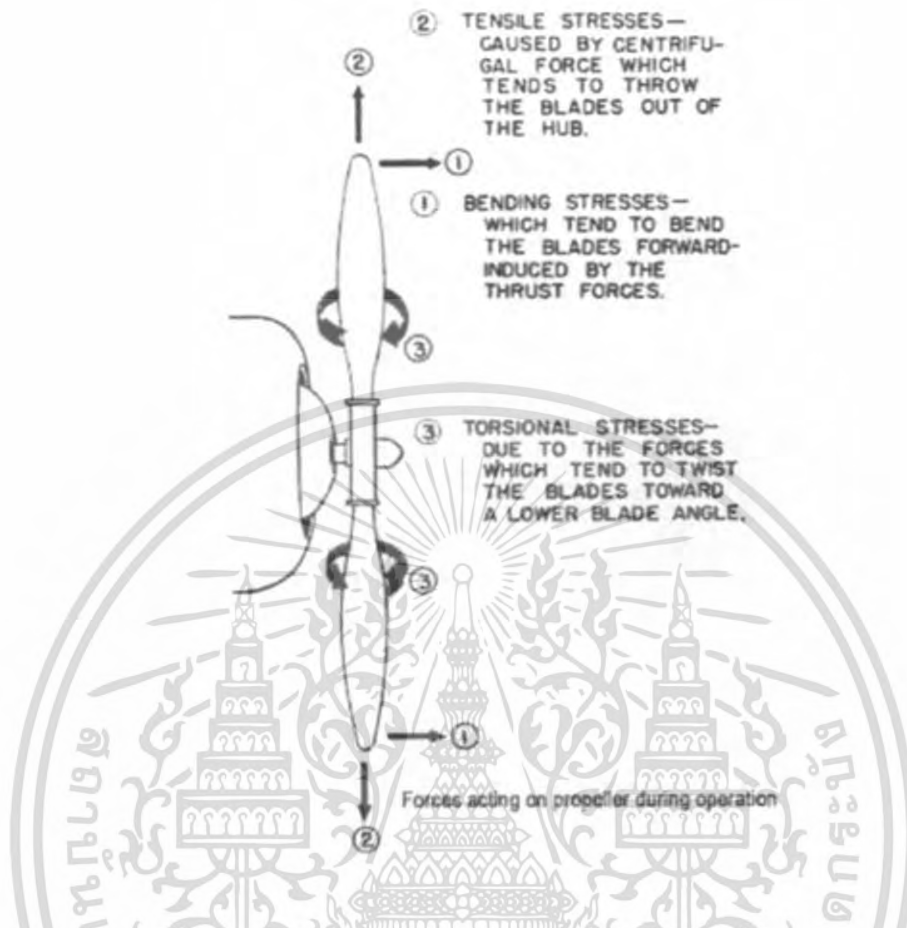
Effective Pitch เป็นระยะทางในทางปฏิบัติจริงๆ เมื่อใบพัดหมุนหนึ่งรอบ ในขณะที่ทำการบินในอากาศ Effective Pitch จะมีระยะทางสั้นกว่า Geometric Pitch เสมอ เนื่องจาก อากาศที่เป็นของไหล จะ สลัด (Slip)

2.1.3 แรง และ ความต้านทานใบพัด ขณะทำการบิน

แรง (The Force) ที่กระทำต่อใบพัดขณะทำการบิน คือ

1. Thrust เป็นแรงของอากาศบนใบพัด ซึ่ง ขนานกับ ทิศทางที่ไปข้างหน้า และก่อให้เกิดแรงที่ทำให้ใบพัดมีอากาศที่จะงอ
2. Centrifugal force เป็นแรงหนีศูนย์กลาง เกิดจากการหมุนของใบพัด มีอาการที่พยายามเหวี่ยงใบพัด ออกไปจากศูนย์กลาง
3. Torsion or Twisting forces ภายในตัวของใบพัดเอง ซึ่งเกิดจากผลของ แรงที่เกิดจากอากาศ ที่พยายามจะบิด ใบของใบพัด ไปหา มุมที่ต่ำกว่า หรือไปหามุมที่แบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แรงที่กระทำต่อใบพัดขณะทำงาน

ความล้า (The Stress) ที่กระทำต่อใบพัดขณะทำการบินคือ

1. Bending Stresses ซึ่งเกิดจากแรง Thrust ที่กระทำต่อใบพัด ความล้านี้พยายามที่จะโค้งงอ ใบพัดไปข้างหน้า ขณะที่เครื่องบินเคลื่อนที่ไปในอากาศ ด้วยใบพัด
2. Tensile Stresses เกิดจากแรงหนีศูนย์กลางของใบพัดเอง
3. Torsion Stresses แรงบิดนี้เกิดจากการ หมุนของใบพัดเอง ด้วยแรงบิดสองแรง แรงแรกเกิดจากแรงที่กระทำตอบ ได้กับแรงลมที่เกิดจากใบพัด ที่เรียกว่า Aerodynamic Twisting Moment อีกแรงหนึ่ง เกิดจากแรงหนีศูนย์กลาง และเรียกแรงนี้ว่า Centrifugal Twisting Moment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไมโครคอนโทรลเลอร์ MCS-8051

2.2.1 ไมโครคอมพิวเตอร์

เมื่อเราเริ่มต้นเรียนรู้เรื่องคอมพิวเตอร์จะพิจารณาได้อย่างไรว่าชิ้นงานที่เห็นอยู่นั้นเป็นไมโครคอมพิวเตอร์ได้หรือไม่ ให้เราพิจารณาได้จากองค์ประกอบของชิ้นงาน ซึ่งมีส่วนประกอบที่สำคัญดังนี้

1. หน่วยประมวลผลกลาง (CPU: Central Processing Unit) ซึ่งมีคุณสมบัติหลักคือประมวลผลข้อมูลการคำนวณทางคณิตศาสตร์และลอจิก เรารู้จักกันในชื่อของ ไอซี ไมโคร โปรเซสเซอร์ เช่น 8080, 80286, 80486 Pentium ฯลฯ ซึ่งเป็นของบริษัท Intel หรืออาจเป็น CPU รุ่นเก่าที่มีขนาด 8 บิต เช่นเบอร์ Z80 ซึ่งเป็น CPU ของบริษัท ZILOG เป็นต้น

2. หน่วยความจำ (Memory Unit)

หน่วยความจำรอม (ROM Read Only Memory) เป็นหน่วยความจำแบบถาวรที่มีการบันทึกข้อมูลไว้ล่วงหน้าก่อนแล้ว ไม่สามารถเปลี่ยนแปลงข้อมูลหรือคำสั่งใดๆ ได้อีก ตัวอย่างเช่น ไอซีที่เป็นไบออส (BIOS) ของคอมพิวเตอร์ ในขณะที่เริ่มเปิดเครื่องไมโครคอมพิวเตอร์ครั้งแรกสังเกตได้ว่าจะมีการแสดงชื่อผู้ผลิตของบริษัทหรือ คุณสมบัติของเครื่องบนหน้าจอคอมพิวเตอร์ ถ้าหากเป็นเครื่องเล่นวีดีโอ จะเป็นตัวอักษรที่ทำหน้าที่แสดงผลเพื่อ บอกให้ตั้งค่าข้อมูลต่างๆ ซึ่งไม่ว่าจะปิดแล้วเปิดก็ครั้งตัวอักษรเดิมนั้นจะยังคงอยู่

หน่วยความจำแรม (RAM. Random Access Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูลไว้เพียงชั่วคราว อาจเป็นข้อมูลที่ CPU ต้องการประมวลผลในขณะนั้น และเมื่อ CPU ประมวลผลเรียบร้อยแล้วอาจลบหรือเปลี่ยนข้อมูลได้ บางครั้งเมื่อหยุดการจ่ายไฟให้กับวงจรจะทำให้ข้อมูลสูญหายไปได้ในทันที ตัวอย่าง เช่น ขณะที่เรากำลังพิมพ์งานแต่ยังไม่ได้บันทึกข้อมูลไว้ในส่วนใด ข้อมูลนี้จะถูกเก็บไว้ที่หน่วยความจำแรมก่อนหากเกิดเหตุการณ์ไฟฟ้าดับ จะทำให้ข้อมูลสูญหายไปหรือการเก็บค่าของเวลาและอุณหภูมิของเครื่องไมโครเวฟ ที่สามารถเปลี่ยนแปลงค่าได้ตลอด หน่วยความจำแบบแรมนี้จะแตกต่างกับหน่วยความจำแบบรอม โดยหน่วยความจำแบบรอมจะไม่สามารถแก้ไข ข้อมูลได้ในขณะนั้น ในขณะที่หน่วยความจำแบบแรมไม่สามารถเก็บค่าข้อมูลไว้ได้ตลอดดังนั้นหากต้องการให้ข้อมูลคงอยู่ต้องใส่แบตเตอรี่สำรองไฟฟ้าไว้

3. หน่วยอินพุต (Input Unit) เป็นหน่วยที่ใช้สำหรับการรับสัญญาณข้อมูลจากภายนอกเช่น คีย์บอร์ด สแกนเนอร์ หรือที่รับสัญญาณมาจากอุปกรณ์เซนเซอร์ (Sensor)ซึ่งอาจเป็นค่าแรงเสียด

ทานของลอร์ดยนต์ขณะเบรการกดปุ่มสวิชต์ตั้งเวลาของวิดีโอเทป ฯลฯ กล่าวได้ว่าส่วนที่เป็น อินพุต คือส่วนที่ทำหน้าที่ป้อนข้อมูล

4. หน่วยเอาต์พุต (Output Unit) เป็นหน่วยที่ใช้สำหรับการแสดงผลของข้อมูลเช่น จอมอนิเตอร์ พริ้นเตอร์ เครื่องตัดสติ๊กเกอร์หรืออุปกรณ์ประเภทแอลอีดี(LED) ลำโพง มอเตอร์ รีเลย์ หลอดไฟ ฯลฯ

2.2.2 ไมโครคอนโทรลเลอร์

ปัจจุบันการพัฒนาและการแข่งขันทางด้านเทคโนโลยีผลิตชิ้นส่วนสารกึ่งตัวนำ ที่นำไปสร้างเป็นไอซีมี ประสิทธิภาพสูงมากขึ้นและมีเทคโนโลยีที่เกิดจากการผลิตของบริษัทต่างๆ ซึ่งส่งผลให้การผลิตชิพไอซีมีขนาดที่เล็กลง แต่มีประสิทธิภาพและคุณสมบัติต่างๆมากขึ้น ไอซีที่ถูกสร้างเป็นแบบ LSI (Large Scale Integrate Circuit) เป็นเทคโนโลยีการสร้างโดยการนำเอาทรานซิสเตอร์จำนวนมากมาสร้างเป็นไอซีดิจิทัลที่ซับซ้อนโดยทำขึ้นเพื่อหน้าที่เป็นหน่วยประมวลผลข้อมูลหรือ เรียกว่าไมโครโพรเซสเซอร์ (Microprocessor) ที่มีคุณสมบัติหลัก คือการประมวลผลข้อมูลการคำนวณทางคณิตศาสตร์ และลอจิก

ถ้าหากมีการติดต่อกับหน่วยความจำที่เป็นแบบแรมแบบรอม หรืออุปกรณ์ภายนอกที่เป็น อินพุต-เอาต์พุตต้องมีการต่ออุปกรณ์อื่น ๆ ร่วมด้วย เพื่อทำหน้าที่เลือกอุปกรณ์ในการติดต่อหรือ วงจรถอดรหัส (Decoder) ซึ่งสามารถทำงานได้ภายใต้การควบคุมของ โปรแกรมและ ในการที่เรา นำไมโครโพรเซสเซอร์มาเป็นตัวประมวลผลกลางมีหน่วยความจำแบบแรมพอร์ดอินพุตและเอาต์พุต เราเรียกว่า ไมโครคอมพิวเตอร์ เป็นสิ่งไม่คุ้มกับการลงทุนหากนำมาใช้ในงานควบคุมขนาดเล็ก และอาจต้องใช้เนื้อที่มาก ในการออกแบบ ดังนั้นการพัฒนาด้านเทคโนโลยีในการสร้างชิป จึงมีการรวบรวมคุณสมบัติที่ต้องการใช้งานมาอยู่ในตัวเดียวกันคือมีองค์ประกอบเกือบทุกอย่างของคอมพิวเตอร์อยู่ในตัวไอซี ที่เราเรียกว่า ไมโครคอมพิวเตอร์แบบชิปเดี่ยวประกอบด้วยอุปกรณ์พื้นฐานเหมือนไมโครคอมพิวเตอร์ เช่นหน่วยประมวลผลกลางขนาดเล็ก (8บิต -16 บิต) และหน่วยประมวลผล ที่สามารถเข้าข้อมูลแบบบิตหน่วยความจำข้อมูลพื้นฐานแบบแรมขนาด 128 ไบต์ และบรรจุกหน่วยความจำ โปรแกรมประเภทรอม(บางเบอร์) สามารถใช้งานได้ทั้งอินพุตและเอาต์พุตมีวงจรสื่อสารอนุกรมแบบ พูลคูลเพล็กซ์ วงจรCounter/Timcr ที่อยู่ใน สามารถต่ออุปกรณ์ที่ใช้ในการสร้างวงจรกำเนิดสัญญาณนาฬิกาเช่น คริสตอล(Crystal) และตัวเก็บประจุก็สามารถใช้งานได้เป็นต้น เราเรียกกันทั่วไปว่า ไอซีไมโครคอนโทรลเลอร์(Microcontroller) ดังนั้นเมื่อเราต้องการใช้งานควบคุมขนาดเล็ก เช่น เต้าไมโครเวฟ เครื่องซักผ้า เครื่องเล่นวิดีโอเทป และเครื่องใช้ไฟฟ้าอื่นๆ เราจึงนิยมนำไอซีไมโครคอนโทรลเลอร์มาใช้งาน เพราะมีทุกอย่างพร้อมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวเดียวกัน ประกอบกับมีขนาดที่เล็กอุปกรณ์ที่จะนำมาต่อรวมมีน้อยและเหมาะสำหรับใช้งานในการคำนวณที่ไม่ซับซ้อนมากนัก

2.2.3 เหตุผลที่ต้องเลือกใช้ไอซีไมโครคอนโทรลเลอร์ MCS-51

จากความก้าวหน้าของเทคโนโลยีจึงพบว่าอุปกรณ์บางส่วนที่นำมาควบคุมเครื่องใช้ไฟฟ้า ระบบควบคุมทางอิเล็กทรอนิกส์หรือ โครงการในวารสารต่าง ๆ จะนำเสนอในรูปแบบของไอซีไมโครคอนโทรลเลอร์ ซึ่งเทคโนโลยีดังกล่าวมีบริษัทต่างๆ ที่ผลิตออกมาแข่งขันด้านคุณสมบัติและการใช้งานกันในตลาด อาทิเช่น Intel, Atmel, Microchip, Motorola ส่วนทางด้านการศึกษาและการเรียนรู้ นั้น มีการพัฒนาทางการเรียนการสอนวิชาไมโครคอนโทรลเลอร์ โดยบริษัทต่างๆ แข่งขันกันเพื่อสร้างชุดฝึกคอบสนองต่อความต้องการเรียนรู้ ความสะดวกในการสร้างชุดทดลองของนักศึกษาและผู้สนใจ

ในขณะที่เดียวกัน เราซึ่งเป็นผู้ตามเพื่อศึกษาเทคโนโลยีที่ทิ้งห่างทุกขณะ จึงต้องใช้ชุดทดลองที่ถูกกำหนดรูปแบบและราคา ทั้งที่อุปกรณ์ ที่เป็นอุปกรณ์ร่วมกับเทคโนโลยีบางอย่าง เรากลับทิ้งไปและไม่ได้นำมาใช้งาน ยกตัวอย่าง เช่น Main Board ,CD-ROM หรือ Hard Disk ที่ใช้งานไม่ได้แล้ว เรานำอุปกรณ์ภายในมาใช้งานได้ก็ชิ้นบ้าง คลอดจนตัวด้านทานและตัวเก็บประจุ ที่เป็นอุปกรณ์ติดบนผิวหน้าแบบ SMD (Surface Mounted Device) หน่วยความจำชนิดแรม และรอม แผ่นระบายความร้อนสเตปป์มอเตอร์ (Stepping Motor) คอนเนคเตอร์ (Connector) ฯลฯ ซึ่งพร้อมที่จะให้เราใช้งานมากกว่าที่จะได้เพิ่มมาจากการซื้อชุดทดลองใหม่

ดังนั้นไมโครคอนโทรลเลอร์จึงถูกเลือกใช้ในการทดลองด้วยเหตุผลในการใช้อุปกรณ์ที่ต่อรวมในวงจรน้อยชิ้น ราคาถูก หาแหล่งข้อมูลได้ง่ายมีการพัฒนาประสิทธิภาพ ในการประมวลผลเทียบเท่าไมโครโพรเซสเซอร์ขนาด 8 บิต-16บิต และ ใช้ในการพัฒนากับงานที่ไม่ต้องการความซับซ้อนซึ่งจะทำให้มีความสะดวกมากขึ้นปัจจุบันมีไอซีของสองบริษัทที่เรานิยมใช้ในงานออกแบบและสำหรับในการเรียนการสอนในสถาบันหลายแห่ง คือ ไอซีไมโครคอนโทรลเลอร์ของบริษัท Microchip ตระกูล 16FXXX และไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 จากหลายบริษัทที่ผลิตออกมามากมายหลายเบอร์ แต่คำสั่งจะใกล้เคียงกัน ดังนั้น การทดลองในหนังสือเล่มนี้จึงเลือกใช้งานไอซี ไมโครคอนโทรลเลอร์ของตระกูล MCS-51 ด้วยเหตุผลเพราะมีข้อมูล และตัวอย่างที่หลากหลายหามาใช้งานได้ง่าย แต่ก็ไม่ได้ตัดสินใจว่าดีกว่าไอซีในตระกูลอื่นๆ ดังนั้น ในการออกแบบชุดทดลองที่ใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ของ MCS-51 ที่จะแนะนำให้สร้างขึ้นนี้ สามารถที่จะนำมาใช้งานร่วมกับไอซีไมโครคอนโทรลเลอร์ได้ทุกเบอร์ และทุกตระกูล

2.2.4 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่สามารถรวมความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณออกทางเอาต์พุต หน่วยความจำ และวงจรถ่ายเก็บสัญญาณนาฬิกา ทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรรีเลย์อิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลงในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกัน คือ “ไมโคร(Micro)” ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วยหน่วยประมวลผลกลางหรือซีพียู (CPU . Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU . Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรถ่ายเก็บสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์(Controller)” หมายถึงอุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์ จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.2.5 โครงสร้างของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีจำนวนมาก ขึ้นกับโครงสร้างภายใน บางเบอร์มีหน่วยความจำภายในเป็นแบบรอม บางเบอร์เป็นแบบอีพรอม บางเบอร์มีแรมภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

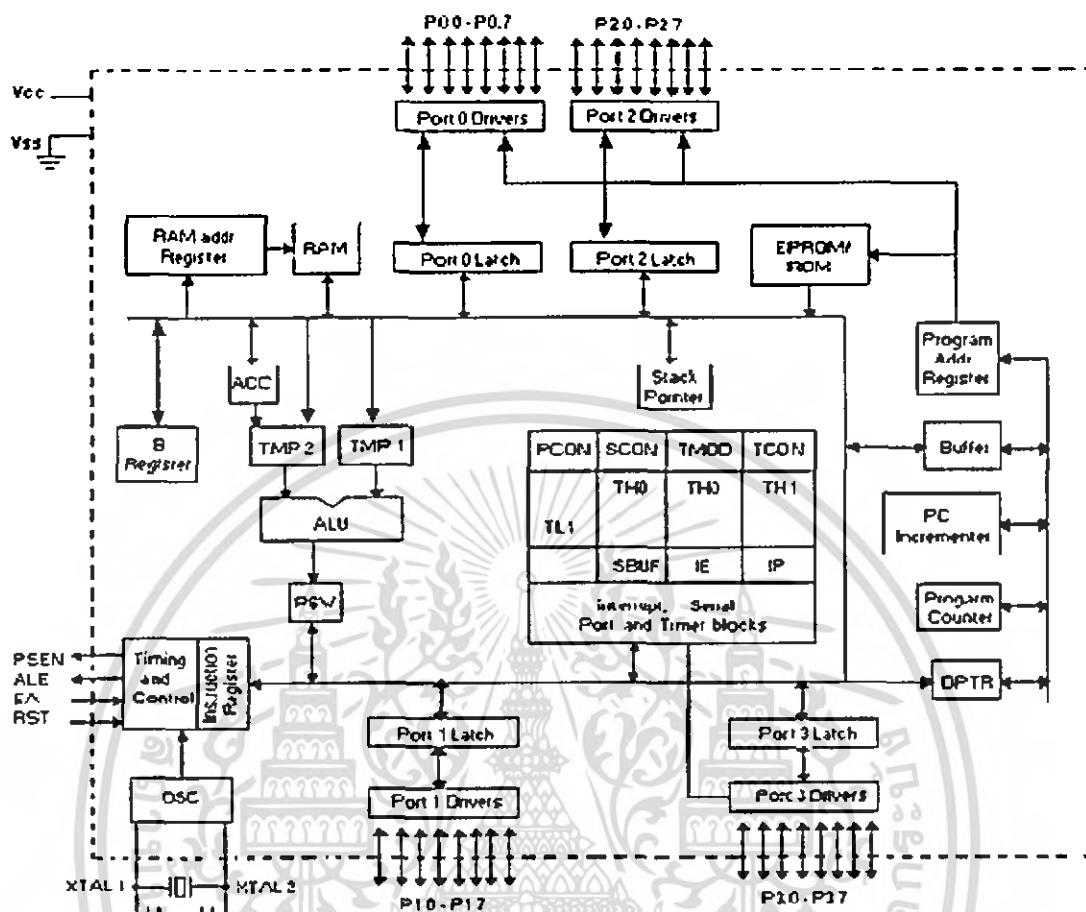
- หน่วยความจำรวม 4 กิโลไบต์
- หน่วยความจำแรม 128 ไบต์
- พอร์ตอินพุตและเอาต์พุต ขนาด 8 บิต 4 พอร์ต
- Timer 16 บิต 2 ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- วงจรออสซิลเลเตอร์และวงจรถ่ายเก็บสัญญาณนาฬิกาบนชิพ
- พอร์ตอนุกรมสามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
- อ้างหน่วยความจำ โปรแกรมภายนอกได้ 64 กิโลไบต์
- อ้างหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์
- สามารถประมวลผลทีละบิต
- อ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 ไมโครวินาทีขณะทำงานด้วย Clock 12 เมกะเฮิร์ต
ตัวอย่างไมโครคอนโทรลเลอร์ตระกูล MCS-51 และลักษณะต่างๆ สามารถแสดงได้ ดัง
ตารางที่ 2.1 สำหรับไออะแกรมโครงสร้างภายในแสดงดังรูปที่ 2.12

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4K ROM	128 bytes	2
8031	-	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	3
8032	-	256 bytes	3
8752	8K EPROM	256 bytes	3

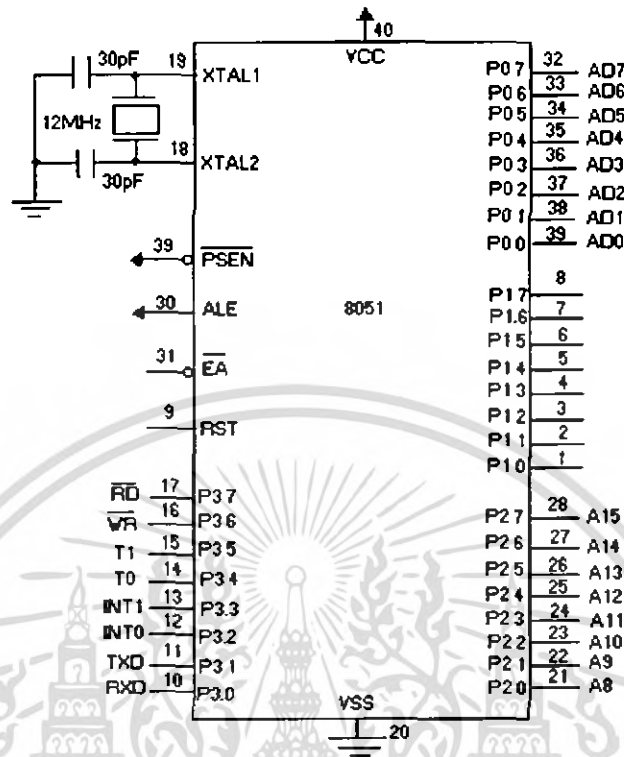
ตารางที่ 2.1 แสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ต่างๆ



รูปที่ 2.12 แสดงโครงสร้างภายในของ MCS-51

2.2.6 การจัดหาต่างๆ ของ MCS-51

ไอซี ไมโครคอนโทรลเลอร์ 8051 โครงสร้าง ไอซี เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่างๆ เป็นขาพอร์ทอินพุต , ขาพอร์ทเอาต์พุต , ขาสัญญาณควบคุม , ขาคำแหน่งหน่วยความจำข้อมูล ดังรูปที่ 2 13



รูปที่ 2.13 แสดงขาต่างๆของ 8051

ความหมายขาต่างๆ มีดังนี้

พอร์ต 0 (Port 0)

พอร์ต 0 ได้แก่ขา 32-39 สามารถเป็นพอร์ตอินพุทและเอาต์พุทได้ ซึ่งในการติดต่อกับหน่วยความจำภายนอกสามารถเป็นขาบัสดำแหน่ง และบัสข้อมูลได้

พอร์ต 1 (Port 1)

พอร์ต 1 ได้แก่ขา 1-8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้ เป็น P1 0, P1.1, etc

พอร์ต 2 (Port 2)

พอร์ต 2 ได้แก่ขา 21-28 สามารถทำงานได้ 2 หน้าที่ เป็นพอร์ต 8 บิต และเป็นขาแอดเดรส 8 บิต ในการอ้างหน่วยความจำภายนอก

พอร์ต 3 (Port 3)

พอร์ต 3 ได้แก่ขา 10-17 สามารถทำงานได้ 2 หน้าที่ เป็นพอร์ตอินพุทและเอาต์พุท และเป็นขาควบคุมต่างๆ ดังตารางที่ 2.2

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ทอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ทอนุกรม
P3.2	$\overline{\text{INT0}}$	อินเทอร์รัพท์ภายนอกหมายเลข 0
P3.3	$\overline{\text{INT1}}$	อินเทอร์รัพท์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา / ตัวนับ ตัวที่ 0
P3.5	T1	ตัวจับเวลา / ตัวนับ ตัวที่ 1
P3.6	$\overline{\text{WR}}$	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	$\overline{\text{RD}}$	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

ตารางที่ 2.2 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3

$\overline{\text{PSEN}}$ (Program Store Enable)

ขา $\overline{\text{PSEN}}$ ได้แก่ขา 29 เป็นขาที่ส่งสัญญาณออก ขานี้ทำการแอกทีฟเมื่อ MCS-51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็นอีพรอมขา $\overline{\text{PSEN}}$ ต่อกับขา Output Enable ($\overline{\text{OE}}$) ของอีพรอม

ALE (Address Latch Enable)

เนื่องจากพอร์ต 0 สามารถเป็นขาอ้างตำแหน่ง และขาข้อมูล MCS-51 ซึ่งมีขา ALE ได้แก่ขา 30 ขานี้ใช้ผสมสัญญาณบัสตำแหน่งของพอร์ต 0 ในการใช้งานระบบ MCS-51 นั้น ต้องมีอุปกรณ์มาต่อกับพอร์ต 0 ที่ทำหน้าที่ Latch สัญญาณบัสตำแหน่ง เมื่อ MCS-51 ติดต่อกับ หน่วยความจำภายนอก MCS-51 ส่งสัญญาณบัสตำแหน่งออกมาก่อนทาง พอร์ต 0 จากนั้นส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่าบัสตำแหน่งของพอร์ต 0 ไว้เพื่อใช้พอร์ต 0 เป็นบัสข้อมูลต่อไป

$\overline{\text{EA}}$ (External Access)

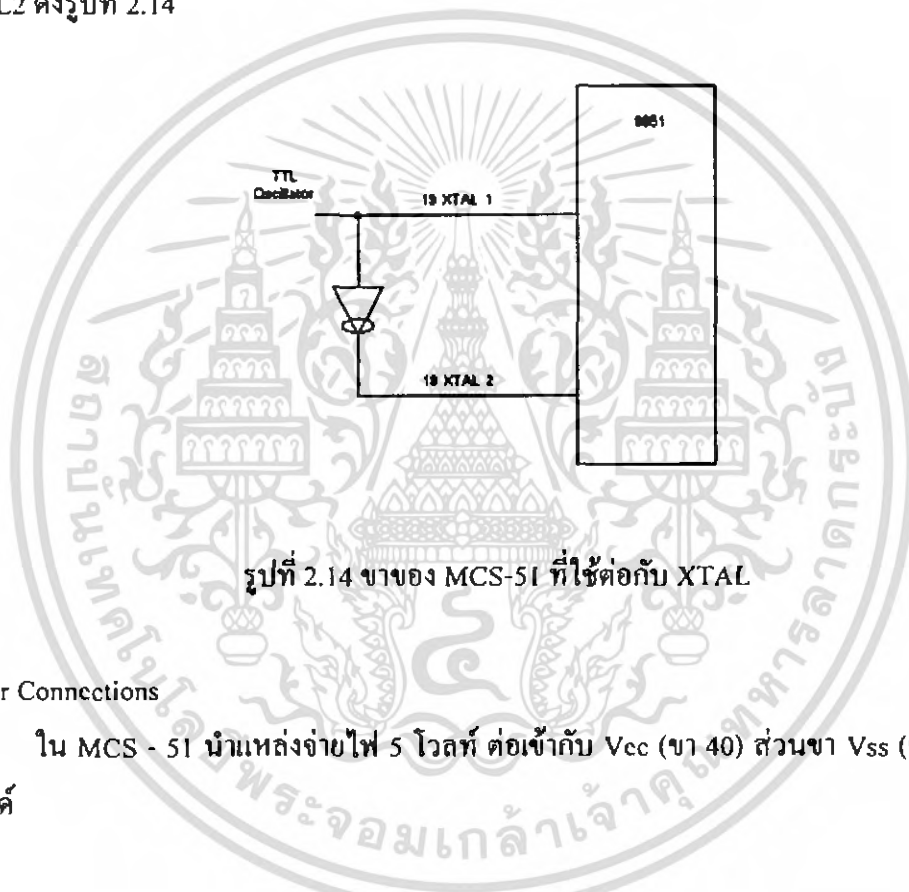
ขา $\overline{\text{EA}}$ ได้แก่ขา 31 ในสถานะเป็นลอจิก "1" ใช้กับเบอร์ 8051/8052 เพื่อให้อ่านโปรแกรมจากหน่วยความจำภายใน และในสถานะเป็นลอจิก "0" ทำให้ MCS-51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก (เมื่อ $\overline{\text{EA}}$ เป็น "0" ขา $\overline{\text{PSEN}}$ ทำการแอกทีฟ) หากเป็นเบอร์ 8031 หรือ 8032 ขา $\overline{\text{EA}}$ เป็น "0" เสมอ เพราะไม่มีโปรแกรมหน่วยความจำภายใน หากใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำภายในและให้ขา $\overline{\text{EA}}$ เป็น " 0 " ซึ่งหยุดการทำงานของรอมภายในและอ่านโปรแกรมจากอีพรอมภายนอกแทน

RST (Reset)

ขา RST ได้แก่ขา 9 เป็นขาเรเซต MCS-51 โดยให้เป็นสภาวะลอจิก "1" อย่างน้อย 2 Machine Cycles จึงสามารถรีเซตระบบได้

ความถี่สัญญาณนาฬิกาบนชิพ (ON-Chip Oscillator Inputs)

ได้แก่ขา 18-19 เป็นวงจรออสซิลเลเตอร์บนชิพ ซึ่งปกติทำการต่อคริสตัล ความถี่ 12 เมกกะเฮิร์ต กับตัวเก็บประจุ หรืออาจใช้สัญญาณนาฬิกาจากต้นกำเนิดชนิด ที่ที่แอลตอกับ XTAL1 และ XTAL2 ดังรูปที่ 2.14



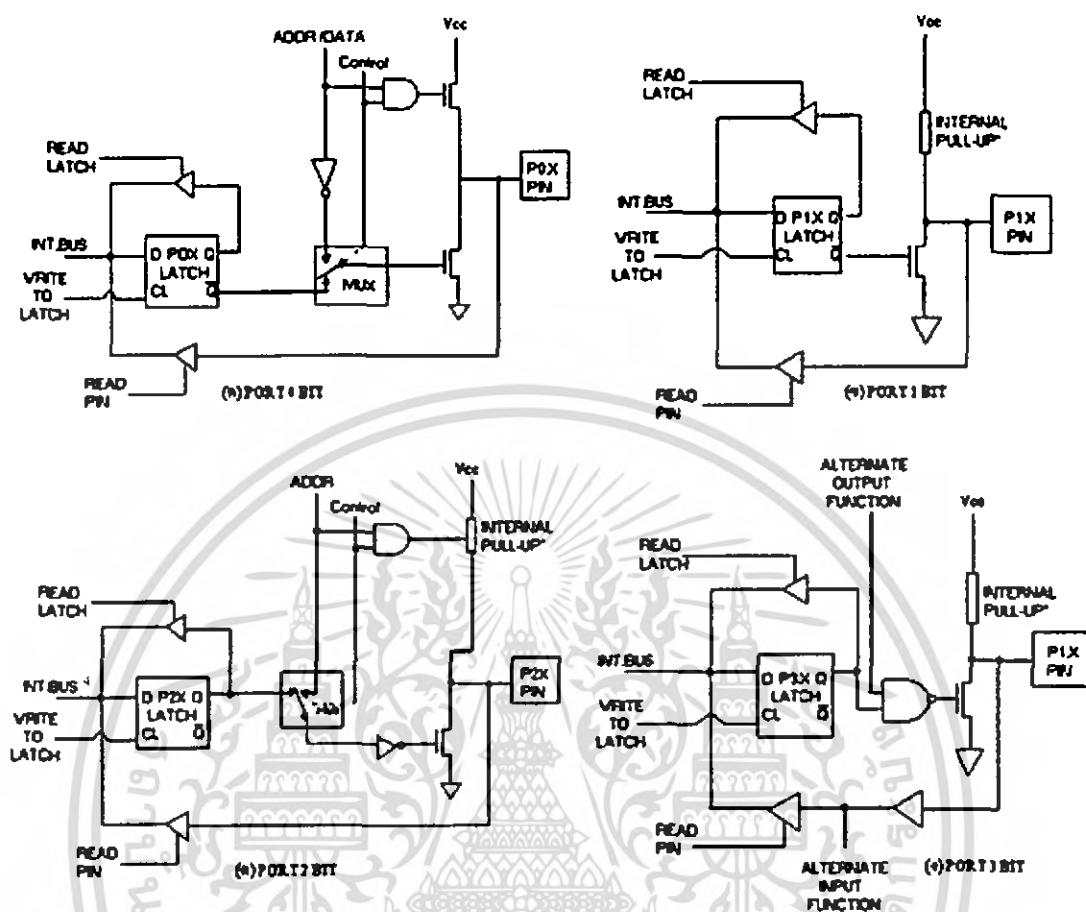
รูปที่ 2.14 ขาของ MCS-51 ที่ใช้ต่อกับ XTAL

Power Connections

ใน MCS - 51 นำแหล่งจ่ายไฟ 5 โวลท์ ต่อเข้ากับ Vcc (ขา 40) ส่วนขา Vss (ขา 20) ต่อกับกราวด์

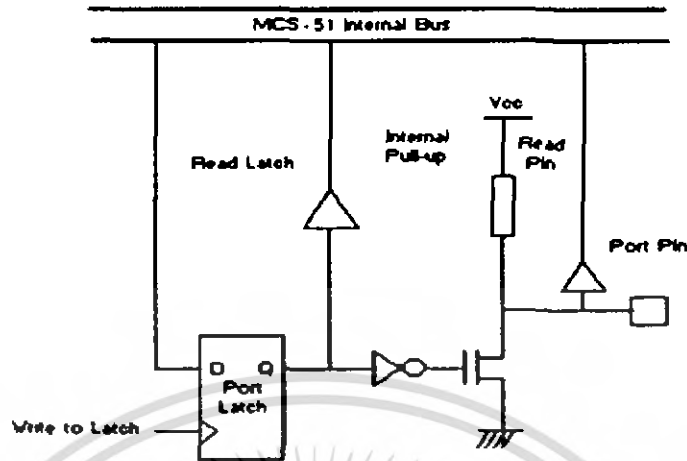
2.2.7 โครงสร้างของพอร์ตอินพุตและเอาต์พุต (I/O Port Structure)

ขาของพอร์ตแสดงโครงสร้างภายในได้ ดังรูปที่ 2.15 โดยมีโครงสร้างเป็น Field-Effect Transistor ต่ออยู่กับขาภายนอก และมีความต้านทานต่อ Pull-Up อยู่สำหรับพอร์ต 1, 2, 3 เมื่อเป็นพอร์ต 0 ไม่มีตัวต้านทาน Pull-Up ภายใน เพราะต้องใช้เป็นขาบัสดำแหน่ง และบัสดข้อมูล



รูปที่ 2.15 โครงสร้างพอร์ตทั้ง 4 ของ MCS-51

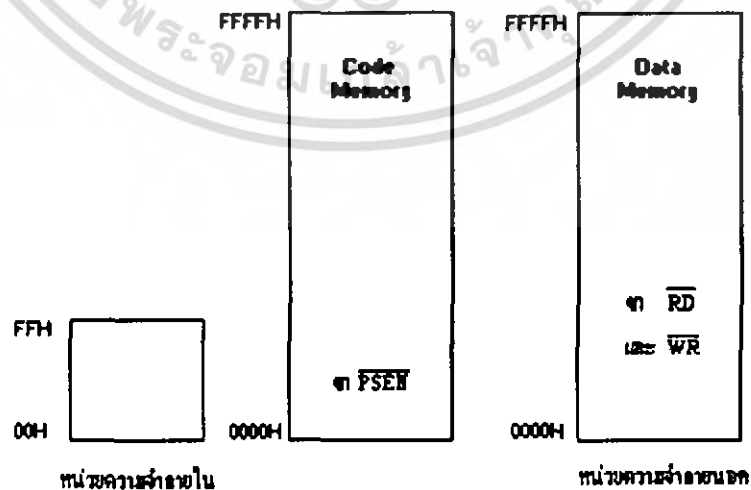
พอร์ตนี้สามารถเป็นอินพุตและเอาต์พุตกับอุปกรณ์ภายนอกได้ ในการอ่านข้อมูลจากพอร์ตอ่านได้ 2 แบบได้แก่ Read Latch และ Read Pin โดย Read Latch หมายถึงการอ่านข้อมูลที่ถูกลatch เอาไว้เข้าสู่บัสภายในของ MCS-51 เช่นการทำคำสั่ง CPL P1.5 ถ้าเป็นการ Read Pin ซึ่งเป็นการใช้พอร์ตเป็นอินพุต โดยอ่านค่าจากขาของไอซีเข้าสู่บัสภายใน โดยการอ่านแบบ Read Latch และ Read Pin มีสัญญาณมาควบคุมที่บัพเฟอร์ดังรูปที่ 2.16



รูปที่ 2.16 การต่อพอร์ทเข้ากับระบบบัสภายในของ MCS-51

2.2.8 โครงสร้างหน่วยความจำ

หน่วยความจำสำหรับ MCS-51 มี 2 ชนิดได้แก่ หน่วยความจำที่ใช้เก็บโปรแกรมรอม กับ หน่วยความจำที่ใช้เก็บข้อมูล ในการประมวลผลแรม MCS-51 บางเบอร์ เช่น 8051, 8052 มี หน่วยความจำภายในชิพ และ MCS-51 ทุกเบอร์สามารถอ้างหน่วยความจำ โปรแกรมภายนอกได้มากที่สุด 64 กิโลไบต์ สำหรับหน่วยความจำแรมภายใน ประกอบไปด้วยพื้นที่ใช้งานทั่วไป, รีจิสเตอร์ชุด, พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ โค้ดแรมของหน่วยความจำของ 8031 แสดงได้ดังรูปที่ 2.17



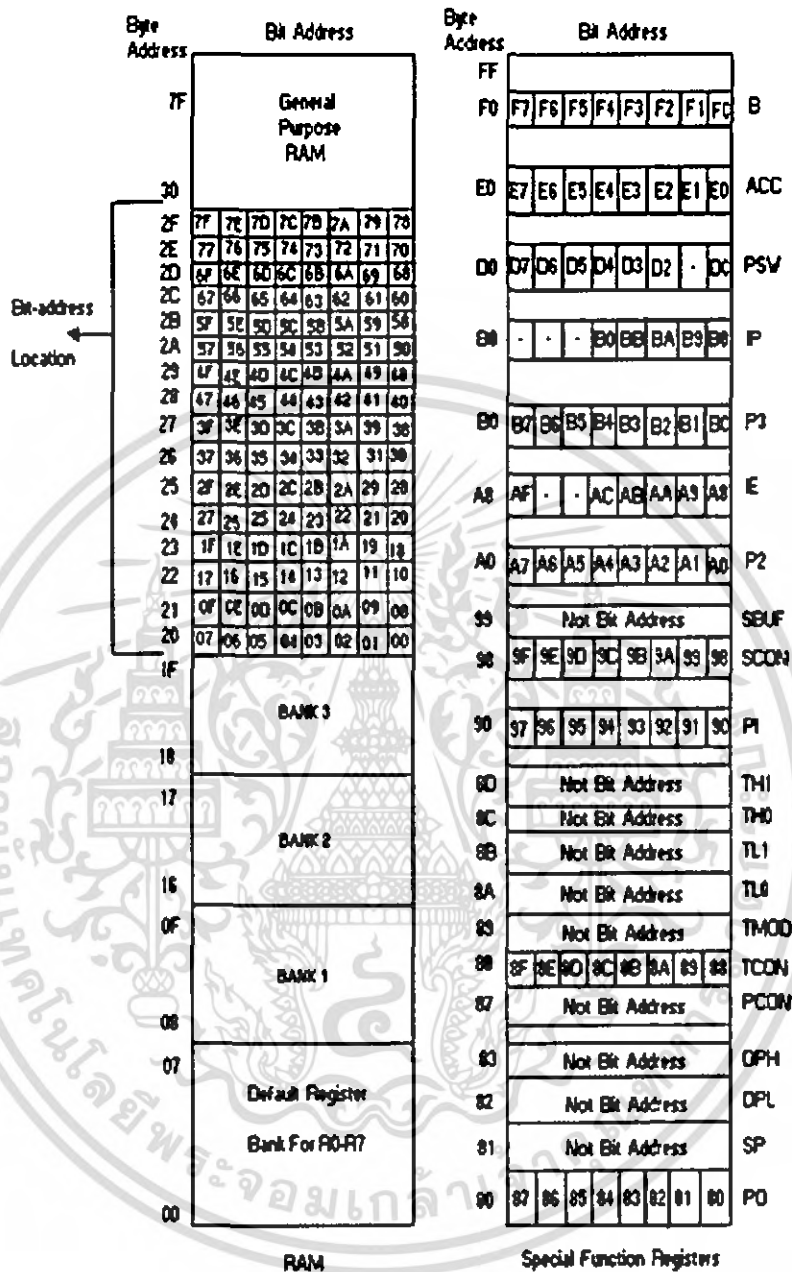
รูปที่ 2.17 การจัดหน่วยความจำของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทางวิชาการเพื่อการศึกษาเท่านั้น เมื่อนุญตเห็นหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน 8031 มีหน่วยความจำภายใน ตั้งแต่ตำแหน่ง 00H ถึง FFH และสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้ 64 กิโลไบต์ ตำแหน่งที่อ่านข้อมูลจากหน่วยความจำโปรแกรมของ PSEN ทำการแอสเซมบลี ซึ่ง 8031 สามารถอ้างหน่วยความจำภายนอกได้ 64 กิโลไบต์ ตำแหน่งโดยการติดต่อกับหน่วยความจำนี้ขา RD และ WR ทำการแอสเซมบลี สำหรับหน่วยความจำข้อมูลภายในสามารถแบ่งได้ดังนี้

- 1 ชุดรีจิสเตอร์ 4 ชุดแต่ละชุดเรียกว่ารีจิสเตอร์ชุด ที่ตำแหน่ง 00H ถึง 1FH โดยแต่ละชุดประกอบด้วยรีจิสเตอร์ R0 ถึง R7
2. หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ ตำแหน่ง 20H ถึง 2FH
- 3 หน่วยความจำใช้งานทั่วไปตำแหน่ง 30H ถึง 7FH
4. รีจิสเตอร์ฟังก์ชันพิเศษ ตำแหน่ง 80H ถึง FFH

แผนผังการจัดการหน่วยความจำข้อมูลภายใน แสดงได้ดังรูปที่ 2.18 จากแผนผังแสดงถึงการอ้างตำแหน่ง หน่วยความจำภายในซึ่งอ้างได้ 2 แบบ ได้แก่ การอ้างไปที่ตำแหน่งของไบต์ (เขียนหมายเลขด้านนอก) หรือการอ้างไปที่ตำแหน่งของบิต (เขียนหมายเลขตำแหน่งด้านใน) โดยตำแหน่งของหน่วยความจำ ที่อ้างเป็นแบบบิตที่แน่นอน



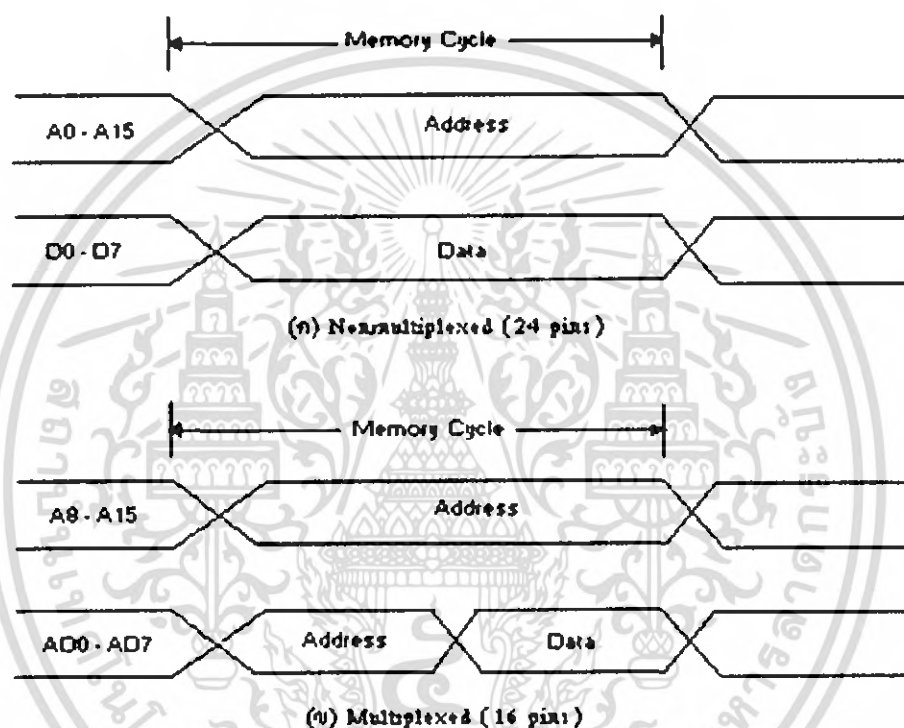
รูปที่ 2 18 ตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต

2.2.9 หน่วยความจำภายนอก

MCS-51 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์ MCS-51 ใช้พอร์ท 0 ในการอ้างตำแหน่งหน่วยความจำ 8 บิตล่าง และใช้พอร์ท 0 เป็นพอร์ทข้อมูลด้วย โดยใช้ขา ALE เป็น Latch ข้อมูลพอร์ท 0 และใช้พอร์ท 2 เป็นขาอ้างตำแหน่ง 8 บิตบน (รวมขาอ้างตำแหน่ง 16 เส้น ซึ่งอ้างได้ 64 กิโลไบต์) นอกจากนี้พอร์ท 0 ใช้งาน 2 หน้าที่ ในการติดต่อกับหน่วยความจำ ใช้วิธี

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้เพื่อใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาตจากทางผู้จัดทำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

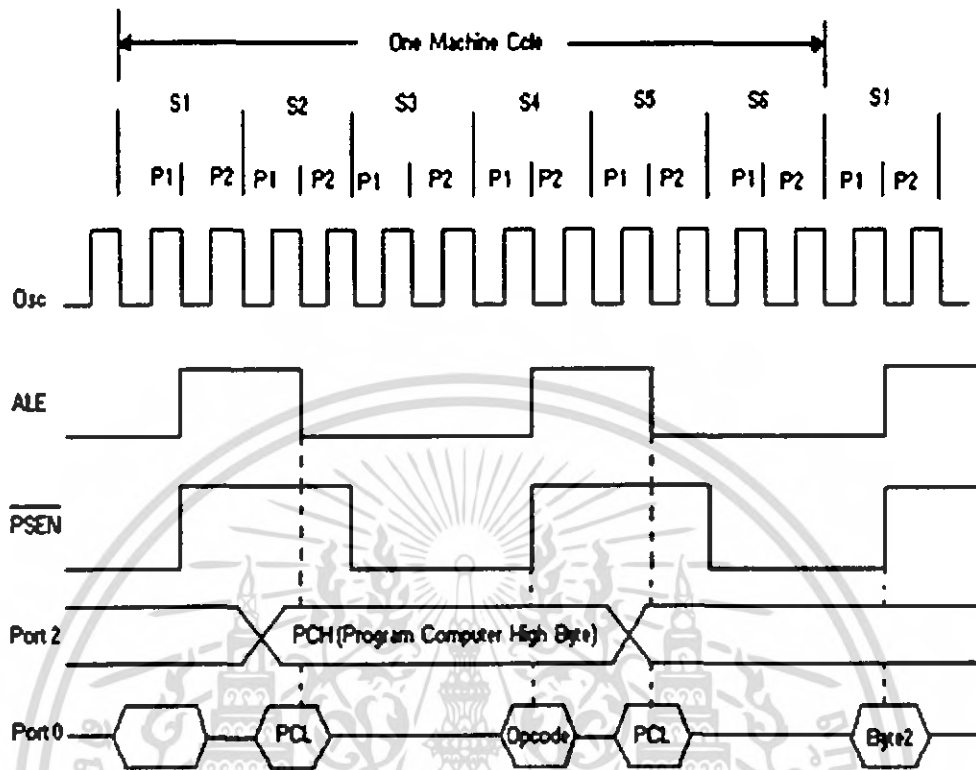
มัลติเพล็กซ์ระหว่างตำแหน่งกับข้อมูล ถ้าต้องการติดต่อกับหน่วยความจำที่เก็บข้อมูล 8 บิต และเก็บได้ 64 กิโลไบต์ ต้องใช้สายสัญญาณ 24 เส้นเป็นขาแอดเดรส 16 เส้น และขาข้อมูล 8 เส้น ดังรูปที่ 19 ถ้าใช้วิธีมัลติเพล็กซ์ ใช้ขา A0 - A7 เป็นขาข้อมูล D0- D7 ใช้สายสัญญาณเพียง 16 เส้น รูปที่ 2.19 เมื่อต้องการติดต่อกับหน่วยความจำ ทำการส่งสัญญาณแอดเดรส A0 -A15 ออก 16 เส้น จากนั้นขา A0-A7 เปลี่ยนเป็น D0- D7 ในการติดต่อกับหน่วยความจำภายนอก MCS - 51 ใช้วิธีนี้



รูปที่ 2.19 ไคอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล

2.2.10 การติดต่อกับหน่วยความจำโปรแกรมภายนอก

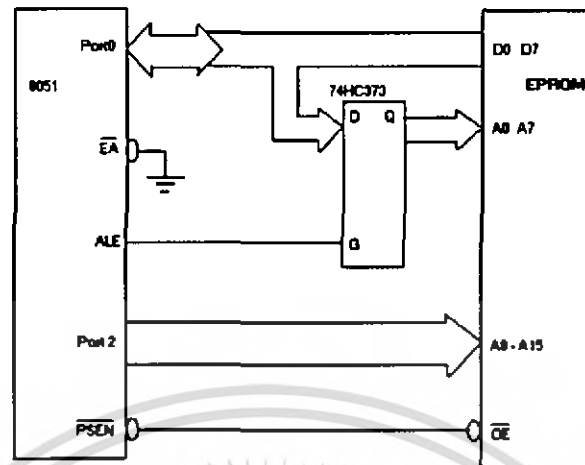
การอ่านข้อมูลที่หน่วยความจำโปรแกรมภายนอก MCS - 51 ส่งค่าตำแหน่งหน่วยความจำออกไป ซึ่งค่าตำแหน่งเก็บอยู่ใน PC ส่งออกทางพอร์ท 0 และพอร์ท 2 จากนั้นส่งขา ALE เป็นลอจิก "0" เพื่อ Latch ขาแอดเดรส ของ 8 บิตค่าคือพอร์ท 0 และส่งสัญญาณทางขา PSEN เป็นลอจิก " 0 " เพื่ออ่านข้อมูลซึ่งได้ Opcode เข้าไปทางขาบัสข้อมูล คือพอร์ท 0 ไคอะแกรมเวลาการอ่านข้อมูลจากหน่วยความจำภายนอก แสดงได้ดังรูปที่ 2 20



รูปที่ 2.20 ไคอะแกรมการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก

รูปที่ 20 ช่วงเวลาการทำงานของ MCS - 51 ใน S2 MCS - 51 ส่งค่าตำแหน่งหน่วยความจำโปรแกรม (ค่า PC) ทางพอร์ต 0 และ พอร์ต 2 จากนั้นส่งขา ALE เป็น " 0 " เพื่อ Latch อุปกรณ์ภายนอกค่าตำแหน่งไบต์ต่ำไว้ (พอร์ต 0) เพื่อใช้พอร์ต 0 เป็นขาข้อมูลต่อไป จากนั้นส่งขา PSEN เป็น " 0 " เพื่ออ่าน Opcode เข้าทางพอร์ต 0

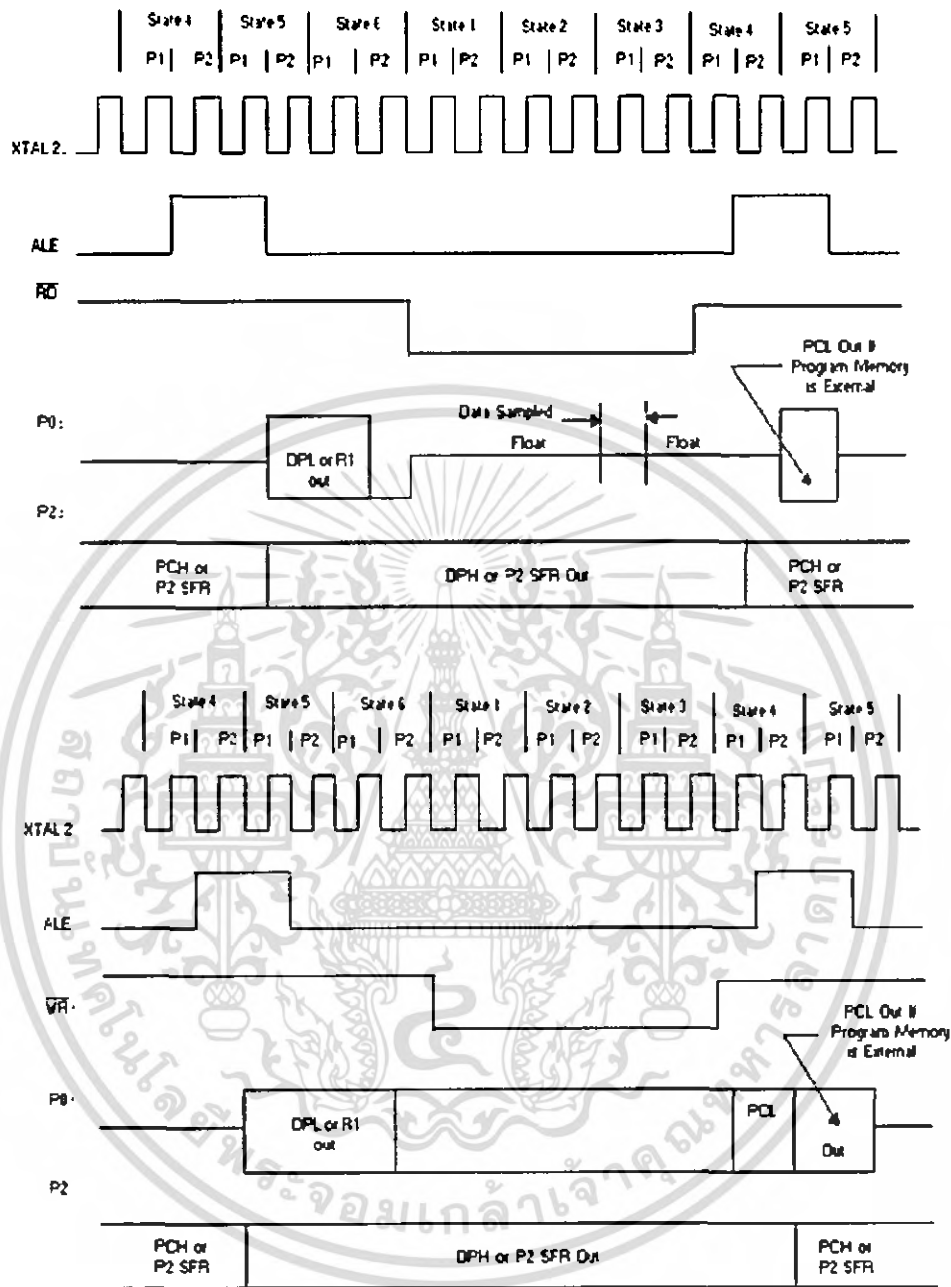
การต่อหน่วยความจำกับ MCS - 51 แสดงได้ดังรูปที่ 2.21 โดยขา \overline{EA} ต่อเป็น "0" เพื่อให้ MCS - 51 อ่านหน่วยความจำโปรแกรมภายนอกสำหรับการ มัลติเพล็กซ์ ใช้ฟลิปฟล็อป 8 ตัวเบอร์ 74373 เก็บค่าตำแหน่ง 8 บิตต่ำไว้ เมื่อ MCS - 51 ส่งค่าตำแหน่งพอร์ทออกไปจากนั้นส่งขา ALE ให้เป็น "0" ซึ่งใช้ขานี้ต่อกับ 74373 เพื่อ Latch ข้อมูลสำหรับขา \overline{PSEN} ต่อกับขา Output Enable (\overline{OE}) ของหน่วยความจำดังรูปที่ 2.21



รูปที่ 2.21 การต่อ MCS - 51 กับหน่วยความจำโปรแกรมภายนอก

2.2.11 การติดต่อกับหน่วยความจำข้อมูลภายนอก

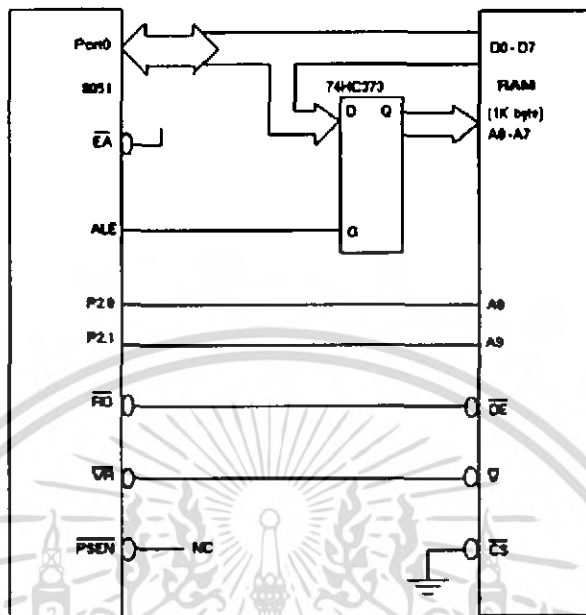
หน่วยความจำข้อมูลภายนอก MCS - 51 สามารถอ่าน และ เขียนได้ ในการติดต่อกับหน่วยความจำข้อมูลภายนอก MCS - 51 ส่งขาแอดเดรส ออกทางพอร์ท 0 และ พอร์ท 2 จากนั้นส่งขา ALE เพื่อไป Latch Address 8 บิตค่า โดยการอ่านเขียนข้อมูลนั้นใช้ขา \overline{RD} หรือ P3 7 และขา \overline{WR} หรือ P3 6 ตามลำดับ โค้ดแอมเวลการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอกแสดงดังรูปที่ 2.22



รูปที่ 2 22 ไตอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอก

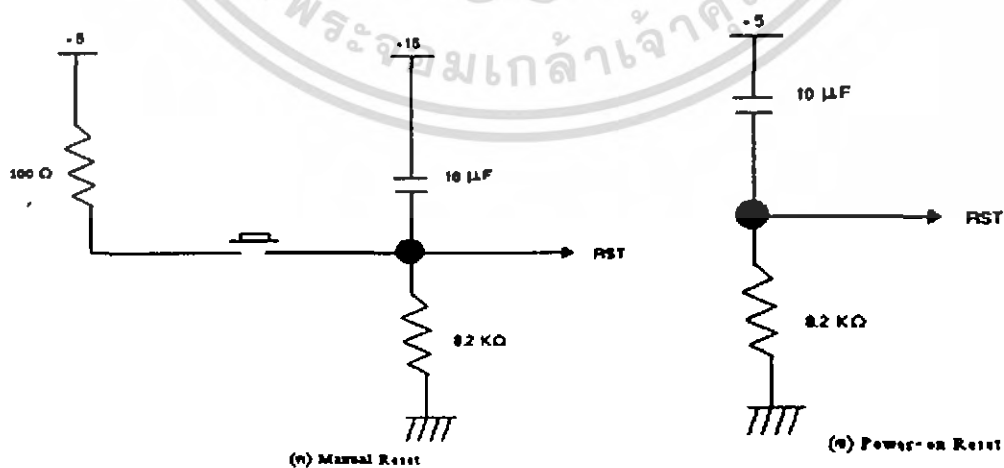
การเชื่อมต่อหน่วยความจำข้อมูลกับ MCS-51 ให้ 8051 ทำงานกับหน่วยความจำแสดงได้ดังรูปที่ 2.23 เป็นการเชื่อมต่อแรมขนาด 1 กิโลไบต์ ใช้ขา แอดเดรส เพียง 10 เส้น ดังนั้น A8 และ A9 ต่อกับ P2.0 และ P2.1 ส่วนขา \overline{EA} ต่อกับลอจิก "1" เพื่อให้อ่านโปรแกรมจากรวมภายใน และขา \overline{PSEN} ไม่ใช่เพราะไม่ได้ต่อรวมแสดงได้ดังรูปที่ 2 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 การต่อหน่วยความจำโปรแกรมกับ MCS-51

Reset Operation การรีเซ็ตหรือเริ่มต้นทำงานใหม่ของ MCS-51 ให้ลอจิก "1" ที่ขา RST เป็นเวลา 2 Machine Cycles (1 Machine Cycle เท่ากับ 12 Clock) จากนั้นให้กลับเป็นลอจิก "0" การรีเซ็ตอาจทำได้โดยใช้สวิทช์กด ดังรูปที่ 2.24 (ก) หรือใช้วิธี Power-up โดยใช้ตัว R-C ต่อเป็นวงจรดังรูปที่ 2.24



รูปที่ 2.24 การรีเซ็ต MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.12 การใช้งานพอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน

Serial Port Buffer (SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมโดยมีอยู่ 2 ตัว พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกทีละบิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานของทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม มีดังตารางที่ 2.3

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/2) ของ CPU Osc
0	1	1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1,2
1	0	2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1,2

ตารางที่ 2.3 โหมดการทำงานของพอร์ตสื่อสารอนุกรม

โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งข้อมูลเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อน ทางขา RxD เนื่องจากไม่มีการส่งบิตเริ่มต้น แต่จะส่ง Shift Clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Osc)

โหมด 1 : พอร์ตสื่อสารข้อมูลอนุกรม 10 บิต ข้อมูล 8 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ ไทเมอร์ 1,2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (1)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ ไทเมอร์ 2} \quad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2 · พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด (TB8 นิยมนำมาใช้ ส่งพาริตีบิต) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 หรือ 1/64 เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON

$$\text{บอดเรท (โหมด 2)} = \frac{(2^{\text{SMOD}}) \text{CPU Osc}}{64} \quad (3)$$

- บอดเรท (โหมด 2) = 1/32 CPU Osc เมื่อ SMOD = 1
- บอดเรท (โหมด 2) = 1/64 CPU Osc เมื่อ SMOD = 0

โหมด 3 พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ ไทเมอร์ 2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (4)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ ไทเมอร์ 2} \quad (5)$$

2.2.13 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51

การส่งข้อมูลออกทางพอร์ตอนุกรมของ MCS-51 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงใน SBUF ข้อมูลนี้ จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกไปครบถ้วนแล้วจะทำให้ค่าแฟลกซ์ TI เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไปต่อคือไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟลกซ์ TI มีค่าเป็น 1 ก่อน จะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่า REN (Receive Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบ และเมื่อบิตสุดท้ายเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และแฟลกซ์ RI ก็จะมีค่าเป็น 1 (ถูกเซต)

1. พอร์ตอนุกรม (โหมด 0)

การทำงานของพอร์ตอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยใช้เพียงขาสัญญาณ RxD เท่านั้น (ขาที่ใช้งาน 2 หน้าที่ใช้ส่งและรับข้อมูล) ส่วนขาสัญญาณ TxD จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะ การเลื่อนข้อมูลกับวงจร เลื่อนบิตภายนอก สำหรับอัตราเร็วจะถูกกำหนดไว้คงที่ที่ค่า $1/12$ เท่าของ CPU Osc จากรูปที่ 25 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆในโหมด 0 เมื่อมีการรับหรือส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดภายในตัว 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่งบิตเริ่มต้นและบิตสิ้นสุด ดังนั้นจึงจำเป็นต้องส่งสัญญาณ Shift clock ออกไป เพื่อใช้ Synchronize ระหว่างฝ่ายรับและฝ่ายส่ง โดยจะใช้ขา TxD ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา RxD และรับ Shift clock เข้าทางขา TxD ถ้า CPU Osc มีค่าเท่ากับ 12 MHz ก็จะส่งได้ถึง 1 ล้านบิต ซึ่งโหมด 0 เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด รายละเอียดผังเวลาในการรับส่งดังแสดงในรูปที่ 2.25

2. พอร์ตอนุกรม (โหมด 1)

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต ข้อมูลจำนวน 8 บิต และบิตสิ้นสุดอีก 1 บิต ดังแสดงในรูปที่ 2.26 โดยข้อมูลจะถูกส่งออกทาง TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออกจะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก ทางฝ่ายรับค่าของบิตสิ้นสุดจะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจิสเตอร์ SCON สำหรับอัตราเร็วในการส่งข้อมูลของโหมด 1 นั้น สามารถเลือกได้จาก ไทเมอร์ 1 ผังเวลาการทำงานแสดงดังในรูปที่ 2.26

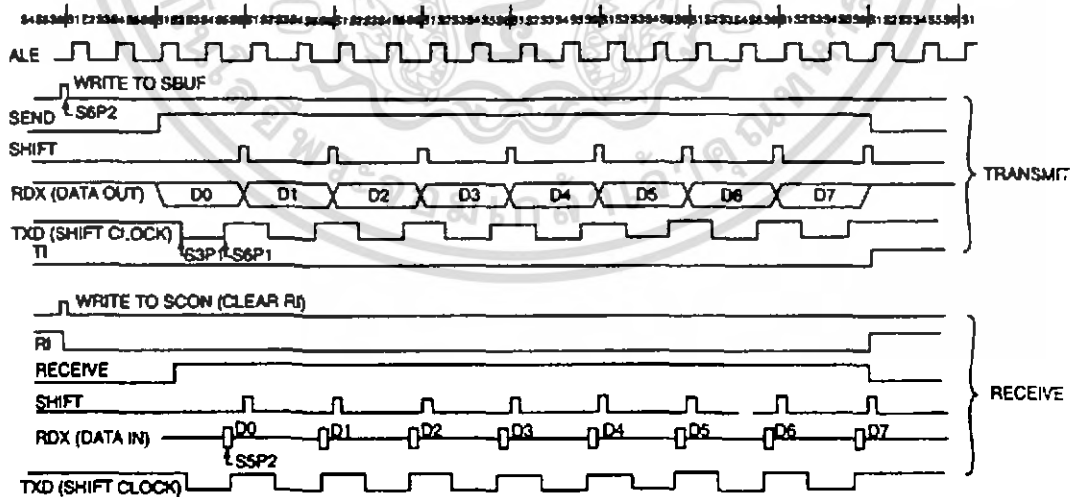
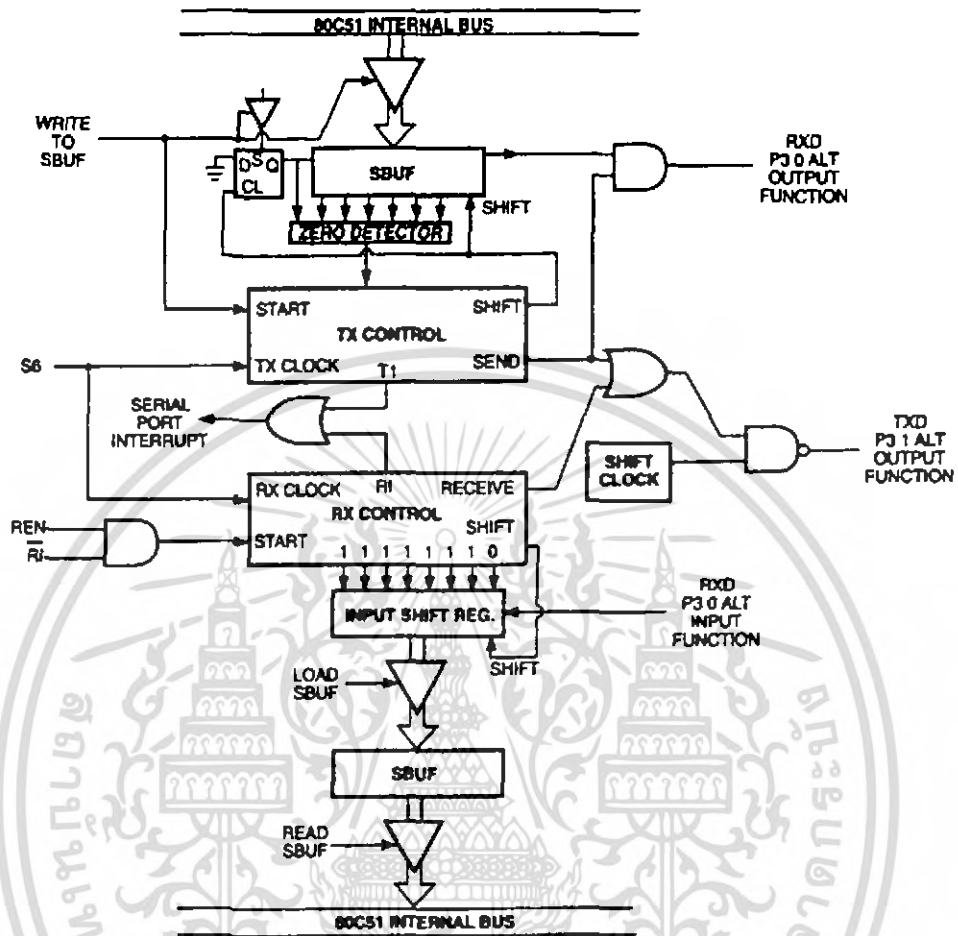
3. พอร์ตอนุกรม (โหมด 2)

โหมด 2 ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น บิตเริ่มต้น 9 บิตข้อมูล และบิตสิ้นสุด โดยบิตที่ 9 ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปใส่ไว้ในบิต TB8 ในรีจิสเตอร์ SCON ส่วนมากผู้ใช้นักจะนำบิตนี้มาใช้เป็นพาริตีบิต โดยรับค่ามาจากพาริตีแฟลกซ์ใน PSW ส่วนทางด้านรับ บิตที่ 9 จะถูกนำไปเก็บไว้ใน RB8 อัตราเร็วในการส่ง/รับข้อมูลกับ CPU Osc และค่า SMODซึ่งอยู่ในบิต 7 ใน SCON ผังเวลาการทำงานแสดงดังในรูปที่ 2.27

4. พอร์ตอนุกรม (โหมด 3)

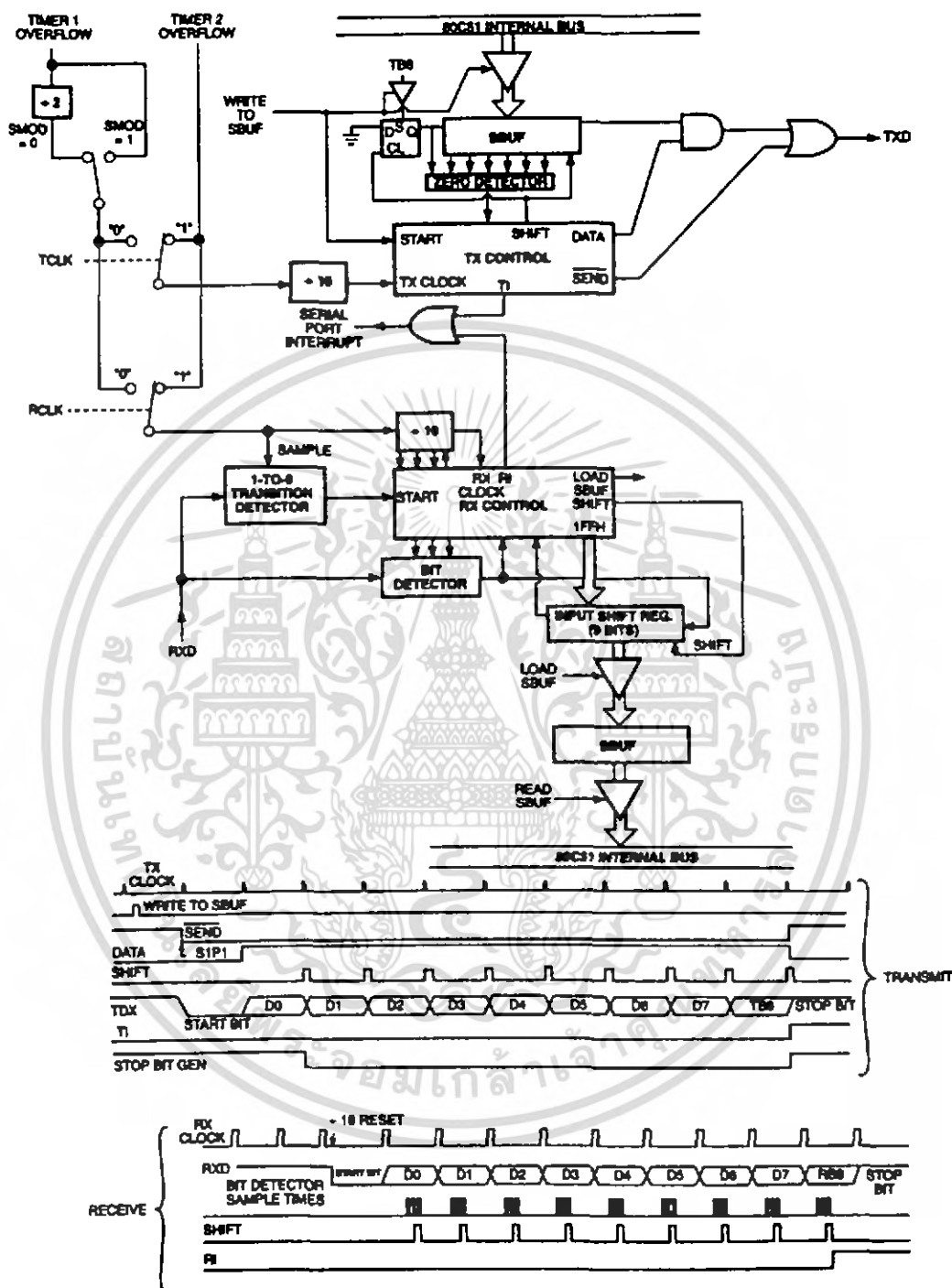
การทำงานเหมือนกับโหมด 2 ทุกอย่าง จะแตกต่างกันที่ความเร็วในการรับส่งข้อมูลจะขึ้นอยู่กับอัตราโอเวอร์โพล์วของไทมเมอร์ 1 หรือ ไทเมอร์ 2 โดยมีผังการทำงานแสดงดังในรูปที่

2.28



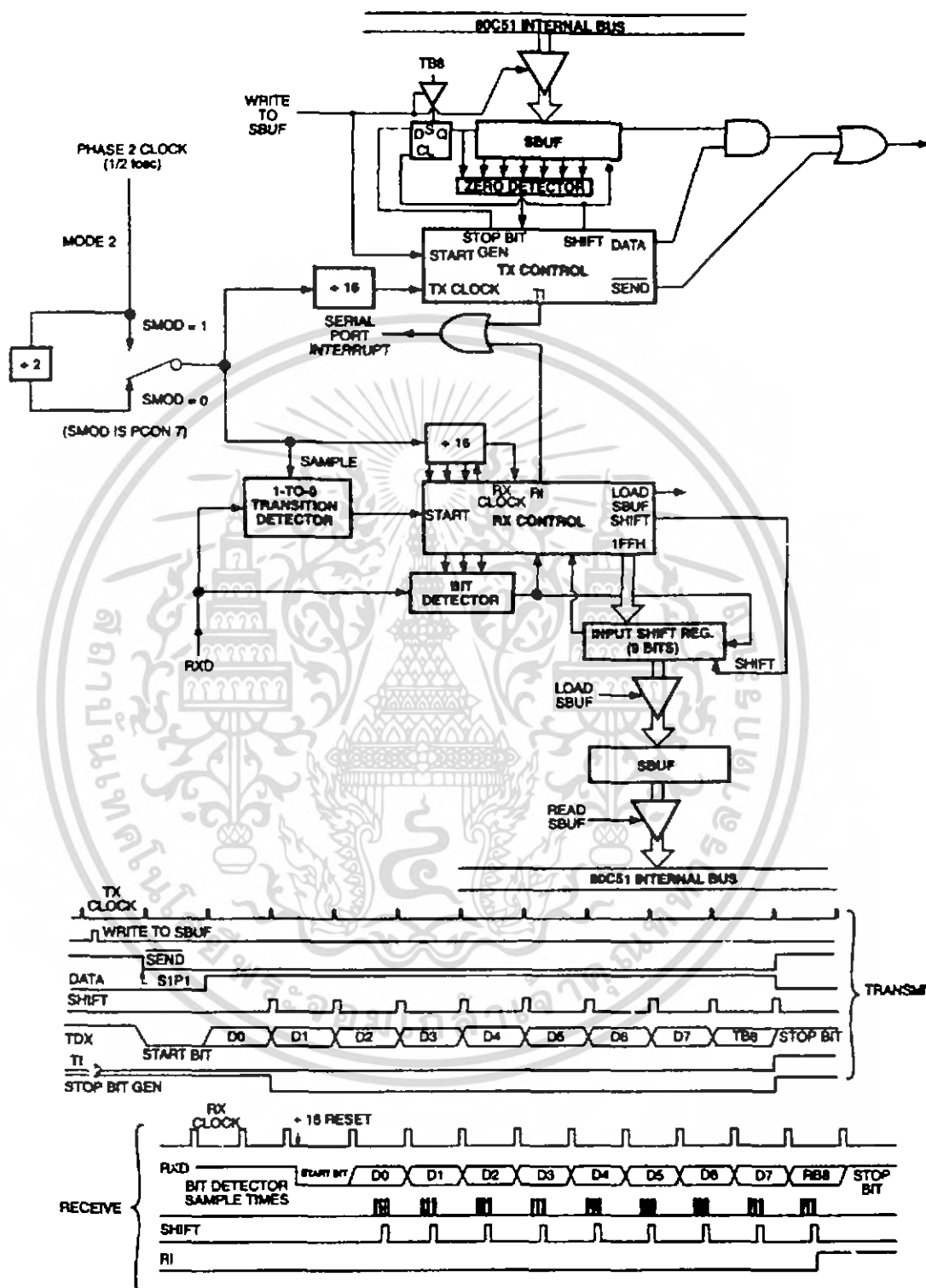
รูปที่ 2.25 ฟังก์ชันการทำงานโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



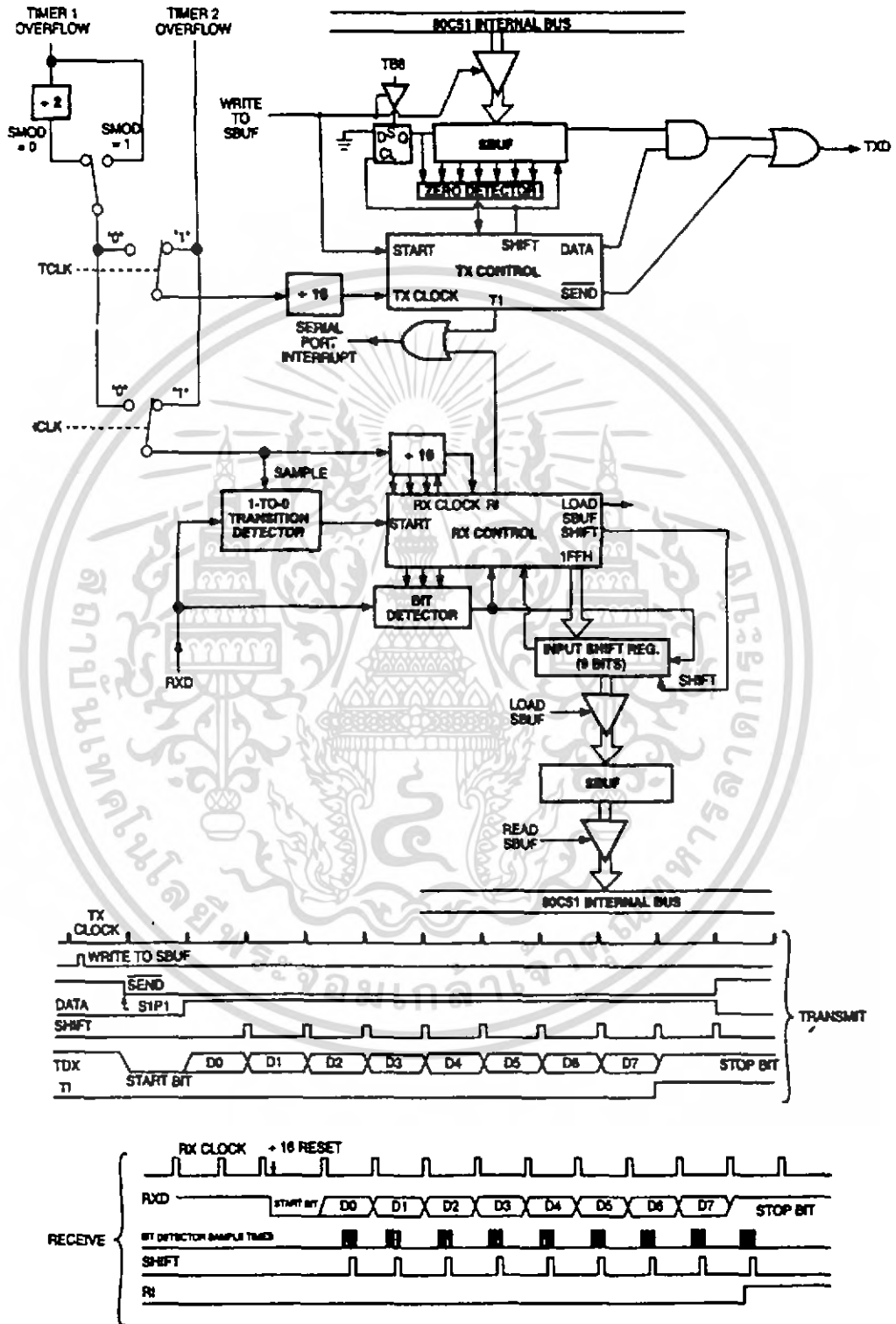
รูปที่ 2 26 ฟังก์การทำงานโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 ผังการทำงานโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

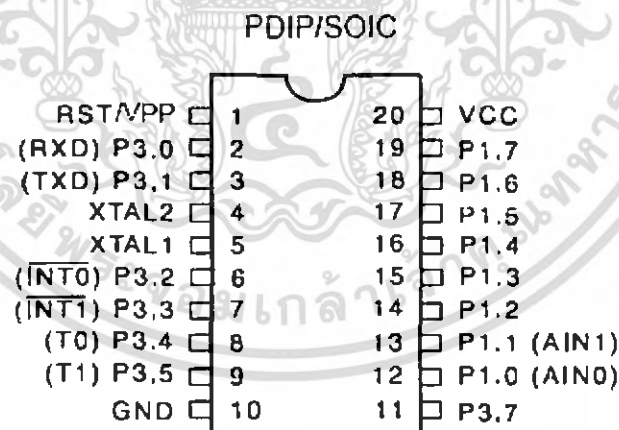


รูปที่ 2 28 ฟังก์ชันการทำงานโหมด 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.14 ข้อมูลและคุณลักษณะสำคัญของไอซี AT89C4051

1. มีหน่วยประมวลผล ขนาด 8 บิต Microcontroller with 4 kBytes FLASH Program Memory [ตัวถึง 20 ขา]
2. โปรแกรมเมโมรี่ จะมีอยู่ในชิพเป็นแบบชนิด Flash Memory AT89C4051=4K
3. แหล่งจ่ายไฟ ใช้ได้ตั้งแต่ = 2.7V ถึง 6.0V
4. ความถี่คริสตอลใช้ได้ถึง 24 MHz = วงจร Dongle ใช้คริสตอล 10 ถึง 15 Mhz
5. ล็อกบิตโปรแกรมเมโมรี่ได้ 2 ระดับ
6. หน่วยความจำของแรมภายใน AT89C4051 = 128 ไบท์
7. มี 15 อินพุตเอาต์พุตพอร์ต
8. มี 16 บิตไทม์เมอร์เคาท์เตอร์ 2 ตัว
9. มี 6 อินเทอร์รับเซอร์ท
10. ต่อ LED ได้ตรงโดยไม่ต้องใช้วงจรช่วย
11. มีวงจรคอมพาราเตอร์เปรียบเทียบแรงสัญญาณอะนาล็อก
12. ระบบประหยัดพลังงาน 2 ระดับ Idle และ Power Down
13. Brown-Out Detection



รูปที่ 2.29 แสดงขาต่างๆ ของไอซี AT89C4051

การใช้งานอินพุต เอาท์พุตพอร์ตของ IC นั้นจะแบ่งพอร์ตออกเป็น 2 ชุดคือ P1 และ P3 พอร์ต P1 มีขนาด 8 บิต โดยที่ขา P1.2 ถึง P1.7 Port I Drives สามารถจะเป็นได้ทั้ง อินพุต และ เอาท์พุต มีอาร์พูลอ์อยู่ภายในแล้ว แต่ P1.0 และ P1.1 ต้องการอาร์พูลอ์ภายนอก และถ้าจะใช้

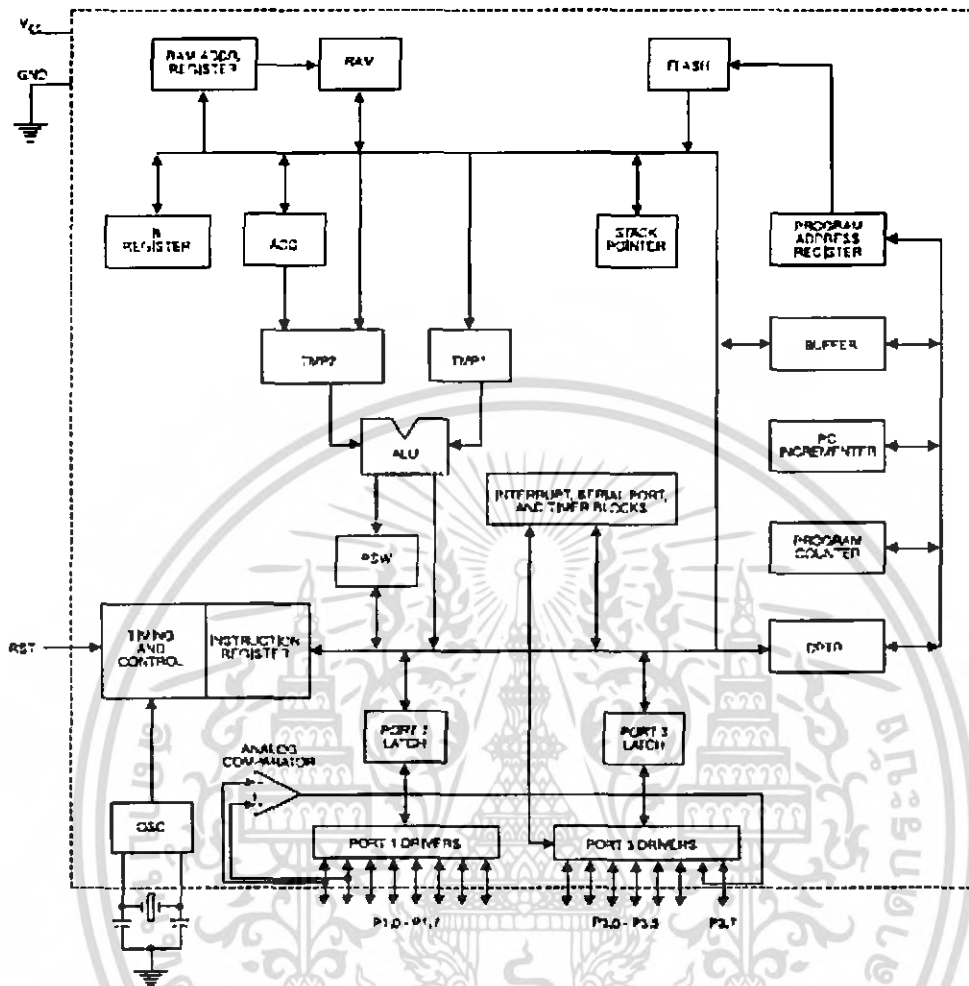
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P1.0 (AIN0) และ P1.1 (AIN1) ที่เป็นอนาล็อก คอมพาราเตอร์ P1.0 จะเป็นขั้วบวก P1.1 จะเป็นขั้วลบ เอาต์พุตจะออกที่พอร์ต P3.6 เช่น แรงดัน P1.0=2V, P1.1=1V จะทำให้ P3.6 เป็น " 1 " การไครฟ์กระแสพอร์ต P1 สามารถซิงค์กระแสได้ในแต่ละขาได้สูงสุด 20mA แต่กระแสรวมทุกพอร์ตจะต้องไม่เกิน 80mA

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

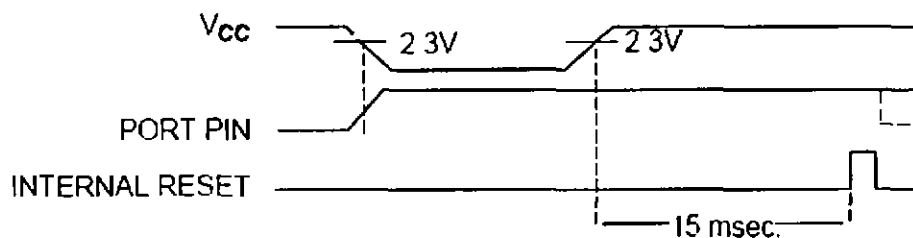
ตารางที่ 2.4 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3

ทางด้าน Port 3 Drives หรือพอร์ต P3 ที่ขา P3.0 ถึง P3.5 และ P3.7 นั้นก็สามารถเป็นได้ทั้งอินพุต และเอาต์พุต และมีอาร์พูล์พอย์ภายในแล้ว ส่วนที่ขา P3.6 นั้นเป็นเอาต์พุต ของวงจรถอมพาราเตอร์ จากขา P1.0, P1.1 การซิงค์กระแสนั้นมีเงื่อนไขเดียวกับพอร์ต P1 พอร์ต P3 ยังมีหน้าที่พิเศษอีกคือ



รูปที่ 2.31 แสดงโครงสร้างภายในของ AT89C4051

Brown-Out Detection ของ AT89C4051 นั้นมีการตรวจสอบแรงดันของแหล่งจ่าย ถ้าแรงดันมีค่าต่ำกว่า 2.3 V CPU จะทำให้พอร์ตทั้งหมด เป็นลอจิก "1" เมื่อแรงดันแหล่งจ่าย กลับมาสู่ 2.3 V +/- 10% จะทำการหน่วงเวลา 15mS แล้วทำการรีเซ็ต CPU

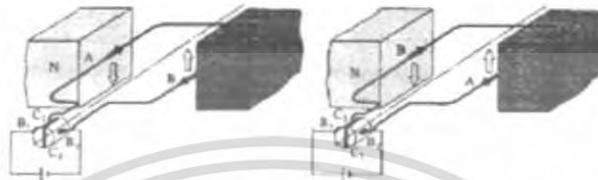


รูปที่ 2.32 โค้ดอะแกรมสัญญาณ Brown-Out Detection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มอเตอร์กระแสตรง (DC MOTOR)

มอเตอร์กระแสตรงจะมีหลักการทำงาน โดยวิธีการผ่านกระแสให้กับขดลวดในสนามแม่เหล็ก ซึ่งจะทำให้เกิดแรงแม่เหล็ก โดยส่วนของแรงนี้จะขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก



รูปที่ 2.33 แสดง โครงสร้างทั่วไปของมอเตอร์กระแสตรง

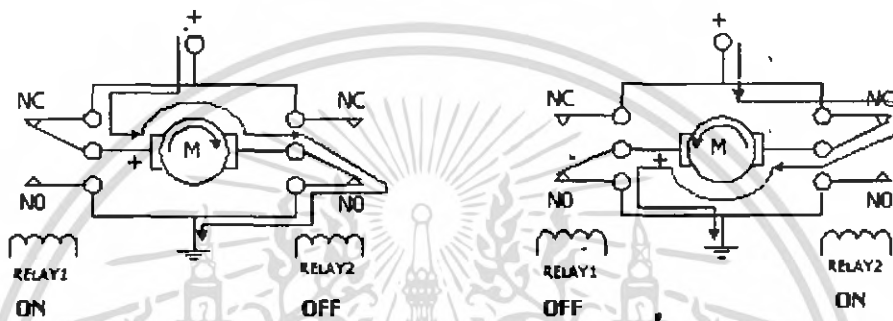
จากรูปที่ 2.29 ทางเดินของของฟลักซ์แม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเพอร์ไรต์ 2 ชิ้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถัง ได้พอดี เพื่อที่จะ ให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ได้ ดังนั้นความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กด้วย ซึ่งส่งผลให้ฟลักซ์แม่เหล็กวิ่ง ไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับขั้ว โรเตอร์ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวร จึงเกิดเป็นแรงบิดเพื่อที่จะหมุนขั้ว โรเตอร์ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่าน ไปยังขั้ว โรเตอร์ โดยผ่านแปรงถ่าน ซึ่งจะสัมผัสกับแหวนตัวนำในขั้ว โรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวดนั่นเอง

2.3.1 การขับและกลับทิศทางของมอเตอร์กระแสตรง (DC MOTOR)

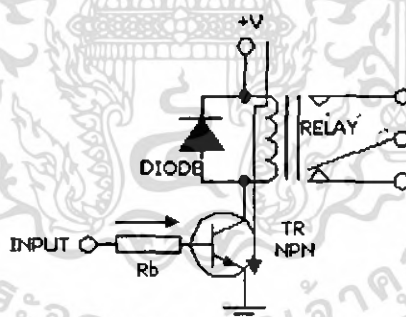
ในการใช้ไอซีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจร ที่เรียกว่าวงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางของมอเตอร์นั้น สามารถที่จะใช้รีเลย์ต่อวงจร สวิตช์เพื่อกลับทิศทางของขั้วไฟกระแสตรง หรืออาจใช้อุปกรณ์สารกึ่งตัวนำที่เป็นวงจรขับกำลังเช่น ทรานซิสเตอร์ มอสเฟต แล้วแต่วิธีที่เราจะเลือกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นการใช้รีเลย์ควบคุมการเปลี่ยนทิศทางการหมุนของมอเตอร์ โดยการควบคุมการเปิด - ปิดที่รีเลย์ 2 ตัว ซึ่งจะทำหน้าที่กลับทิศทางของขั้วไฟที่ป้อนให้กับมอเตอร์ โดยการสลับการทำงานของรีเลย์ เช่น ให้รีเลย์ตัวที่ 1 ทำงาน (ON) และรีเลย์ตัวที่ 2 หยุดทำงาน (OFF) จะทำให้มอเตอร์หมุนไปทางซ้าย และในทำนองเดียวกันถ้าหากรีเลย์ตัวที่ 1 หยุดทำงาน (OFF) และรีเลย์ตัวที่ 2 ทำงาน (ON) ก็จะทำให้มอเตอร์หมุนไปทางขวา



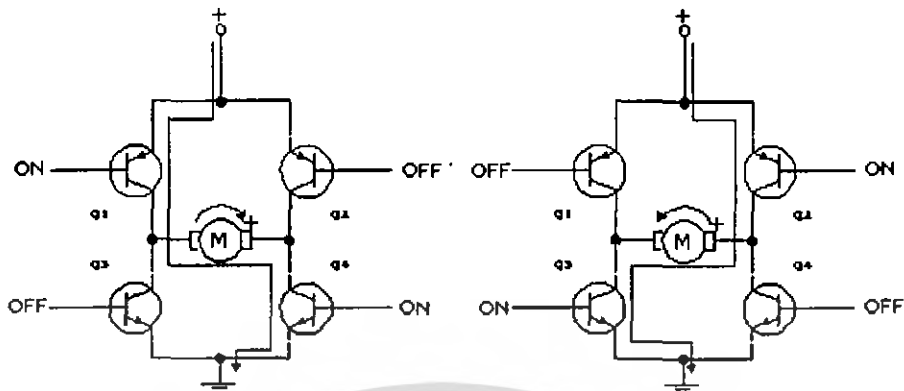
รูปที่ 2.34 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์



รูปที่ 2.35 แสดงการใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน

จากรูปเป็นวงจรขับรีเลย์โดยใช้ทรานซิสเตอร์ทำหน้าที่ขยายกระแส ด้วยเหตุผลเพราะไม่สามารถจะใช้ขา เอาต์พุตของไมโครคอนโทรลเลอร์ป้อนกระแสไฟที่ขดลวดของรีเลย์โดยตรงได้ เนื่องจากว่ากระแสที่จ่ายออกมาจากขา เอาต์พุตของไมโครคอนโทรลเลอร์มีค่าน้อยเกินไป ดังนั้นเราจึงต้องมีส่วนของวงจรถานซิสเตอร์เพื่อที่จะทำการขยายกระแสให้เพียงพอในการป้อนให้กับขดลวดของรีเลย์ ส่วนไดโอดนำมาต่อไว้สำหรับป้องกันแรงดันย้อนกลับที่เกิดจากการเหนี่ยวนำของสนามแม่เหล็กในขณะที่เกิดการขุดตัว ซึ่งอาจจะทำให้ทรานซิสเตอร์เสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 แสดงการใช้ทรานซิสเตอร์เป็นวงจรถับและกำหนดทิศทางของมอเตอร์กระแสตรง

จากรูปเป็นวงจรถับรีเวิร์มที่ประกอบด้วยทรานซิสเตอร์กำลัง 4 ตัวที่ทำหน้าที่ขับ และควบคุมทิศทางการหมุนของมอเตอร์ ถ้าหากกำหนดให้ทรานซิสเตอร์ Q1 และ Q4 อยู่ในสถานะทำงาน (Active) กระแสไฟฟ้าจะไหลผ่านทรานซิสเตอร์จากซ้ายไปขวา โดยผ่านมอเตอร์ กระแสตรงทำให้มอเตอร์หมุนไปทางขวา ในทำนองเดียวกันถ้าหากเราทำให้ทรานซิสเตอร์ Q2 และ Q3 อยู่ในสถานะทำงาน (Active) กระแสไฟฟ้าก็จะไหลจากทางขวาไปทางซ้ายซึ่งจะส่งผลให้มอเตอร์กลับทิศทางการหมุนจากทางขวาไปทางซ้าย

2.3.2 การควบคุมความเร็วของมอเตอร์กระแสตรง

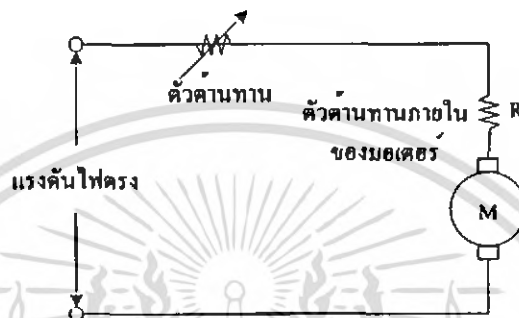
การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุม โดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุม โดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

2.3.3 การควบคุมความเร็วของมอเตอร์ขั้นพื้นฐาน

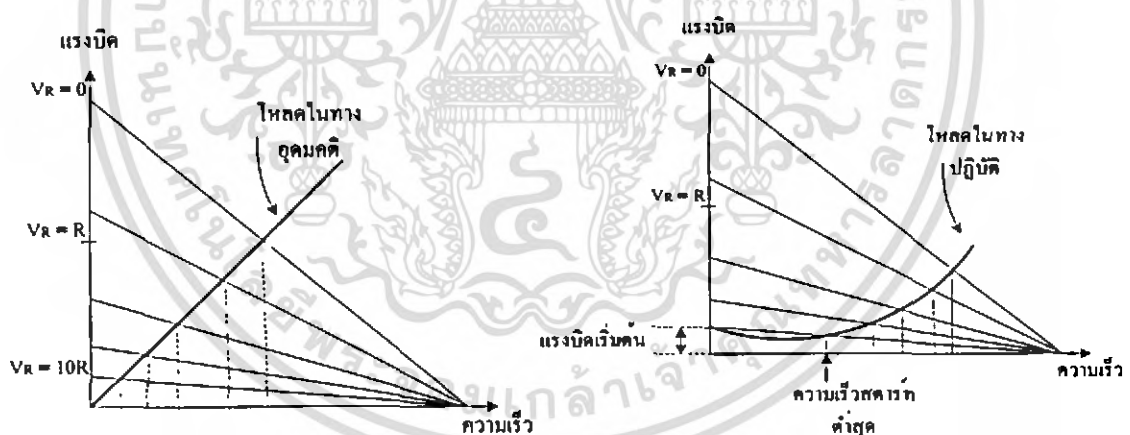
1. การควบคุมด้วยตัวต้านทานที่ปรับค่าได้ เป็นรูปแบบพื้นฐานที่สุดของการควบคุมมอเตอร์คือ ใช้ตัวต้านทานปรับค่าได้อนุกรมกับมอเตอร์ โดยตัวต้านทานที่ปรับค่าได้จะเป็น

ตัวกำหนดความเร็วในการหมุนของมอเตอร์ การบังคับแบบนี้ไม่มีประสิทธิภาพเพราะกำลังไฟเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์การค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูญเสียไปในตัวความต้านทาน มักนิยมใช้กับมอเตอร์ตัวเล็กๆ การบังคับแบบนี้ให้คุณสมบัติการสตาร์ทดี (ให้แรงบิดสูงที่ความเร็วต่ำ) แต่จะให้ความเร็วสูงมากเมื่อมอเตอร์อยู่ในภาวะที่มีโหลดน้อยๆ ดังนั้นการบังคับแบบนี้มีประโยชน์เฉพาะภาวะที่แรงต้านคงที่ เช่น การบังคับความเร็วของเครื่องจักรเย็บผ้า เป็นต้น



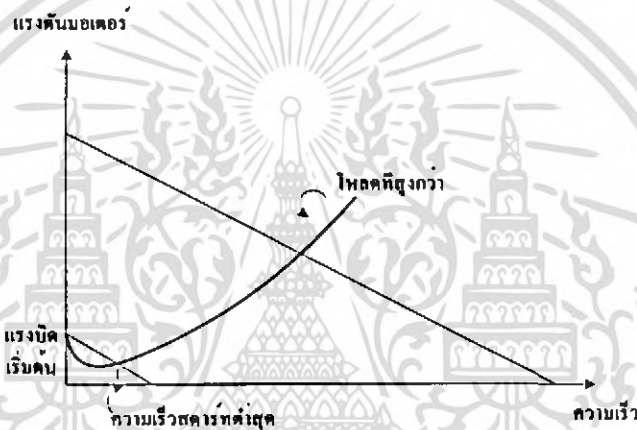
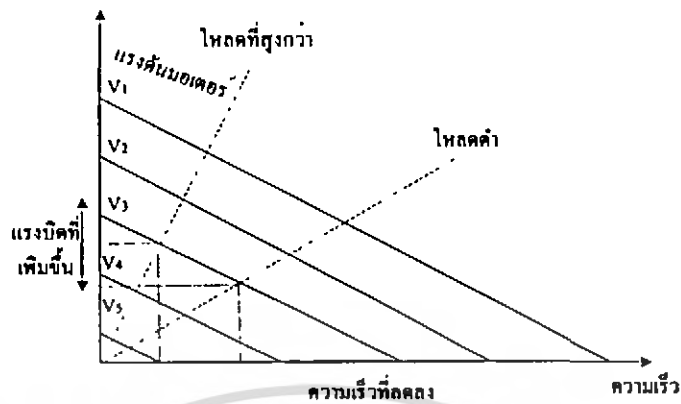
รูปที่ 2 37 วงจรควบคุมความเร็วของมอเตอร์กระแสตรงแบบใช้ตัวต้านทานอนุกรม



รูปที่ 2 38 กราฟแสดงคุณสมบัติการควบคุมด้วยตัวต้านทานที่ปรับค่าได้

2 การควบคุมด้วยวิธีเปลี่ยนค่าแรงดัน วิธีการนี้ดีกว่าวิธีการแรกแต่จะซับซ้อนกว่าต้องใช้อุปกรณ์อิเล็กทรอนิกส์ที่อัตราขยายกำลังสูง และ มอเตอร์จะถูกป้อนด้วยแรงดันที่เปลี่ยนแปลงค่าได้ จากแหล่งจ่ายที่มีอิมพีแดนซ์ต่ำ ข้อดีของการควบคุมวิธีนี้คือ ถ้าความเร็วลดลงจากผลของแรงบิด แรงดันที่ป้อนให้กับมอเตอร์จะเพิ่มขึ้นเพื่อรักษาระดับความเร็ว ส่วนข้อเสียจากการควบคุมวิธีนี้คือ เมื่อมอเตอร์มีความเร็วต่ำแรงดันที่ป้อนให้กับมอเตอร์จะมีค่าต่ำเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 การควบคุมความเร็วโดยเปลี่ยนค่าแรงดัน

3. การควบคุมแบบ PWM (Pulse Width Modulation) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์ โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนที่สัดส่วน และความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มี การเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดีวตี้ไซเคิล (Duty Cycle) นั่นเอง ซึ่งค่าของดีวตี้ไซเคิล คือช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดีวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึงใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูป 33 และในทำนองเดียวกันถ้าหากค่าดีวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดีวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่า จะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดีวตี้ไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ค่าความถี่ไซเคิล} = (\text{ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ}) \times 100\% \quad (6)$$

การทำงานของสัญญาณ PWM

รูปที่ 1 แสดงสัญญาณ PWM ที่แตกต่างกัน 3 สัญญาณ

- โดย 36a แสดงสัญญาณ PWM ที่ 10% duty cycle คือ สัญญาณในการอนจะเป็น 10% ของคาบสัญญาณ และ จะออฟเป็น 90% ของคาบสัญญาณ
- โดย 36b แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณในการอนจะเป็น 50% ของคาบสัญญาณ และ จะออฟเป็น 50% ของคาบสัญญาณ
- โดย 36c แสดงสัญญาณ PWM ที่ 90% duty cycle คือ สัญญาณในการอนจะเป็น 90% ของคาบสัญญาณ และ จะออฟเป็น 10% ของคาบสัญญาณ

เช่น ถ้า Power Supply มี 9V และ duty cycle เป็น 10% จะได้อาท์พุท 0.9V



รูปที่ 2.40a



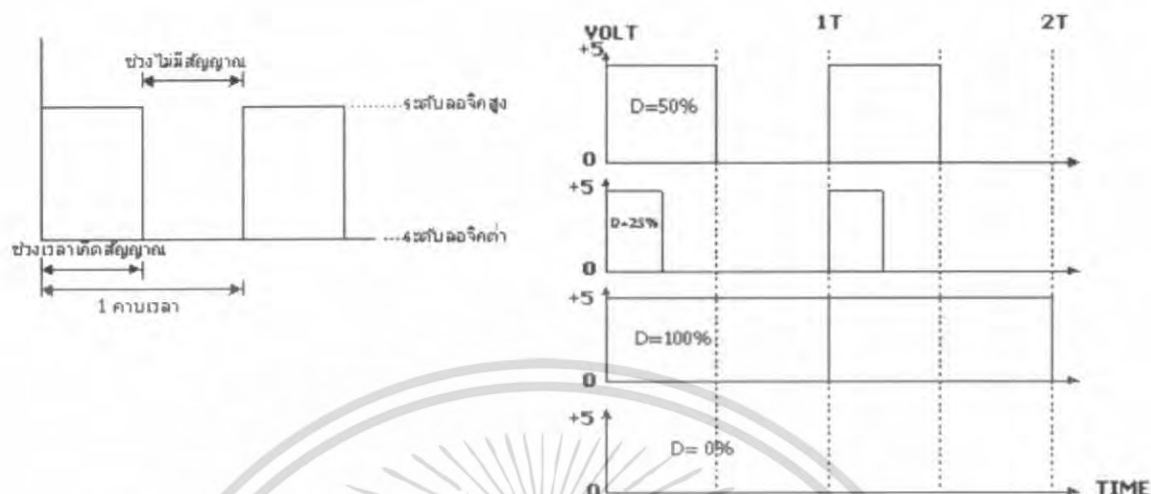
รูปที่ 2.40b



รูปที่ 2.40c

รูปที่ 2.40 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycles ที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 แสดงความกว้างของพัลส์ขนาดต่างๆ และค่าคิวตี้ไซเคิล ของช่วงพัลส์ที่มีความถี่คงที่

2.4 ทรานซิสเตอร์สนามไฟฟ้า

ทรานซิสเตอร์สนามไฟฟ้า (Field-Effect Transistor หรือ FET) คือ ทรานซิสเตอร์ที่ใช้สนามไฟฟ้าในการเปลี่ยนแปลงสภาพของสารกึ่งตัวนำเพื่อให้เกิดการนำกระแสเมื่อได้รับแรงดันไฟฟ้าที่เหมาะสมโดยทั่วไปมักเรียกชื่อย่อว่า "เฟต"

เฟตแบ่งออกเป็น 2 กลุ่ม คือ ทรานซิสเตอร์สนามไฟฟ้าแบบรอยต่อ (Junction Field Effect Transistor) หรือ เจเฟต (JFET) และ ทรานซิสเตอร์สนามไฟฟ้าแบบโลหะ-ออกไซด์-สารกึ่งตัวนำ (Metal-Oxide-Semiconductor Field Effect Transistor) หรือ มอสเฟต (MOSFET)

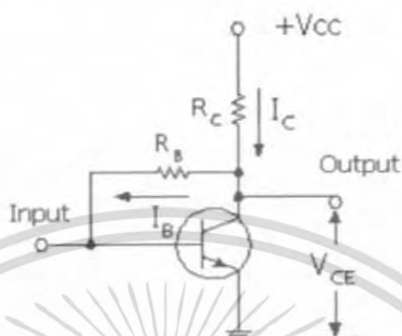
2.4.1 ชนิดของมอสเฟต

ในที่นี้จะกล่าวถึงมอสเฟตเท่านั้น มอสเฟตจะแบ่งออกเป็น 2 ชนิด คือ คีพลีชัน (Depletion) และเอนฮานซ์เมนต์ (Enhancement) แต่แต่ละประเภทยังแบ่งออกเป็น 2 แบบ คือ แบบแชนแนล n และแบบแชนแนล p

มอสเฟตประเภท คีพลีชันหรือคิมอสเฟต (D-MOSFET) ทั้ง 2 แบบจะทำงานได้ 2 โหมด คือ โหมดคีพลีชัน (Depletion Mode) และ โหมดเอนฮานซ์เมนต์ (Enhancement Mode) กล่าวคือ ถ้าจ่ายแรงดันลบให้กับคิมอสเฟตแชนแนล n จะทำงานในโหมดคีพลีชัน แต่ถ้าจ่ายแรงดันบวกจะทำงานในโหมดเอนฮานซ์เมนต์ ส่วนคิมอสเฟตแชนแนล p ก็ทำงานคล้ายกันเมื่อได้รับแรงดันที่มีขั้วตรงข้ามกับแบบแชนแนล n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอสเฟตประเภทเอนฮานซ์เมนต์หรืออีมอสเฟต (E-MOSFET) มีโครงสร้างบางอย่างคล้ายกับมอสเฟตแบบคิปลิซัน แต่จะทำงานได้เฉพาะโหมดเอนฮานซ์เมนต์เท่านั้น



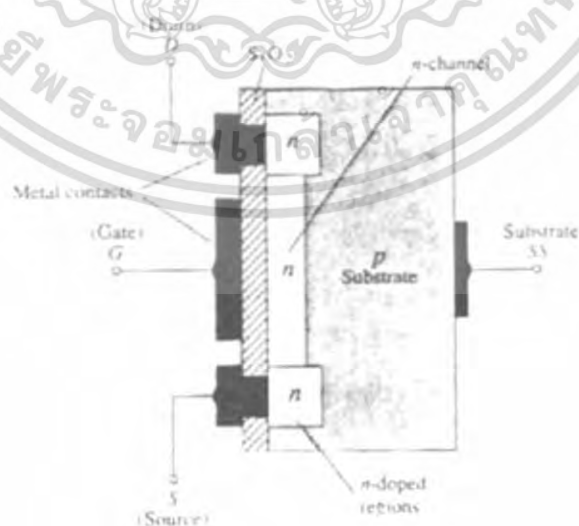
รูปที่ 2.42 วงจรสมมูลของมอสเฟต

2.4.2 ดิมอสเฟต (D-MOSFET)

แบ่งออกเป็น 2 ประเภท คือ

1. ดิมอสเฟตแบบแชนแนล n
2. ดิมอสเฟตแบบแชนแนล p

ดิมอสเฟตแบบแชนแนล n



รูปที่ 2.43 โครงสร้างภายในของดิมอสเฟตแบบแชนแนล n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

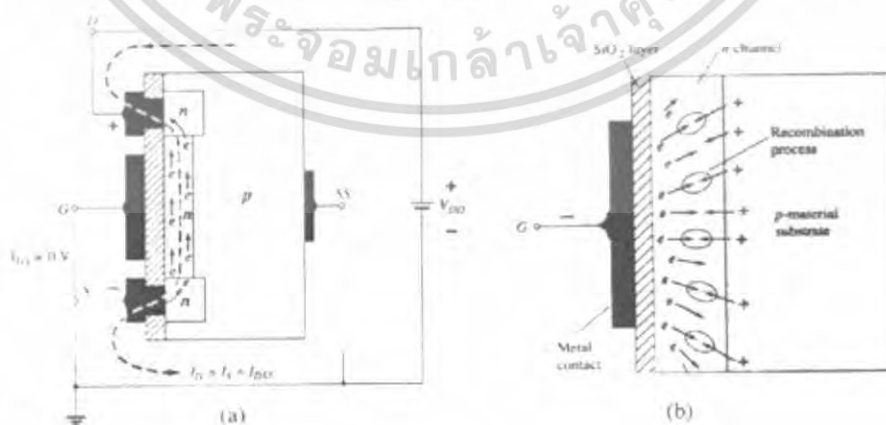
ดิมอสเฟตแบบแซนแนล n ประกอบขึ้นจากแผ่นผลึกฐาน p (p-substrate) ที่เป็นสารกึ่งตัวนำทำจากซิลิกอนซั้ว D และซั้ว S ต่อกับบริเวณที่มีการกระตุ้นหรือโด๊ปให้เป็นบริเวณสารกึ่งตัวนำ n (n-doped region) ทั้งสองส่วนนี้จะเชื่อมกับแซนแนล n สำหรับซั้ว G จะต่อกับวัสดุฉนวนที่เป็นโลหะ โดยมีซิลิกอนไดออกไซด์ (SiO_2) กั้นแซนแนล n กับซั้ว G (ซิลิกอนไดออกไซด์เป็นฉนวนประเภทไดอิเล็กทริก) เมื่อมีสนามไฟฟ้าจ่ายเข้ามาที่ชั้นของ SiO_2 ก็จะสร้างสนามไฟฟ้าด้านและสร้างชั้นฉนวนขึ้นภายในตัวเองเพื่อกั้นซั้วเกตกับแซนแนล แสดงว่า ไม่มีการต่อโดยตรงระหว่างซั้ว G กับแซนแนลของมอสเฟต ชั้นที่เป็นฉนวน SiO_2 จะทำให้ Z_i มีค่าสูงตามความต้องการได้

นอกจากนี้บางครั้งจะต่อแผ่นผลึกจากฐานเข้ากับแหล่งจ่ายจึงมีซั้วเพิ่มขึ้นมาเรียกว่า ซั้วผลึกฐาน SS (Substrat : SS) ทำให้มี ซั้วเพิ่มเป็น 4 ซั้ว และจากข้างต้น จึงสรุปความหมายของคำว่า MOS ในชื่อมอสเฟต(ทรานซิสเตอร์สนามไฟฟ้าโลหะออกไซด์สารกึ่งตัวนำ) ได้ดังนี้

- โลหะ(Metal, M) หมายถึงบริเวณสำหรับการต่อซั้ว D, S และ G กับวัสดุฉนวน
- ออกไซด์(Oxide, O) หมายถึงซิลิกอน ไดออกไซด์ (SiO_2)
- สารกึ่งตัวนำ (Semiconductor, S) หมายถึง โครงสร้างพื้นฐานในบริเวณแพร่กระจายของสารกึ่งตัวนำชนิด p และสารกึ่งตัวนำชนิด n

การทำงานและคุณสมบัติเบื้องต้น (Basic Operation and Characteristics)

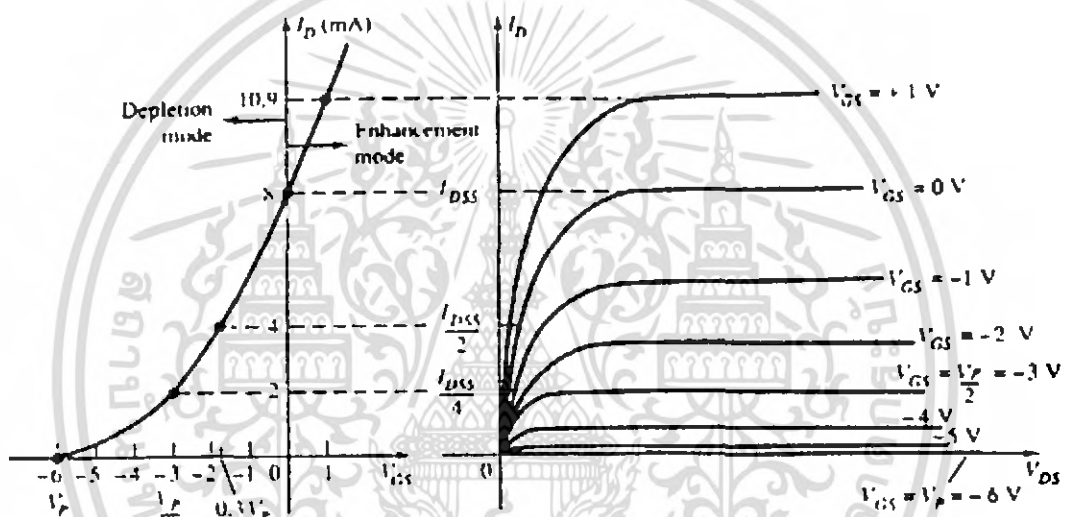
กำหนดให้ V_{GS} ในรูป (a) มีค่าเป็นศูนย์ แล้วจ่าย V_D ที่ซั้ว D และ S ซั้ว D สามารถดึงดูดอิเล็กตรอนอิสระ(e) ผ่านแซนแนล n และทำให้กระแส $I_D = I_S = I_{DSS}$ ไหลผ่านแซนแนล n ได้ (คล้ายกับการไหลของกระแสไฟฟ้าในแซนแนลของเจฟท์ขณะ $V_{GS} = 0 \text{ V}$)



รูปที่ 2.44 การทำงานของดิมอสเฟตแบบแซนแนล n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าจ่าย V_{GS} ที่มีค่าเป็นลบให้กับขั้วเกต (รูป b) เช่น -1 V ความต่างศักย์ที่ขั้วเกตจะผลักดันให้อิเล็กตรอนอิสระเคลื่อนไปยังแผ่นผลึกฐาน p และดึงคู่อิเล็กตรอนโฮลจากแผ่นผลึกฐาน p ทำให้อิเล็กตรอนและโฮลรวมตัวกันใหม่ (Recombination Process) จึงเกิดการลดจำนวนอิเล็กตรอนอิสระในแชนแนล n ที่มีไว้สำหรับการนำกระแส เมื่อมีค่า V_{GS} เป็นลบมากเท่าใดก็จะเกิดการรวมตัวกันใหม่มากขึ้นเท่านั้นและอิเล็กตรอนอิสระที่แชนแนล n ก็จะมีจำนวนลดลง จึงกล่าวได้ว่าถ้า V_{GS} เป็นลบมากขึ้น I_D จะมีค่าน้อยลง เขียนเป็นเทอร์พควดลักษณะได้ ดังรูปต่อไปนี้ การทำงานขณะ V_{GS} เป็นลบนี้ เราเรียกว่า การทำงานในโหมดคิฟลิซัน

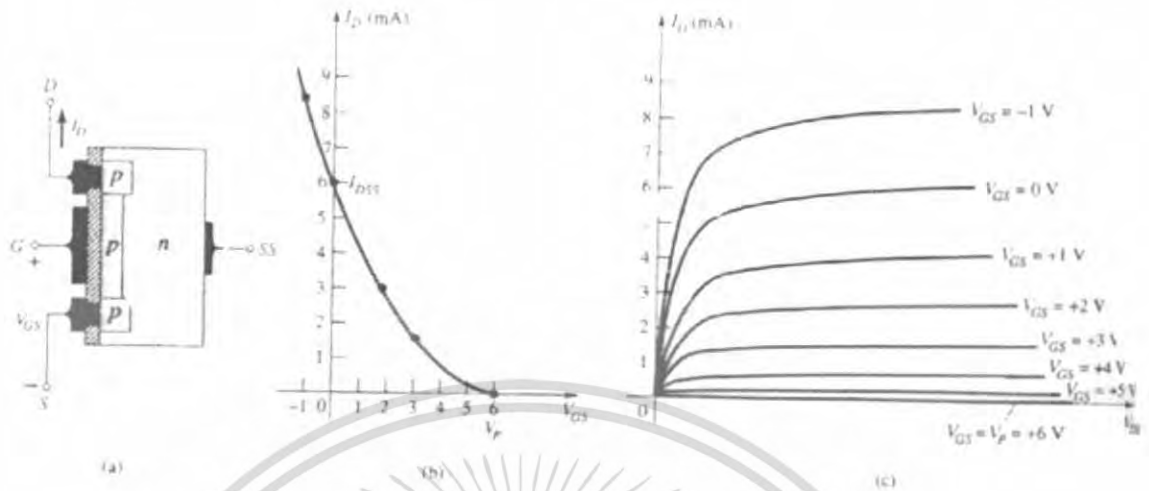


รูปที่ 2.45 กราฟแสดงคุณสมบัติของคิมอสเฟตแบบแชนแนล n

ถ้าจ่าย V_{GS} ที่มีค่าเป็นบวกให้กับขั้วเกตความต่างศักย์ที่ขั้วเกตจะดึงคู่อิเล็กตรอนจากผลึกฐาน p มายังบริเวณชั้น SiO_2 ทำให้พาหะนำกระแสและสภาพนำกระแสของแชนแนลเพิ่มขึ้น ดังนั้นกระแส I_D จึงเพิ่มขึ้นจนมีค่ามากกว่า I_{DSS} การทำงานขณะ V_{GS} เป็นบวกนี้เราเรียกว่า การทำงานในโหมดเอนเอนซ์เมนต์

คิมอสเฟตแบบแชนแนล p

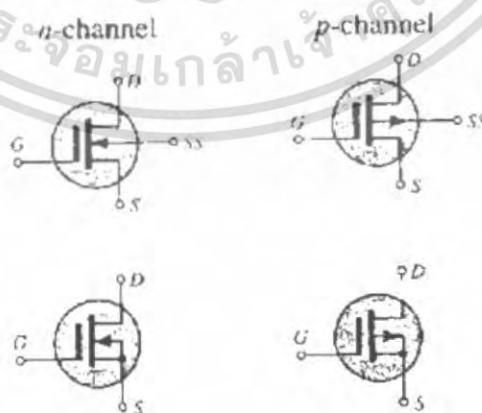
โครงสร้างของคิมอสเฟตแบบแชนแนล p มีลักษณะตรงข้ามกับรูปของคิมอสเฟตแบบแชนแนล n คือประกอบด้วยแผ่นผลึกฐาน n และแชนแนล p ดังรูป



รูปที่ 2.46 โครงสร้างและกราฟแสดงคุณสมบัติของคิมอสเฟตแบบแชนแนล p

จากการเปรียบเทียบคุณลักษณะของคิมอสเฟตแบบแชนแนล กับคิมอสเฟตแบบแชนแนล p (ตามรูป b และ c) เราพบว่าทิศทางของกระแสแรงดันต่าง ๆ กลับกันทำให้คุณลักษณะกลับกันด้วย I_D จะเริ่มเพิ่มขึ้นจากจุด Cutoff ที่ $V_{GS} = V_p$ และขณะที่ V_{GS} มีค่าเป็นบวกลดลง I_D จะเพิ่มขึ้นจนถึง I_{DSS} และเพิ่มอย่างต่อเนื่องจนเลขค่า I_{DSS} เมื่อ V_{GS} มีค่าเป็นลบเพิ่มขึ้นยังคงใช้สมการของขอคได้แต่ควรระวังเครื่องหมาย V_{GS} และ V_p ในสมการให้ถูกต้อง (คือจะต้องมีเครื่องหมายเป็นบวก)

สัญลักษณ์ (Symbols)



รูปที่ 2.47 แสดงสัญลักษณ์คิมอสเฟตแบบแชนแนล n และ p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

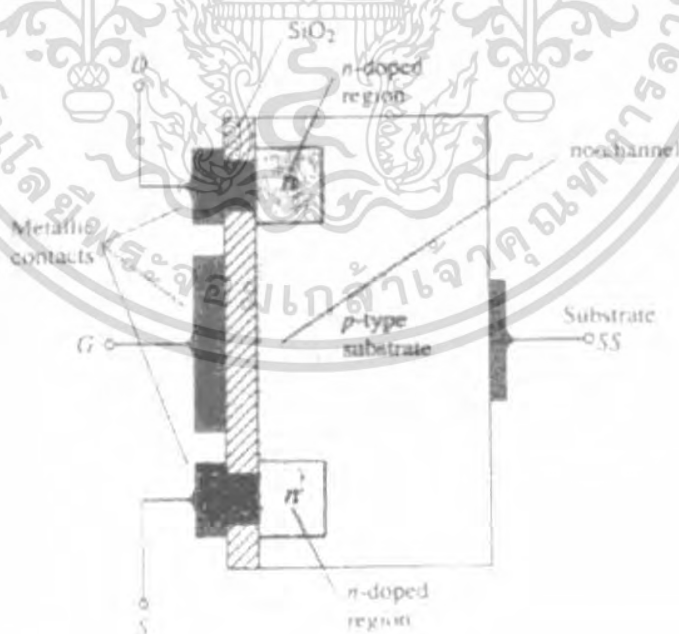
ถ้าสังเกตให้ดีจะเห็นว่า สัญลักษณ์สื่อความหมายถึงโครงสร้างแท้จริงของอุปกรณ์ ช่องว่างระหว่างขั้ว G กับขั้ว D (ที่ต่อกับขั้ว S) แสดงว่าไม่มีการต่อกันโดยตรงระหว่างขั้วทั้งสามเนื่องจากมีฉนวนกันที่ขั้ว G ส่วนขั้วผลึกฐาน SS ในมอสเฟต บางครั้งมีบางครั้งไม่มีจึงเขียนสัญลักษณ์ได้ทั้ง 2 แบบ คือ กรณีที่มีขั้ว SS และในกรณีไม่มีขั้ว SS ในการวิเคราะห์ลำดับต่อไปมักจะไม่มีขั้ว SS ดังนั้นสัญลักษณ์ที่อยู่ข้างล่างจะเป็นสัญลักษณ์ที่ใช้ทั่ว ๆ ไป

2.4.3 อิมอสเฟต (EMOSFET)

แม้ว่า โครงสร้างและขอบเขตการทำงานของดีมอสเฟตกับอิมอสเฟตจะมีลักษณะคล้ายกันอยู่บ้าง แต่กราฟคุณลักษณะของอิมอสเฟตจะแตกต่างออกไปโดยสิ้นเชิง กล่าวคือไม่สามารถนำสมการของชอกเลย์มาเขียนกราฟถ่ายโอนของอิมอสเฟตได้และกระแส I_D ยังไม่เกิดขึ้นจนกระทั่ง V_{GS} มีค่าสูงถึงค่าเฉพาะค่าหนึ่งอิมอสเฟตแบ่งออกเป็น 2 ประเภทคือ

1. อิมอสเฟตแบบเซนแนล n
2. อิมอสเฟตแบบเซนแนล p

อิมอสเฟตแบบเซนแนล n



รูปที่ 2.48 โครงสร้างเบื้องต้นของอิมอสเฟตแบบเซนแนล n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิมอสเฟตแบบแซนแนล n ประกอบขึ้นจากแผ่นผลึกฐาน p ที่เป็นสารกึ่งตัวนำทำจากซิลิคอน ขั้ว D และขั้ว S ต่อกับ บริเวณที่มีการกระตุ้น n โดยผ่านวัสดุฉนวนที่เป็น โลหะ นอกจากนี้ บางครั้งจะต่อแผ่นผลึกฐาน p เข้ากับแหล่งจ่ายจึง มีขั้ว SS เพิ่มขึ้นมาคล้ายกับคิมอสเฟต

ถ้าสังเกตรูป 1 ให้จะเห็นได้ว่าไม่มีเส้นทางเชื่อมหรือ ไม่มีแซนแนล (no-channel) ระหว่างบริเวณที่มีการกระตุ้น n ทั้ง 2 แห่ง นี่คือความแตกต่างเบื้องต้นระหว่าง โครงสร้างของ อิมอสเฟตและคิมอสเฟต

การทำงานและคุณลักษณะเบื้องต้น

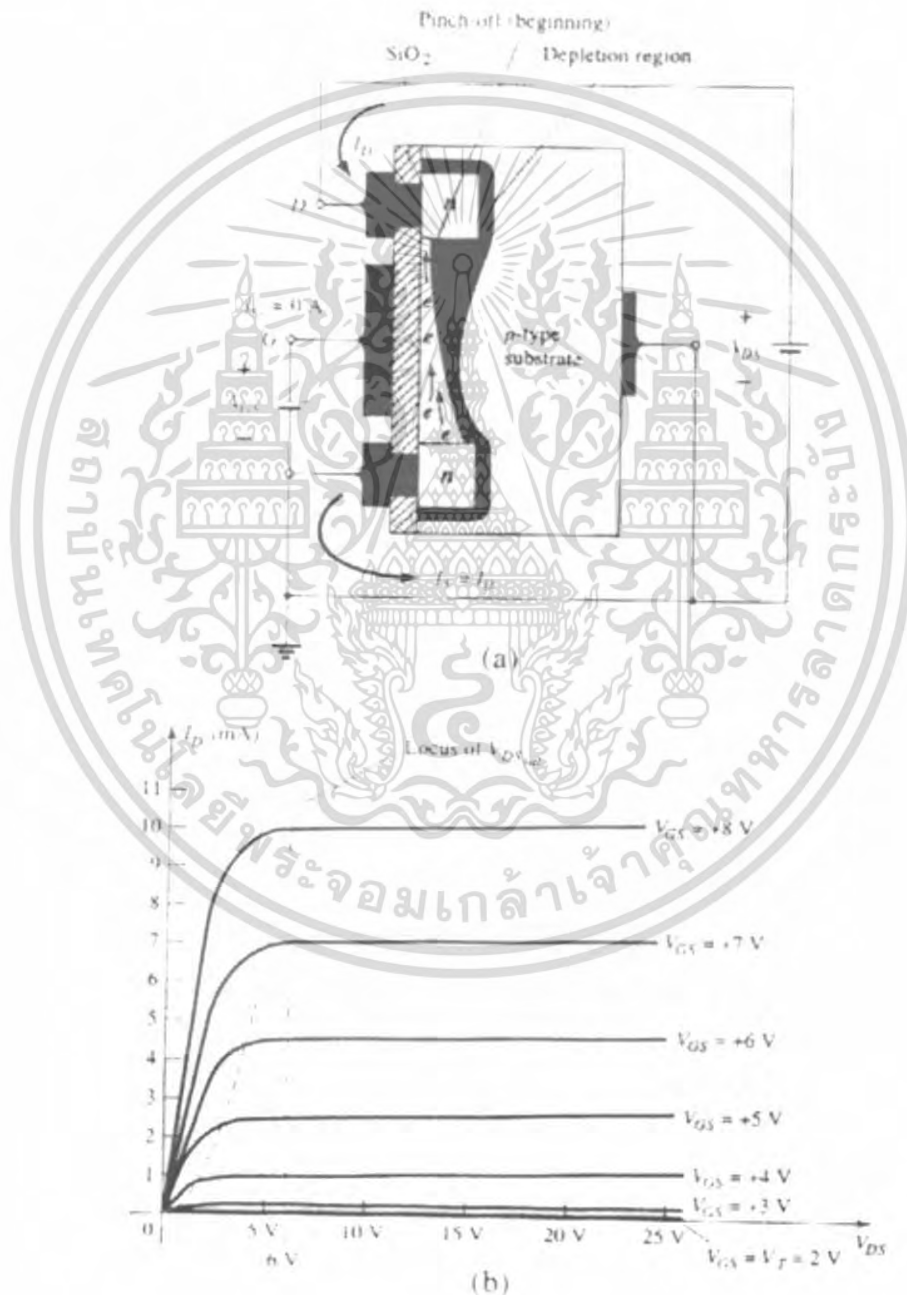
กำหนดให้ $V_{GS} = 0V$ และจ่าย V_{DS} ที่มีค่าเป็นบวกให้กับขั้ว S กับขั้ว D โดยขั้ว SS ต่อกับขั้ว S ดังรูป จะเกิดจาก โปสิทีฟที่รอยต่อ p-n (บริเวณที่มีการกระตุ้น n กับผลึกฐาน p) เนื่องจากไม่มีเส้นทางเชื่อม หรือ แซนแนลระหว่างขั้ว D และขั้ว S ทำให้เกิดการต้านการไหลของ อิเล็กตรอน] กระแส $I_D = 0$ แตกต่างจาก คิมอสเฟตและเจเฟตซึ่งมี $I_D = I_{DSS}$



รูปที่ 2.49 แสดงการทำงานของอิมอสเฟต

ถ้าจ่าย V_{DS} และ V_{GS} ที่มีค่าเป็นบวกดังรูป ทำให้ขั้ว D และขั้ว G มีความต่างศักย์เป็นบวกการที่ขั้ว G มีความต่างศักย์เป็นบวกนี้จะผลักดันให้ โอลิโนผลึกฐาน p เข้าไปสู่บริเวณภายในผลึกฐาน p และดึงดูดอิเล็กตรอนในผลึกฐาน p (เป็นพาหะข้างน้อยรวมตัวอยู่ในบริเวณใกล้กับผิวของ SiO_2) ซึ่งมีคุณลักษณะเป็นฉนวนและป้องกันอิเล็กตรอนไม่ให้ดึงดูดไปยังขั้วเกต ขณะที่ V_{GS} เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มขึ้น การรวมตัวของอิเล็กตรอนใกล้กับชั้นของ SiO_2 ก็เพิ่มมากขึ้นตามลำดับขณะเดียวกันบริเวณที่มีการกระตุ้น n เกิดการเหนี่ยวนำจากแรงดัน V_{GS} ทำให้มีอิเล็กตรอนหรือ ID (มีทิศทางตรงข้ามกับอิเล็กตรอน) ไหลระหว่าง ขั้ว D กับขั้ว S ระดับ V_{GS} ทำให้ ID ไหลเราเรียกว่า แรงดันเชรสโโลด์ (Threshold Voltage; V_T) ในสเปคกำหนดให้ V_T เป็น $V_{GS(th)}$



รูปที่ 2.50 กราฟแสดงคุณสมบัติของอิมอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเพิ่ม V_{GS} ให้สูงขึ้น I_D ก็จะเพิ่มขึ้นด้วย แต่ถ้า V_{GS} มีค่าคงที่ และ เพิ่มค่า V_{DS} จะทำให้ I_D ถึงจุดอิ่มตัว (เช่นเดียวกับคิมอสเฟต) เนื่องจากขั้วบวกของ V_{DS} ดึงคู่อิเล็กตรอน จึงจะทำให้ปลายของช่องทางเหนี่ยวนำบริเวณใกล้ขั้ว D แคบลงใกล้ระดับพินช์ออฟ [Pinch-Off (Beginning)] ดังรูป (a) เมื่อนำ KVL มารวมพิจารณา จะได้แรงดันไฟฟ้าระหว่างขั้ว D กับขั้ว G (V_{DG}) ดังนี้

$$V_{DG} = V_{DS} - V_{GS} \quad (7)$$

ถ้ากำหนดให้ $V_{GS} = 8 \text{ V}$ และ $V_{DS} = 2 \text{ V}$ ก็จะได้ $V_{DG} = -6 \text{ V}$ แต่ถ้าเพิ่ม V_{GS} เป็น 5 V ค่า V_{DG} จะเป็น -3 V (ตามสมการที่ 1) การลดลงของ V_{DG} ทำให้แรงดึงดูด (จากขั้วบวกของ V_{DS}) ที่มีต่ออิเล็กตรอนอิสระในบริเวณช่องทางเหนี่ยวนำลดลงด้วย ทำให้ช่องทางเหนี่ยวนำแคบลง ถ้าความกว้างของช่องทางดังกล่าวลดลงเรื่อยๆ จนกระทั่งถึงจุดพินช์ออฟ I_D ก็จะถึงจุดอิ่มตัว ดังที่ได้อธิบายในคิมอสเฟต

คุณลักษณะของขั้วเกรนของมอสเฟตใน รูป (a) เป็นดังรูป (b) ขณะที่ $V_T = 2 \text{ V}$ ที่ $V_{GS} = 8 \text{ V}$ ทำให้เกิด V_{DSsat} อิ่มตัว (V_{DSsat}) = 6 V ทำให้ได้ความสัมพันธ์ระหว่าง V_{DSsat} กับ V_{GS} ดังนี้

$$V_{DSsat} = V_{GS} - V_T \quad (8)$$

สมการที่ 2 ทำให้ทราบว่า เมื่อ V_T คงที่และ V_{GS} ยิ่งสูงขึ้นเท่าใด V_{DSsat} ก็ยิ่งสูงขึ้นเท่านั้น ในรูป (b) ขณะที่ V_T เป็น 2 V ณ ตำแหน่งนี้ $I_D = 0 \text{ mA}$ ดังนั้นจึงทำให้ทราบว่า ถ้า V_{GS} มีค่าต่ำกว่า V_T ค่า I_D ของมอสเฟตจะเป็นศูนย์หรือไม่มีการแสไหลนั่นเอง

ถ้าค่า V_{GS} เพิ่มขึ้นจาก V_T เป็น 8 V จะทำให้ระดับการอิ่มตัวของ I_D เพิ่มขึ้นจาก 0 mA เป็น 10 mA แต่เนื่องจากช่วงของพัลส์ V_{GS} มีระยะห่างไม่เท่ากัน ดังนั้น I_D ที่เพิ่มขึ้น จึงมีความสัมพันธ์กับ V_{GS} ในลักษณะไม่เป็นเชิงเส้นดังสมการต่อไปนี้

$$I_D = k (V_{GS} - V_T)^2 \quad (9)$$

เมื่อ k เป็นค่าคงที่ของโครงสร้างมอสเฟต ซึ่งหาค่าได้จาก

$$k = I_{D(on)} / (V_{GS(on)} - V_T)^2 \quad (10)$$

เมื่อ $I_{D(on)}$ และ $V_{GS(on)}$ เป็นกระแสและแรงดันที่ทำให้เกิดจุดเฉาะบน

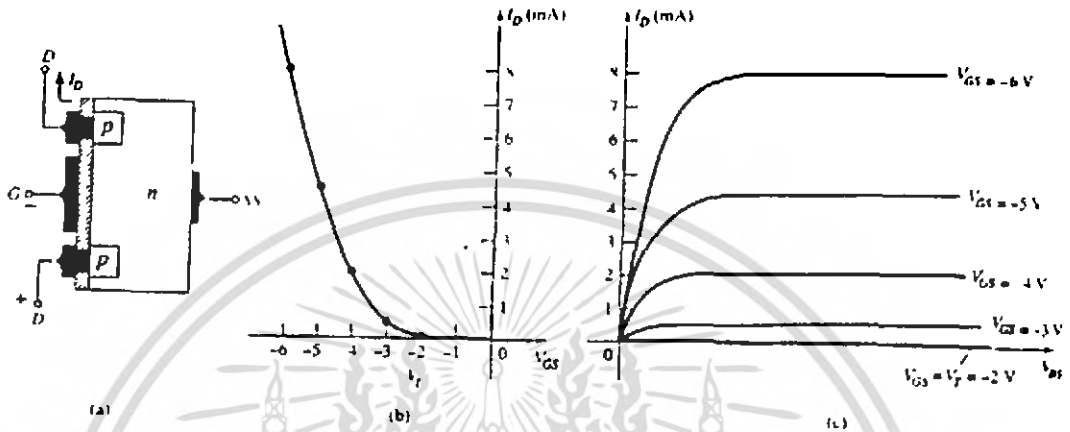
เคอร์ฟคุณลักษณะของมอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิมอสเฟทแบบแซนแนล p

๔

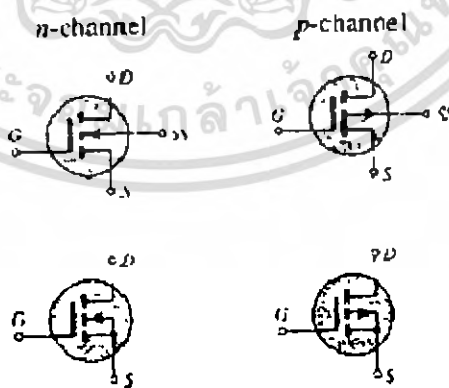
โครงสร้างของอิมอสเฟทแบบแซนแนล p มีลักษณะตรงข้ามกับแบบแซนแนล n



รูปที่ 2.52 แสดง โครงสร้างของอิมอสเฟทแบบแซนแนล p

กล่าวคือ ขั้ว D และขั้ว S ต่อกับผลึกฐาน n และบริเวณที่มีการกระตุ้น p (p-doped regions) แต่ขั้วของแรงดันและทิศทางการกระแสตรงข้ามกับแบบแซนแนล n นอกจากนี้คุณลักษณะของเทอร์ฟ ถ่ายไออน ก็จะมีค่าที่ด้านตรงข้าม

สัญลักษณ์ (Symbol)



รูปที่ 2.53 แสดงสัญลักษณ์ของอิมอสเฟทแบบแซนแนล p และแซนแนล n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าสัญลักษณ์แสดงโครงสร้างแท้จริงของอุปกรณ์เส้นประเชื่อมระหว่างขั้ว D กับ ขั้ว S แสดงว่าไม่มี แชนแนลระหว่างขั้วทั้งสอง (ขณะไม่ได้รับการไบอัส) ซึ่งเป็นความแตกต่างประการเดียวระหว่างสัญลักษณ์ของคิมอสเฟทกับอิมอสเฟท

2.5 การมอดูเลตสัญญาณดิจิทัล (Digital Modulation)

การส่งสัญญาณดิจิทัลด้วยคลื่นวิทยุ

การส่งสัญญาณ PCM นั้น ถ้าเป็นการส่งแบบมีสายแล้ว สัญญาณ PCM จะถูกส่งในลักษณะ baseband อย่างไรก็ตามสำหรับการส่งแบบไร้สายนั้น คลื่นพาห้จะถูกมอดูเลตกับสัญญาณ PCM หรือ อาจกล่าวได้ว่าจำเป็นต้องเปลี่ยนสัญญาณพัลส์ ให้เป็นสัญญาณในรูปแบบความถี่วิทยุ (RF) ซึ่งก็คือการมอดูเลตชนิดหนึ่ง วิธีการมอดูเลชัน มีด้วยกัน 3 วิธี คือ

Amplitude Modulation (AM)

Frequency Modulation (FM)

Phase Modulation (PM)

ถ้าจะกล่าวถึงการมอดูเลชันสัญญาณพัลส์กับคลื่นความถี่วิทยุนั้น ก็เป็นเพียงการเปลี่ยน (เปลี่ยนแปลง) แอมพลิจูด ความถี่ หรือเฟส ของคลื่นพาห้ไปยังสภาวะหนึ่งใน 2 สภาวะ ตามการมีหรือไม่มีสัญญาณพัลส์ของ PCM เท่านั้น ดังนั้นชื่อของการมอดูเลชันจะเปลี่ยนเป็น

Amplitude Shift Keying (ASK)

Frequency Shift Keying (FSK)

Phase Shift Keying (PSK)

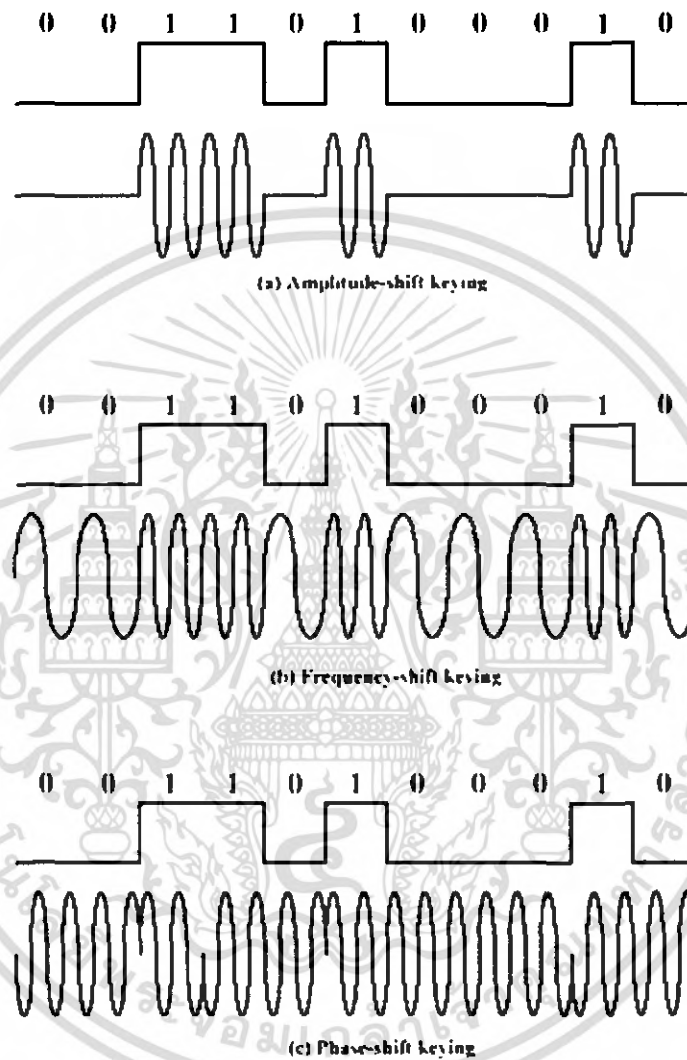
ASK อาจจะถูกเรียกว่า “ON-OFF keying” เนื่องจากได้ถูกนำมาใช้ในการส่งทางด้านโทรเลขกล่าวคือ จะทำการ ON และ OFF คลื่นพาห้ตามค่าของรหัส “1” และ “0”

FSK จะเป็นวิธีการเปลี่ยนความถี่ของคลื่นพาห้ตามรหัส “1” และ “0” ของพัลส์ โดยจะจัดสรรให้ความถี่ของ f_1 และ f_2 สอดคล้องกับรหัส “1” และ “0” แล้วทำการสับเปลี่ยนความถี่ 2 ความถี่นี้ ตามรหัสที่สอดคล้อง

PSK เป็นวิธีการเปลี่ยนเฟสของคลื่นพาห้ตามรหัส “1” และ “0” ของพัลส์ ตัวอย่างเช่น ในการส่งรหัส “0” และ “1” นั้น โดยทั่วไปจะใช้คลื่นพาห้ เฟส 0° และ 180° แทนรหัส

ในกรณีที่ใช้คลื่นไมโครเวฟสำหรับเครื่องทวนสัญญาณ (Repeater) แบบไร้สายนั้นเครื่องทวนสัญญาณจะรับเอาการเปลี่ยนแปลงของกำลังของคลื่นที่รับได้มาสร้างเป็นสัญญาณใหม่ แต่ผลจากการที่คลื่นไมโครเวฟถูกลดทอน เนื่องจากอิทธิพลของ fading (การจางหาย) และฝน ดังนั้นในการส่งสัญญาณพัลส์ ที่ต้องการเสถียรภาพนั้น จะไม่เหมาะถ้าใช้ระบบ ASK ส่วนระบบ FSK นั้น ก็

ต้องการ bandwidth ที่กว้าง เพื่อใช้ในการส่ง ดังนั้นจึงมีคุณลักษณะที่ด้อยกว่าระบบ PSK ดังนั้น
วิธีการส่งสัญญาณ PCM แบบไร้สายนั้น โดยทั่วไปจึงใช้ระบบ phase modulation หรือ PSK นั่นเอง



รูปที่ 2.54 ลักษณะของสัญญาณ ASK – FSK– PSK

2.5.1 Amplitude Shift Keying (ASK)

ASK คือ การเปลี่ยนความสูง (ระดับ) ของสัญญาณตามบิตข้อมูล เช่น ระดับ A_0 เมื่อข้อมูลเป็น 0 (ถ้า $A_0 = 0$ เรียกว่า On/Off Keying-OOK) และระดับ A_1 เมื่อเป็นข้อมูล 1 ตามสมการ

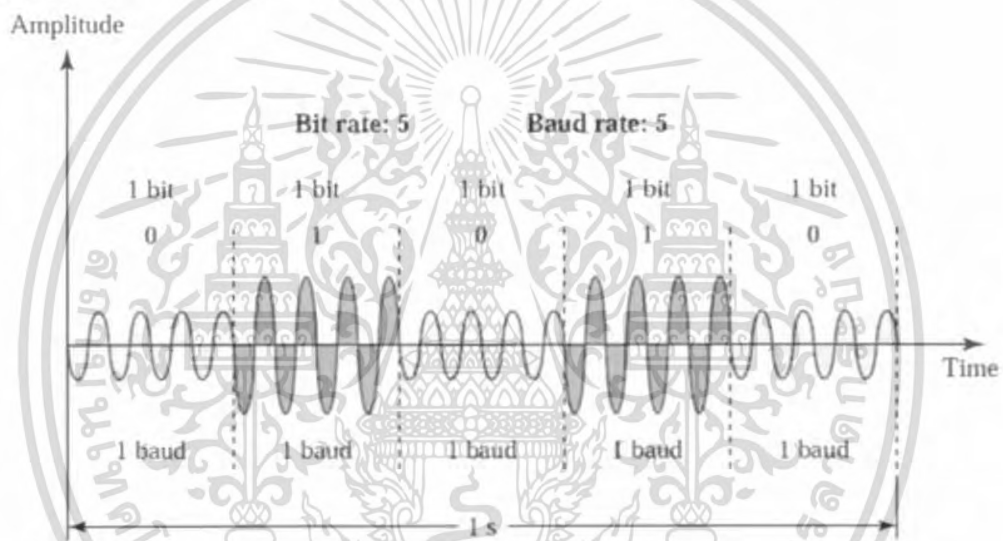
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$ASK(t) = \begin{cases} A_0 \sin(2\pi ft + \phi) & , data(t) = 0 \\ A_1 \sin(2\pi ft + \phi) & , data(t) = 1 \end{cases}$$

$$OOK(t) = \begin{cases} 0 & , data(t) = 0 \\ A_1 \sin(2\pi ft + \phi) & , data(t) = 1 \end{cases}$$
(12)

จุดเด่นของ ASK คือ สามารถออกแบบได้ง่าย เนื่องจากการคูณกันของฟังก์ชัน 2 ชนิด
 ดังสมการด้านล่าง ทำให้การคำนวณคุณสมบัติไม่ซับซ้อน

จุดด้อยของ ASK คือ มีภูมิคุ้มกันต่อสัญญาณรบกวนต่ำ อาจเกิดความผิดพลาดในการ
 สื่อสารได้ง่าย ASK ที่พบจะใช้งานในอุปกรณ์ผสมสัญญาณความเร็วต่ำ (<1200 Bauds) และการส่ง
 ข้อมูลด้วย LED



รูปที่ 2.55 การผสมสัญญาณของข้อมูล 01010 ด้วยวิธีการ ASK ในกรณีนี้ Bit Rate = Baud Rate

แถบความถี่ (Bandwidth) ของการผสมสัญญาณแบบ ASK สามารถหาได้จากความสัมพันธ์

$$BW_{ASK} = \left(1 + \frac{A_1 - A_0}{A_1 + A_0}\right) \times N_{baud} = (1 + d) \times N_{baud}$$
(13)

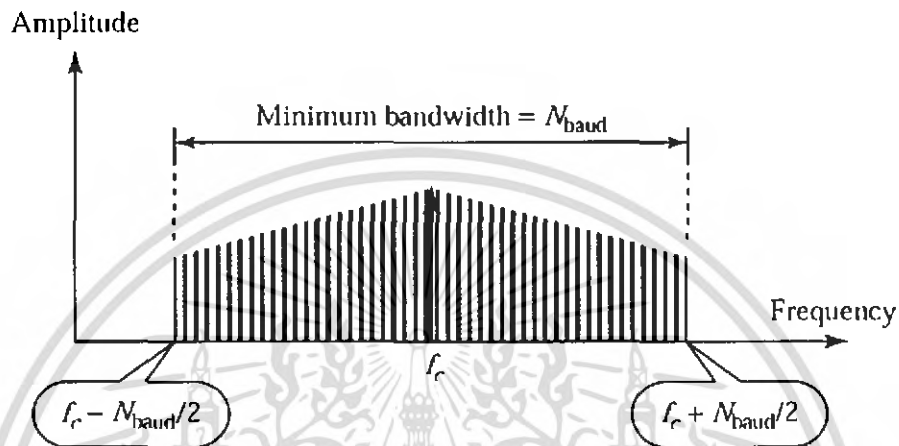
โดยที่ d คือ Modulating Factor ซึ่งมีคุณสมบัติดังนี้

- ถ้าค่า d น้อย (A_1 ต่างจาก A_0 น้อย) แล้ว Bandwidth จะแคบ และสัญญาณมีโอกาส
 เสียหายจาก Noise สูง
- ถ้าค่า d มาก (A_1 ต่างจาก A_0 มาก) แล้ว Bandwidth จะกว้าง และสัญญาณมีโอกาส
 เสียหายจาก Noise ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bandwidth ของการผสมสัญญาณแบบ ASK ด้วยความถี่ของสัญญาณพาหะ f_c และอัตรา Baud

N_{baud} มีความถี่อยู่ในช่วง $f_c \pm (N_{\text{baud}}/2)$ ตามลำดับ ดังรูปที่ 2.56

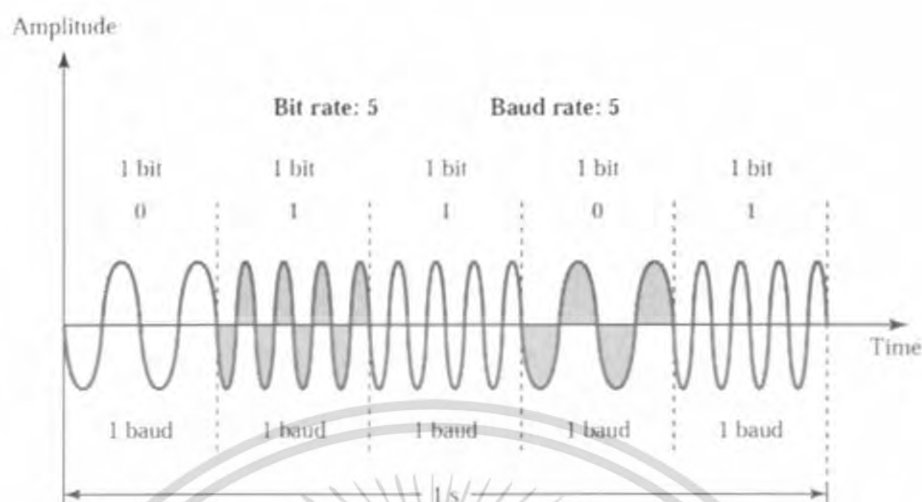


รูปที่ 2.56 Spectrum ของสัญญาณ ASK ที่มีความถี่ของสัญญาณพาหะ f_c และอัตราเร็วของข้อมูล Baud N_{baud}

2.5.2 Frequency Shift Keying (FSK)

FSK คือ การเปลี่ยนความถี่ของสัญญาณพาหะตามบิตข้อมูล เช่น ความถี่ f_{c0} เมื่อข้อมูลเป็น 0 ความถี่ f_{c1} เมื่อข้อมูลเป็น 1 ตามสมการ

$$FSK(t) = \begin{cases} A \sin(2\pi f_{c0}t + \phi) & , data(t) = 0 \\ A \sin(2\pi f_{c1}t + \phi) & , data(t) = 1 \end{cases} \quad (14)$$



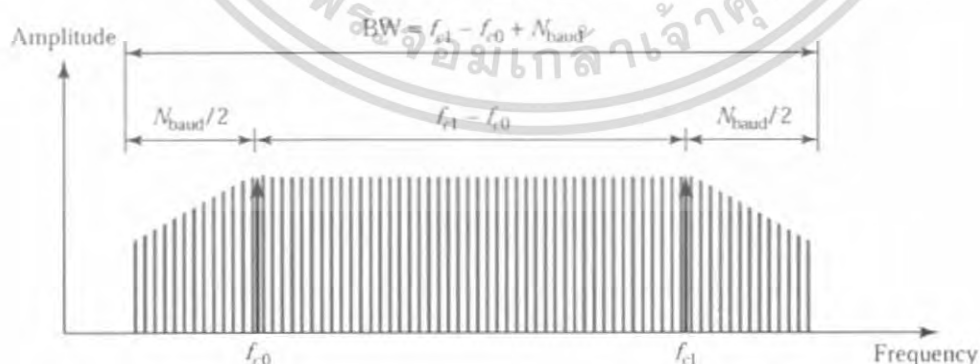
รูปที่ 2.57 การผสมสัญญาณของข้อมูล 01101 ด้วยวิธีการ FSK ในกรณีนี้ Bit Rate = Baud Rate

จุดเด่นของ FSK คือ มีภูมิคุ้มกันต่อสัญญาณรบกวนมากกว่าวิธี ASK เนื่องจากอุปกรณ์ด้านรับมองหาความถี่เฉพาะที่อยู่ในช่วงเวลาหนึ่ง โดยไม่สนใจ Noise กระชากกระยะสั้น (Transient Noise)

จุดด้อยของ FSK ก็คือ ต้องการ Bandwidth กว้างกว่าวิธี ASK เมื่อส่งข้อมูลที่มี Baud Rate เท่ากัน แล่นความถี่ของ FSK สามารถหาได้จาก ผลรวมระหว่างความแตกต่างของความถี่พาหะและอัตราเร็วของข้อมูล ดังสมการ

$$BW_{FSK} = (f_{c1} - f_{c0}) + N_{baud} \quad (15)$$

เมื่อ N_{baud} คือ อัตราเร็วของข้อมูล และ f_{c1} และ f_{c0} คือ ความถี่สูงสุดและต่ำสุด ตามลำดับ ดังรูป



รูปที่ 2.58 Spectrum ของสัญญาณ FSK ที่มีความถี่ของสัญญาณพาหะ f_{c1} และ f_{c0}

และอัตราเร็วของข้อมูล Baud N_{baud}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Phase Shift Keying (PSK)

PSK คือ การเปลี่ยนเฟส (ตำแหน่งเริ่มต้น) ของสัญญาณพาหะตามบิตข้อมูล เช่น เฟส ϕ_0 เมื่อข้อมูลเป็น 0 เฟส ϕ_1 เมื่อข้อมูลเป็น 1 ดังสมการ

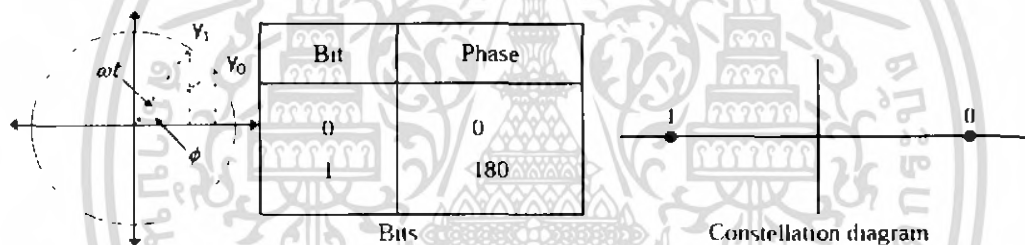
$$PSK(t) = \begin{cases} A\sin(2\pi f_c t + \phi_0) & , data(t) = 0 \\ A\sin(2\pi f_c t + \phi_1) & , data(t) = 1 \end{cases} \quad (16)$$

จุดเด่นของ PSK คือ สามารถทนทานต่อสัญญาณรบกวนมากกว่า ASK ในขณะที่แถบความถี่แคบกว่า FSK นอกจากนี้ยังสามารถแทนสัญลักษณ์ได้มากกว่า 1 บิตต่อหน่วยสัญญาณ

จุดด้อยของ PSK คือ การออกแบบเครื่องรับ-ส่ง และวิเคราะห์ ทำได้ยากกว่า ASK และ FSK

PSK Characteristics (Constellation Diagram)

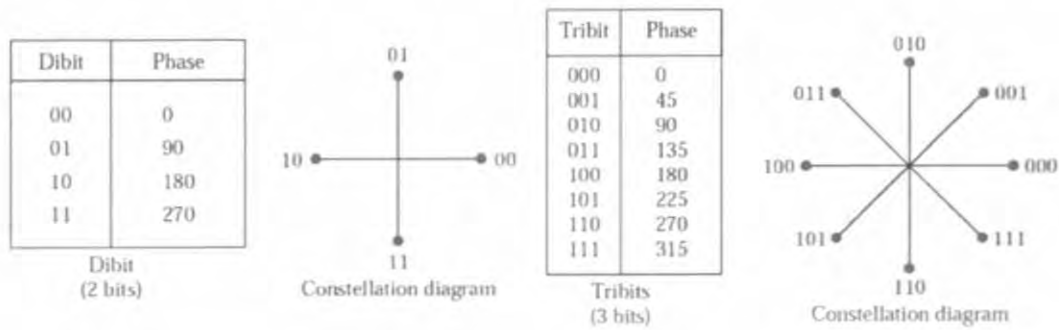
การผสมสัญญาณแบบ PSK ในรูปแบบต่างๆ สามารถแสดงโดยแผนผัง Constellation ซึ่งบอกตำแหน่ง Phase ของสัญลักษณ์ จากตัวอย่าง 1 Symbol = 1 Bit เรียกว่า Binary PSK ดังรูป



รูปที่ 2 59 Constellation Diagram ของการผสมสัญญาณแบบ PSK แบบ Binary PSK ซึ่งใช้เฟส 2 เฟส แทนข้อมูล 0 และ 1

ในการออกแบบหรือวิเคราะห์ Modulation PSK มีขั้นตอนดังนี้

- 1 กำหนดจำนวนของบิต (N) ต่อ 1 สัญลักษณ์ (2 = Dibit, 3 = Tribit ..)
- 2 แบ่งเฟส (จากทั้งหมด 360°) ออกเป็น 2^N ช่วง เท่าๆ กัน
3. แทนค่าสัญลักษณ์แต่ละแบบด้วยค่าเฟสแต่ละช่วง
4. แสดงการจับคู่สัญลักษณ์และเฟส ด้วยตารางหรือแผนผัง Constellation

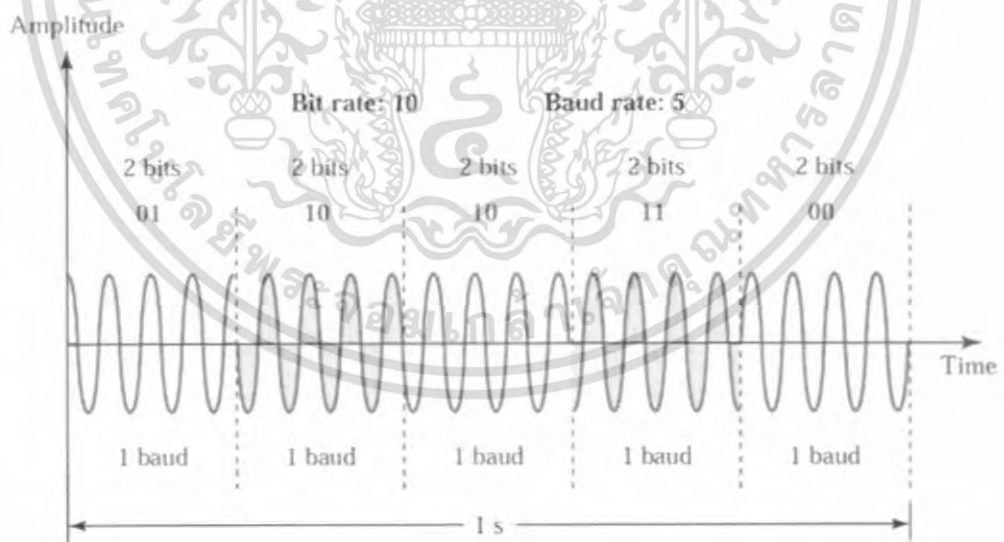


รูปที่ 2.60 Constellation Diagram ของการผสมสัญญาณแบบ PSK แบบ Dibit (ซ้าย) และ Tribit (ขวา) ตามลำดับ

รูปที่ 8 แสดงตัวอย่างของ PSK ชนิด Dibit ซึ่งสามารถแทนสัญลักษณ์ได้ 2 บิต (4 รูปแบบ) ต่อ 1 หน่วยสัญญาณ ซึ่งจะได้ว่า Baud Rate = 2 × Bit Rate

แถบความถี่ของ PSK มีค่าเท่ากับแถบความถี่ ASK เมื่อส่งด้วยอัตรา Baud เท่ากันและค่าสัมประสิทธิ์ $\beta = 0$ (แต่ Bit Rate ของ PSK จะมากกว่าหรือเท่ากับ ASK เสมอ) ดังสมการ

$$BW_{PSK} = N_{baud} \quad (17)$$



รูปที่ 2.61 การผสมสัญญาณของข้อมูล 01101100 ด้วยวิธีการ PSK ในกรณีนี้ Bit Rate มีค่าเป็น 2 เท่าของ Baud Rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

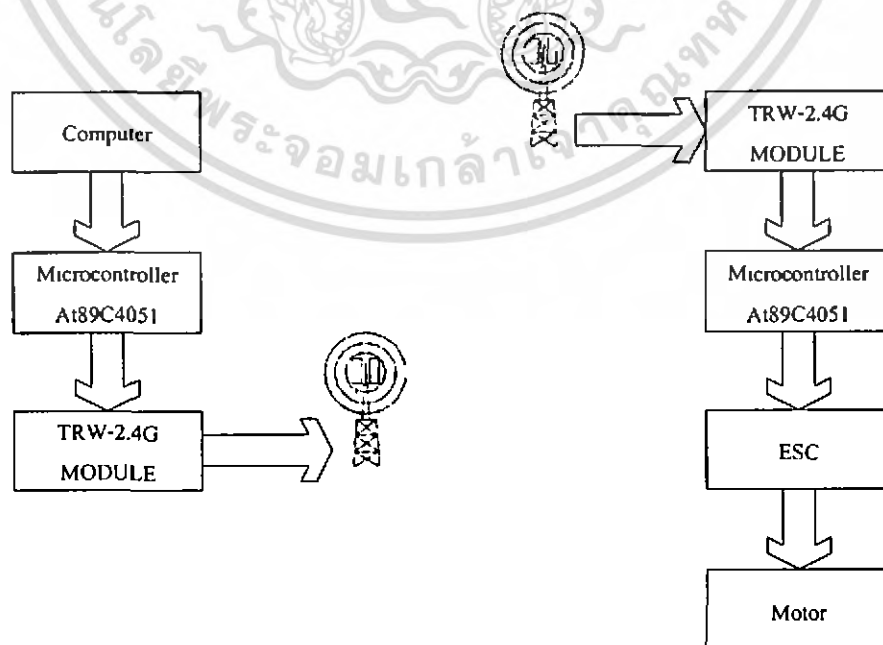
บทที่ 3

การออกแบบโครงสร้าง

ในการออกแบบโครงงานนี้จะแยกการออกแบบเป็น 2 ส่วนคือ ทางด้านฮาร์ดแวร์ (Hardware) และทางด้านซอฟต์แวร์ (Software) เพื่อง่ายต่อการทดลอง โดยส่วนฮาร์ดแวร์จะอธิบายถึงการออกแบบวงจร การสร้างวงจร และการออกแบบรูปลักษณะภายนอกของโครงงาน ส่วนทางด้านซอฟต์แวร์ จะกล่าวถึงแนวทางในการเขียนโปรแกรมเพื่อที่จะให้เป็นไปตามรูปแบบของโครงงานที่กำหนดไว้

3.1 โครงสร้างทางฮาร์ดแวร์

โครงงานจะทำงานจากการป้อนข้อมูลทางคอมพิวเตอร์ เพื่อรับคำสั่งว่าให้อากาศยานทำงานแบบใด แล้วทำการส่งข้อมูลดังกล่าวไปยังตัวไมโครคอนโทรลเลอร์เบอร์ AT89C4051 ซึ่งไมโครคอนโทรลเลอร์จะทำการประมวลผลข้อมูล เพื่อให้การทำงานเป็นไปตามที่กำหนด และส่งคำสั่งนั้นไปยังโมดูล TRW 2.4G ทางภาคส่ง ซึ่งเป็นตัวรับ-ส่งสัญญาณที่ใช้ควบคุมอากาศยาน และโมดูล TRW 2.4G ทางภาครับก็จะรับคำสั่งนั้นส่งให้กับไมโครคอนโทรลเลอร์เบอร์ AT89C4051 ทางภาครับ ซึ่งจะควบคุมการทำงานของมอเตอร์ โดยจะมีชุดวงจรอิเล็กทรอนิกส์เข้ามาช่วยในการทำงานนั้นเป็นไปตามต้องการ เช่น วงจร ESC (Electronic Speed Control) ซึ่งภายในประกอบไปด้วยไอซี ทำหน้าที่เป็นวงจรควบคุมความเร็วของมอเตอร์

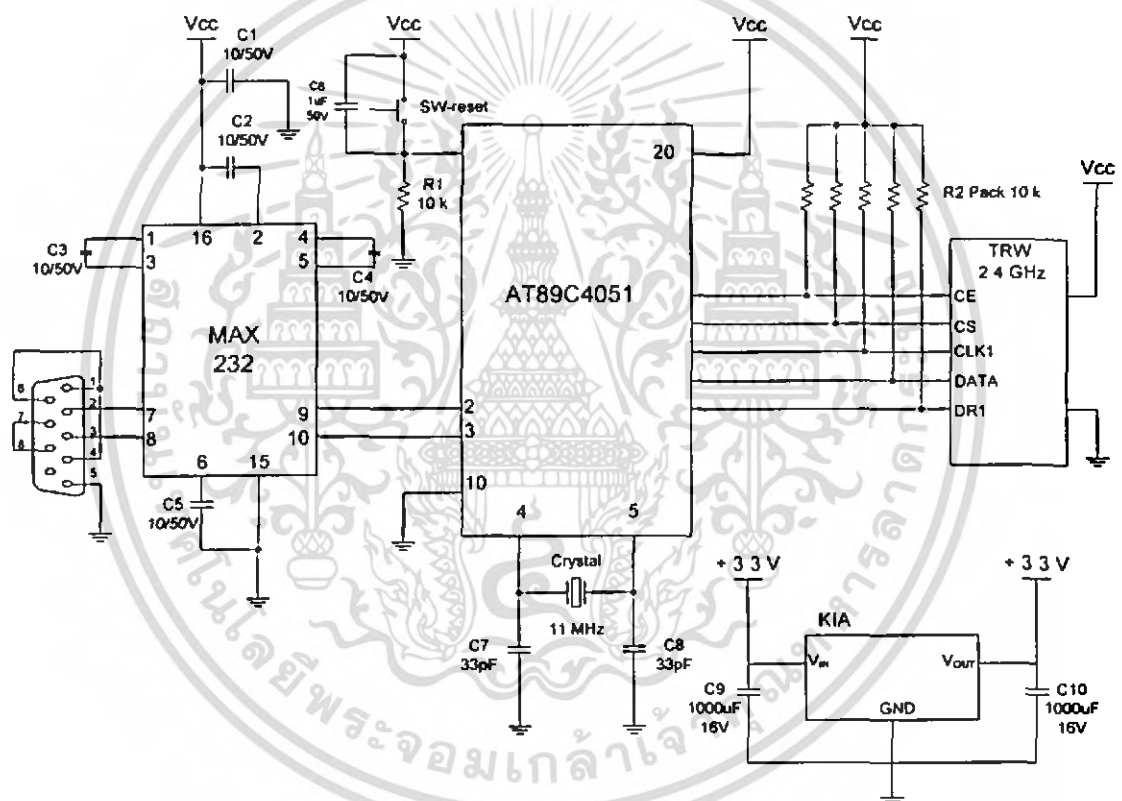


รูปที่ 3.1 บล็อกไดอะแกรมของอากาศยานแบบ 4 ใบพัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นใบใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ส่วนของชุดควบคุม

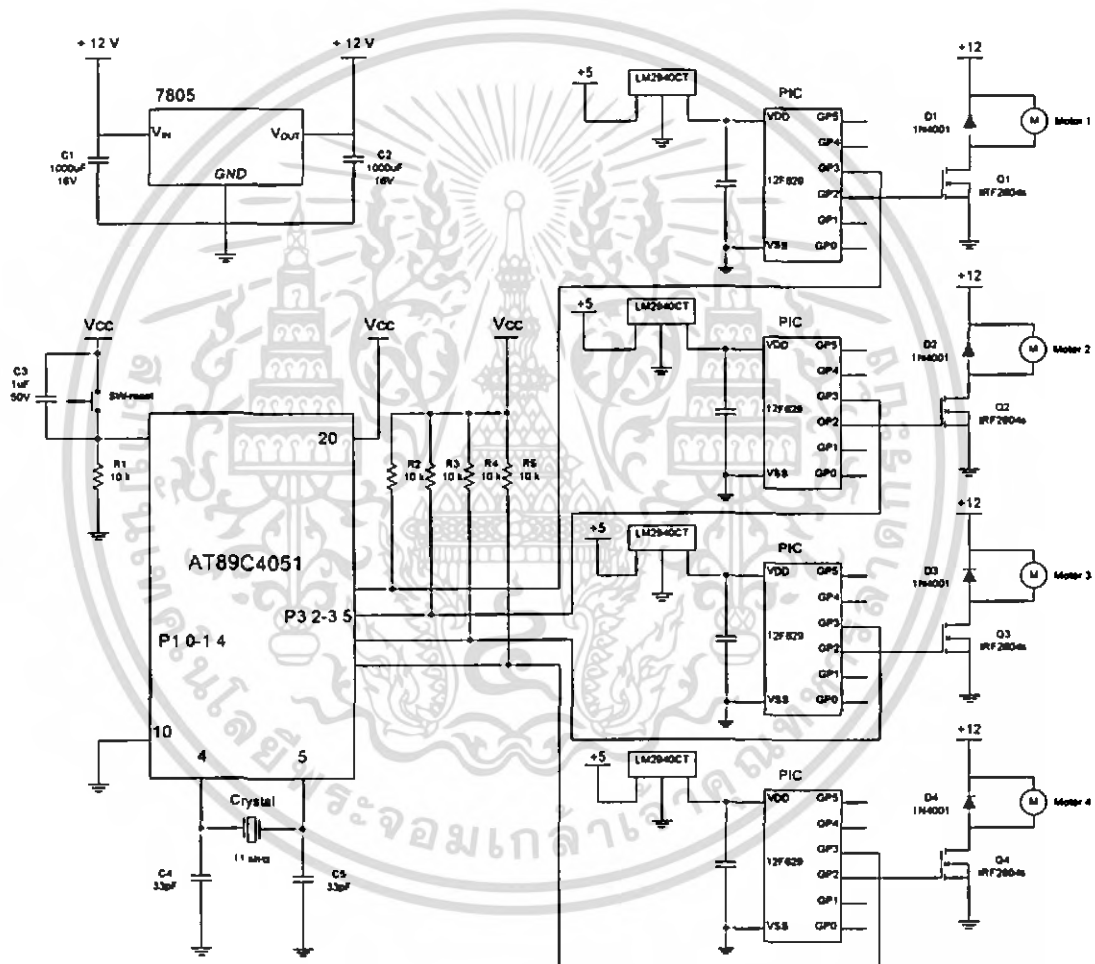
1. การออกแบบวงจรส่วนภาคส่ง จะประกอบด้วย ไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C4051 โดยพอร์ต P1.0-P1.4 จะใช้ต่อกับไอซีโมดูลเบอร์ TRW 2.4 G ซึ่งเป็นไอซีสำเร็จรูปใช้รับ-ส่งข้อมูลในแบบอนุกรม ใช้กับความถี่ 2.4 GHz มีเสาอากาศในตัว และพอร์ต P3.0-P3.1 จะใช้ต่อกับไอซี MAX 232 ซึ่งทำหน้าที่แปลงระดับแรงดันตามมาตรฐาน RS-232 และรับข้อมูลจากคอมพิวเตอร์โดยผ่านพอร์ต RS-232 ผ่านไอซี MAX 232 แล้วจึงส่งต่อไปให้ไอซี AT89C4051 เพื่อประมวลผลต่อไป ส่วนการเชื่อมต่อวงจรแสดงดังรูปที่ 63



รูปที่ 3.2 แสดงวงจรส่วนภาคส่ง

2. การออกแบบภาครับ จะคล้ายกับวงจรส่วนภาคส่ง โดยไมโครคอนโทรลเลอร์ที่ใช้เป็นเบอร์ AT89C4051 โดยพอร์ต P1.0-P1.4 จะใช้ต่อกับวงจรมอดคอนโทรลเพื่อรับข้อมูลเข้ามาแล้วทำการประมวลผล พอร์ต P3.2-P3.5 จะต่อกับวงจร ESC ซึ่งภายในประกอบด้วยไอซีเบอร์ LM2940CT เป็นไอซีเร็กกูเลเตอร์ (Regulator) ทำหน้าที่จ่ายไฟให้กับวงจร ESC ส่วนไอซีเบอร์ PIC12F629 เป็นไมโครคอนโทรลเลอร์อีกตัวหนึ่งที่จะประมวลผลข้อมูลที่ได้จาก MCS-51

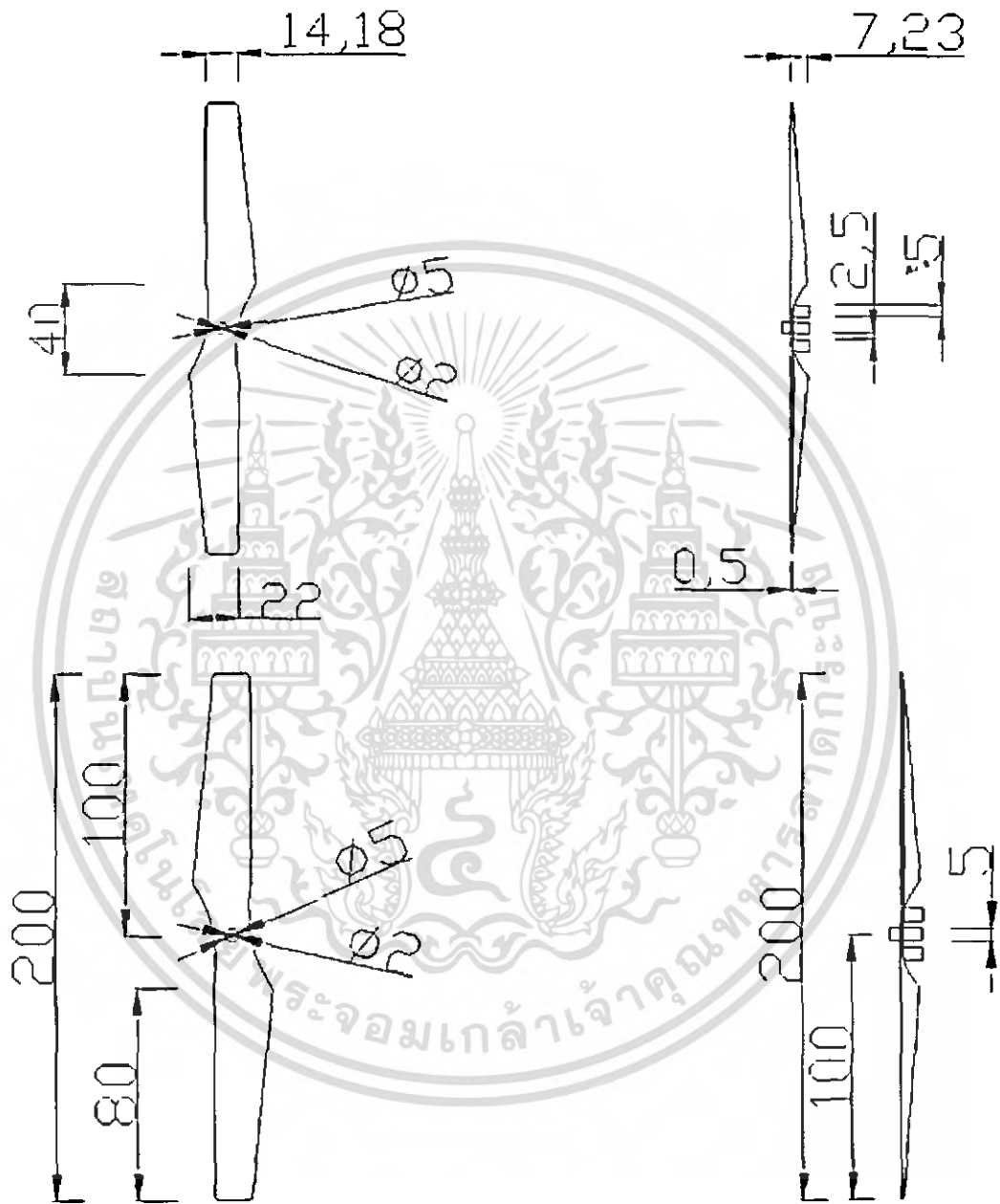
3. การออกแบบส่วนวงจรควบคุมความเร็วมอเตอร์ เป็นวงจรที่เรียกว่า Electronics Speed Control (ESC) เป็นอุปกรณ์ควบคุมความเร็วของมอเตอร์ที่ใช้ในการหมุนของใบพัด ให้มีความเร็วได้ตามความต้องการ เพื่อให้ตัวอากาศยานสามารถขึ้น-ลงได้ โดย ESC นี้จะมี FET ใน การขยายกระแสเพื่อส่งผ่านไปยังมอเตอร์ให้มีกระแสมากขึ้นซึ่งก็จะทำให้ออเตอร์หมุนได้เร็วขึ้นด้วย โดยจะแบ่งวงจรออกเป็นสองส่วนคือ ส่วนที่เป็นไมโครคอนโทรลเลอร์และส่วนที่เป็นวงจรควบคุมความเร็วมอเตอร์



รูปที่ 3 4 วงจรควบคุมความเร็วมอเตอร์

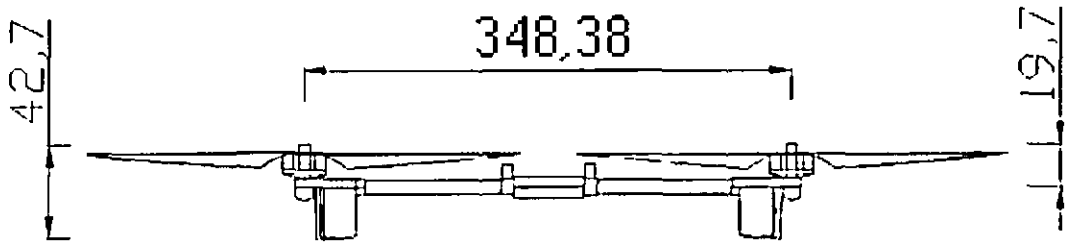
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ส่วนของโครงสร้าง

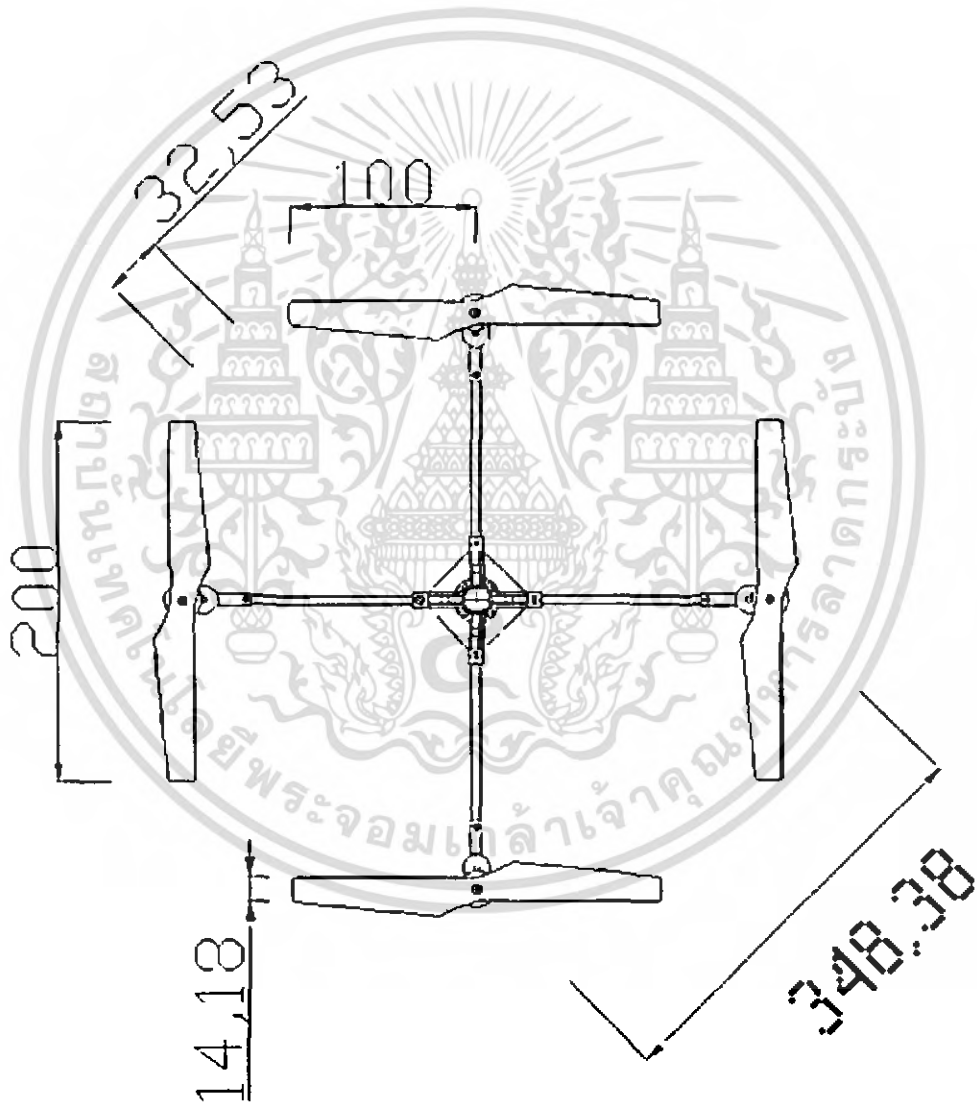


รูปที่ 3.5 แสดงโครงสร้างส่วนใบพัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



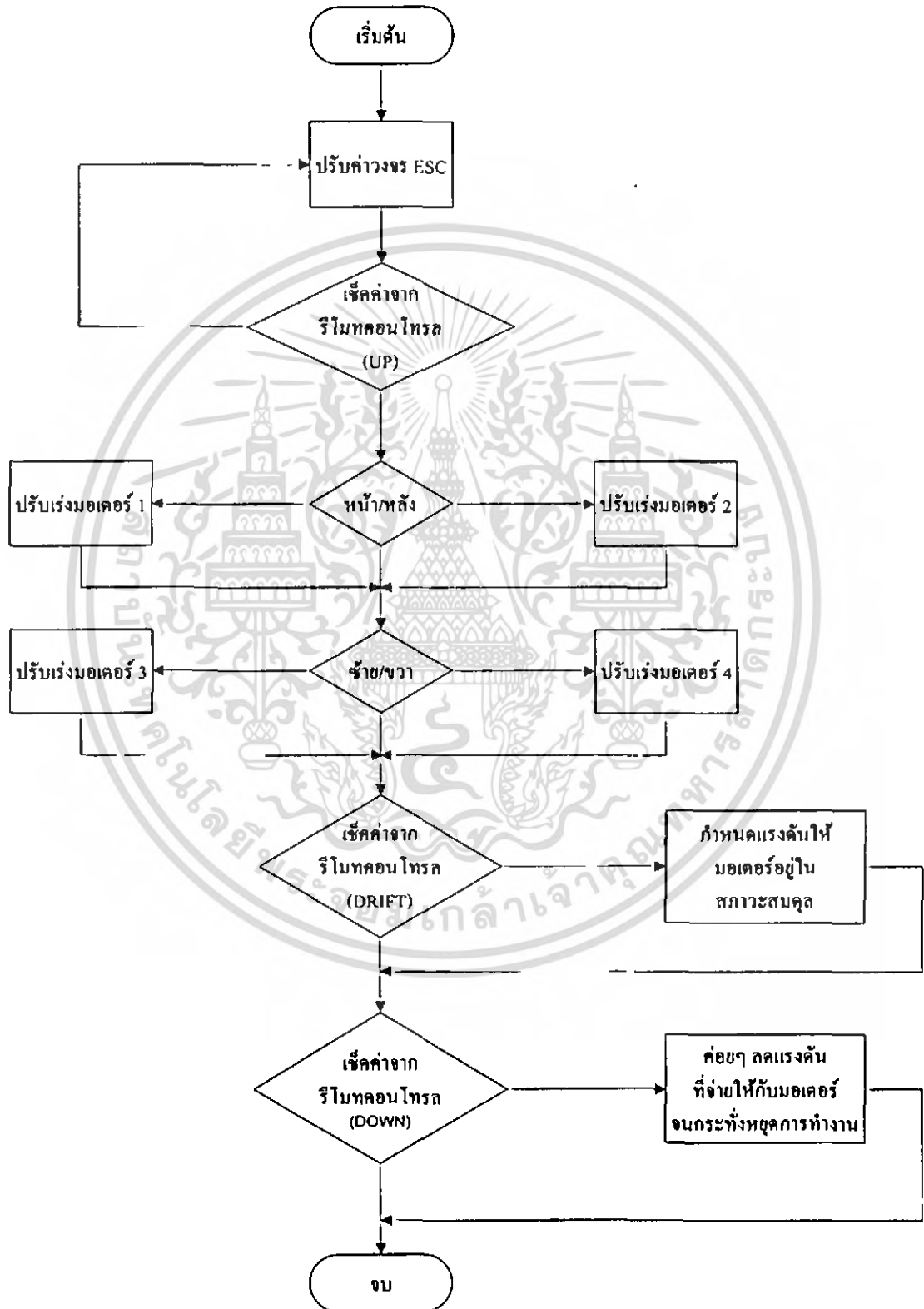
รูปที่ 3.6 โครงสร้างด้านข้างของอากาศยาน



รูปที่ 3.7 แสดงโครงสร้างทั้งหมดของอากาศยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างทางซอฟต์แวร์



รูปที่ 3 8 Flow Chart แสดงสถานะการทำงานของโปรแกรมควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ดูแลเห็นใบแจ้งประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 การทดลองวงจรขั้วมอเตอร์

จุดประสงค์ในการทดลองวงจรขั้วมอเตอร์

1. เพื่อทดสอบความเร็วรอบของมอเตอร์
2. เพื่อทดสอบแรงบิดของมอเตอร์
3. การกินกระแสของมอเตอร์

ในขั้นตอนการทดลอง จะต้องติดตั้ง ESC เข้ากับส่วนของไมโครคอนโทรลเลอร์ให้เรียบร้อยก่อน และต่อมอเตอร์แต่ไม่ต้องใส่ใบพัดเพื่อความปลอดภัยในการทดลอง ก่อนที่จะต่อ ESC ควรตรวจสอบสวิทช์ของ ESC ว่าปิดอยู่หรือไม่ แล้วจึงต่อเข้ากับแบตเตอรี่ แล้วทำการทดลองตามขั้นตอน ดังนี้

1. ทำการจ่ายแรงดันให้กับ ESC
2. ต่อสายสัญญาณ แดง-ขาว เข้ากับมอเตอร์
3. ต่อสายสัญญาณ แดง-ดำ เข้ากับแหล่งจ่ายไฟ
4. ทำการเปิดสวิทช์รีโมทบังคับวิทยุ และเปิดสวิทช์ ESC
5. เลื่อนคันบังคับวิทยุ ให้อยู่ที่ตำแหน่งต่ำสุด จะได้ยินเสียง บี๊ป 1 ครั้ง
6. ทำการโยกคันบังคับเพื่อทำการวัดสัญญาณ

รูปที่ได้จากการทดลองตามขั้นตอนข้างต้น เป็นรูปที่อยู่ทางด้าน Input และ Output ของ ESC แสดงดังนี้

4.2 ผลการทดลอง

สัญญาณทางด้านอินพุตที่วัดได้

เป็นการทดลองส่งสัญญาณเข้าเพื่อเช็คค่า ESC

ค่าความเร็วรอบ ของมอเตอร์(%)	t_{on}	t_{off}
0	1.00ms	22.00 ms
10	1.00ms	22.00 ms
20	1.20ms	21.80 ms
25	1.25ms	21.75 ms
30	1.27 ms	21.73 ms
35	1.30 ms	21.70 ms
40	1.32 ms	21.68 ms
45	1.34 ms	21.66 ms
50	1.35 ms	21.65 ms
55	1.40 ms	21.60 ms
60	1.42 ms	21.58 ms
65	1.45 ms	21.55 ms
70	1.47 ms	21.53 ms
75	1.50 ms	21.50 ms
80	1.53 ms	21.47 ms
85	1.60 ms	21.40 ms
90	1.65 ms	21.35 ms
95	1.67 ms	21.33 ms
100	2.00 ms	21.00 ms

ตารางที่ 4.1 แสดงค่าระดับสัญญาณเข้าเพื่อเช็คค่า ESC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

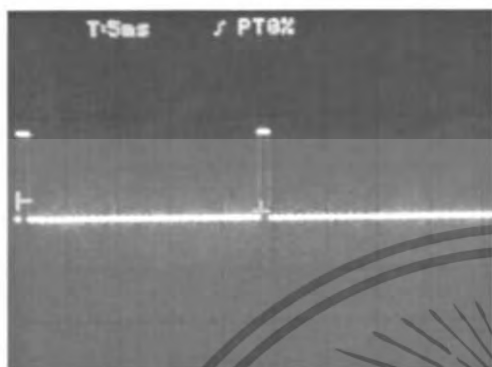
เป็นการทดลองสร้างสัญญาณที่เขียนขึ้นมาจาก MCS-51 โดยใช้การวนของช่วงเวลา
ซึ่งการวนของช่วงเวลา= เวลา $10 \mu\text{s}$ \times จำนวนรอบที่ทำการวนซ้ำ

จำนวนการวนรอบ (รอบ) ค่าความเร็วมอเตอร์(%)	t_{on}	t_{off}
0	50	1200
10	50	1200
20	60	1185
25	65	1175
30	67	1175
35	69	1175
40	70	1165
45	71	1165
50	72	1165
55	75	1165
60	76	1155
65	77	1155
70	78	1155
75	80	1155
80	82	1145
85	85	1145
90	88	1140
95	89	1140
100	106	1125

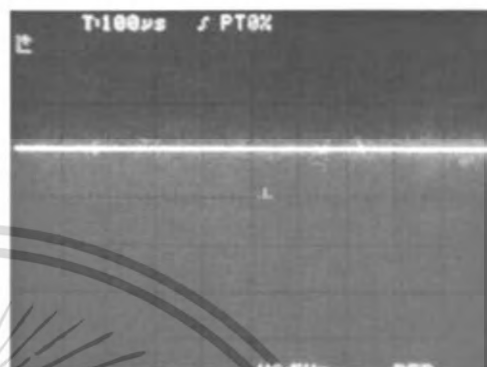
ตารางที่ 4 2 แสดงค่าระดับสัญญาณที่เขียนขึ้นมาจาก MCS51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจะได้รูปสัญญาณดังนี้
สัญญาณทางด้าน Input



รูปที่ 4.1 ที่ค่า Input 0 %



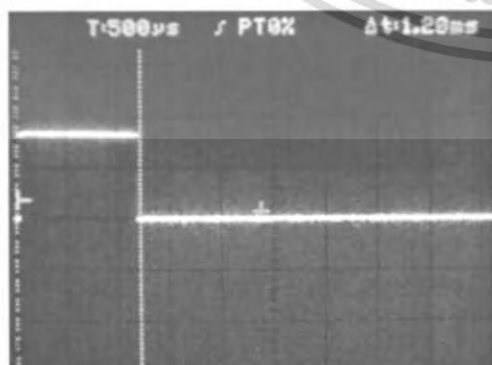
รูปที่ 4.2 ที่ค่า Output 0 %



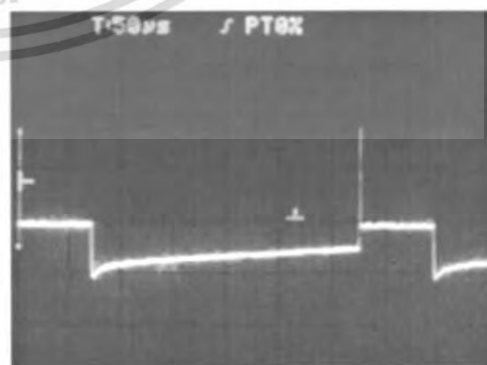
รูปที่ 4.3 ที่ค่า Input 10 %



รูปที่ 4.4 ที่ค่า Output 10 %

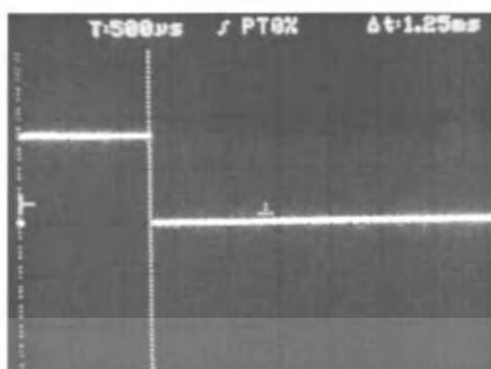


รูปที่ 4.5 ที่ค่า Input 20 %

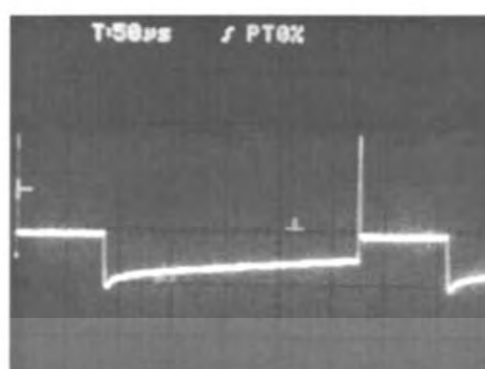


รูปที่ 4.6 ที่ค่า Output 20 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ที่ค่า Input 25 %



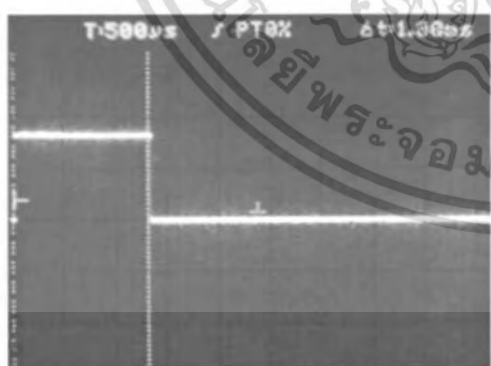
รูปที่ 4.8 ที่ค่า Output 25 %



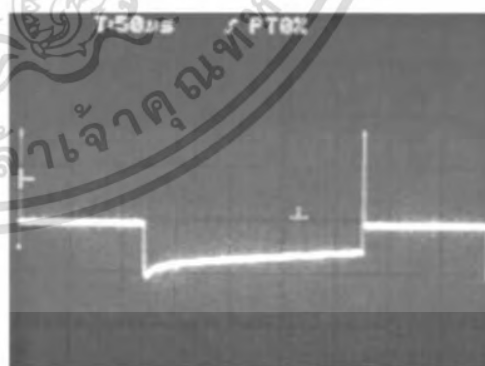
รูปที่ 4.9 ที่ค่า Input 30 %



รูปที่ 4.10 ที่ค่า Output 30 %

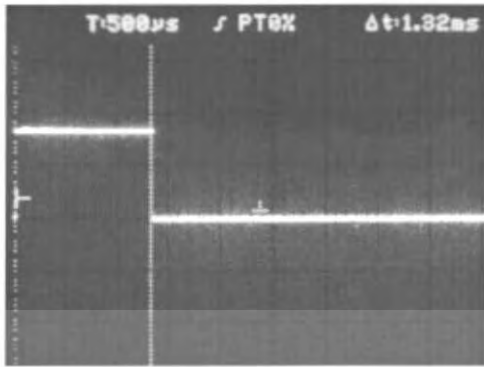


รูปที่ 4.11 ที่ค่า Input 35 %

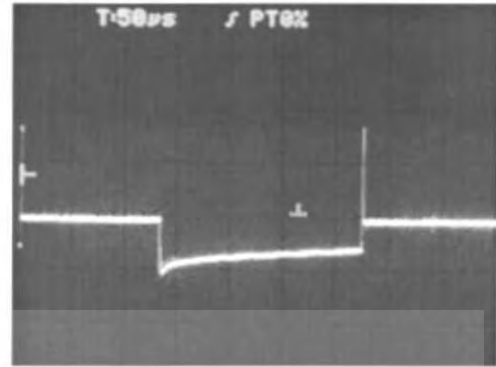


รูปที่ 4.12 ที่ค่า Output 35 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



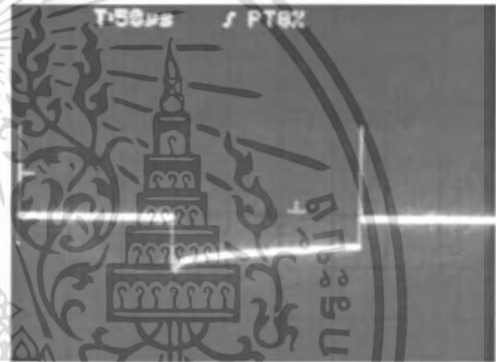
รูปที่ 4.13 ที่ค่า Input 40 %



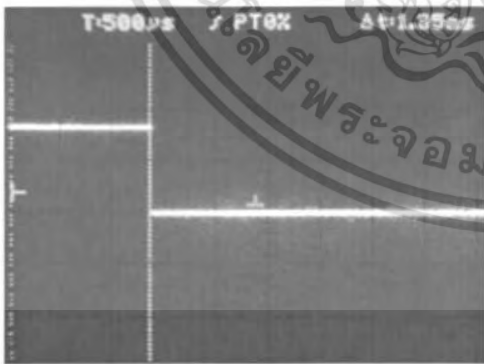
รูปที่ 4.14 ที่ค่า Output 40 %



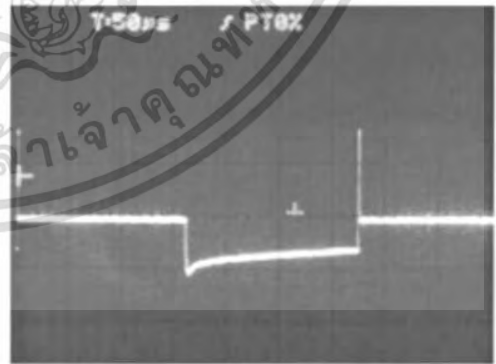
รูปที่ 4.15 ที่ค่า Input 45 %



รูปที่ 4.16 ที่ค่า Output 45 %

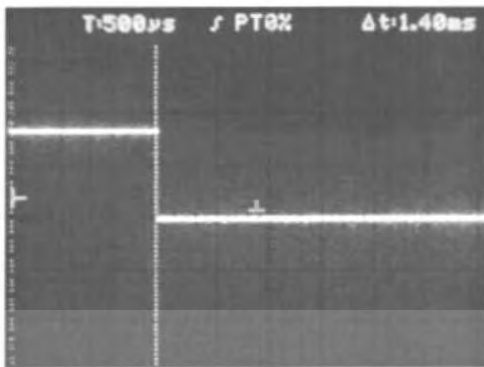


รูปที่ 4.17 ที่ค่า Input 50 %

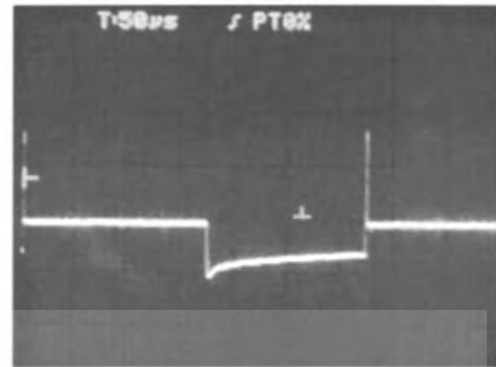


รูปที่ 4.18 ที่ค่า Output 50 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



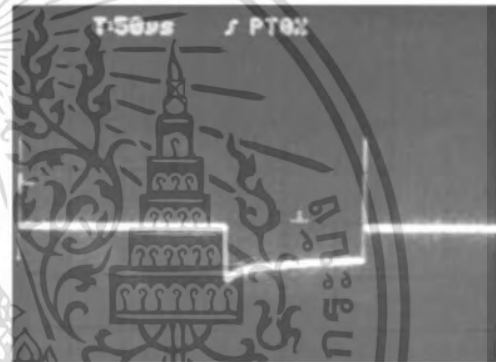
รูปที่ 4.19 ที่ค่า Input 55 %



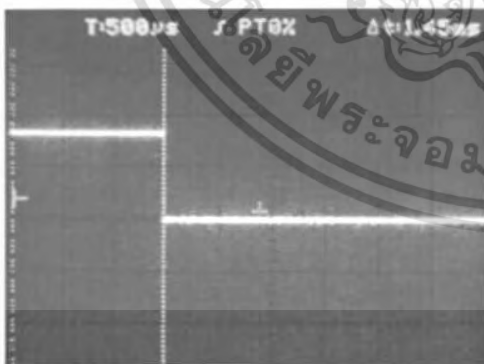
รูปที่ 4.20 ที่ค่า Output 55 %



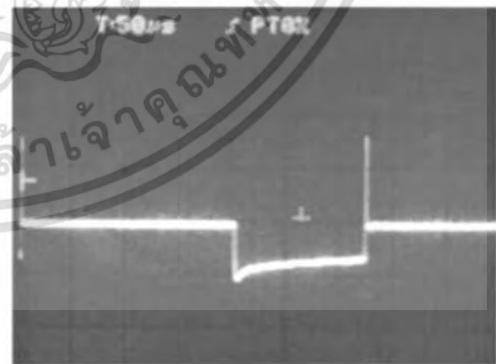
รูปที่ 4.21 ที่ค่า Input 60 %



รูปที่ 4.22 ที่ค่า Output 60 %

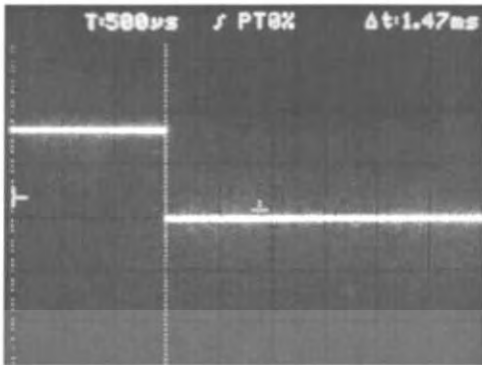


รูปที่ 4.23 ที่ค่า Input 65 %

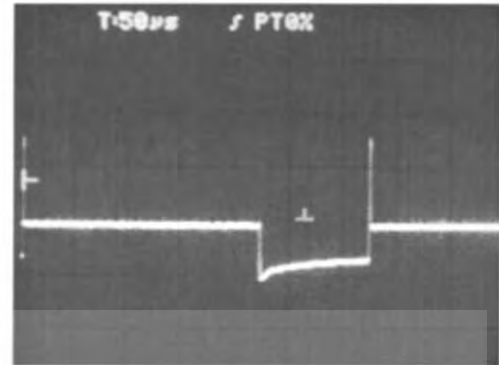


รูปที่ 4.24 ที่ค่า Output 65 %

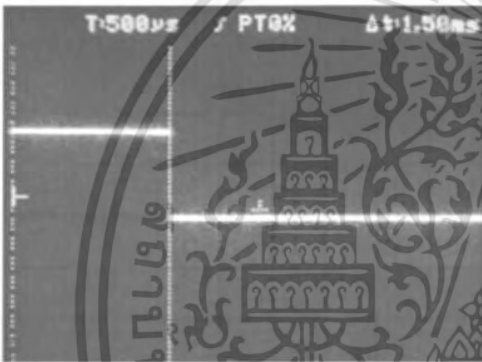
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



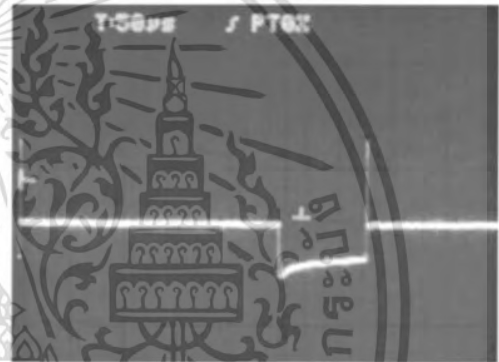
รูปที่ 4.25 ที่ค่า Input 70 %



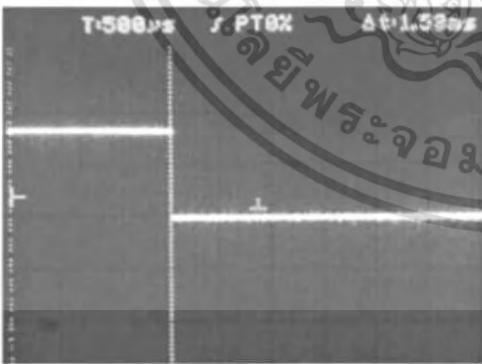
รูปที่ 4.26 ที่ค่า Output 70 %



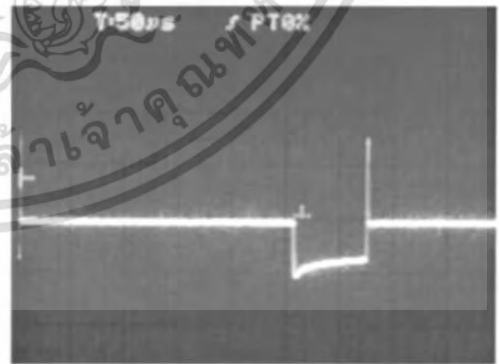
รูปที่ 4.27 ที่ค่า Input 75 %



รูปที่ 4.28 ที่ค่า Output 75 %

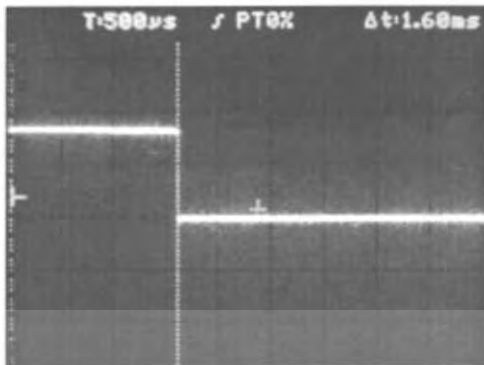


รูปที่ 4.29 ที่ค่า Input 80 %

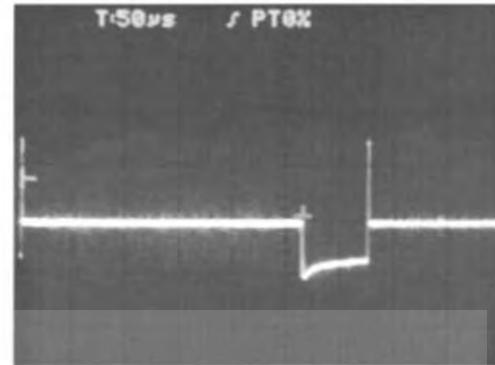


รูปที่ 4.30 ที่ค่า Output 80 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 ที่ค่า Input 85 %



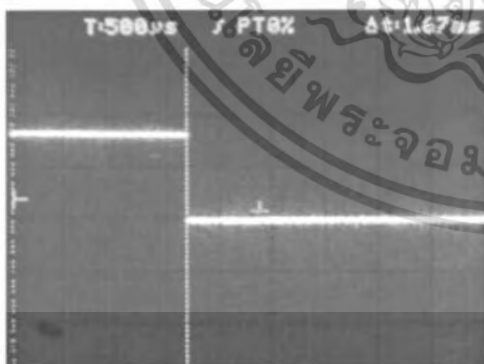
รูปที่ 4.32 ที่ค่า Output 85 %



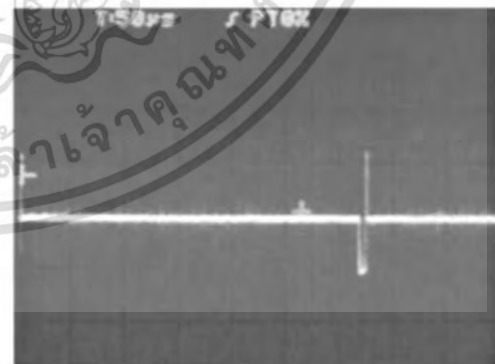
รูปที่ 4.33 ที่ค่า Input 90 %



รูปที่ 4.34 ที่ค่า Output 90 %

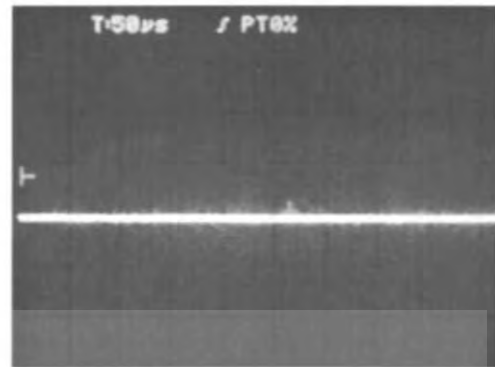
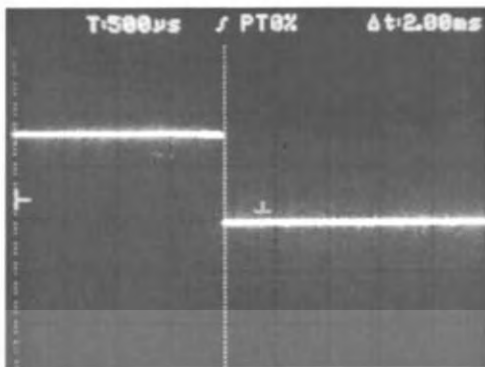


รูปที่ 4.35 ที่ค่า Input 95 %



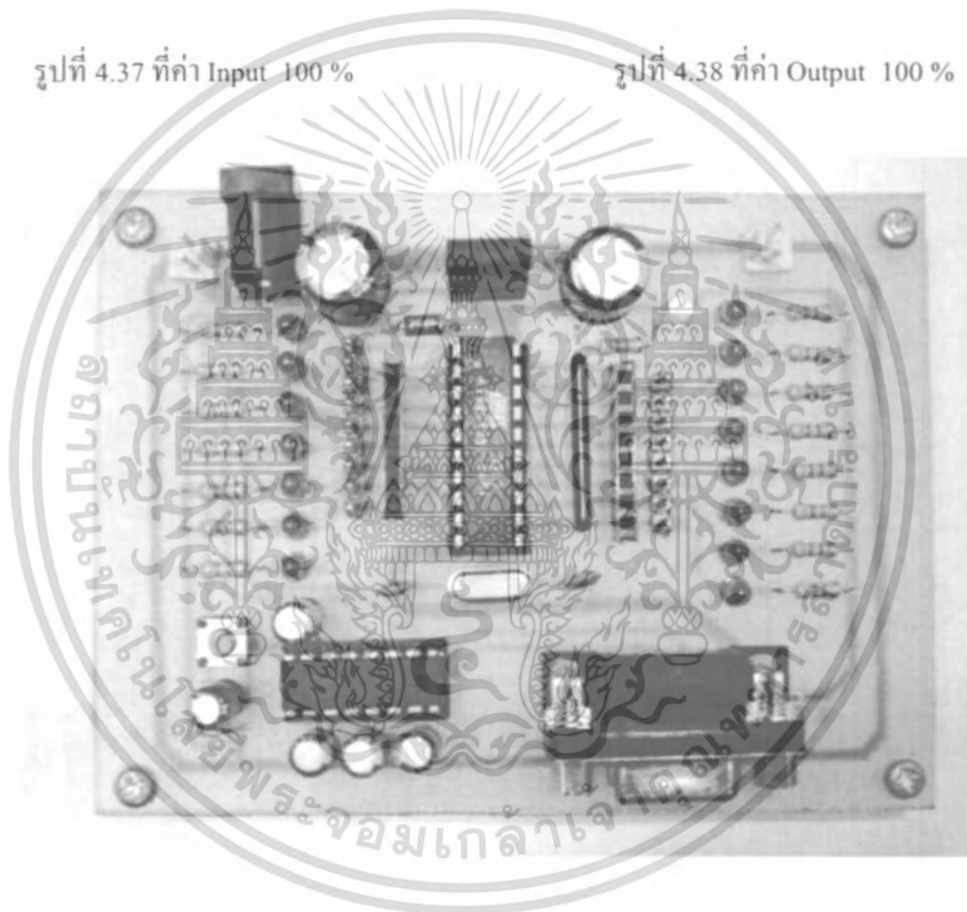
รูปที่ 4.36 ที่ค่า Output 95 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 ที่ค่า Input 100 %

รูปที่ 4.38 ที่ค่า Output 100 %



รูปที่ 4.39 แสดงการต่ออุปกรณ์ต่างๆ ลงบอร์ดทางด้านส่ง

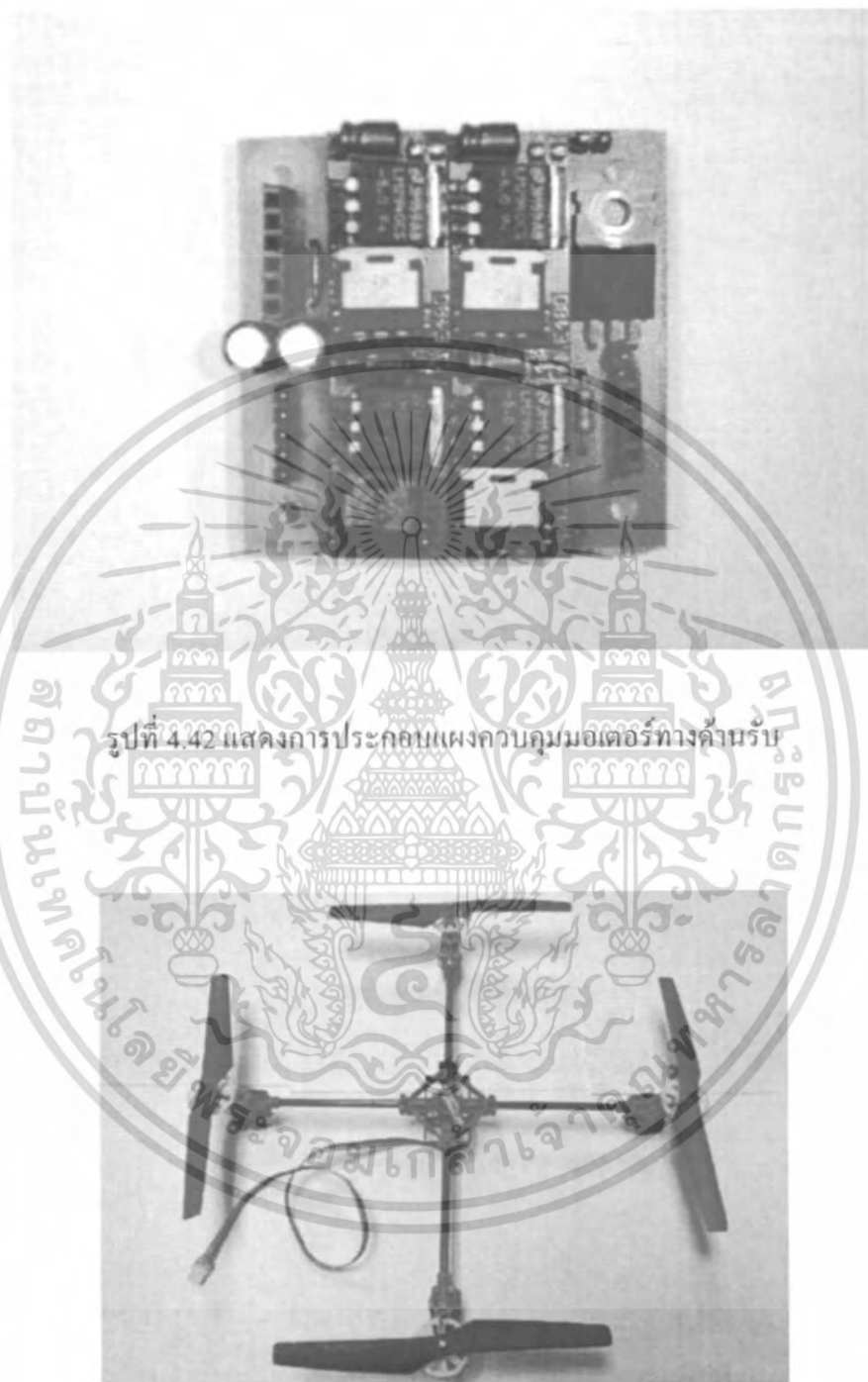
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 แสดงการต่อแผงควบคุมไอซี TRW 2.4 G ทางด้านรับ

รูปที่ 4.41 แสดงการต่อแผงควบคุมไอซี TRW 2.4 G ทางด้านส่ง

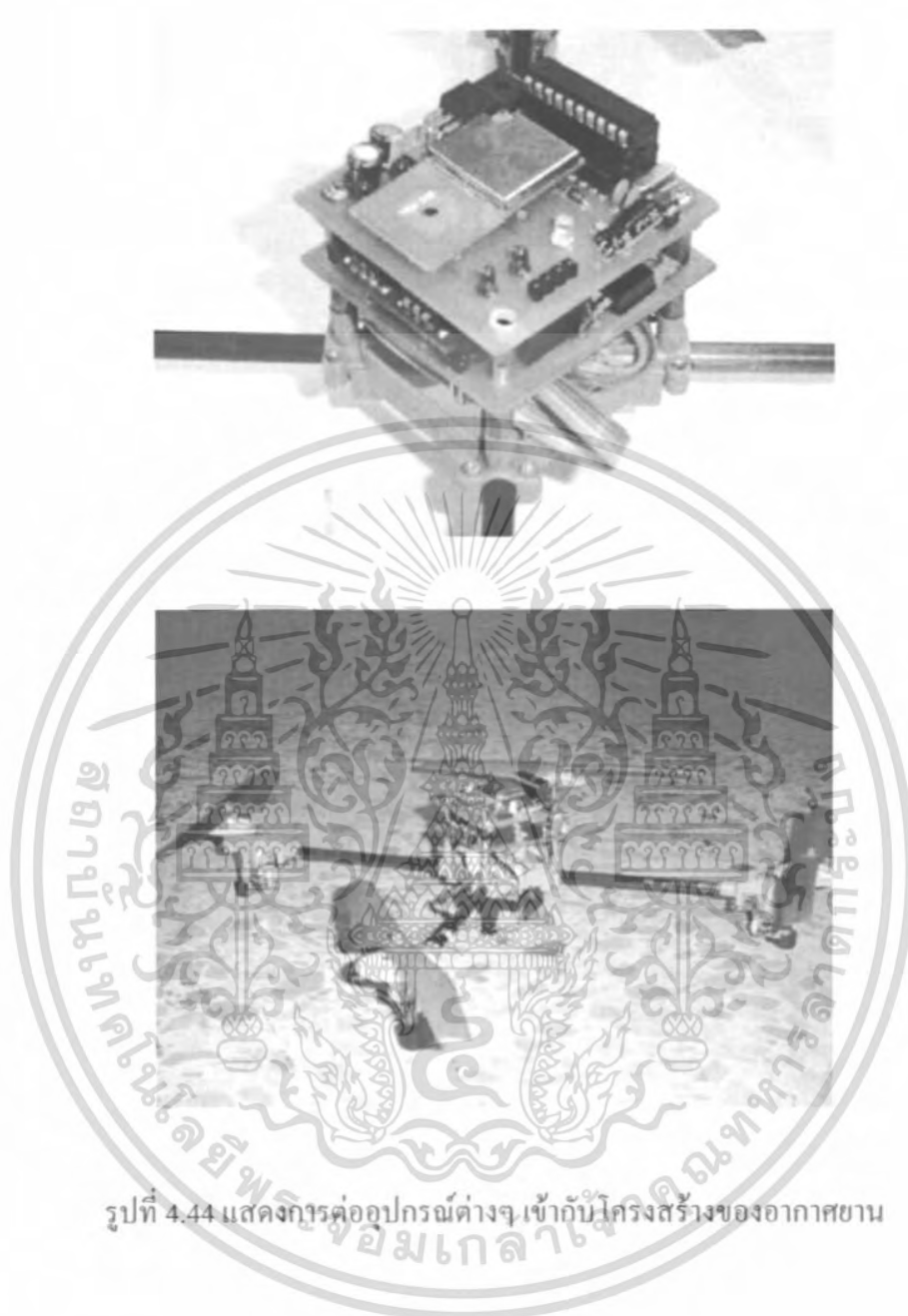
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 แสดงการประกอบแผงควบคุมมอเตอร์ทางด้านรับ

รูปที่ 4.43 แสดงการประกอบ โครงสร้างของอากาศยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 แสดงการต่ออุปกรณ์ต่างๆ เข้ากับ โครงสร้างของอากาศยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและปัญหาการทำโครงการ

5.1 สรุปผลการทำโครงการ

จากแนวความคิดที่จะต้องการสร้างอากาศยานแบบ 4 ใบพัด มาเพื่อใช้ประโยชน์ในด้านการสำรวจพื้นที่ซึ่งยากต่อการเข้าถึง ซึ่งปรากฏว่าสามารถควบคุมให้อากาศยานบินขึ้นไปในอากาศได้แต่ยังไม่สมบูรณ์เท่าที่ควร เนื่องจากยังไม่สามารถเคลื่อนที่ไปทางซ้ายและขวาได้ตามขอบเขตที่กำหนดไว้แต่แรก

5.2 ปัญหาที่พบและการแก้ไข

1. โครงสร้างของอากาศยาน ไม่มีความสมดุล ซึ่งแก้ไขได้โดยการใช้โปรแกรมไปควบคุมความเร็วรอบของมอเตอร์ทั้ง 4 ตัว ให้มีความสมดุลกันทั้งหมด และได้ทำการเปลี่ยนโครงสร้างบางส่วนของอากาศยานให้มีความสมดุลมากขึ้น
2. เกิดความร้อนขึ้นที่ไอซีเบอร์ L293D ซึ่งเป็นไอซีที่ใช้ควบคุมมอเตอร์ การเกิดความร้อนนี้ส่งผลให้เครื่องอากาศยานหยุดการทำงานลง เมื่อความร้อนอยู่ในระดับวิกฤต การแก้ไขทำได้โดยการเปลี่ยนไปใช้ไอซีเบอร์ IRF2804S ซึ่งเป็น MOSFET ที่สามารถทนความร้อนได้ดีกว่าไอซีเบอร์ L293D
3. ความเร็วรอบของมอเตอร์แต่ละตัวไม่เท่ากัน การแก้ไขคือ เปลี่ยนไปใช้วงจร ESC ในการควบคุมความเร็วของมอเตอร์แทนไอซีเบอร์ L293D
4. ไอซีไมโครคอนโทรลเลอร์ที่ใช้เป็นเบอร์ AT89C52 ซึ่งเป็นไอซีที่มีขนาดใหญ่และใช้กระแสไฟจำนวนมากเพื่อทำให้ไอซีทำงานได้ การแก้ไข คือ เปลี่ยนไอซีไมโครคอนโทรลเลอร์เป็นเบอร์ AT89C4051 เพราะมีขนาดเล็กกว่าและใช้กระแสไฟจำนวนไม่มากนักที่จะทำให้ไอซีไมโครคอนโทรลเลอร์ทำงานได้ ส่งผลให้แผงวงจรที่ตัวอากาศยานเล็กลงด้วย ทำให้น้ำหนักโดยรวมน้อยลง



รูปที่ 5.1 แสดงโครงสร้างของอากาศยาน เมื่อมีการเปลี่ยนแปลงความยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของแกนใบพัดทั้ง 4 ด้าน

5.3 แนวทางในการพัฒนาต่อ

ในการดำเนินงานในการสร้างปริญญานิพนธ์นี้ ยังสามารถที่จะประยุกต์และคิดแปลงในส่วนต่างๆ ของโครงงานนี้ เพื่อนำมาใช้ประโยชน์อื่นๆ ตามความประสงค์ได้อย่างกว้างขวางและก่อให้เกิดการพัฒนาเครื่องมือหรืออุปกรณ์ต่างๆ เพื่อนำมาใช้ให้ประโยชน์ในชีวิตประจำวันได้

5.3.1 เพิ่มอุปกรณ์ในการสำรวจ

อากาศยานสามารถพัฒนาต่อให้มีความสามารถในการสำรวจได้ โดยการเพิ่มอุปกรณ์ในการสำรวจที่ตัวอากาศยาน เช่น กล้องวงจรปิดแบบไร้สาย และอุปกรณ์ตรวจสอบก๊าซต่างๆ

5.3.2 ใช้อุปกรณ์ที่มีน้ำหนักเบา

การที่จะทำให้อากาศยานสามารถเคลื่อนที่ได้อย่างคล่องตัว อากาศยานนั้นจะต้องมีน้ำหนักที่เบา จึงต้องเลือกใช้อุปกรณ์ที่มีน้ำหนักเบา ซึ่งในโครงงานนี้สามารถลดน้ำหนักของอากาศยานได้หลายส่วน แต่ก็ยังมีส่วนที่ลดน้ำหนักไม่ได้คือแบตเตอรี่

5.3.3 พัฒนาให้อากาศยานมีรัศมีการควบคุมที่ไกลมากขึ้น

ยิ่งอากาศยานสามารถควบคุมได้ไกลมากขึ้นเท่าใด ก็จะทำให้ขีดความสามารถในการสำรวจพื้นที่ไปได้ไกลมากขึ้น

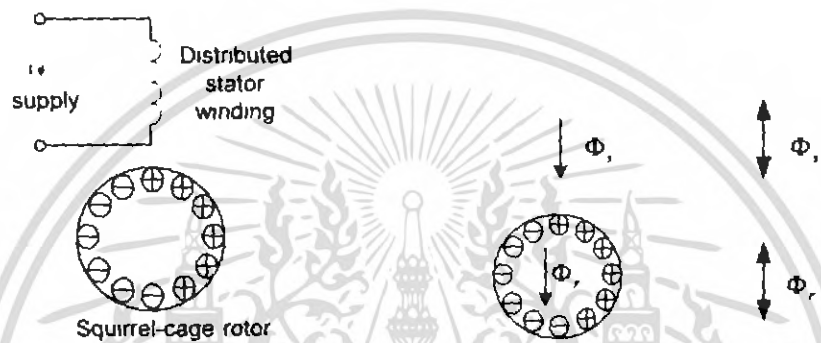
บรรณานุกรม

1. อุดม รานอก ภาษา C สำหรับงานควบคุม ไมโครคอนโทรลเลอร์ MCS-51. นนทบุรี . ไอดีซีฯ, 2548
2. วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฉบับ AT89C5x ของ Atmel. กรุงเทพฯ . อินโนเวทีฟ เอ็ดจิวเอร์เมนท์ จำกัด, 2542
3. ชัยวัฒน์ ประกอบผล การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. กรุงเทพฯ . สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2548
4. คอนสัน ปงผาว. ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน. กรุงเทพฯ . สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2549
5. สัจจะ จรัสรุ่งรวิวาร คู่มือเขียนโปรแกรม VISUAL BASIC 6 ฉบับ ผู้เริ่มต้น. นนทบุรี ไอดีซี อินโฟฯ, 2548
6. http://www.electron.rmutphysics.com/physicsglossary/index.php?option=com_frontpage&Itemid=1
7. <http://www.bingun.com/article.php?id=2070&lang=th>
8. http://www.geocities.com/aon_micro/
9. <http://www.inex.co.th/micro/whatismicro.html>
10. <http://www.sut.ac.th/Engineering/Telecom/about/student2/ST51.htm>
11. <http://www.cpe.ku.ac.th/~yuen/204471/micro/mcs51/>
12. http://www.geocities.com/ch_kwang/12.html
13. <http://micro-research.co.th/MCS-51-2.htm>
14. <http://micro-research.co.th/MCS-51-1.htm>
15. <http://schoolnet.nectec.or.th/library/webcontest2003/100team/dlncs137/am/Microcontroller.html>
16. www.csie.nctu.edu.tw/~java/jdbc/z_others/jboss/fivedots.cocpsu.ac.th/Software.coc/240-380/PDF/report.doc
17. http://www.cpe.ku.ac.th/~yuen/204471/device/diode_transistor/mosfet.htm
18. <http://61.19.247.211/~adisak51/ebook.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

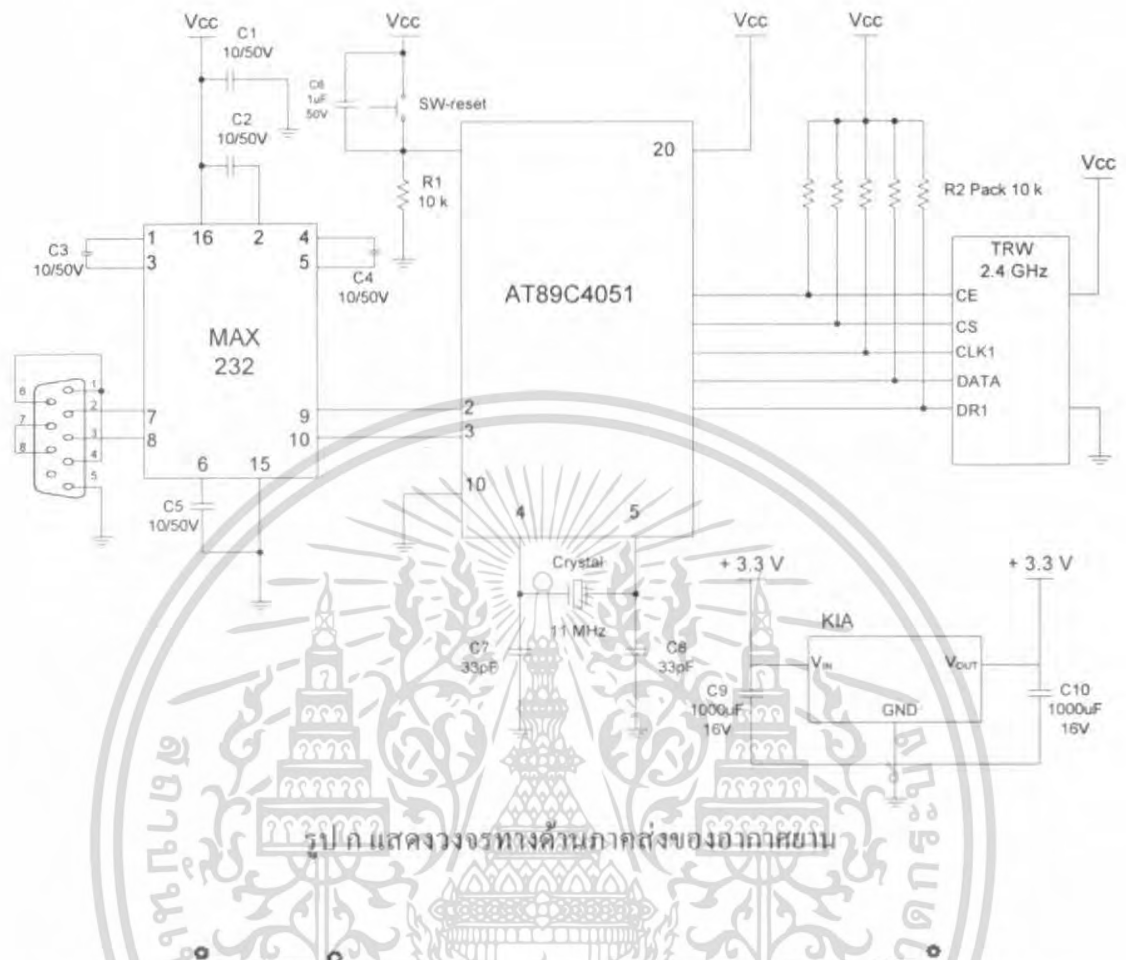


(ก) ลักษณะโครงสร้าง

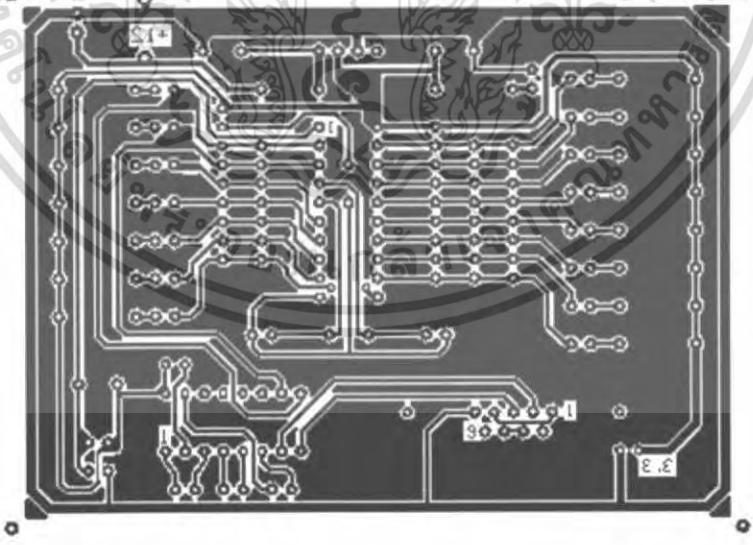
(ข) เส้นแรงแม่เหล็ก

<http://mylesson.swu.ac.th/ee420/4-2.htm>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

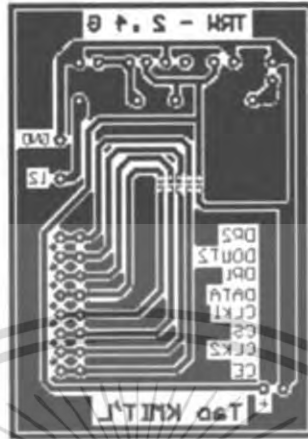


รูป ก แสดงวงจรทางด้านภาคส่งของอากาศยาน



รูป ข แสดงลายปรินวงจรทางด้านภาคส่งของอากาศยานส่วนที่เป็น ไมโครคอนโทรลเลอร์

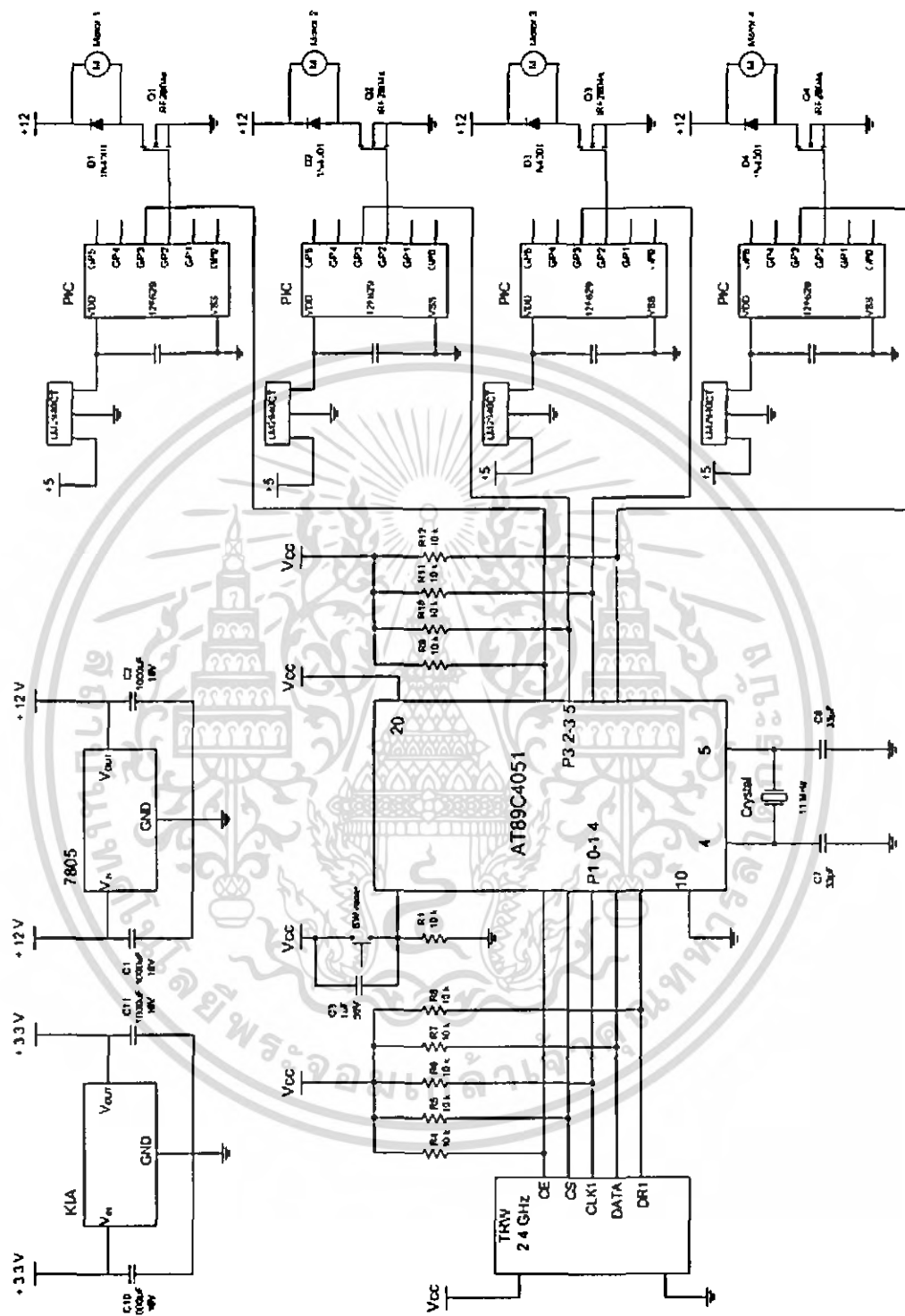
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ค แสดงลายปริ้น วงจรทางด้านภาคส่งของอากาศยานส่วนที่เป็น
ไอซีโมดูลรับ-ส่ง TRW 2.4 G

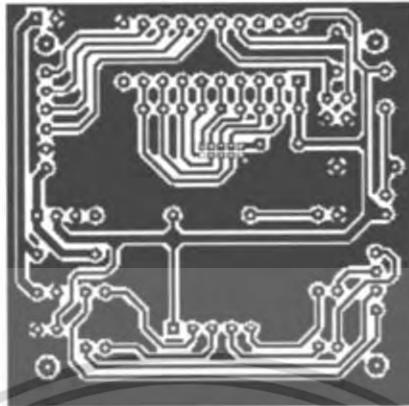


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

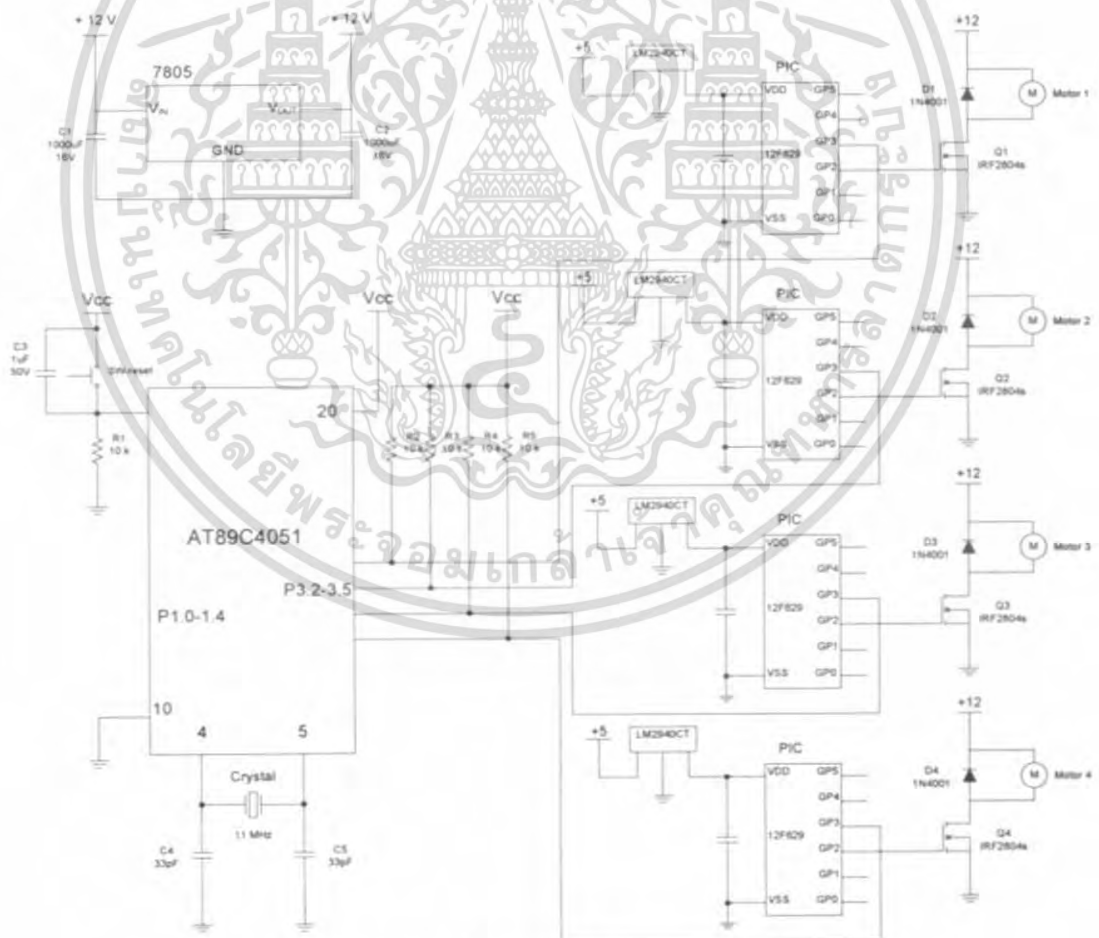


รูป แสดงวงจรทางค่านภาครับของอากาศยาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



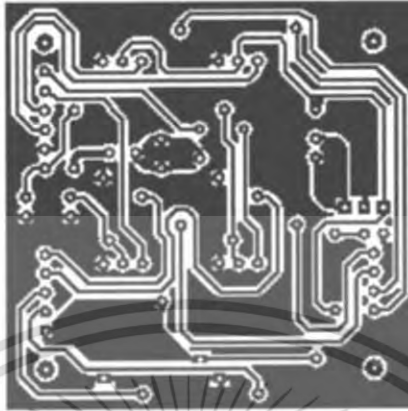
รูป จ แสดงลายปริ้นวงจรทางด้านภาครับของอากาศยานส่วนที่เป็น ไมโครคอนโทรลเลอร์และ ไอซี ในคู่มือ-ส่ง TRW 2.4 G



รูป ฉ แสดงวงจรทางด้านภาครับของอากาศยานส่วนที่เป็น

วงจรควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ช แสดงลายปริ้นวงจรทางค่านภาครับของอภาสยวนส่วนที่เป็น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในไมโครคอนโทรลทางด้านภาคส่ง

```
/******  
  
#pragma code  
  
#include <reg51 h>  
#include <intrins h>  
#include <stdio h>  
#include <trw_tx.c>  
#include <delay.c>  
#include <serial c>  
  
unsigned char ans,comp=0;  
unsigned char rj,  
bit w,  
  
/******  
/* interrupt rs-232 */  
/******  
void serial_inter()interrupt 4  
{  
if(RI)  
{  
ans=SBUF,  
if(comp==ans)  
{  
w=~w;  
}  
else  
{  
w=1,  
}  
comp=ans;  
RI=0,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(TI)
{
TI=0;
}
}

/*****/

/*      Main Program      */
/*****/

void main (void)
{
    serial();
    delay_1ms(500),
    TX_LED = 0;
    delay_1ms(500),
    TX_LED = 1,

    Data = 0,
    CLK1 = 0;
    CE = 0;
    CS = 0;
    delay_1ms(250),

    config_TRW24(1), // mode 1 to send mode

    delay_1ms(100),
    while(1)
    {
        switch(ans)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0x61. // go
if(w==1)
{
    dclay_1ms(50);
if(ans==0x61)
{
    TX_LED = 0,
send_TRW24(0x41);
SBUF = 0x41,
TX_LED = 1;
}
}
break,
//=====//
case 0x62: // back
if(w==1)
{
    dclay_1ms(50);
if(ans==0x62)
{
    TX_LED = 0;
send_TRW24(0x42),
SBUF = 0x42,
TX_LED = 1;
}
}
break;
//=====//
case 0x63 // left
if(w==1)
{
    dclay_1ms(50);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(ans==0x63)
{
    TX_LED = 0;
    send_TRW24(0x43);
    SBUF = 0x43;
    TX_LED = 1,
}
}

```

break;

```

//=====//

```

```

case 0x64: // right

```

```

if(w==1)

```

```

{

```

```

    delay_1ms(50);

```

```

    if(ans==0x64)

```

```

    {

```

```

        TX_LED = 0;

```

```

        send_TRW24(0x44);

```

```

        SBUF = 0x44;

```

```

        TX_LED = 1,

```

```

    }

```

```

}

```

break;

```

//=====//

```

```

case 0x65 // motor 1

```

```

if(w==1)

```

```

{

```

```

    delay_1ms(50);

```

```

    if(ans==0x65)

```

```

    {

```

```

        TX_LED = 0,

```

```

        send_TRW24(0x45).

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SBUF = 0x45;
        TX_LED = 1,
    }
}

break;

//=====//

case 0x66.    // motor 2
    if(w==1)
    {
        delay_1ms(50);
        if(ans==0x66)
        {
            TX_LED = 0;
            send_TRW24(0x46);
            SBUF = 0x46;
            TX_LED = 1;
        }
    }
break;

//=====//

case 0x67.    // motor 3
    if(w==1)
    {
        delay_1ms(50),
        if(ans==0x67)
        {
            TX_LED = 0,
            send_TRW24(0x47),
            SBUF = 0x47,
            TX_LED = 1,
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

//=====//
case 0x68:    // motor 4
if(w==1)
{
    delay_1ms(50);
if(ans==0x68)
{
    TX_LED = 0,
    send_TRW24(0x48),
    SBUF = 0x48;
    TX_LED = 1;
}
}
break,
//=====//
case 0x69    // set csc
if(w==1)
{
    delay_1ms(50);
if(ans==0x69)
{
    TX_LED = 0;
    send_TRW24(0x49);
    SBUF = 0x49,
    TX_LED = 1,
}
}
break;

//=====//
case 0x6A:    // up
if(w==1)

```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        delay_1ms(50),
    if(ans==0x6A)
    {
        TX_LED = 0,
        send_TRW24(0x50),
        SBUF = 0x50;
        TX_LED = 1,
    }
    }
break;

//=====//
case 0x6B: // down
if(w==1)
{
    delay_1ms(50);
if(ans==0x6B)
{
    TX_LED = 0;
    send_TRW24(0x51);
    SBUF = 0x51;
    TX_LED = 1;
}
}
break;

//=====//
case 0x6C: // test 4 rotor
if(w==1)
{
    delay_1ms(50);
if(ans==0x6C)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        TX_LED = 0;
        send_TRW24(0x52);
        SBUF = 0x52,
        TX_LED = 1,
    }
}
break,
//=====//
case 0x6D // 4 rotor stop
if(w==1)
{
    delay_1ms(50),
if(ans==0x6D)
{
    TX_LED = 0,
    send_TRW24(0x53);
    SBUF = 0x53,
    TX_LED = 1;
}
}
break;
//=====//
default
SBUF=0x61;
TX_LED = 0,
send_TRW24(0x61),
TX_LED = 1;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/*interrupt RS-232 */
/*****/

void serial()
{
    SCON=0x50,
    TMOD=0x20;
    TH1=0xFD, // 9600 bps //

    EA=1;
    ES=1,
    TR1=1,
    TI=1,
}

//=====//

/*****/
/* set TRW 2.4 GHz */
/*****/

sbit Data = P1^4; // Port send data to module 2.4G chanel 1
sbit CLK1 = P1^3; // Port Clock befor send data chanel 1
sbit CS = P1^2; // Port chip select
sbit CE = P1^0; // Port Enable Shift

sbit TX_LED = P3^7;

void dmsec(unsigned int count),

//=====//

void Wait(unsigned int x)
{
    unsigned int i,
    for (i=0,i<x;i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
//=====//
void CLK_TRW24(void)
{
    CLK1 = 0,
    dmsec(1);
    CLK1 = 1,
    dmsec(1),
}
//=====//
void Write_TRW24(unsigned char Dat)
{
    unsigned char i,
    bit Out;
    for (i=0,i<8,i++)
    {
        Out = Dat & 0x80,
        Data = Out,
        CLK_TRW24(),
        Dat = Dat << 1;
    }
}
//=====//
void config_TRW24( unsigned char Mode)
{
    Wait(500),
    CE = 0;
    CS = 1;
    Write_TRW24(0x8E), /* Reserved for testing */
    Write_TRW24(0x08), /* Reserved for testing */
    Write_TRW24(0x1C); /* Reserved for testing */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Write_TRW24(0x08); /* Length of Bit Ch 2 */
Write_TRW24(0x08), /* Length of Bit Ch 1 */

Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
Write_TRW24(0xAA);
Write_TRW24(0x55),
Write_TRW24(0xAA),
Write_TRW24(0x55);

```

```

Write_TRW24(0xAA), /* Address 5 Byte Ch 1 */
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(0xAA),

Write_TRW24(0xA3); /* Number of Address bit + CRC */
Write_TRW24(0x4F), /* RF Programming */

```

```

if (Mode == 1) /* Tx Mode */
{
Write_TRW24(0x14);
}

```

```

else /* Rx_Mode */
{

```

```

Write_TRW24(0x15);

```

```

Data=1,

```

```

CE=1;

```

```

}

```

```

CS = 0.

```

```

Wait(200),

```

```

}

```

```

=====

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void send_TRW24(unsigned char dat)
{
    Wait(100);
    CS = 0;
    CE = 1;
    Write_TRW24(0xAA), // address 5 byte send
    Write_TRW24(0x55); // address 5 byte send
    Write_TRW24(0xAA), // address 5 byte send
    Write_TRW24(0x55); // address 5 byte send
    Write_TRW24(0xAA), // address 5 byte send

    Write_TRW24(dat), // data
    Wait(100),
    CLK1 = 0,
    CE = 0;
    Wait(100);
}
//=====//
void dmsec (unsigned int count) // mSec Dclay 11.0592 Mhz
{
    unsigned int i,
    while (count)
        {
            i = 225, while (i>0) i--,
            count--,
        }
}
//=====//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/*delay 11.0592MHz*/
/*****/

/*void delay_5us(void)//delay 5us
{int i;
for(i=0,i<1;i++);
}

=====//

void delay_10us(int x)//delay 10us
{
int i;
for(i=0,i<x,i++)
_nop_(),
}
=====//

void delay_200us(int x)//delay 200us
{int i,j,
for(i=0;i<x,i++)
for(j=0,j<20,j++),
}*/
=====//

void delay_1ms(int x)//delay 1ms
{
int i,j,
for(i=0,i<x,i++)
for(j=0,j<100,j++),
}
=====//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในไมโครคอนโทรลทางด้านภาครับ

```
/*-----*/
#include <reg51.h>
#include <intrins.h>
#include <stdio.h>
//#include <scrial.c>
#include <sct_esc.c>
#include <delay_10us.c>

/*-----*/
/* Define I/O Port */
/*-----*/
unsigned char i,j,mode;

sbit Data = P1^4; // Port send data to module 2 4G chanel 1
sbit CLK1 = P1^3, // Port Clock befor send data chanel 1
sbit DR1 = P1^5; // port select data to chanel 1

sbit CS = P1^2, // Port chip select
sbit CE = P1^0;

sbit motor1 = P3^5;
sbit motor2 = P3^4;
sbit motor3 = P3^3,
sbit motor4 = P3^2,

//=====//

void Wait(unsigned int x)
{
    unsigned int i;
    for (i=0,i<x,i++)
    {}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/* Subrutine for Delay time      */
/*****/
void dmsec (unsigned int count) // mSec Delay 11.0592 Mhz
{
    unsigned int i,
    while (count) {
        i = 225; while (i>0) i--,
        count--,
    }
}
//=====//
void CLK_TRW24(void)
{
    CLK1 = 0;
    dmsec(1);
    CLK1 = 1;
    dmsec(1);
}
//=====//
void Write_TRW24(unsigned char Dat)
{
    unsigned char i,
    bit Out,
    for (i=0;i<8;i++)
    {
        Out = Dat & 0x80;
        Data = Out;
        CLK_TRW24();
        Dat = Dat << 1,
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//=====//
```

```
void config_TRW24( unsigned char Mode)
```

```
{
```

```
    Wait(500),
```

```
    CE = 0,
```

```
    CS = 1,
```

```
    Write_TRW24(0x8E), /* Reserved for testing */
```

```
    Write_TRW24(0x08), /* Reserved for testing */
```

```
    Write_TRW24(0x1C), /* Reserved for testing */
```

```
    Write_TRW24(0x08); /* Length of Bit Ch 2 */
```

```
    Write_TRW24(0x08); /* Length of Bit Ch 1 */
```

```
    Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
```

```
    Write_TRW24(0xAA);
```

```
    Write_TRW24(0x55);
```

```
    Write_TRW24(0xAA);
```

```
    Write_TRW24(0x55);
```

```
    Write_TRW24(0xAA), /* Address 5 Byte Ch 1 */
```

```
    Write_TRW24(0x55);
```

```
    Write_TRW24(0xAA),
```

```
    Write_TRW24(0x55);
```

```
    Write_TRW24(0xAA);
```

```
    Write_TRW24(0xA3), /* Number of Address bit + CRC */
```

```
    Write_TRW24(0x4F), /* RF Programming */
```

```
    if (Mode == 1) /* Tx Mode */
```

```
    {
```

```
        Write_TRW24(0x14);
```

```
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else /* Rx_Mode */
{
    Write_TRW24(0x15);
    Data=1; DR1=1; CE=1;
}
CS = 0;
Wait(500),
}
//=====//
unsigned char Read_TRW24(void)
{
    unsigned char i,Temp;
    bit Out;

    Data = 1; //P1^4 input
    for (i=0,i<8;i++)
    {
        Temp = Temp << 1; //Temp = 0x00
        CLK1 = 1;
        Wait(50);
        Out = Data;

        if (Out)
        {
            Temp = Temp + 0x01,
        }

        CLK1 = 0,
        Wait(50),
    }

    rcturn(Temp);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//=====//
/*****/
/*      Mian Program      */
/*****/

void main (void)
{
    unsigned char DATA;
    //serial(),
    //printf("ite");
    delay_10us(1),
    DATA = 0xFF;

    DR1 = 0;
    CLK1 = 0;
    CE = 0;
    CS = 0;

    config_TRW24(0), // mode rx
    set_esc();
    while(1)
    {
        while (DRI == 0);
        DATA = Rcad_TRW24();
        //printf("%c",DATA),
        delay_10us(1),
        switch(DATA)
        {
            case 0x41 //go
                for(i=0;i<3;i++)
                    motor1=0x01; // m1
                    delay_10us(76),
                    motor1=0x00;

```

เอกสารนี้เป็นเอกสารที่สวจนไวสำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_10us(1155);
motor2=0x01; // m2
delay_10us(80);
motor2=0x00;
delay_10us(1155),
motor3=0x01, // m3
delay_10us(80);
motor3=0x00,
delay_10us(1155),
motor4=0x01, // m4
delay_10us(84),
motor4=0x00;
delay_10us(1145);
break;
//=====//
case 0x42: // back
for(i=0;i<3;i++)
    motor1=0x01, // m1
    delay_10us(82),
    motor1=0x00;
    delay_10us(1145);
    motor2=0x01; // m2
    delay_10us(76),
    motor2=0x00,
    delay_10us(1155);
    motor3=0x01, // m3
    delay_10us(80),
    motor3=0x00;
    delay_10us(1155),
    motor4=0x01; // m4
    delay_10us(84);
    motor4=0x00;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        delay_10us(1145);

break,
//=====//

case 0x43          // left
for(i=0;i<3;i++)
    motor1=0x01; // m1
    delay_10us(82);
    motor1=0x00,
    delay_10us(1145),
    motor2=0x01, // m2
    delay_10us(80);
    motor2=0x00;
    delay_10us(1155);
    motor3=0x01; // m3
    delay_10us(80);
    motor3=0x00;
    delay_10us(1155),
    motor4=0x01; // m4
    delay_10us(76),
    motor4=0x00;
    delay_10us(1155);

break;
//=====//

case 0x44.        // right
for(i=0;i<3;i++)
    motor1=0x01, // m1
    delay_10us(82);
    motor1=0x00;
    delay_10us(1145),
    motor2=0x01; // m2
    delay_10us(80),
    motor2=0x00;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_10us(1155),
motor3=0x01; // m3
delay_10us(76),
motor3=0x00,
delay_10us(1155);
motor4=0x01, // m4
delay_10us(83),
motor4=0x00;
delay_10us(1145),
break;
//=====//
/*case 0x45: // motor 1
for(i=0;i<3,i++)
motor1=0x01; // m1
delay_10us(67);
motor1=0x00,
delay_10us(1175);
motor2=0x01, // m2
delay_10us(67),
motor2=0x00;
delay_10us(1175);
motor3=0x01; // m3
delay_10us(67),
motor3=0x00,
delay_10us(1175),
motor4=0x01, // m4
delay_10us(67);
motor4=0x00,
delay_10us(1175);
break,
//=====//
case 0x46: // motor 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=0,i<3,i++)

    motor1=0x01; // m1
    delay_10us(78);
    motor1=0x00,
    delay_10us(1155);
    motor2=0x01, // m2
    delay_10us(76);
    motor2=0x00;
    delay_10us(1155),
    motor3=0x01; // m3
    delay_10us(76),
    motor3=0x00;
    delay_10us(1155),
    motor4=0x01; // m4
    delay_10us(79),
    motor4=0x00,
    delay_10us(1155);

break,
//-----//
case 0x47: // motor 3
for(i=0,i<3;i++)
    motor1=0x01; // m1
    delay_10us(82);
    motor1=0x00,
    delay_10us(1145),
    motor2=0x01, // m2
    delay_10us(80);
    motor2=0x00,
    delay_10us(1155);
    motor3=0x01; // m3
    delay_10us(80);
    motor3=0x00;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_10us(1155),
motor4=0x01; // m4
delay_10us(83);
motor4=0x00,
delay_10us(1145),
break;
//=====//

case 0x48 // motor 4
for(i=0,i<3;i++)
    motor1=0x01; // m1
    delay_10us(106),
    motor1=0x00,
    delay_10us(1125);
    motor2=0x01, // m2
    delay_10us(104),
    motor2=0x00;
    delay_10us(1125);
    motor3=0x01, // m3
    delay_10us(104);
    motor3=0x00,
    delay_10us(1125),
    motor4=0x01, // m4
    delay_10us(107);
    motor4=0x00;
    delay_10us(1125),

break;*/
//=====//

case 0x49 //set esc
    set_esc(),
break,
//=====//

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*case 0x50:    // up
for(i=0,i<3,i++)
    motor1=0x01; // m1
    delay_10us(72);
    motor1=0x00;
    delay_10us(1165),
    motor2=0x01; // m2
    delay_10us(70);
    motor2=0x00;
    delay_10us(1165),
    motor3=0x01; // m3
    delay_10us(70),
    motor3=0x00,
    delay_10us(1165);
    motor4=0x01; // m4
    delay_10us(73);
    motor4=0x00;
    delay_10us(1165);
break,
//-----//
case 0x51:    // down
for(i=0,i<3,i++)
    motor1=0x01; // m1
    delay_10us(80);
    motor1=0x00,
    delay_10us(1155),
    motor2=0x01; // m2
    delay_10us(78),
    motor2=0x00,
    delay_10us(1155),
    motor3=0x01, // m3
    delay_10us(78);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        motor3=0x00,
        delay_10us(1155);
        motor4=0x01. // m4
        delay_10us(82),
        motor4=0x00;
        delay_10us(1155);

break,
//=====//

//case 0x52. // test 4 rotor /**test**/
/*for(i=0;i<3,i++)
        motor1=0x01, // m1
        delay_10us(72),
        motor1=0x00,
        delay_10us(1165);
        motor2=0x01; // m2
        delay_10us(70);
        motor2=0x00;
        delay_10us(1165);
        motor3=0x01; // m3
        delay_10us(70);
        motor3=0x00;
        delay_10us(1165);
        motor4=0x01; // m4
        delay_10us(73);
        motor4=0x00;
        delay_10us(1165),

break,
//=====//

case 0x53: // 4 rotor stop
for(i=0;i<3,i++)
        motor1=0x01, // m1
        delay_10us(83);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

motor1=0x00;
dclay_10us(1145),
motor2=0x01; // m2
dclay_10us(80);
motor2=0x00;
delay_10us(1155),
motor3=0x01, // m3
dclay_10us(81),
motor3=0x00,
delay_10us(1155),
motor4=0x01; // m4
delay_10us(81);
motor4=0x00,
delay_10us(1155),
break,*/
//=====//
default.
set_esc(),
}
}
}

```

SET ESC

```

void set_esc();
void time_set(int count),

```

```

sbit m_1=P3^2,
sbit m_2=P3^3;
sbit m_3=P3^4,
sbit m_4=P3^5,

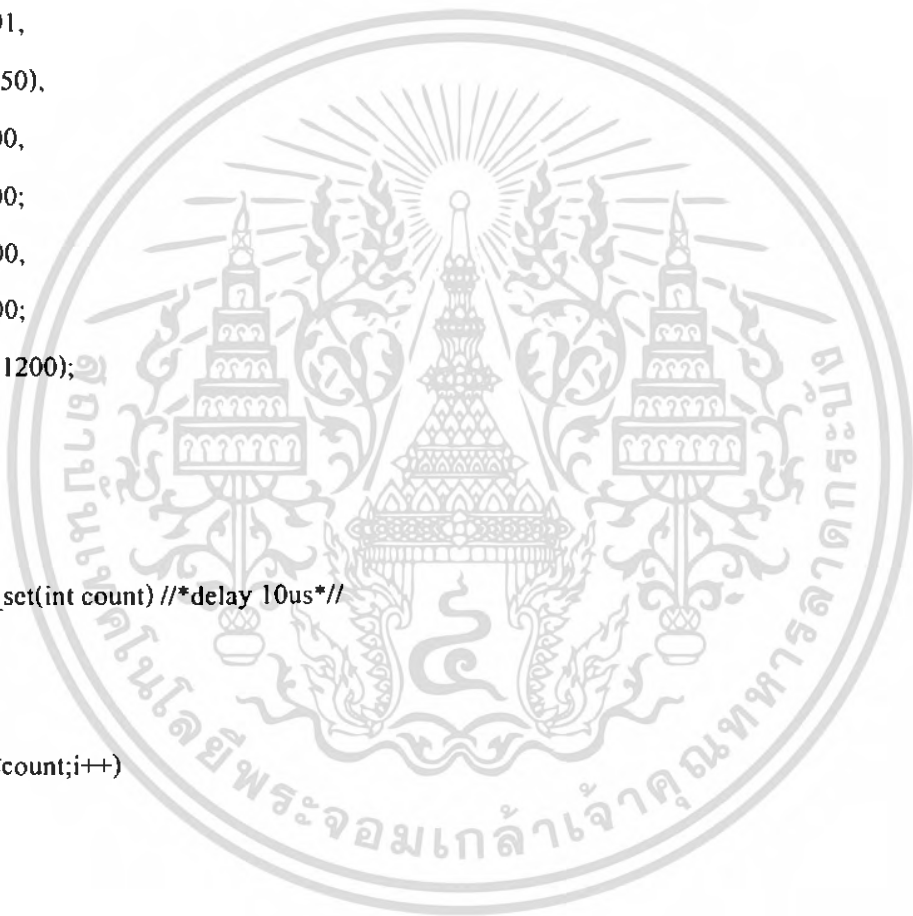
```

```

void sct_csc() /**set csc**/
{
    int i;
    for(i=0,i<10;i++)
    {
        m_1=0x01,
        m_2=0x01;
        m_3=0x01,
        m_4=0x01,
        time_set(50),
        m_1=0x00,
        m_2=0x00;
        m_3=0x00,
        m_4=0x00;
        time_set(1200);
    }
}

void time_set(int count) /**delay 10us**/
{
    int i;
    for(i=0,i<count;i++)
        _nop_(),
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

ข้อมูลรายละเอียดของไมโครคอนโทรลเลอร์เบอร์ AT89C4051

ข้อมูลรายละเอียดของไอซีเบอร์ TRW 2.4 G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

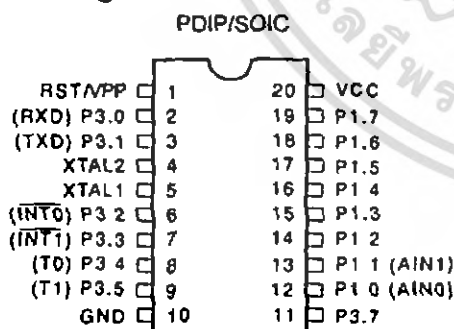
- Compatible with MCS-51™ Products
- 4K Bytes of Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- 3.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Brown-Out Detection

Description

The AT89C4051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash programmable and erasable read only memory (PEROM) The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C4051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C4051 provides the following standard features: 4K Bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C4051 is designed with static logic for operation down to zero frequency and supports two software-selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Configuration



ATMEL

8-Bit Microcontroller with 4K Bytes Flash

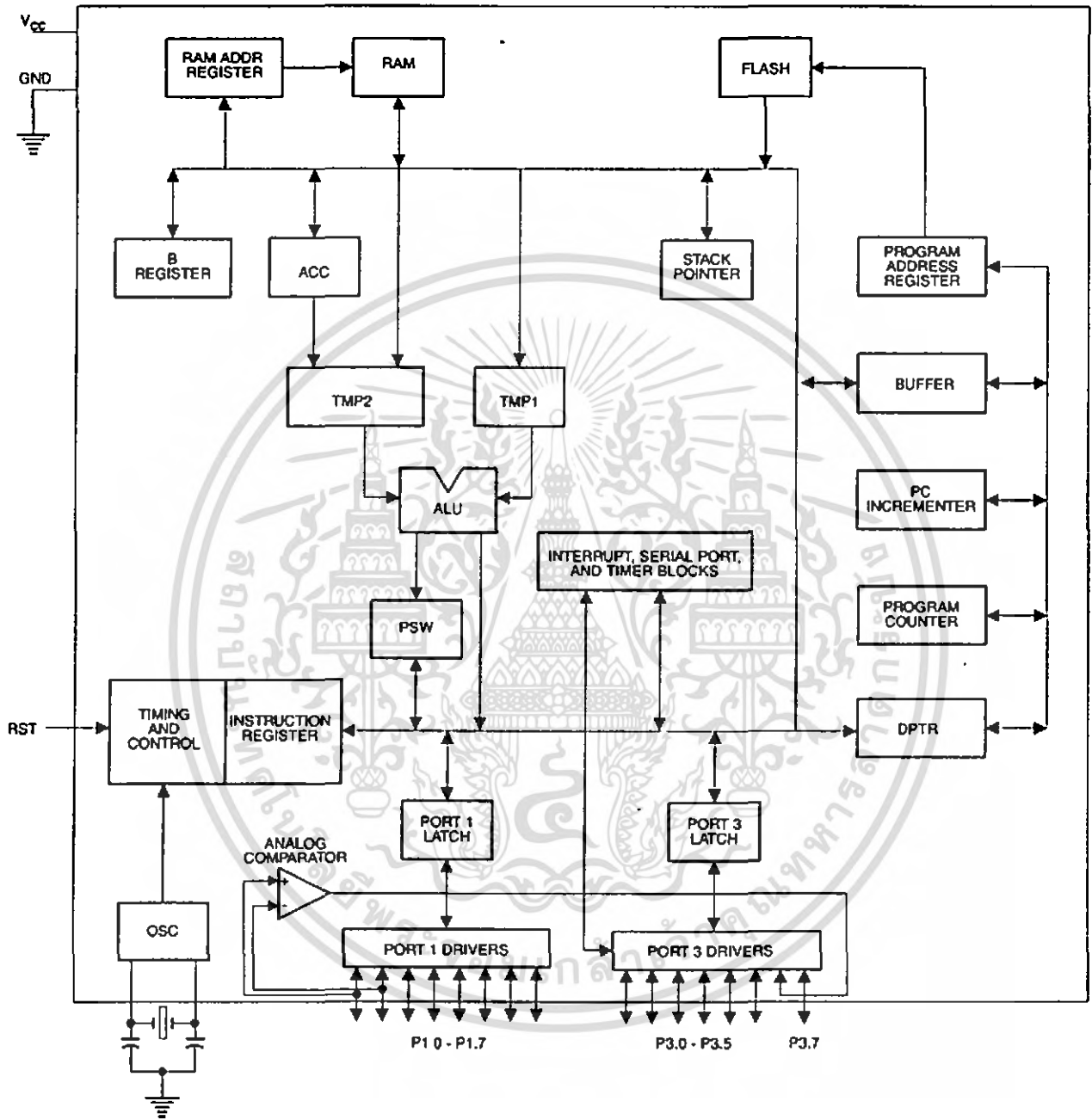
AT89C4051 Preliminary

Rev 1001A-02/98

ATMEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



Pin Description

V_{CC}
Supply voltage

GND
Ground.

Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C4051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and verification

RST

Reset input All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit

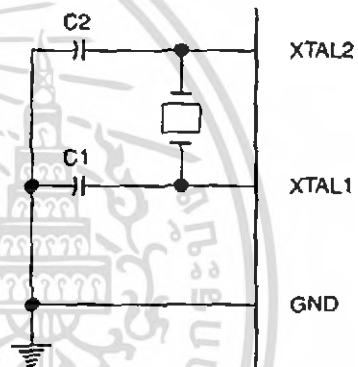
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

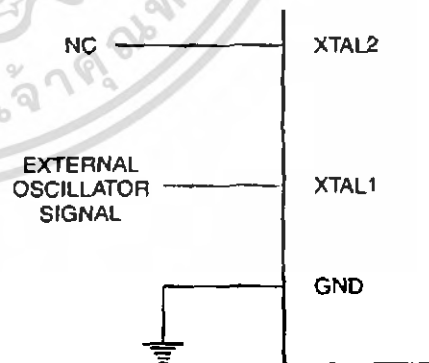
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration





Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C4051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H		SP 00000111	DPL 00000000	DPH 00000000			PCON 0XX00000	87H

Restrictions on Certain Instructions

The AT89C4051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 4K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 4K for the AT89C4051. This should be the responsibility of the software programmer. For example, LJMP 0FE0H would be a valid instruction for the AT89C4051 (with 4K of memory), whereas LJMP 1000H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to FFFH for the 89C4051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related Instructions, Data Memory:

The AT89C4051 contains 128 bytes of internal data memory. Thus, in the AT89C4051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes⁽¹⁾

Program Lock Bits			Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note. 1 The Lock Bits can only be erased with the Chip Erase operation.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

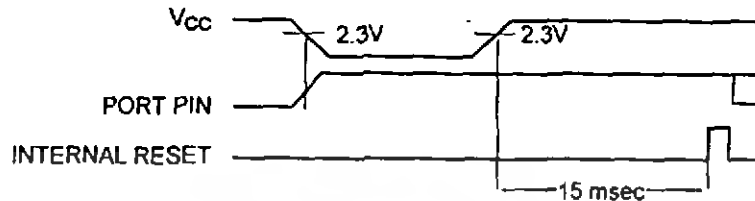




Brown-Out Detection

When V_{CC} drops below the detection threshold, all port pins (except P1.0 and P1.1) are weakly pulled high. When

V_{CC} goes back up again, an internal Reset is automatically generated after a delay of typically 15 msec. The nominal brown-out detection threshold is $2.3V \pm 10\%$.



Programming The Flash

The AT89C4051 is shipped with the 4K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. Once the array

10. Power-off sequence:
 - set XTAL1 to 'L'
 - set RST to 'L'
 - Turn V_{CC} power off

is programmed, to re-program the AT89C4051 contains an internal address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C4051, the following sequence is recommended.

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins
 - Set RST and XTAL1 to GND
2. Set pin RST to 'H'
3. Set pin P3.2 to 'H'
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 4K bytes array or until the end of the object file is reached.

Data Polling: The AT89C4051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification.

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (4K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all '1's in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.





- (000H) = 1EH indicates manufactured by Atmel
- (001H) = 41H indicates 89C4051

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data ⁽¹⁾⁽³⁾		12V		L	H	H	H
Read Code Data ⁽¹⁾		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V		H	L	L	L
Read Signature Byte		H	H	L	L	L	L

- Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin
 2. Chip Erase requires a 10-ms PROG pulse.
 3. P3.1 is pulled Low during programming to indicate RDY/BSY.



Figure 3. Programming the Flash Memory

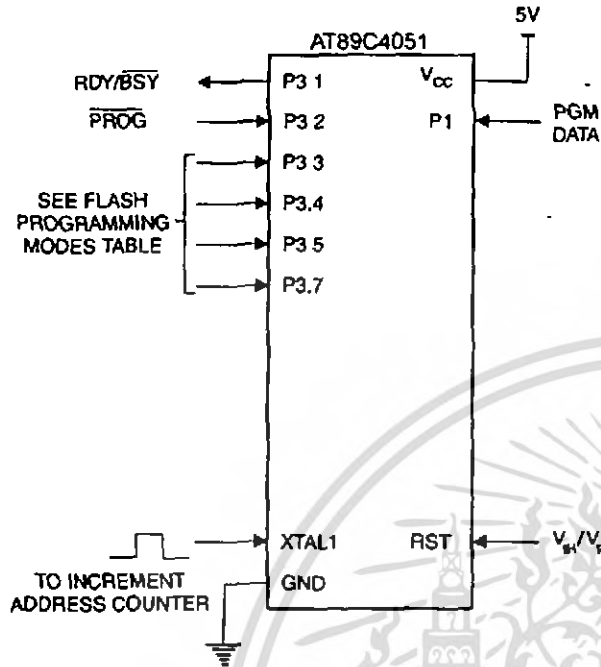
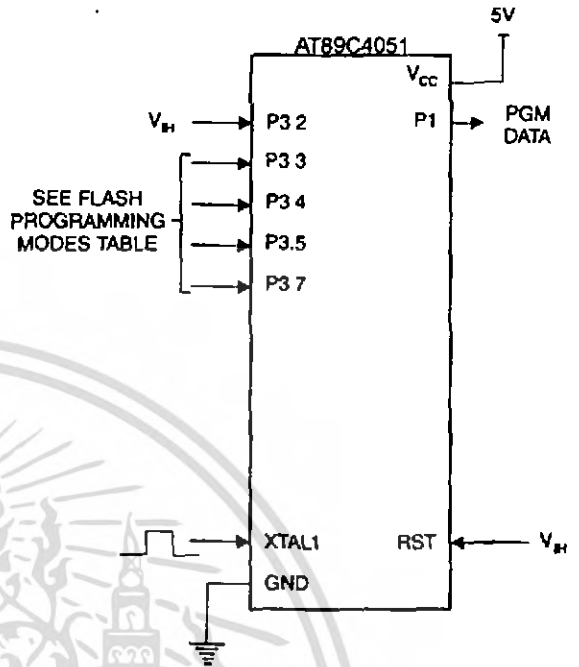


Figure 4. Verifying the Flash Memory



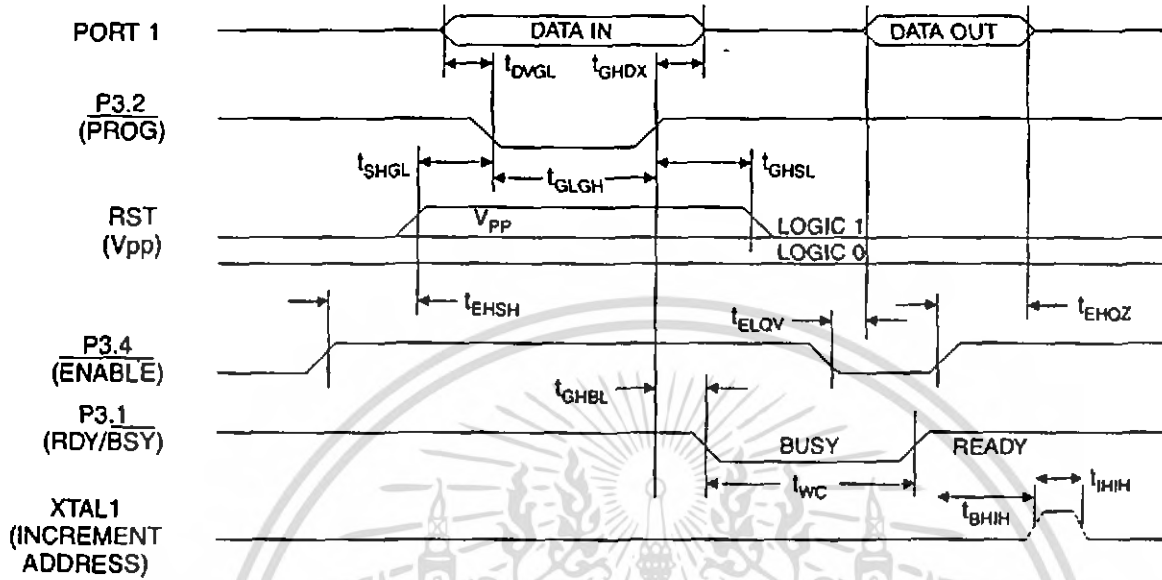
Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	μA
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	1.0		μs
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	1.0		μs
t_{EHS}	P3.4 (ENABLE) High to V_{PP}	1.0		μs
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		10	μs
t_{EHOZ}	Data Float After $\overline{\text{ENABLE}}$	0	10	μs
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		20	ms
t_{BHIH}	$\overline{\text{RDY/BSY}}$ to Increment Clock Delay	10		μs
t_{HIL}	Increment Clock High	200		ns

Note 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current.....	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.



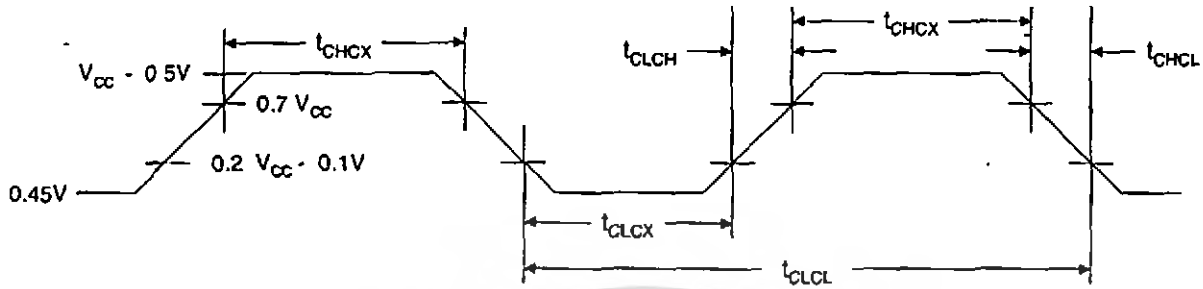
DC Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 3.0\text{V}$ to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	$0.2 V_{CC} - 0.1$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1, 3)	$I_{OL} = 20\text{ mA}$, $V_{CC} = 5\text{V}$ $I_{OL} = 10\text{ mA}$, $V_{CC} = 2.7\text{V}$		0.5	V
V_{OH}	Output High Voltage (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1, 3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-750	μA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		± 10	μA
V_{OS}	Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pulldown Resistor		50	300	$\text{k}\Omega$
C_{IO}	Pin Capacitance	Test Freq = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$ P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{V}$ P1.0 & P1.1 = 0V or V_{CC}		100	μA
		$V_{CC} = 3\text{V}$ P1.0 & P1.1 = 0V or V_{CC}		20	μA

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin 20 mA
 Maximum total I_{OL} for all output pins 80 mA
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2V

External Clock Drive Waveforms



External Clock Drive

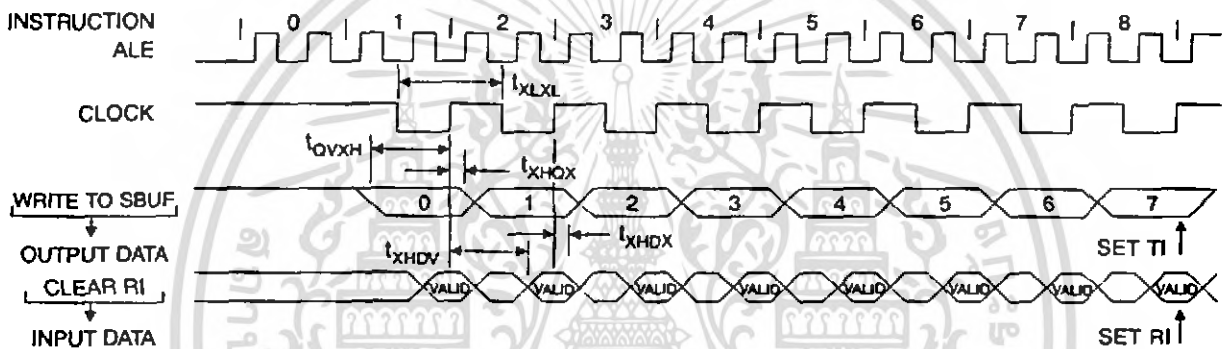
Symbol	Parameter	$V_{CC} = 3.0V \text{ to } 6.0V$		$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

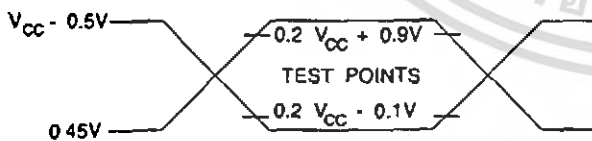
($V_{CC} = 5.0V \pm 20\%$, Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDX}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾ Float Waveforms⁽¹⁾

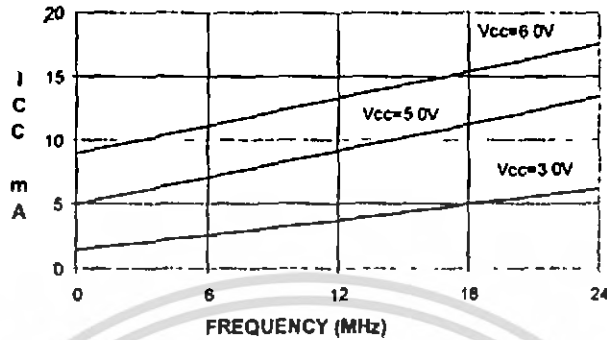


Note: 1 AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at $V_{IH \min}$ for a logic 1 and $V_{IL \max}$ for a logic 0.

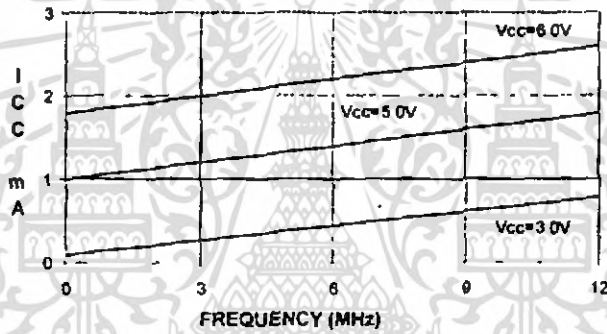
Note 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.



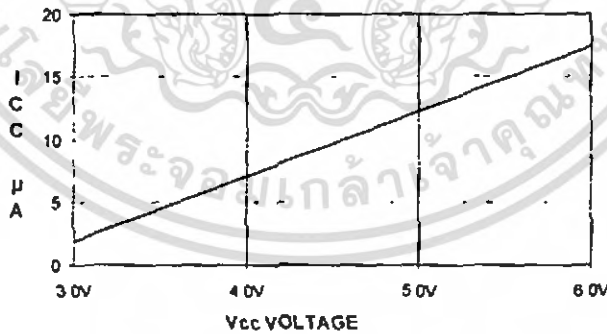
AT89C4051
TYPICAL ICC - ACTIVE (85°C)



AT89C4051
TYPICAL ICC - IDLE (85°C)



AT89C4051
TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



- Notes
- 1 XTAL1 tied to GND for I_{CC} (power down)
 - 2 P1.0 and P1.1 = V_{CC} or GND
 - 3 Lock bits programmed





Ordering Information

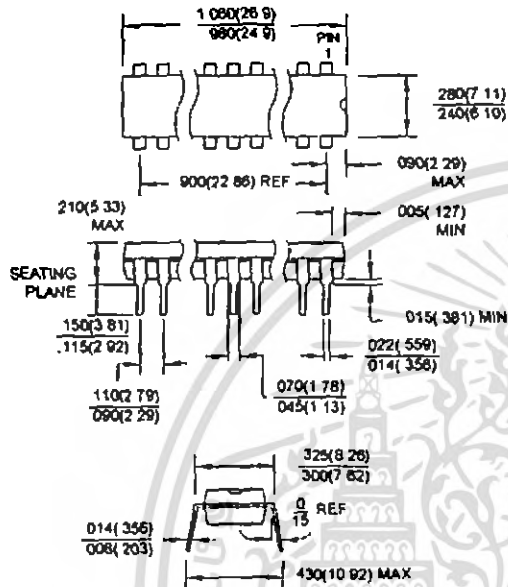
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	3.0V to 6.0V	AT89C4051-12PC AT89C4051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C4051-12PI AT89C4051-12SI	20P3 20S	Industrial (-40°C to 85°C)
		AT89C4051-12PA AT89C4051-12SA	20P3 20S	Automotive (-40°C to 105°C)
24	4.0V to 6.0V	AT89C4051-24PC AT89C4051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C4051-24PI AT89C4051-24SI	20P3 20S	Industrial (-40°C to 85°C)



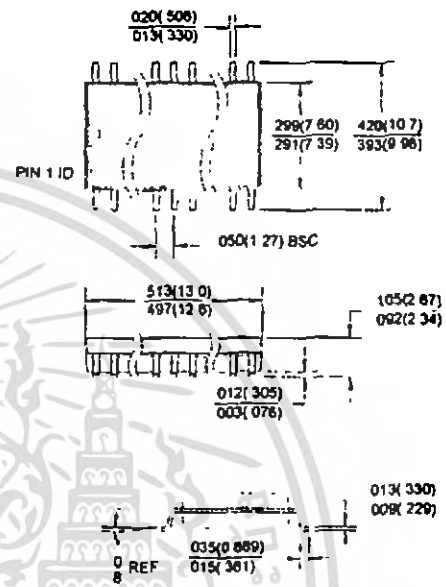
Package Type	
20P3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

Packaging Information

20P3, 20-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
 Dimensions in Inches and (Millimeters)
 JEDEC STANDARD MS-001 AD



20S, 20-Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
 Dimensions in Inches and (Millimeters)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TRF-2.4G Transceiver Data Sheet

**High frequency 2.4G Wireless Transceiver
Antenna, Codec and CRC built in
Data Rate up to 1Mbps**

High frequency TRF-2.4G Transceiver module

Specification

- Frequency Range: 2.4~2.524 GHz ISM band
- Modulate Mode: GFSK
- Data Rate: 1Mbps; 250Kbps
- Multi channel operation: 125 channels, Channel switching time<200uS, Support frequency hopping
- Emulated full duplex RF link due to the 1Mbits/s on the air data rate
- Simultaneous dual receiver
- Data slicer / clock recovery of data
- Including decoder, encoder and data buffer and CRC computation
- ShockBurst mode for ultra-low power operation and relaxed MCU performance
- Sensitivity: -90dBm
- Built in antenna
- Power supply range: 1.9 to 3.6 V
- Low supply current (TX), typical 10.5mA peak@ -5dBm output power
- Low supply current (RX), typical 18mA peak in receive mode
- Supply current in Power Down Mode: 1 uA
- Operating Temperature: -40~+85 Centigrade
- Size: 20.5*36.5*2.4mm
- 100% RF tested
- Competitive price

Applications

- Wireless mouse, keyboard, joystick
- Wireless data communication
- Alarm and security systems
- Home automation
- Wireless Earphone
- Telemetry
- Surveillance
- Automotive

GENERAL DESCRIPTION

Laipac TRF-2.4G Module is an easy to use radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of an antenna, a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator and a modulator. Output power and frequency channels are easily programmable by use of the 3-wire serial interface. Current consumption is very low, only 10.5mA at an output power of -5dBm and 18mA in receive mode. Built-in Power Down modes makes power saving easily realizable.

ELECTRICAL SPECIFICATIONS

Conditions: VCC = +3V, VSS = 0V, TA = - 40°C to + 85°C

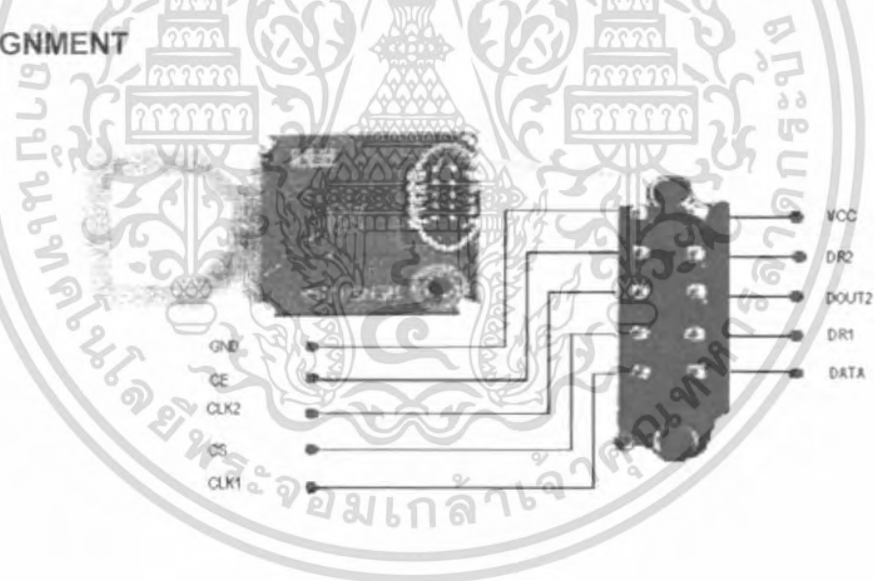
Symbol	Parameter (condition)	Notes	Min.	Ttp.	Max.	Units
Operating conditions						
VCC	Supply voltage		1.9	3.0	3.6	V
TEMP	Operating Temperature		-40	+27	+85	°C
Digital input pin						
VIH	HIGH level input voltage		VCC-0.3		VCC	V
VIL	LOW level input voltage		Vss		0.3	V
Digital output pin						
VOH	HIGH level output voltage (IOH=-0.5mA)		VCC-0.3		VCC	V
VOL	LOW level output voltage (IOL=0.5mA)		Vss		0.3	V
General RF conditions						
fOP	Operating frequency	1)	2400		2524	MHz
Δf	Frequency deviation			±156		kHz
RGFSK	Data rate ShockBurst		>0		1000	kbps
RGFSK	Data rate Direct Mode	2)	250		1000	kbps
FCHANNEL	Channel spacing			1		MHz
Transmitter operation						
PRF	Maximum Output Power	3)		0	+4	dBm
PRFC	RF Power Control Range		16	20		dB
PRFCR	RF Power Control Range Resolution				±3	dB
PBW	20dB Bandwidth for Modulated Carrier				1000	kHz
PRF2	2nd Adjacent Channel Transmit Power 2MHz				-20	dBm
PRF3	3rd Adjacent Channel Transmit Power 3MHz				-40	dBm
IVCC	Supply current @ 0dBm output power	4)		13		mA
IVCC	Supply current @ -20dBm output power	4)		8.8		mA
IVCC	Average Supply current @ -5dBm output power, ShockBurst	5)		0.8		mA
IVCC	Average Supply current in stand-by mode	6)		12		μA
IVCC	Average Supply current in power down			1		μA
Receiver operation						
IVCC	Supply current one channel 250kbps			18		mA
IVCC	Supply current one channel 1000kbps			19		mA
IVCC	Supply current two channels 250kbps			23		mA
IVCC	Supply current two channels 1000kbps			25		mA
RXSSENS	Sensitivity at 0.1%BER (@250kbps)			-90		dBm
RXSSENS	Sensitivity at 0.1%BER (@1000kbps)			-80		dBm

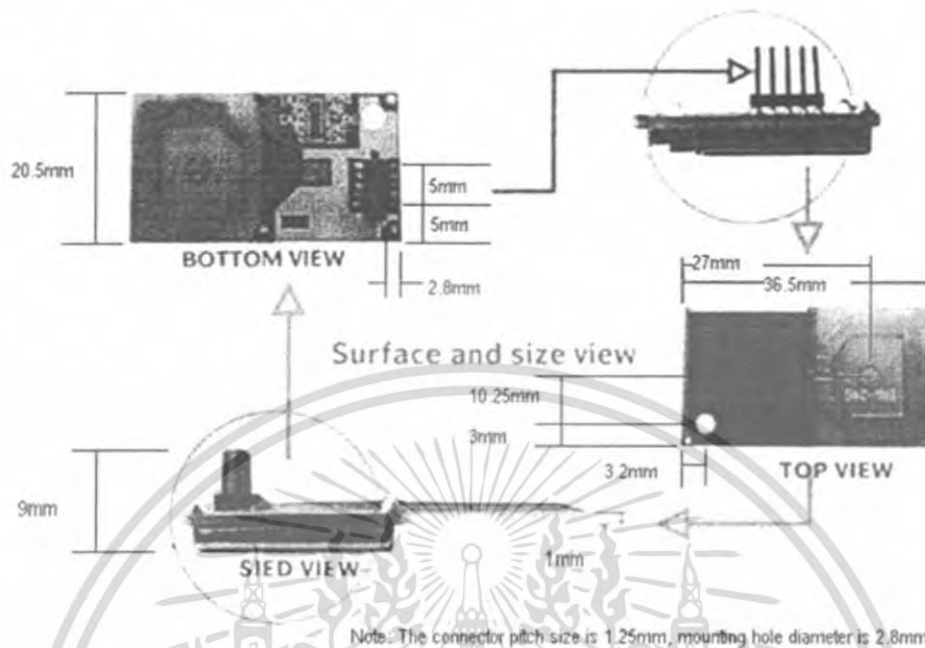
C/I _{CO}	C/I Co-channel		6		dB
C/I _{1ST}	1 st Adjacent Channel Selectivity C/I 1MHz		-1		dB
C/I _{2ND}	2 nd Adjacent Channel Selectivity C/I 2MHz		-16		dB
C/I _{3RD}	3 rd Adjacent Channel Selectivity C/I 3MHz		-26		dB
RXB	Blocking Data Channel 2		-41		dB

- 1) Usable band is determined by local regulations
- 2) Data rate must be either 250kbps or 1000kbps.
- 3) De-embedded Antenna load impedance = 400
- 4) De-embedded Antenna load impedance = 400 . Effective data rate 250kbps or 1Mbps.
- 5) De-embedded Antenna load impedance = 400 . Effective data rate 10kbps.
- 6) Current if 4 MHz crystal is used.

Table 1 TRF-2.4G RF specifications

PIN ASSIGNMENT





PIN FUNCTIONS

Pin	Name	Pin function	Description
1	GND	Power	Ground (0V)
2	CE	Input	Chip Enable activates RX or TX mode
3	CLK2	I/O	Clock output/input for RX data channel 2
4	CS	Input	Chip Select activates Configuration mode
5	CLK1	I/O	Clock Input(TX)&I/O(RX) for data channel 1 3-wire interface
6	DATA	I/O	RX data channel 1/TX data input /3-wire interface
7	DR1	Output	RX data ready at data channel 1 (ShockBurst only)
8	DOU2	Output	RX data channel 2
9	DR2	Output	RX data ready at data channel 2 (ShockBurst only)
10	VCC	Power	Power Supply (+3V DC)

Table 2 TRF-2.4G pin function

MODE OF OPERATION

TRF-2.4G can be set in the following main mode:

Mode	CE	CS
Active (RX /TX)	1	0
Configuration	0	1
Stand by	0	0

Table 3 TRF-2.4G main modes

TRF-2.4G has two active (RX /TX) modes:

- ShockBurst
- Direct Mode

The device functionality in these modes is decided by the content of a configuration word. This configuration word is presented in configuration section.

Absolute Maximum Ratings

Supply voltages

VCC.....- 0.3V to + 3.6V

VSS0V

Input/Output voltages

V_I.....- 0.3V to VCC + 0.3V

V_O.....- 0.3V to VCC + 0.3V

Total Power Dissipation

P_D (T_A=85°C).....90mW

Temperatures

Operating Temperature.....- 40°C to + 85°C

Storage Temperature.....- 40°C to + 125°C

ShockBurst Mode

The ShockBurst technology uses on-chip FIFO to clock in data at a low data rate and transmit at a very high rate thus enabling extremely power reduction.

When operating the TRF-2.4G in ShockBurst, you gain access to the high data rates (1 Mbps) offered by the 2.4 GHz band without the need of a costly, high-speed micro controller (MCU) for data processing.

By putting all high speed signal processing related to RF protocol on-chip, the TRF-2.4G offers the following benefits:

- Highly reduced current consumption
- Lower system cost (facilitates use of less expensive micro controller)
- Greatly reduced risk of 'on-air' collisions due to short transmission time

The TRF-2.4G can be programmed using a simple 3-wire interface where the data rate is decided by the speed of the micro controller.

By allowing the digital part of the application to run at low speed while maximizing the data rate on the RF link, the nRF ShockBurst mode reduces the average current consumption in applications considerably.

ShockBurst principle

When the TRF-2.4G is configured in ShockBurst, TX or RX operation is conducted in the following way (10 kbps for the example only).

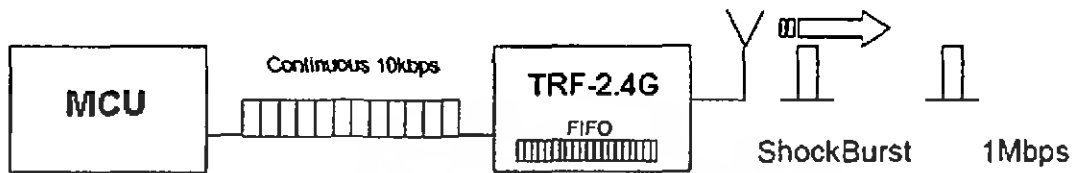


Figure 0 Clocking in data with MCU and sending with ShockBurst technology

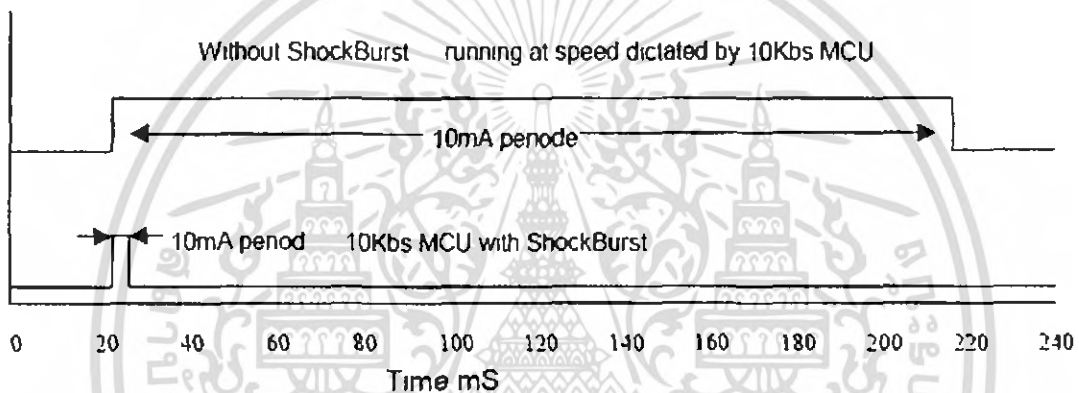


Figure 1 Current consumption with & without ShockBurst technology

TRF-2.4G ShockBurst Transmit:

MCU interface pins: CE, CLK1, DATA

1. When the application MCU has data to send, set CE high. This activates TRF-2.4G on-board data processing.
2. The address of the receiving node (RX address) and payload data is clocked into the TRF-2.4G. The application protocol or MCU sets the speed <1Mbps (ex: 10kbps).
3. MCU sets CE low, this activates a TRF-2.4G ShockBurst transmission.
4. TRF-2.4G ShockBurst:
 - RF front end is powered up
 - RF package is completed (preamble added, CRC calculated)
 - Data is transmitted at high speed (250 kbps or 1 Mbps configured by user).
 - TRF-2.4G return to stand-by when finished

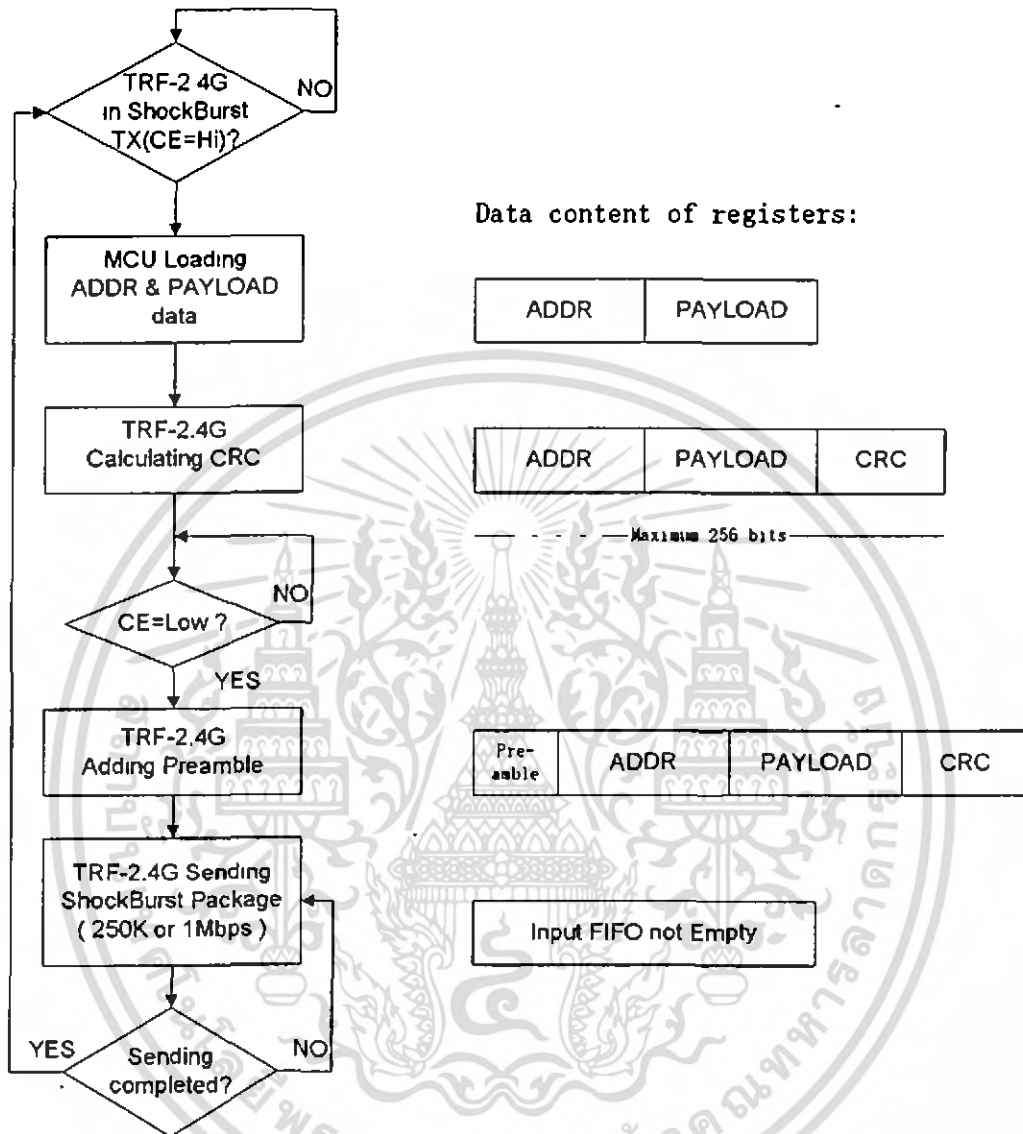
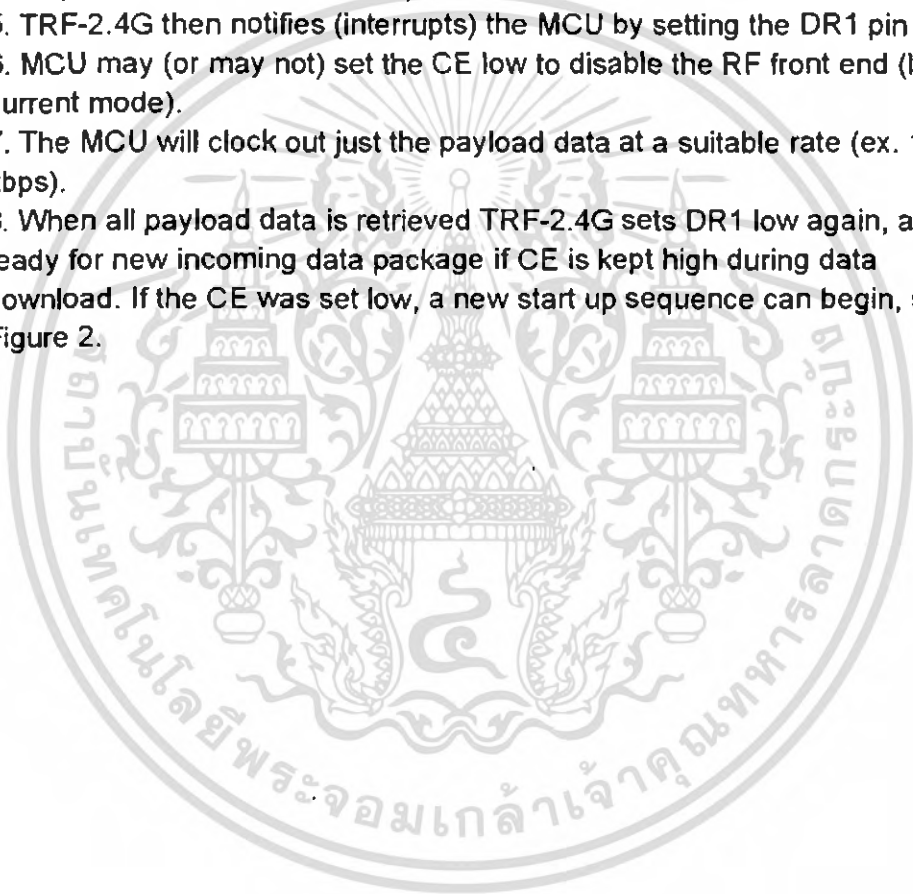


Figure 2 Flow Chart ShockBurst Transmit of TRF-2.4G

TRF-2.4G ShockBurst Receive:

MCU interface pins: CE, DR1, CLK1 and DATA (one RX channel receive)

1. Correct address and size of payload of incoming RF packages are set when TRF-2.4G is configured to ShockBurst RX
2. To activate RX, set CE high.
3. After 200 μ s settling, TRF-2.4G is monitoring the air for incoming communication.
4. When a valid package has been received (correct address and CRC found), TRF-2.4G removes the preamble, address and CRC bits.
5. TRF-2.4G then notifies (interrupts) the MCU by setting the DR1 pin high.
6. MCU may (or may not) set the CE low to disable the RF front end (low current mode).
7. The MCU will clock out just the payload data at a suitable rate (ex. 10 kbps).
8. When all payload data is retrieved TRF-2.4G sets DR1 low again, and is ready for new incoming data package if CE is kept high during data download. If the CE was set low, a new start up sequence can begin, see Figure 2.



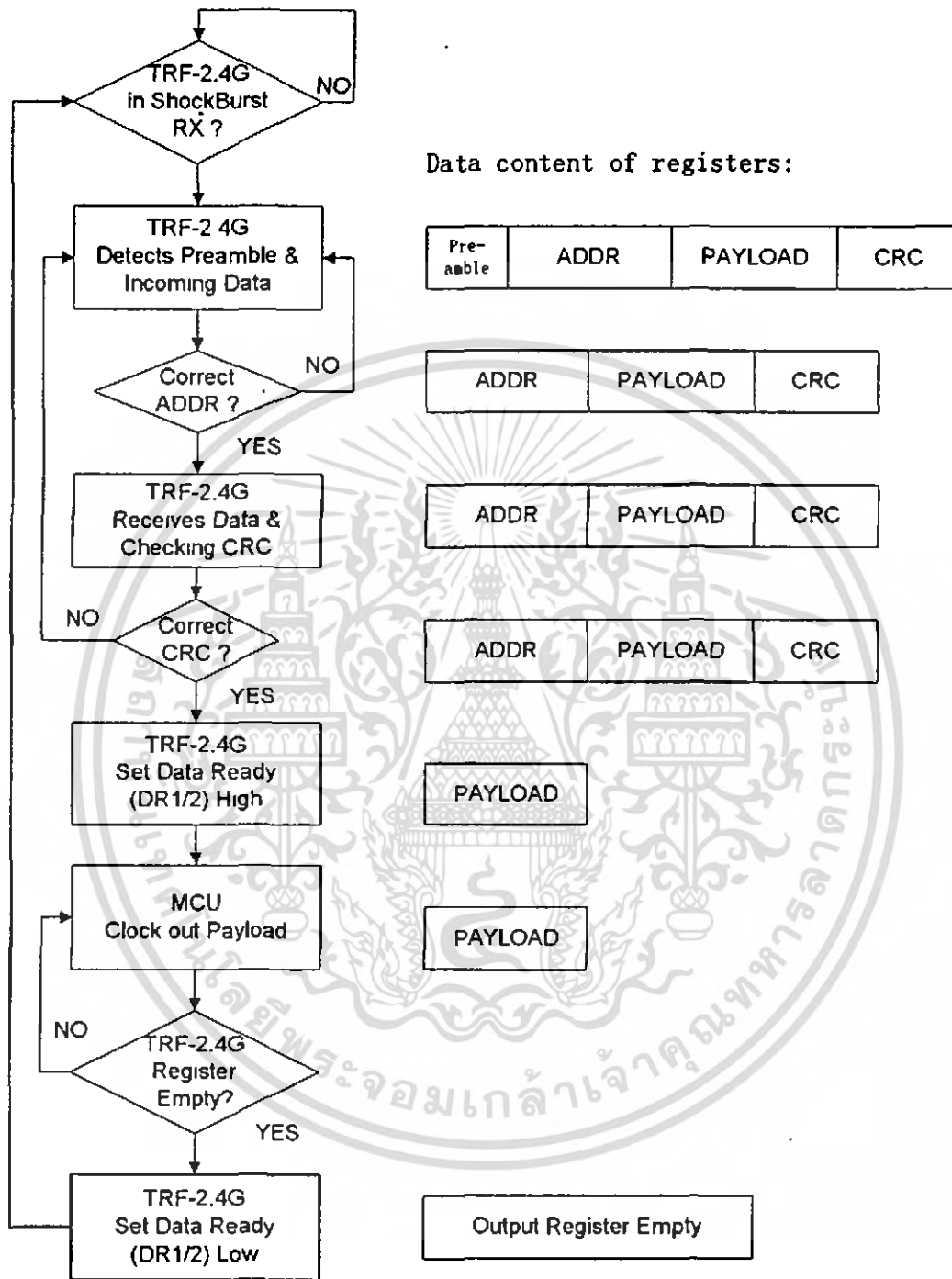


Figure 3 Flow Chart ShockBurst Receive of TRF-2.4G

TRF-2.4G Direct Mode:

In direct mode the TRF-2.4G works like a traditional RF device. Data must be at 1Mbps, or 250kbps at low data rate setting, for the receiver to detect the signals.

Direct Mode Transmit:

MCU interface pins: CE, DATA

1. When application MCU has data to send, set CE high
2. The TRF-2.4G RF front end is now immediately activated, and after 200 seconds settling time, data will modulate the carrier directly.
3. All RF protocol parts must hence be implemented in MCU firmware (preamble, address and CRC).

Direct Mode Receive:

MCU interface pins: CE, CLK1, and DATA

1. Once the TRF-2.4G is configured and powered up (CE high) in direct RX mode, DATA will start to toggle due to noise present on the air.
2. CLK1 will also start to toggle as TRF-2.4G is trying to lock on to the incoming data stream.
3. Once a valid preamble arrives, CLK1 and DATA will lock on to the incoming signal and the RF package will appear at the DATA pin with the same speed as it is transmitted.
4. To enable the demodulator to re-generate the clock, the preamble must be 8 bits toggling hi-low, starting with low if the first data bit low.
5. In this mode no data ready (DR) signals is available. Address and checksum verification must also be done in the receiving MC.

DuoCeiver Simultaneous Two Channel Receive Mode

In both ShockBurst & Direct modes the TRF-2.4G can facilitate simultaneous reception of two parallel independent frequency channels at the maximum data rate.

This means:

- TRF-2.4G can receive data from two 1 Mbps transmitters, 8 MHz (8 frequency channels) apart through one antenna interface.
- The output from the two data channels is fed to two separate MCU interfaces.
 - Data channel 1: CLK1, DATA, and DR1
 - Data channel 2: CLK2, DOUT2, and DR2
 - DR1 and DR2 are available only in ShockBurst.

The DuoCeiver technology provides 2 separate dedicated data channels for RX and replaces the need for two, stand alone receiver systems.

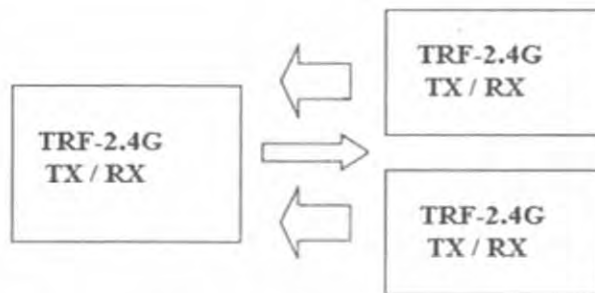
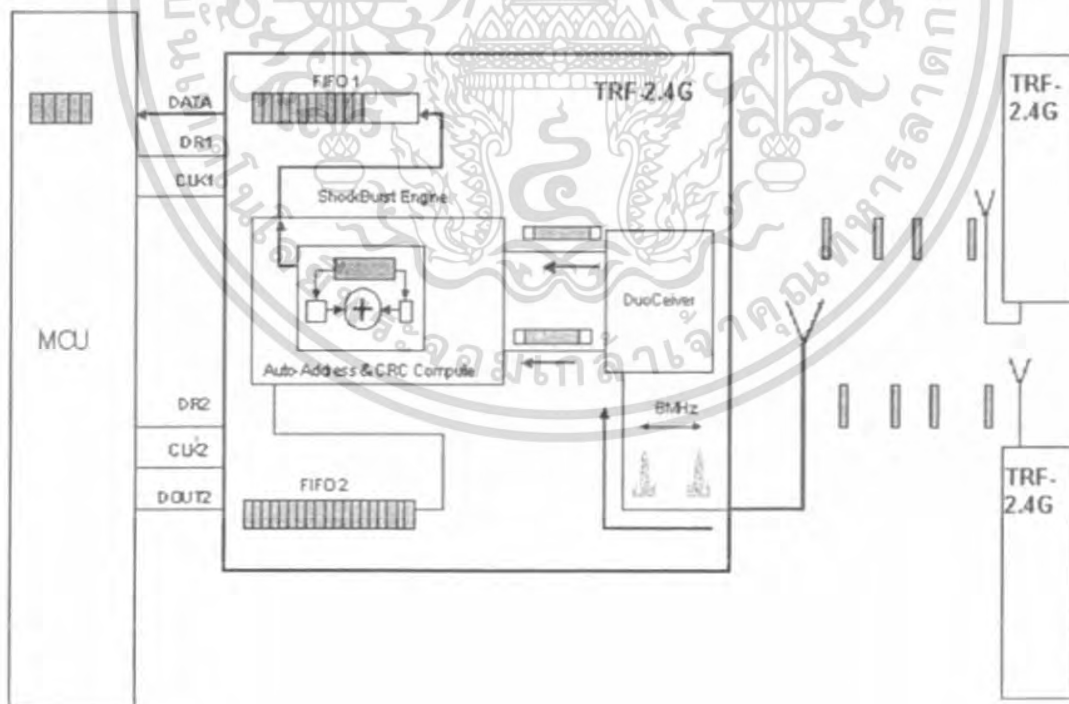


Figure 4 Simultaneous 2 channel receive on TRF-2.4G

There is one absolute requirement for using the second data channel. For the TRF-2.4G to be able to receive at the second data channel the frequency channel must be 8MHz higher than the frequency of data channel 1. The TRF-2.4G must be programmed to receive at the frequency of data channel 1. No time multiplexing is used in TRF-2.4G to fulfil this function. In direct mode the MCU must be able to handle two simultaneously incoming data packets if it is not multiplexing between the two data channels. In ShockBurst it is possible for the MCU to clock out one data channel at a time while data on the other data channel waits for MCU availability, without any lost data packets, and by doing so reduce the needed performance of the MCU.



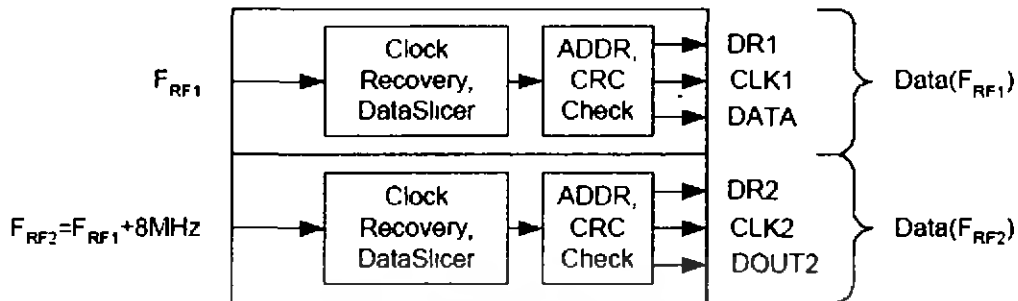


Figure 5 DuoCeiver with two simultaneously independent receive channels.

Configuration Mode

In configuration mode a configuration word of up to 15 bytes is downloaded to TRF-2.4G. This is done through a simple 3-wire interface (CS, CLK1 and DATA). For more information on configuration please refer to the TRF-2.4G Device configuration chapter on next 2nd page.

Stand-By Mode

Stand by mode is used to minimize average current consumption while maintaining short start up times. In this mode, part of the crystal oscillator is active. Current consumption is dependent on crystal frequency (Ex: 12uA @ 4 MHz, 32uA @ 16MHz). The configuration word content is maintained during stand by.

Power Down Mode

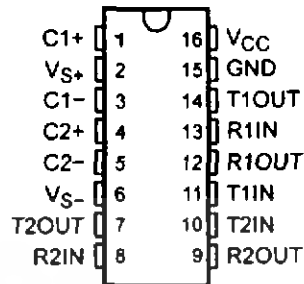
In power down the TRF-2.4G is disabled with minimal current consumption, typically less than 1 A. Entering this mode when the device is not active minimizes average current consumption, maximizing battery lifetime. The configuration word content is maintained during power down.

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- \pm 30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept \pm 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING	
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N	
		Tube of 40	MAX232D	MAX232	
	SOIC (D)	Reel of 2500	MAX232DR		
		SOIC (DW)	Tube of 40	MAX232DW	MAX232
			Reel of 2000	MAX232DWR	
SOP (NS)	Reel of 2000	MAX232NSR	MAX232		
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN	
		Tube of 40	MAX232ID	MAX232I	
	SOIC (D)	Reel of 2500	MAX232IDR		
		SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
			Reel of 2000	MAX232IDWR	

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



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

LinASIC is a trademark of Texas Instruments.

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

Copyright © 2004 Texas Instruments Incorporated



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

Function Tables

EACH DRIVER

INPUT T1N	OUTPUT TOUT
L	H
H	L

H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)

