สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมยานลาดตระเวนระยะไกล

PATROL ROBOT CONTROL SYSTEM

โดย

นายภูมเดชา สุขยืน

นายยุทธพงษ์ คงสว่าง

ส.ม.
ส1661 ร
2550

เลขหมู่.....
เลขทะเบียน 82997
วัน,เดือน,ปี 30 ก.ค. 2551

b 11957803
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมยานลาดตระเวนระยะไกล

PATROL ROBOT CONTROL SYSTEM

โดย

นายภูมเดชา สุขยืน รหัส 47010574

นายยุทธพงษ์ คงสว่าง รหัส 47010605

อาจารย์ที่ปรึกษา

อาจารย์พลศาสตร์ เลิศประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานเรื่อง ระบบควบคุมยานลาดตระเวนระยะไกล

PATROL ROBOT CONTROL SYSTEM

จัดทำโดย นายภูมเดชา สุขยืน รหัส 47010574

นายยุทธพงษ์ คงสว่าง รหัส 47010605

อาจารย์ที่ปรึกษา อาจารย์พลศาสตร์ เลิศประเสริฐ

รายงานฉบับนี้ได้ผ่านการตรวจสอบ

ลงชื่อ.....อาจารย์ที่ปรึกษา

(อาจารย์พลศาสตร์ เลิศประเสริฐ)

วันที่ 17 / 3 / 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมยานลาดตระเวนระยะไกล

นายภูมเดชา สุขยืน 47010574
นายยุทธพงษ์ คงสว่าง 47010605
อาจารย์ที่ปรึกษา อ.พลศาสตร์ เลิศประเสริฐ
ปีการศึกษา 2550

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอการออกแบบและสร้างระบบควบคุมยานลาดตระเวนระยะไกลด้วยคลื่นวิทยุ (Radio Frequency) ซึ่งควบคุมทิศทางการเคลื่อนที่ด้วยจอยสติก โดยใช้ไมโครคอนโทรลเลอร์ตระกูล ในการประมวลผลเพื่อนำไปขับมอเตอร์ทั้ง 2 ตัวในการขับเคลื่อนรถ โดยการขับมอเตอร์อาศัยหลักการพัลส์วิดมอดดูเลชั่น (PWM) เพื่อควบคุมความเร็วมอเตอร์ ซึ่งทำให้มอเตอร์หมุนพร้อมกัน ซึ่งมีแหล่งจ่ายไฟให้แก่มอเตอร์เป็นแบตเตอรี่ 12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PATROL ROBOT CONTROL SYSTEM

Mr. Phumdecha Sukyuen 47010574
Mr. Yuttapong Kongsawang 47010605
Mr. Polsart Lertprasert Advisor
Education Year 2007

Abstract

This project is model of the Patrol Robot control system. It is controlled with RF (Radio Frequency). By using Microcontroller for car driving motors process. The direction of this Patrol Robot Control System is controlled by joystick. Motor is controlled a velocity by pulse width modulation (PWM) and the supplied by a 12 Volts battery.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สามารถลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือจากบุคคลหลายท่าน โดยเฉพาะอย่างยิ่ง อาจารย์พลศาสตร์ เลิศประเสริฐ(อาจารย์ที่ปรึกษา) ที่ให้คำปรึกษาและแนะนำเกี่ยวกับโครงงานระบบลาดตระเวนควบคุมระยะไกล อีกทั้งรุ่นพี่และเพื่อนๆทุกคนที่ช่วยเหลือในการปฏิบัติงานเป็นอย่างดีมาโดยตลอด จนทำให้โครงงานนี้สำเร็จโดยสมบูรณ์ได้

นายภูมเดชา สุขยืน รหัส 47010574

นายยุทธพงษ์ คงสว่าง รหัส 47010605

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ I

Abstract II

กิตติกรรมประกาศ III

สารบัญ

สารบัญรูป

สารบัญตาราง

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีและหลักการ 3

2.1จอยสติก (Joystick) 3

2.1.1 โครงสร้างจอยสติก

2.2 มอเตอร์กระแสตรง (DC MOTOR) 4

2.3 ไมโครคอนโทรลเลอร์ ชนิด PIC 6

2.3.1 คุณสมบัติของ dsPIC 30F4012

2.3.1.1 โมดูล Analog to Digital Converter 11

2.3.1.2 การสื่อสารอนุกรม UART 14

2.3.2 คุณสมบัติของ PIC16F876 17

2.3.2.1 โมดูล Analog to Digital Converter 21

2.3.2.2 การสื่อสารอนุกรม UART 26

2.4 Pulse Width Modulation (PWM) 31

2.5 ไทเมอร์(TIMER) 33

2.6 แบตเตอรี่ 38

2.7 GPS 39

บทที่ 3 การออกแบบ 45

3.1 ภาคส่งสัญญาณ 45

3.1.1 จอยสติก 45

3.1.2 วงจรไมโครคอนโทรลเลอร์ 46

3.1.3 วงจรเครื่องส่งคลื่นวิทยุ 47

3.2 ภาครับสัญญาณ 47

3.2.1 วงจรเครื่องรับวิทยุ 48

3.2.2 วงจรไมโครคอนโทรลเลอร์ 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 วงจรขับมอเตอร์ 49
บทที่ 4 ผลการทดลอง 50
4.1 การทดลองที่ 1 การแปลงค่าแรงดันของจอยสติกจากอนาลอกเป็นดิจิตอล 50
4.2 การทดลองที่ 2 ทดสอบสมรรถนะของตัวรถ 50
4.3 การทดลองที่ 3วัดสัญญาณ PWM ที่วงจร Drive Motor 51
บทที่ 5 วิเคราะห์และสรุปผลการทดลอง 63
บรรณานุกรม 64
ภาคผนวก 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

บทที่ 2	รูปที่ 2.1 จอยสติก	3
	รูปที่ 2.2 วงจรภายในจอยสติก	3
	รูปที่ 2.3 แสดงสถาปัตยกรรมของไมโครคอนโทรลเลอร์ dsPIC30F4012	7
	รูปที่ 2.4 แสดงตัวถังของ CPU dsPIC30F4012	8
	รูปที่ 2.5 บล็อกไดอะแกรมภายในของโมดูล A/D	11
	รูปที่ 2.6 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART	15
	รูปที่ 2.7 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART	15
	รูปที่ 2.8 แสดงสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F876	18
	รูปที่ 2.9 แสดงตัวถังของ CPU PIC16F876	19
	รูปที่ 2.10 บล็อกไดอะแกรมภายในของโมดูล A/D	23
	รูปที่ 2.11 แสดงการเก็บผลลัพธ์จากการแปลงข้อมูล A/D	24
	รูปที่ 2.12 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART	29
	รูปที่ 2.13 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART	29
	รูปที่ 2.14 ความกว้างของพัลส์ขนาดต่างๆและค่าดิวตี้ไซเคิล	31
	รูปที่ 2.15 สัญญาณเอาต์พุตของ PWM	31
	รูปที่ 2.16 การทำงานของไทเมอร์ในฐานเวลาแบบ A	34
	รูปที่ 2.17 การทำงานของไทเมอร์ในฐานเวลาแบบ B	35
	รูปที่ 2.18 การทำงานของไทเมอร์ในฐานเวลาแบบ C	35
	รูปที่ 2.19 ไดอะแกรมของไทเมอร์ 32 บิต	37
	รูปที่ 2.20 ตัวแปรของเอลลิพซอยด์ที่กำหนดมาตรฐาน	41
	รูปที่ 2.21 แสดงรูปแบบจำลองของพื้นผิวโลก	41
	รูปที่ 2.22 แสดงระดับของจีออยด์	42
	รูปที่ 2.23 การกำหนดอีเควเตอร์และไพรม์เมอริเดียน	43
	รูปที่ 2.24 แสดงการกำหนดพิกัด ละติจูด ลองติจูดและความสูง	43
	รูปที่ 2.25 แสดงระบบ ECEF X, Y, Z	44
บทที่ 3	รูปที่ 3.1 วงจรรวมภาคส่งสัญญาณ	45
	รูปที่ 3.2 วงจรภายในจอยสติก	46
	รูปที่ 3.3 วงจรไมโครคอนโทรลเลอร์ PIC16F876 ของภาคส่ง	46
	รูปที่ 3.4 RF Module Radiometrix Tx2-433	47
	รูปที่ 3.5 วงจรรวมของภาครับสัญญาณ	47

เอกสารนี้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 RF Module Radiometrix Rx2-433 48
รูปที่ 3.7 วงจรไมโครคอนโทรลเลอร์ dsPIC30F4012 ของภาครับ 48
รูปที่ 3.8 วงจรขับมอเตอร์ 49
บทที่ 4 รูปที่ 4.1 FORWARD (duty cycle = 50%) 51
รูปที่ 4.2 FORWARD (duty cycle = 100%) 52
รูปที่ 4.3 BACKWARD (duty cycle = 50%) 53
รูปที่ 4.4 BACKWARD (duty cycle = 100%) 54
รูปที่ 4.5 TURN RIGHT (duty cycle = 50%) 55
รูปที่ 4.6 TURN RIGHT (duty cycle = 100%) 56
รูปที่ 4.7 TURN LEFT (duty cycle = 50%) 57
รูปที่ 4.8 TURN LEFT (duty cycle = 100%) 58
รูปที่ 4.9 เดินหน้าเฉียง 25 องศา 59
รูปที่ 4.10 ถอยหลังเฉียง 25 องศา 60
รูปที่ 4.11 เดินหน้าเฉียง 60 องศา 61
รูปที่ 4.12 ถอยหลังเฉียง 60 องศา 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

บทที่ 2	ตารางที่ 2.1 การทำงานของขาสัญญาณ dsPIC30F4012	9
	ตารางที่ 2.2 รีจีสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการใช้งาน A/D ของ dsPIC30F4012	13
	ตารางที่ 2.3 การทำงานของขาสัญญาณ PIC16F876	20
	ตารางที่ 2.4 การกำหนดสภาวะการทำงานของขาสัญญาณต่าง ๆ	22
	ตารางที่ 2.5 รีจีสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการใช้งาน A/D ของPIC16F876	25
บทที่4	ตารางที่ 4.1 การแปลงค่าแรงดันของจอยสติกจากอนาลอกเป็นดิจิตอล	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ปัจจุบันมีการนำวงจรอิเล็กทรอนิกส์มาประยุกต์ใช้งานกันอย่างมากมาย เพื่ออำนวยความสะดวก ประหยัดเวลาและเป็นการพัฒนาความรู้ ซึ่งโครงงานนี้ได้นำเอาวงจรอิเล็กทรอนิกส์มาประยุกต์ใช้กับระบบลาดตระเวนระยะไกลโดยสร้างเป็นรถลาดตระเวนจำลอง ซึ่งจากเดิมการลาดตระเวนจะใช้คนเข้าไปลาดตระเวนซึ่งมีความไม่ปลอดภัย จึงมีการใช้วงจรอิเล็กทรอนิกส์เข้าไปในพื้นที่ที่เสี่ยงภัยเพื่อลดความเสียหาย โดยโครงงานนี้ใช้การควบคุมคันบังคับ(Joystick) ในการควบคุมให้รถเคลื่อนที่ตามต้องการ ส่วนตัวรถลาดตระเวนเป็นการจำลองขนาดของรถลาดตระเวน เพื่อเป็นต้นแบบในการพัฒนาต่อไป ซึ่งเปรียบเสมือนการควบคุมรถลาดตระเวนจากฐานควบคุมในการตรวจจับสิ่งแปลกปลอมหรือสิ่งที่อันตรายผ่านกล้องที่ติดไว้บนรถลาดลาดตระเวน

1.2 วัตถุประสงค์

วัตถุประสงค์ของโครงงานนี้จะเป็นการจำลองระบบลาดตระเวนระยะไกลด้วยการสร้างรถลาดตระเวนจำลอง ซึ่งขนาดและรูปร่างของรถลาดตระเวนอาจจะไม่ตรงตามความเป็นจริง โดยโครงงานนี้เน้นที่หลักการทำงานของรถลาดตระเวนและหลักการทำงานซึ่งสามารถประยุกต์ให้เข้ากับรถลาดตระเวนของจริงในพื้นที่จริงได้

1.3 ขอบเขตของโครงงาน

เทอม 1/2550

ในเทอมแรกนี้จะทำการศึกษาภาษา C รวมถึงศึกษาคอมไพเลอร์ CCS C ที่ใช้ในการเขียนโปรแกรม รวมทั้งศึกษาโปรแกรม MPLAB ที่ใช้ในการ Burn โปรแกรม ส่วนจอยสติกนั้นจะศึกษาว่าวงจรของจอยสติกเป็นอย่างไร เอาท์พุทที่ได้เป็นอย่างไร เมื่อเราโยกจอยสติกแล้วค่าของเอาท์พุทเป็นอย่างไรแล้วเขียนโปรแกรมการแปลงค่าอนาลอกเป็นดิจิตอล และออกแบบฮาร์ดแวร์ในส่วนของตัวรถลาดตระเวนจำลอง

เทอม 2/2550

ในเทอม 2 นี้ จะการเขียนโปรแกรมในส่วนของภาครับ-ส่งของไมโครคอนโทรเลอร์เพื่อส่งสัญญาณและนำไปขับมอเตอร์ทั้ง 2 ตัวให้เสร็จ และได้เพิ่มกล้องเข้ามาติดบนตัวรถลาดตระเวนเพื่อที่จะตรวจจับสิ่งต่างๆ ในสถานที่ที่อันตราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 โครงสร้างของโครงงาน

โครงงานนี้ได้อธิบายขั้นตอนและวิธีในการออกแบบรวมทั้งวงจรและผลการทดลองต่างๆ ของรถลาดตระเวนจำลอง โดยมีเนื้อหาแบ่งออกเป็นบทต่างๆ ดังนี้

บทที่ 2 ทฤษฎีจะกล่าวถึงหลักการพื้นฐานต่างๆที่เกี่ยวกับการออกแบบและสร้างระบบลาดตระเวน

บทที่ 3 การออกแบบ จะกล่าวถึงขั้นตอนในการออกแบบส่วนต่างๆ ตั้งแต่ส่วนอุปกรณ์ควบคุมระยะไกล จอยสติกไปจนถึงวงจรภาครับบนตัวรถลาดตระเวนจำลอง

บทที่ 4 การทดลองและผลการทดลอง จะกล่าวถึงการทดลองและผลการทดลอง เมื่อทำการทดสอบ

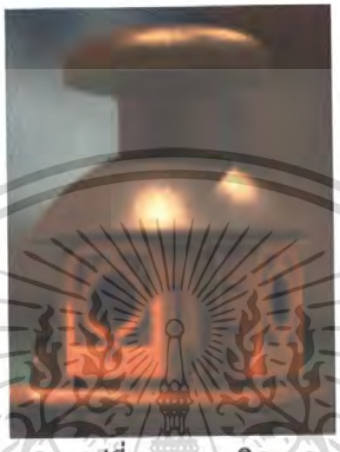
บทที่ 5 สรุปผล จะกล่าวถึงผลการทดลอง ปัญหาที่เกิดขึ้นและการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 จอยสติก (Joystick)

โครงสร้างจอยสติก



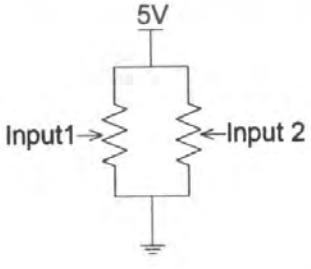
รูปที่ 2.1 จอยสติก

โครงสร้างภายนอกของจอยสติกประกอบด้วย

- คันโยก
- ปุ่มเคลื่อนปรับความต้านทานในแนวแกน X
- ปุ่มเคลื่อนปรับความต้านทานในแนวแกน Y

โครงสร้างภายในและการทำงานของจอยสติก

ในการใช้งานจอยสติกจะต้องศึกษาถึงวงจรภายในของจอยสติกที่ทำหน้าที่เปลี่ยนการโยก ซึ่งเป็นงานทางกลให้เป็นค่าทางไฟฟ้าโดยแสดงวงจรภายใน ดังรูป



รูปที่ 2.2 วงจรภายในจอยสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโยกจอยสติกไปในทิศทางต่างๆ ทำให้เกิดการเปลี่ยนค่าความต้านทานในแนวแกน Y และแนวแกน X โดยค่าความความต้านทานในแนวแกน Y จะเกิดการเปลี่ยนแปลงเมื่อโยกจอยสติกไปข้างหน้าหรือข้างหลัง ส่วนความต้านทานในแนวแกน X นั้นจะเกิดการเปลี่ยนแปลงเมื่อโยกจอยสติกไปทางขวาหรือซ้าย จุดกึ่งกลางเป็นจุดที่ค่าความต้านทานของแกน X และแกน Y อยู่กึ่งกลางการเปลี่ยนแปลงค่าความต้านทาน

2.2 มอเตอร์กระแสตรง (DC Motor)

นิยามสั้นๆ ของมอเตอร์ คือ อุปกรณ์ไฟฟ้าที่ทำหน้าที่เปลี่ยนพลังงานมาเป็นพลังงานกล โดยอาศัยหลักการดูดและผลักของสนามแม่เหล็ก การหมุนของมอเตอร์โดยสมมติ ถ้ามีขั้วแม่เหล็กอยู่ตรงข้ามขั้วต่างกัน เรียกว่า โพล (Pole) ซึ่งจะให้สนามแม่เหล็กออกมาเรียกว่า ฟิลด์ฟลักซ์(Field flux) เหล็กแท่งหนึ่งที่ติดอยู่ที่แกนหมุนก็ให้สนามแม่เหล็กออกมาเรียกว่า อาร์เมเจอร์ฟลักซ์ (Armature flux) ถ้าเริ่มวางตำแหน่งให้เยื้องนิดๆพอให้มีแรงเริ่มผลัก แท่งแม่เหล็กจะถูกขั้วเหล็กหัวท้ายทั้งสองผลักให้หมุนตามเข็มนาฬิกาไป เมื่อแท่งแม่เหล็กหมุนไปได้ 180 องศา เราก็จะทำการสลับขั้วแม่เหล็กที่หมุนนั้นกลับกันแท่งแม่เหล็กก็จะถูกผลักให้หมุนต่อไปเรื่อยๆ มอเตอร์ไฟตรงที่เรารู้จักกันทั่วไปคือมอเตอร์ที่สตาร์ทเครื่องยนต์นั่นเองหรือที่ช่างมักเรียกกันว่า ไดสตาร์ท ลักษณะโครงสร้างทั่วไปของมอเตอร์ไฟตรงจะเหมือนกับเจเนอเรมอเตอร์ไฟตรง ฉะนั้นการศึกษาก็สามารถเข้าใจได้ไม่ยาก

ชนิดของมอเตอร์ไฟตรง

1. **มอเตอร์ไฟตรงแบบอนุกรม(D.C.series motor)**

เป็นมอเตอร์ที่มีขดลวดสนามแม่เหล็กพันด้วยลวดเส้นใหญ่ เรียกว่าขดซีรีส์ฟิลด์และต่ออนุกรมเข้ากับอาร์เมเจอร์ ขดซีรีส์ฟิลด์จะพันรอบโพลเพื่อสร้างสนามแม่เหล็ก และขดซีรีส์มีความต้านทานต่ำ กระแสที่ไหลผ่านขดลวดฟิลด์จะเท่ากับกระแสอาร์เมเจอร์เพราะต่ออนุกรมกันอยู่ มอเตอร์ไฟตรงชนิดนี้จะให้คุณสมบัติแตกต่างออกไปจากมอเตอร์ไฟตรงแบบชันต์ งานที่ใช้กับมอเตอร์ชนิดนี้ส่วนใหญ่ต้องการแรงหมุนเริ่มแรกสูงถึง 500% ของค่าเต็มพิกัด ซึ่งมากกว่าแรงหมุนสูงสุดที่มอเตอร์แบบชันต์ทำได้

2. **มอเตอร์ไฟตรงแบบชันต์**

ชันต์มอเตอร์เป็นมอเตอร์ที่มีขดลวดสนามแม่เหล็กพันด้วยลวดเส้นเล็ก เรียกว่าขดชันต์ฟิลด์ ขดลวดที่ต่อขนานอยู่กับอาร์เมเจอร์ ขดชันต์ฟิลด์มีความต้านทานสูง ซึ่งจะก่อให้เกิดสนามแม่เหล็กคงที่เป็นสนามแม่เหล็กคงที่เป็นฟิลด์ฟลักซ์ทำหน้าที่ออกแรงผลักทุ่นอาร์เมเจอร์ให้หมุนไป มอเตอร์ไฟตรงแบบชันต์มีเทคนิคการใช้งานเพิ่มเติม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้ในการควบคุมแรงหมุนให้คงที่แต่ความเร็วรอบเปลี่ยนได้ ทำโดยรักษาสนามแม่เหล็กของขดลวดชันต์ให้คงที่ แต่กลับมาปรับแรงดันที่ป้อนให้แก่ขดลวดอาร์เมเจอร์ การทำแบบนี้จะทำให้ใช้แหล่งจ่ายไฟตรงถึง 2 ชุด ชุดหนึ่งจ่ายให้กับชันต์ อีกชุดหนึ่งจ่ายให้กับอาร์เมเจอร์

- ใช้ในการควบคุมโหลดที่มีความเร็วตกลงตามแรงหมุน โหลดบางชนิด เช่น พัดลมและปั๊มน้ำมันจะมีความเร็วลดลงเมื่อแรงหมุนลดลง ซึ่งขัดกับลักษณะของมอเตอร์ที่ว่าโหลดมากขึ้นรอบจะต่ำลงการปรับกระแสทำได้โดยการต่อความต้านทานอนุกรมกับขดลวดอาร์เมเจอร์

3. มอเตอร์ไฟตรงแบบคอมปาวด์ (D.C. compound motor)

คอมปาวด์มอเตอร์ เป็นมอเตอร์ที่มีขดลวดสนามแม่เหล็ก 2 ชุด พันอยู่บนขั้วแม่เหล็กอันเดียวกันโดยชุดแรก คือ ขดชันต์ซึ่งจะพันไว้ด้านในสุดของขั้วแม่เหล็ก ส่วนชุดที่สองคือขดซีรีส์ฟิลด์จะพันไว้ด้านนอกเป็นเส้นลวดเส้นใหญ่และมีจำนวนรอบเพียงเล็กน้อย

คุณลักษณะและการนำไปใช้งานของมอเตอร์ไฟฟ้ากระแสตรง

1. มอเตอร์ชนิดนี้ขณะได้รับโหลดมากๆ มอเตอร์จะหมุนช้า แต่ในขณะที่มอเตอร์มีโหลดน้อยๆ ความเร็วของมอเตอร์จะสูง แต่ถ้าไม่มีโหลดเลยจะมีความเร็วสูงจนเกิดอันตราย เพราะในขณะที่มีโหลดน้อยหรือไม่มีโหลดเลยกระแสจะไหลผ่านขดซีรีส์น้อยมาก ดังนั้นต้องไม่นำไปใช้กับชุดสายพานใดๆ เพราะบางครั้งอาจเผลอปลดเอาสายพานออกมอเตอร์อาจชำรุดได้

2. ชันต์มอเตอร์เมื่อจ่ายแรงดันไฟฟ้าให้กับมอเตอร์คงที่ จะทำให้เกิดสนามแม่เหล็กคงที่ ในขณะที่ไม่มีโหลดมอเตอร์จะใช้กระแสเพียงเล็กน้อยเพื่อเอาชนะความฝืด เมื่อมอเตอร์มีโหลดกระแสในอาร์เมเจอร์จะสูงขึ้น แต่ความเร็วจะลดลงเล็กน้อย กระแสในอาร์เมเจอร์จะเปลี่ยนตามโหลดคือ ถ้าโหลดมากกระแสจะสูง ถ้าลดโหลดน้อยลงกระแสจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไมโครคอนโทรลเลอร์ ชนิด dsPIC

โครงสร้างและสถาปัตยกรรม

dsPIC เป็นชื่อของไมโครคอนโทรเลอร์ 16 บิต โดยชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรเลอร์อนุกรมใหม่นี้ว่า Digital Signal Controller หรือ DSC นั่นก็คือ dsPIC เป็นไมโครคอนโทรเลอร์ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิตอลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง ไมโครคอนโทรเลอร์ได้มีการคิดค้นพัฒนาอย่างต่อเนื่อง จนมีไมโครคอนโทรเลอร์ตระกูลใหม่ๆ เกิดขึ้นมากมาย แต่ในที่นี้เราจะศึกษาการใช้งานไมโครคอนโทรเลอร์ตระกูล dsPIC ของบริษัท Microchip ซึ่งได้รับความนิยมอย่างแพร่หลายในบ้านเรา เนื่องจากเป็นไมโครคอนโทรเลอร์ที่มีความสามารถ และ เพียบพร้อมไปด้วยทรัพยากรหรือ ฟังก์ชันการใช้งานต่าง ๆ มากมาย เช่น โมดูล Analog to Digital , Timer/Counter , UART , SPI , I²C , PWM และอื่นๆ ซึ่งส่วนต่าง ๆ เหล่านี้ถูกสร้างรวมอยู่ภายใน CPU เพียงตัวเดียว ทำให้ CPU เพียงตัวเดียวนี้ สามารถทำงานได้หลายอย่างและสามารถลดในส่วนของฮาร์ดแวร์บางอย่างลง

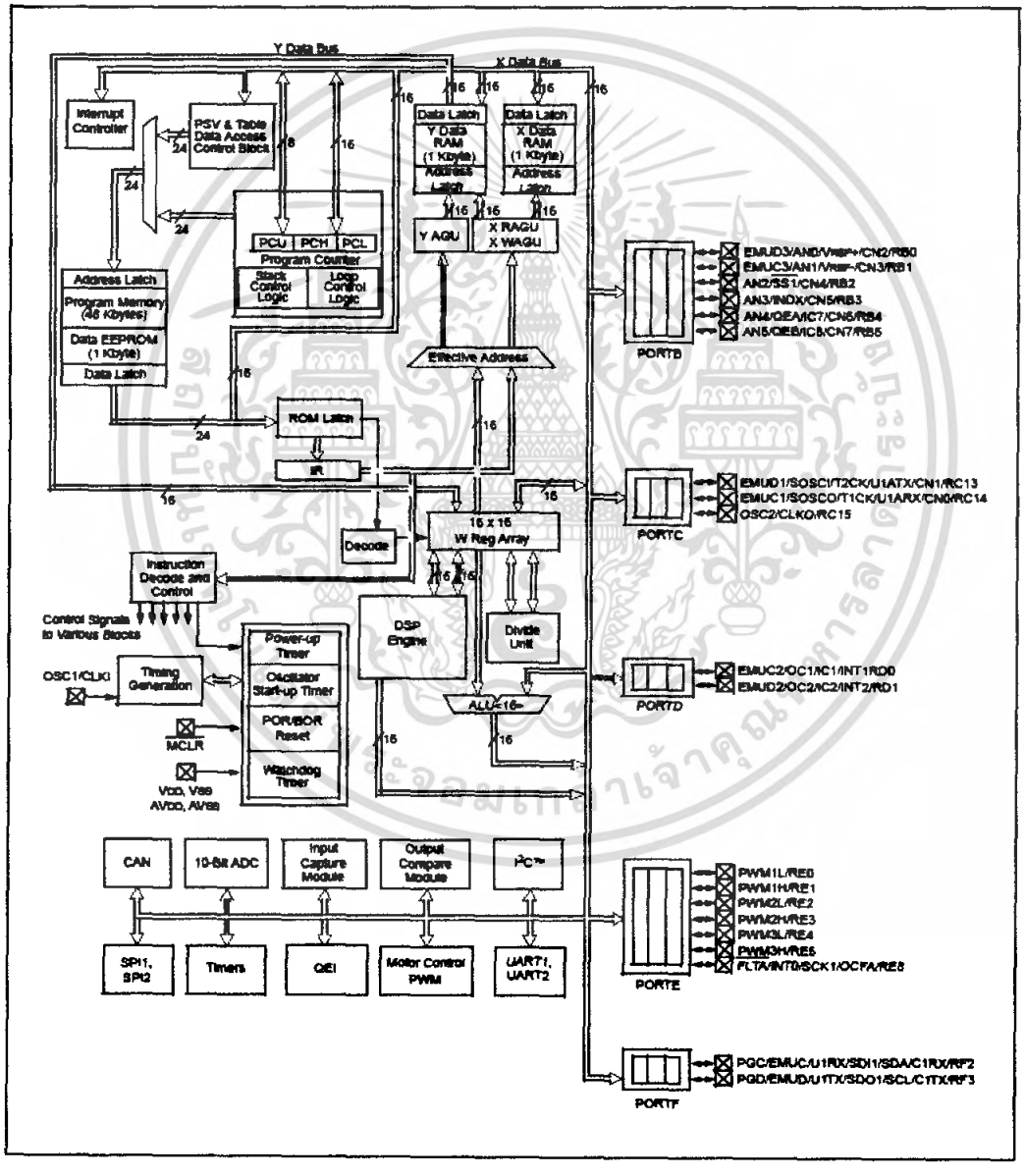
2.3.1 คุณสมบัติของ dsPIC 30F4012

คุณสมบัติต่าง ๆ ของไมโครคอนโทรเลอร์ dsPIC 30F4012 สรุปอย่างคร่าวๆ ได้ดังนี้

- 83 Base Instructions
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต
- มีอินเตอร์รัปต์เวคเตอร์ 30 ตำแหน่ง
- DC to 40 MHz external clock input
- 4 MHz – 10 MHz oscillator input
- หน่วยความจำ Program Memory มีขนาด 48K
- หน่วยความจำ Static RAM 2048 Bytes
- เพาเวอร์ออนรีเซต (POR),เพาเวอร์อัพไทเมอร์ (PWRT) และ Oscillator Start-Up Timer
- Extended Watcthdog Timer (WDT)
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- โหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของ สัญญาณนาฬิกาได้หลายโหมด
- ทำงานที่ไฟเลี้ยง 2.5 V ถึง 5.5 V
- กระแสทั้งซิงก์และซอร์สของพอร์ต คือ 25 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

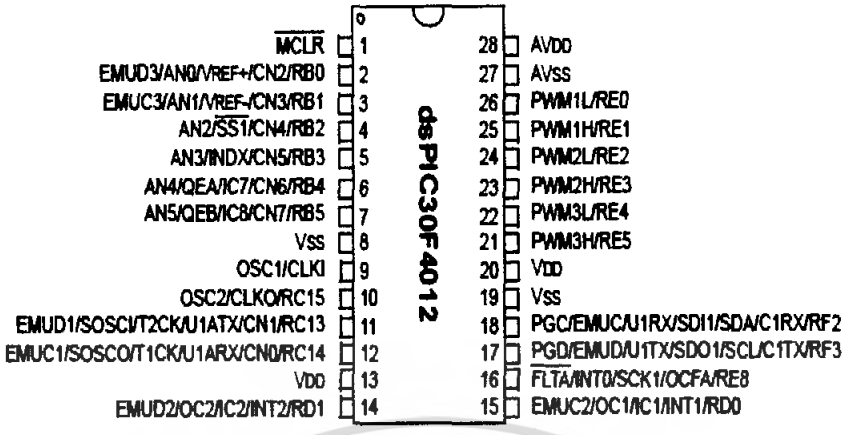
- Timer/Counter 16 บิต จำนวน 5 ตัว
- โมดูล Capture/Compare/PWM จำนวน 6 แชนแนล
- มีโมดูลสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์ 6 ช่อง
- Analog to Digital Converter ความละเอียด 10 บิต 6 แชนแนล
- อัตราการสุ่มและแปลงสัญญาณ 1000 กิโลแซมเปิลต่อวินาที
- มีโมดูลการสื่อสาร UART
- มีพอร์ต I/O 5 พอร์ตประกอบด้วย B,C,D,E และ F แต่ละพอร์ตมีจำนวนบิตไม่เท่ากันซึ่งรวมแล้ว จะมี จำนวน 20 บิต



รูปที่ 2.3 แสดงสถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ dsPIC 30F4012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28-Pin SPDIP and SOIC



รูปที่ 2.4 แสดงตัวถังของ dsPIC 30F4012 และ การจัดวางตำแหน่งขาสัญญาณต่าง ๆ

ขาสัญญาณของ dsPIC 30F4012 นี้จะมีทั้งหมด 28 ขา ประกอบไปด้วยขาที่ทำหน้าที่ต่าง ๆ โดยสามารถนำไปใช้เป็นอินพุต/เอาต์พุตได้ทั้งหมดทุกขา สามารถใช้งานได้ตามปกติ นอกจากขาสัญญาณ I/O แล้ว ยังประกอบไปด้วยขาสัญญาณอื่น ๆ อีกคือ ขาไฟเลี้ยง ,กราวด์ ,ขารีเซท และขาออสซิลเลเตอร์ สามารถสรุปหน้าที่การทำงานของขาสัญญาณในแต่ละขาได้ดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 การทำงานของขาสัญญาณ dsPIC30F4012

Pin Name	Pin Type	Buffer Type	Description
AN0-AN5	I	Analog	Analog input channels. AN0 and AN1 are also used for device programming data and clock inputs, respectively.
AVDD	P	P	Positive supply for analog module.
AVSS	P	P	Ground reference for analog module.
CLKI CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN7	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
C1RX C1TX	I O	ST —	CAN1 bus receive pin. CAN1 bus transmit pin.
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2 EMUD3 EMUC3	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST ST ST	ICD Primary Communication Channel data input/output pin. ICD Primary Communication Channel clock input/output pin. ICD Secondary Communication Channel data input/output pin. ICD Secondary Communication Channel clock input/output pin. ICD Tertiary Communication Channel data input/output pin. ICD Tertiary Communication Channel clock input/output pin. ICD Quaternary Communication Channel data input/output pin. ICD Quaternary Communication Channel clock input/output pin.
IC1, IC2, IC7, IC8	I	ST	Capture inputs 1, 2, 7 and 8.
INDX QEA QEB	I I I	ST ST ST	Quadrature Encoder Index Pulse input. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode.
INT0 INT1 INT2	I I I	ST ST ST	External interrupt 0. External interrupt 1. External interrupt 2.
$\overline{FLTA}$ PWM1L PWM1H PWM2L PWM2H PWM3L PWM3H	I O O O O O O	ST — — — — — —	PWM Fault A input. PWM1 low output. PWM1 high output. PWM2 low output. PWM2 high output. PWM3 low output. PWM3 high output.
$\overline{MCLR}$	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active-low Reset to the device.
OCFA OC1, OC2	I O	ST —	Compare Fault A input (for Compare channels 1, 2, 3 and 4). Compare outputs 1 and 2.

Legend: CMOS = CMOS compatible input or output; Analog = Analog input; ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input; P = Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 การทำงานของขาสัญญาณ dsPIC30F4012 (ต่อ)

Pin Name	Pin Type	Buffer Type	Description
OSC1	I	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
OSC2	I/O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.
PGD	I/O	ST	In-Circuit Serial Programming™ data input/output pin.
PGC	I	ST	In-Circuit Serial Programming clock input pin.
RB0-RB5	I/O	ST	PORTB is a bidirectional I/O port.
RC13-RC15	8I/O	8ST	PORTC is a bidirectional I/O port.
RD0-RD1	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE5, RE8	I/O	ST	PORTE is a bidirectional I/O port.
RF2-RF3	I/O	ST	PORTF is a bidirectional I/O port.
SCK1	I/O	ST	Synchronous serial clock input/output for SPI1.
SDI1	I	ST	SPI1 Data In.
SDO1	O	—	SPI1 Data Out.
$\overline{SS1}$	I/O	ST	SPI1 Slave Synchronization
SCL	I/O	ST	Synchronous serial clock input/output for I^2C™.
SDA	I/O	ST	Synchronous serial data input/output for I^2C.
SOSCO	O	—	32 kHz low-power oscillator crystal output.
SOSCI	I	ST/CMOS	32 kHz low-power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
U1RX	I	ST	UART1 receive.
U1TX	O	—	UART1 transmit.
U1ARX	I	ST	UART1 alternate receive.
U1ATX	O	—	UART1 alternate transmit.
VDD	P	—	Positive supply for logic and I/O pins.
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog voltage reference (high) input.
VREF-	I	Analog	Analog voltage reference (low) input.

Legend: CMOS = CMOS compatible input or output Analog = Analog input
ST = Schmitt Trigger input with CMOS levels O = Output
I = Input P = Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 โมดูล Analog to Digital Converter

การศึกษาการใช้งานโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ซึ่งเป็นโมดูลที่อยู่ในตัวไมโครคอนโทรลเลอร์ dsPIC 30F4012 จะมี 10 เซนแนล สำหรับไมโครคอนโทรเลอร์ dsPIC 30F4012 ที่มี 28 ขา แต่ทั้ง 10 เซนแนลนี้ จะไม่สามารถทำงานพร้อมกันได้ทั้งหมด โดนการทำงานจะเป็นลักษณะของการมัลติเพล็กซ์ ส่วนเรื่องของความละเอียดโมดูล A/D นี้มีความละเอียด 10 บิต โดยการใช้งานในโมดูล A/D นี้จะมีรีจิสเตอร์ใช้งานหลัก ๆ 5 ตัวคือ

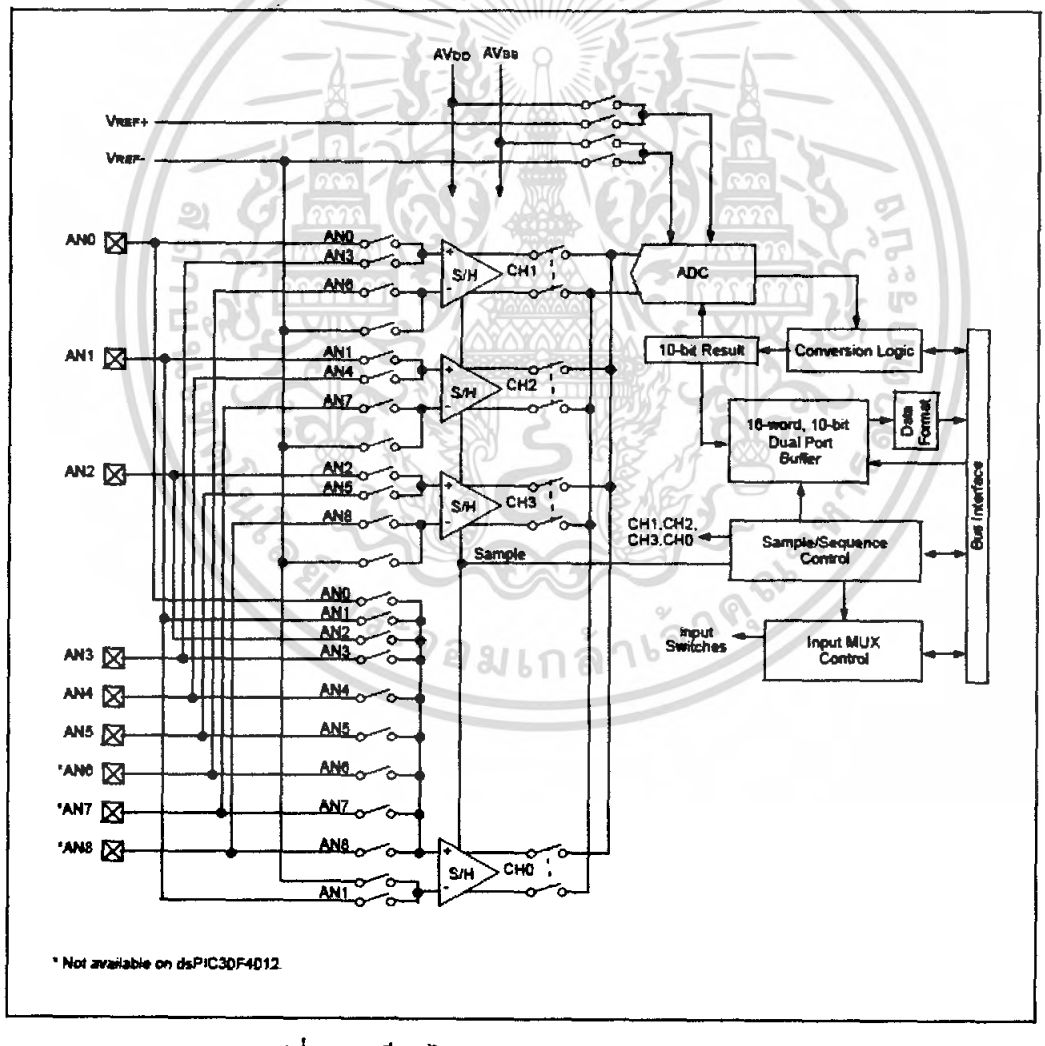
ADPCFG : เป็นรีจีสเตอร์ Port Configuration Register

ADCSSL : เป็นรีจีสเตอร์ Input Scan Selection Register

ADCHS : เป็นรีจีสเตอร์ Input Select Register

ADCON1 : เป็นรีจีสเตอร์ควบคุมการทำงานของ A/D ตัวที่ 1

ADCON2 : เป็นรีจีสเตอร์ควบคุมการทำงานของ A/D ตัวที่ 2



รูปที่ 2.5 บล็อกไดอะแกรมภายในของโมดูล A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขั้นตอนในการใช้งานโมดูลแปลงสัญญาณ A/D จะเป็นดังนี้

1. กำหนดค่าการทำงานต่างๆของโมดูล A/D
    - กำหนดหน้าที่การทำงานของขาสัญญาณต่างๆในรีจิสเตอร์ ADCONx
    - กำหนดแชนแนลหรือขาสัญญาณที่ใช้เป็นขาอินพุตอนาลอกในรีจิสเตอร์ ADCONx
    - กำหนดค่าเวลาในการแปลงสัญญาณ A/D ในรีจิสเตอร์ ADCONx
    - เปิดการทำงานของโมดูล A/D ในรีจิสเตอร์ADCONx ที่บิต ADON
2. กำหนดค่าการทำงานในส่วนของการอินเตอร์รัพท์ (เฉพาะกรณีที่ใช้งานอินเตอร์รัพท์)
    - เคลียร์บิตสถานะของการอินเตอร์รัพท์ ADIF เป็น '0'
    - เซตบิต ADIE เพื่ออนุญาตการอินเตอร์รัพท์จากโมดูล A/D
    - เซตบิต PEIE เพื่ออนุญาตการอินเตอร์รัพท์แบบ Peripheral Interrupt
    - เซตบิต GIE เพื่ออนุญาตการอินเตอร์รัพท์ทั้งหมด
3. รอเวลาให้โมดูล A/D พร้อมทำงาน
4. เริ่มการแปลงสัญญาณ(Start conversion) ด้วยการเซตบิต GO/DONE ใน ADCONx
5. รอการแปลงสัญญาณ เนื่องจากการแปลงสัญญาณ A/D นั้นต้องใช้เวลาช่วงหนึ่งซึ่งเราสามารถตรวจสอบได้ว่าการแปลงสัญญาณนี้เสร็จสิ้นแล้วหรือยัง ด้วยวิธีการตรวจสอบต่อไปนี้
    - วนเช็คค่าที่บิต GO/DONE ในรีจิสเตอร์ ADCONx โดยบิตนี้จะมีค่าเป็น '0' เมื่อการแปลงสัญญาณ A/D เสร็จสิ้น
    - ตรวจสอบโดยรอให้เกิดอินเตอร์รัพท์ ซึ่งเมื่อมีการแปลงสัญญาณ A/D เสร็จสิ้น บิต ADIF จะเซตเป็น '1' ทำให้เกิดการอินเตอร์รัพท์ (หากมีการEnableการอินเตอร์รัพท์นี้ไว้)
6. เมื่อกระบวนการแปลงสัญญาณเสร็จสิ้นผลลัพธ์ที่ได้จะอยู่ในรีจิสเตอร์ADRESH และ ADRESL และจะต้องทำการเคลียร์บิต ADIF ด้วยเพื่อไม่ให้มีการอินเตอร์รัพท์ซ้อนเกิดขึ้น
7. หากต้องการแปลงสัญญาณในครั้งต่อไปก็ไม่สามารถทำได้โดยกลับไปเริ่มทำในขั้นที่ 1 หรือ 2 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.2 รีจีสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการใช้งาน A/D ของ dsPIC 30F4012

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
ADCBUF0	0280	—	—	—	—	—	—	ADC Data Buffer 0										0000 00uu uuuu uuuu
ADCBUF1	0282	—	—	—	—	—	—	ADC Data Buffer 1										0000 00uu uuuu uuuu
ADCBUF2	0284	—	—	—	—	—	—	ADC Data Buffer 2										0000 00uu uuuu uuuu
ADCBUF3	0286	—	—	—	—	—	—	ADC Data Buffer 3										0000 00uu uuuu uuuu
ADCBUF4	0288	—	—	—	—	—	—	ADC Data Buffer 4										0000 00uu uuuu uuuu
ADCBUF5	028A	—	—	—	—	—	—	ADC Data Buffer 5										0000 00uu uuuu uuuu
ADCBUF6	028C	—	—	—	—	—	—	ADC Data Buffer 6										0000 00uu uuuu uuuu
ADCBUF7	028E	—	—	—	—	—	—	ADC Data Buffer 7										0000 00uu uuuu uuuu
ADCBUF8	0290	—	—	—	—	—	—	ADC Data Buffer 8										0000 00uu uuuu uuuu
ADCBUF9	0292	—	—	—	—	—	—	ADC Data Buffer 9										0000 00uu uuuu uuuu
ADCBUFA	0294	—	—	—	—	—	—	ADC Data Buffer 10										0000 00uu uuuu uuuu
ADCBUFB	0296	—	—	—	—	—	—	ADC Data Buffer 11										0000 00uu uuuu uuuu
ADCBUFC	0298	—	—	—	—	—	—	ADC Data Buffer 12										0000 00uu uuuu uuuu
ADCBUFD	029A	—	—	—	—	—	—	ADC Data Buffer 13										0000 00uu uuuu uuuu
ADCBUFE	029C	—	—	—	—	—	—	ADC Data Buffer 14										0000 00uu uuuu uuuu
ADCBUFF	029E	—	—	—	—	—	—	ADC Data Buffer 15										0000 00uu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>					ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>				CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	—	—	—	—	—	—	—	PCFG8*	PCFG7*	PCFG6*	PCFG5*	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	—	—	—	—	—	—	—	CSSL8*	CSSL7*	CSSL6*	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

Legend: u = uninitialized bit

* Not available on dsPIC30F4012

Note: Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2 การสื่อสารอนุกรม UART

การสื่อสารอนุกรมUART (Universal Asynchronous Receiver Transmitter) เป็นโมดูลการสื่อสารที่มีอยู่ภายในตัวไมโครคอนโทรลเลอร์ ซึ่งจะมีโหมดการทำงานอยู่ 2 โหมด คือ โหมดการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous Mode) และโหมดการสื่อสารอนุกรมแบบอะซิงโครนัส (Asynchronous Mode) ซึ่งการใช้งานจะใช้งานได้เพียงโหมดใดโหมดหนึ่งเท่านั้น ไม่สามารถใช้งานพร้อมกันได้ 2 โหมดได้ โดยสามารถนำไปใช้งานติดต่อกับอุปกรณ์ต่างๆ เช่น ติดต่อกับไมโครคอนโทรลเลอร์ด้วยกันหรือเครื่อง PC และอุปกรณ์ที่มีรูปแบบการสื่อสารเหมือนกัน โดยสามารถแบ่งการทำงานของโมดูลการสื่อสาร UART ออกเป็นโหมดต่างๆ ดังนี้

- โหมดการสื่อสาร Asynchronous แบบ Full - Duplex
- โหมดการสื่อสาร Synchronous- Master แบบ Half - Duplex
- โหมดการสื่อสาร Synchronous- Slave แบบ Half - Duplex

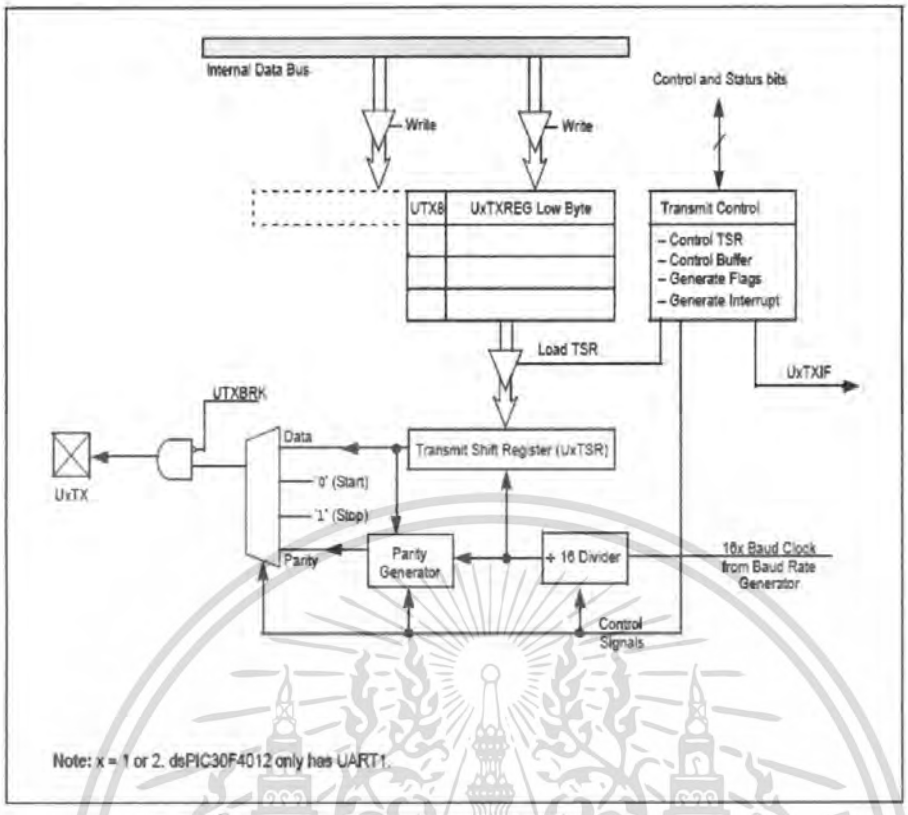
สำหรับ dsPIC 30F4012 มี UART Module Overview ดังนี้

1. การสื่อสารแบบ Full – Duplex สำหรับการสื่อสารแบบ 8 หรือ 9 บิต
2. มี Stop bit จำนวน 1 หรือ 2 บิต
3. อัตรา Baud Rates อยู่ในช่วง 38 bps ถึง 1.875 Mbps ที่ 30 MHz
4. Interrupts ส่วนรับและส่งสัญญาณข้อมูลแยกออกจากกัน
5. มีAddress สำหรับตรวจสอบ interrupts (9^{th} bit = 1)

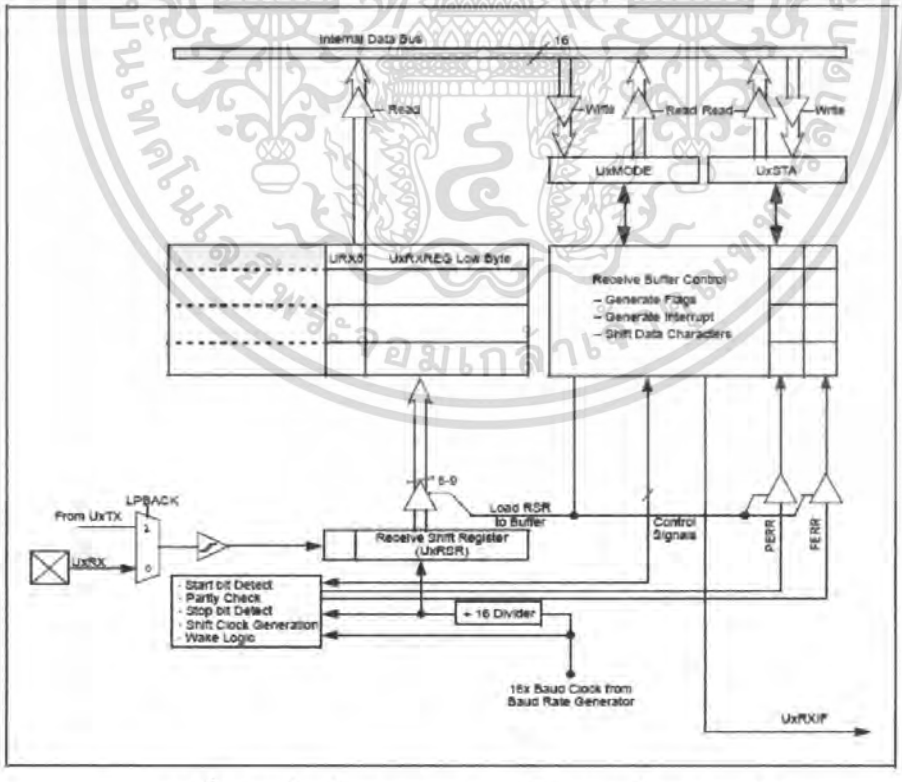
การสื่อสารในโหมดอะซิงโครนัส (Asynchronous Mode)

การสื่อสารในโหมดนี้จะเป็นลักษณะของการสื่อสารแบบ NRZ (non-return-to-zero) ในการส่งข้อมูลใน 1 เฟรมจะประกอบไปด้วย บิตstart จำนวน 1 บิตและบิตข้อมูล 8 บิตหรือ 9 บิต ปิดด้วยบิตstop อีกหนึ่งบิต ซึ่งรวมแล้วหากเป็นข้อมูลแบบ 8 บิต การส่งข้อมูลใน 1 เฟรม จะมีข้อมูล 10 บิต ส่วนการส่งข้อมูลแบบ 9 บิตเฟรมจะมีขนาด 11 บิต โดยในการส่งข้อมูลแบบ 9 บิตนี้ส่วนใหญ่ จะใช้บิตที่ 9 เป็นบิต พาริตี้(Parity bit) และเมื่ออยู่ในโหมดประหยัดพลังงาน(Sleep Mode) การสื่อสารนี้จะหยุดลง คือไม่สามารถทำงานในโหมด Sleep ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART



รูปที่ 2.7 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารแบบซิงโครนัส (Synchronous Mode)

การสื่อสารในโหมดซิงโครนัสนี้เป็นการสื่อสารแบบ ฮาล์ฟดูเพล็กซ์ (Half-duplex) ทำให้ไม่สามารถทำการส่งหรือรับข้อมูลในเวลาเดียวกันได้ ซึ่งก็คือเมื่อตัวใดตัวหนึ่งทำการส่งข้อมูล อีกตัวหนึ่งจะต้องทำการรับข้อมูล โดยในการสื่อสารแบบซิงโครนัสนี้จะประกอบด้วย ขาสัญญาณข้อมูล(DT:data) และขาสัญญาณนาฬิกา(CK:clock) การรับข้อมูลจะสัมพันธ์กับสัญญาณนาฬิกาทั้งตัวรับและตัวส่ง ทำให้การสื่อสารแบบนี้มีประสิทธิภาพมากกว่าการสื่อสารแบบอะซิงโครนัส แต่ในเรื่องของความเร็วในการส่งข้อมูลแบบนี้จะมีความเร็วต่ำกว่าแบบอะซิงโครนัส ซึ่งในการสื่อสารแบบซิงโครนัสนี้ จะแบ่งการทำงานออกเป็น 2 โหมด คือ

1. โหมดมาสเตอร์(Master Mode)

การสื่อสารในโหมดมาสเตอร์นี้ อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์จะเป็นที่ตัวกำหนดสัญญาณนาฬิกาในการสื่อสารข้อมูลเข้าสู่การทำงานในโหมดนี้จะต้องเซตบิต SYNC (TXSTA<4>) และทำการ Enable การสื่อสารข้อมูลโดยการเซตบิต SPEN เป็น “1” จะทำให้ขาสัญญาณ RC6/TX/CX และ RC7/RX/CK ทำหน้าที่เป็นขาสัญญาณนาฬิกา (CK:clock) และขาสัญญาณข้อมูล (DT:data) จากนั้นกำหนดการทำงานให้อยู่ในโหมดมาสเตอร์โดยการเซตบิตCSRC(TXSTA<7>) อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์นี้จะเป็นตัวหลักที่ทำหน้าที่ควบคุมการรับส่งข้อมูลและการกำเนิดสัญญาณนาฬิกา(Clock) ที่ใช้ในระบบ

2. โหมดสเลฟ(Slave Mode)

การสื่อสารแบบซิงโครนัสในโหมดสเลฟนี้จะต่างจากการสื่อสารในโหมดมาสเตอร์ในเรื่องของแหล่งกำเนิดสัญญาณนาฬิกาในการสื่อสารข้อมูล โดยในโหมดสเลฟนี้จะอาศัยสัญญาณนาฬิกาในการรับส่งข้อมูลจากภายนอกที่ขาสัญญาณ RC6/TX/CK ซึ่งในโหมดมาสเตอร์จะใช้สัญญาณนาฬิกาภายใน โดยสัญญาณนาฬิกาที่ส่งให้ขา RC6/TX/CK มาจากอุปกรณ์ที่เป็นมาสเตอร์นั้นเอง การรับส่งข้อมูลในโหมดสเลฟนี้สามารถเกิดขึ้นได้แม้ว่า CPU จะอยู่ในโหมดประหยัดพลังงาน (Sleep Mode) การกำหนดการทำงานในโหมดสเลฟทำได้โดยการเคลียร์บิต CSRC (TXSTA<7>) ซึ่งจะเป็นการเลือกรับสัญญาณนาฬิกาจากภายนอกนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ PIC 16F876

คุณสมบัติต่าง ๆ ของไมโครคอนโทรเลอร์ PIC 16F876 สามารถสรุปอย่างคร่าวๆ ได้ดังนี้

- 35 Instruction คำสั่ง
- ความถี่สูงสุดที่ทำงานคือ 20 HMz (16F877-20/P)
- หน่วยความจำโปรแกรม FLASH Program Memory มีขนาด 8k (14-Bit Words)
- หน่วยความจำข้อมูล (RAM) 368 Bytes
- หน่วยความจำข้อมูล (EEPROM) 256 Bytes
- สามารถตอบสนองการอินเตอร์รัพได้ถึง 14 แหล่ง
- STACK 8 ระดับ
- เพาเวอร์ออนรีเซต (POR),เพาเวอร์อัพไทเมอร์ (PWRT) และ Oscillator Start-Up Timer
- Watcthdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- โหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของ สัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- ฟังก์ชันการโปรแกรมแบบ ICSP (In-Circuit Serial Programming)
- ทำงานที่ไฟเลี้ยง 2.0V ถึง 5.5V
- กระแสทั้งซิงก์และซอร์สของพอร์ตคือ 25mA
- Timer/Counter จำนวน 3 ตัว คือ Timer0, Timer1และ Timer2
- โมดูล Capture/Compare/PWM จำนวน 2 ชุด
- Analog to Digital Converter ความละเอียด 10 บิต 8 แชนแนล
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับแรงดันไฟเลี้ยง Brown – out reset (BOR)
- มีพอร์ต I/O 5 พอร์ตประกอบด้วย A,B,C,D และ E แต่ละพอร์ตจะมาจำนวนบิตไม่เท่ากันซึ่งรวมแล้ว จะมี จำนวน 33 บิต

82997
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

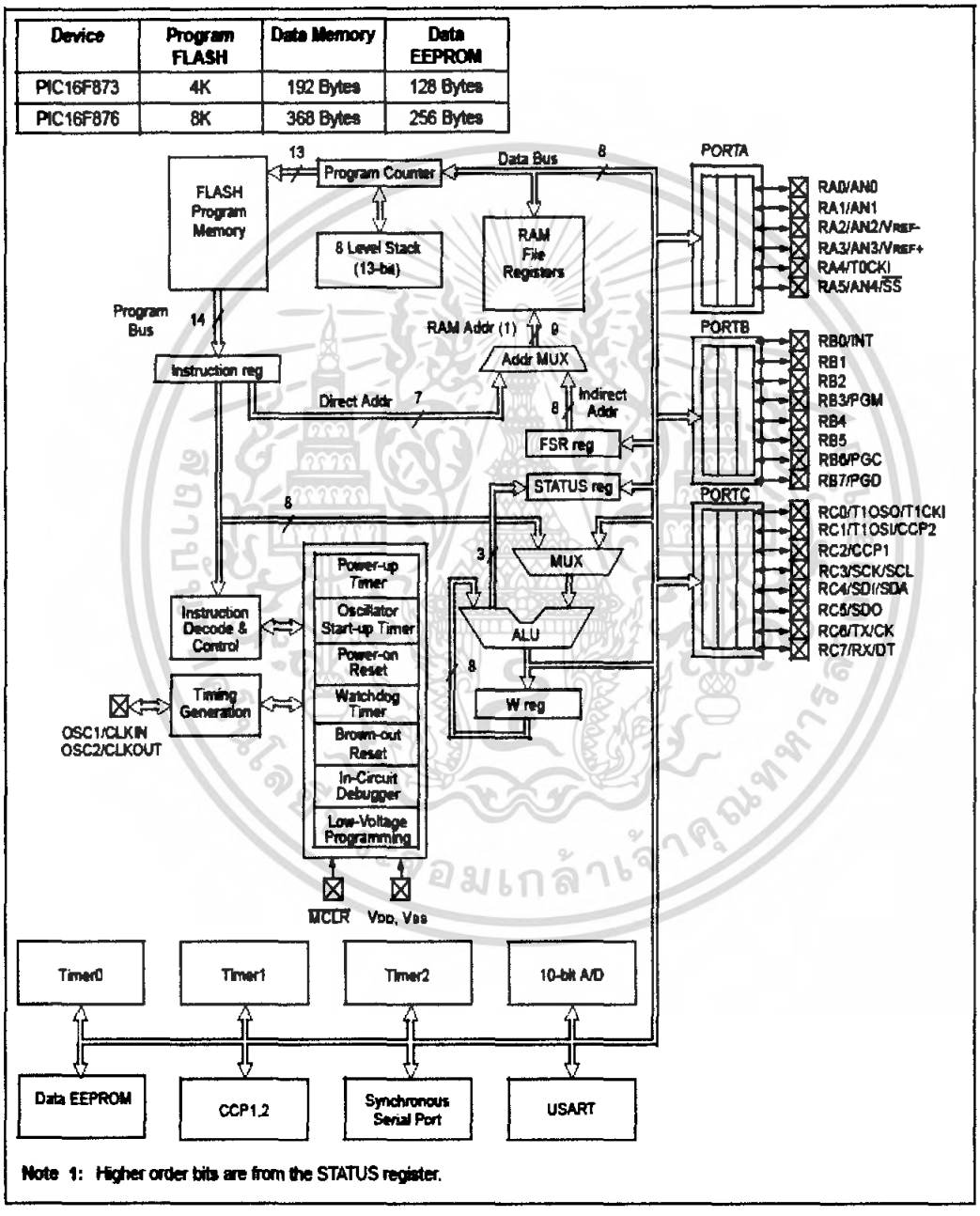
PORTA = RA5 – RA0 จำนวน 6 บิต

PORTB = RA7 – RA0 จำนวน 8 บิต

PORTC = RA7 – RA0 จำนวน 8 บิต

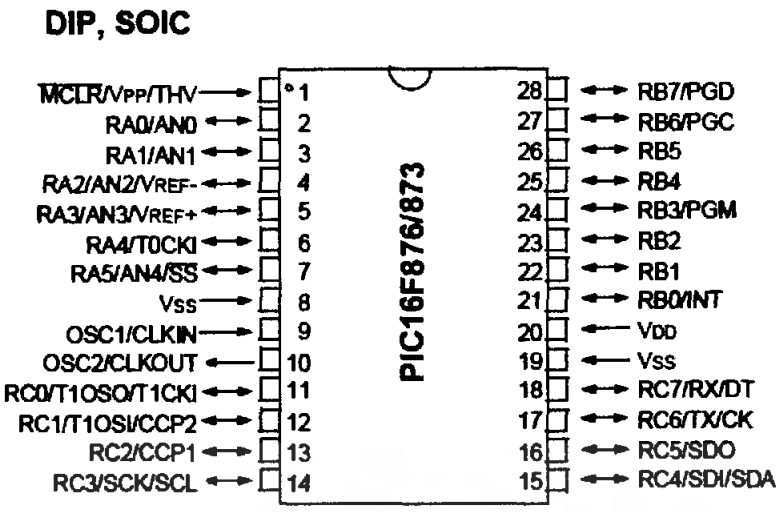
PORTD = RA7 – RA0 จำนวน 8 บิต

PORTE = RA2 – RA0 จำนวน 3 บิต



รูปที่ 2.8 แสดงสถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ PIC 16F876

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงตัวถังของ CPU PIC 16F876 และ การจัดวางตำแหน่งขาสัญญาณต่าง ๆ

ขาสัญญาณของ PLC เบอร์ PIC 16F876 นี้จะมีทั้งหมด 28 ขาจะประกอบไปด้วยขาที่ทำหน้าที่ต่าง ๆ โดยจะมีขาสัญญาณ I/O Ports ทั้งหมดจำนวน 22 ขา โดยสามารถนำไปใช้เป็นอินพุต/เอาต์พุตได้ทั้งหมดทุกขา ยกเว้นขา RA4 ซึ่งโครงสร้างภายในเป็นแบบ Open Drain ดังนั้นหากต้องการนำไปใช้เป็นขาสัญญาณเอาต์พุต จะต้องต่อตัวต้านทานพูลอัพ (Pull-Up) ไว้ด้วย ส่วนขาที่เหลือ สามารถใช้งานได้ตามปกติ นอกจากขาสัญญาณ I/O แล้ว ยังประกอบด้วยขาสัญญาณอื่น ๆ อีกคือ ขาไฟเลี้ยง ,กราวด์ ,ขารีเซ็ท และขาออสซิลเลเตอร์ สามารถสรุปหน้าที่การทำงานของขาสัญญาณในแต่ละขาได้ดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.3 การทำงานของขาสัญญาณ PIC 16F876

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP/THV	1	1	I/P	ST	Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input
RB4	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	I/O	TTL	Interrupt on change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โมดูล Analog to Digital Converter

การศึกษาการใช้งานโมดูลแปลงสัญญาณ อนาลอกเป็นดิจิตอล ซึ่งเป็นโมดูลที่อยู่ในตัวไมโครคอนโทรลเลอร์ PIC เบอร์ PIC 16F876 จะมี 8 เชนแนล (AN0-An7) แต่ทั้ง 8 เชนแนลนี้ จะไม่สามารถทำงานพร้อมกันได้ทั้งหมด โดนการทำงานจะเป็นลักษณะของการมัลติเพล็กซ์ ส่วนเรื่องของความละเอียดโมดูล A/D นี้มีความละเอียด 10 บิต และยังสามารถเลือกแรงดันอ้างอิงในการแปลงสัญญาณได้ ทั้งจากแรงดันไฟเลี้ยงภายใน CPU หรือ แรงดันจากแหล่งจ่ายไฟภายนอกโดยผ่านทางขาสัญญาณ RA2 และ RA3 โมดูล A/D นี้ยังลักษณะที่พิเศษอีกคือ สามารถทำงานได้แม้ว่าจะอยู่ในโหมดประหยัดพลังงาน (Sleep Mode) โดยการใช้งานในโมดูล A/D นี้จะมีรีจิสเตอร์ใช้งานหลัก ๆ 4 ตัวคือ

ADRESH: เป็นรีจีสเตอร์เก็บผลลัพธ์จากการแปลงสัญญาณ A/D ในไบต์สูง

ADRESL: เป็นรีจีสเตอร์เก็บผลลัพธ์จากการแปลงสัญญาณ A/D ในไบต์ต่ำ

ADCON0: เป็นรีจีสเตอร์ควบคุมการทำงานของ A/D ตัวที่ 1

ADCON1: เป็นรีจีสเตอร์ควบคุมการทำงานของ A/D ตัวที่ 2

ADCON0 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit7							bit0

Bit 7-6 ADCS1:ADCS0: บิตเลือกอัตราส่วนของสัญญาณนาฬิกาที่ใช้แปลงสัญญาณ A/D

00 = Fosc/2

01 = Fosc/8

10 = Fosc/32

11 = FRC สัญญาณนาฬิกาที่ได้มาจากวงจร RC ภายในโมดูล A/D

Bit 5-3 CHS2:CHS0: เป็นบิตที่ใช้มัลติเพล็ก เพื่อเลือกขาสัญญาณอินพุตสัญญาณอนาลอก

000 = channel 0: เลือกสัญญาณอินพุตจากแชนเนล 0 ที่ขาสัญญาณ RA0/AN0

001 = channel 1: เลือกสัญญาณอินพุตจากแชนเนล 1 ที่ขาสัญญาณ RA1/AN1

010 = channel 2: เลือกสัญญาณอินพุตจากแชนเนล 2 ที่ขาสัญญาณ RA2/AN2

011 = channel 3: เลือกสัญญาณอินพุตจากแชนเนล 3 ที่ขาสัญญาณ RA3/AN3

100 = channel 4: เลือกสัญญาณอินพุตจากแชนเนล 4 ที่ขาสัญญาณ RA5/AN4

101 = channel 5: เลือกสัญญาณอินพุตจากแชนเนล 5 ที่ขาสัญญาณ RE0/AN5

110 = channel 6: เลือกสัญญาณอินพุตจากแชนเนล 6 ที่ขาสัญญาณ RE1/AN6

111 = channel 7: เลือกสัญญาณอินพุตจากแชนเนล 7 ที่ขาสัญญาณ RE2/AN7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 2 GO/DONE: บิตแสดงสถานะการแปลงสัญญาณA/D (กำหนดบิตADCON = 1)

1 = เมื่อยังอยู่ในในระหว่างกระบวนการแปลงสัญญาณ A/D

0 = บิตนี้จะถูกเคลียร์เป็น "0" เองโดยอัตโนมัติเมื่อการแปลงสัญญาณ A/D เสร็จ

Bit 1 Unimplemented: ไม่ใช้งานอ่านค่าได้เท่ากับ "0"

Bit 0 ADON: บิตเปิด – ปิดการทำงานของโมดูล A/D

1 = เปิดการทำงานของโมดูลแปลงสัญญาณ A/D

0 = ปิดการทำงานของโมดูลแปลงสัญญาณ A/D

ADCON1 REGISTER

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit7							bit0

Bit 7 ADFM: เป็นบิตเลือกรูปแบบการเก็บผลลัพธ์จากการแปลงสัญญาณ A/D

1 = กำหนดการเก็บผลลัพธ์แบบชิดขวา (Right justified) คือ ข้อมูล 8 บิตล่างจะเก็บใน ADRESH ส่วน 2 บิตบนจะเก็บใน ADRESH <1:0> ทำให้ 6 บิตบนของ ADRESH <7:2> อ่านค่าเป็น "0"

0 = กำหนดการเก็บผลลัพธ์แบบชิดซ้าย (Left justified) คือ ข้อมูล 8 บิตบนจะเก็บใน ADRESH ส่วน 2 บิตล่างจะเก็บใน ADRESL <7:6>ทำให้ 6 บิตล่างของ ADRESL <5:0> อ่านค่าเป็น "0"

Bit 6-4 Unimplemented: ไม่ใช้งานอ่านค่าได้เท่ากับ "0"

Bit 3-4 PCFG3:PCFG0: บิตเลือกการทำงานของขาสัญญาณต่าง ๆ

ตาราง 2.4 การกำหนดสภาวะการทำงานของขาสัญญาณต่าง ๆ

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / REFS(2)
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input

D = Digital I/O

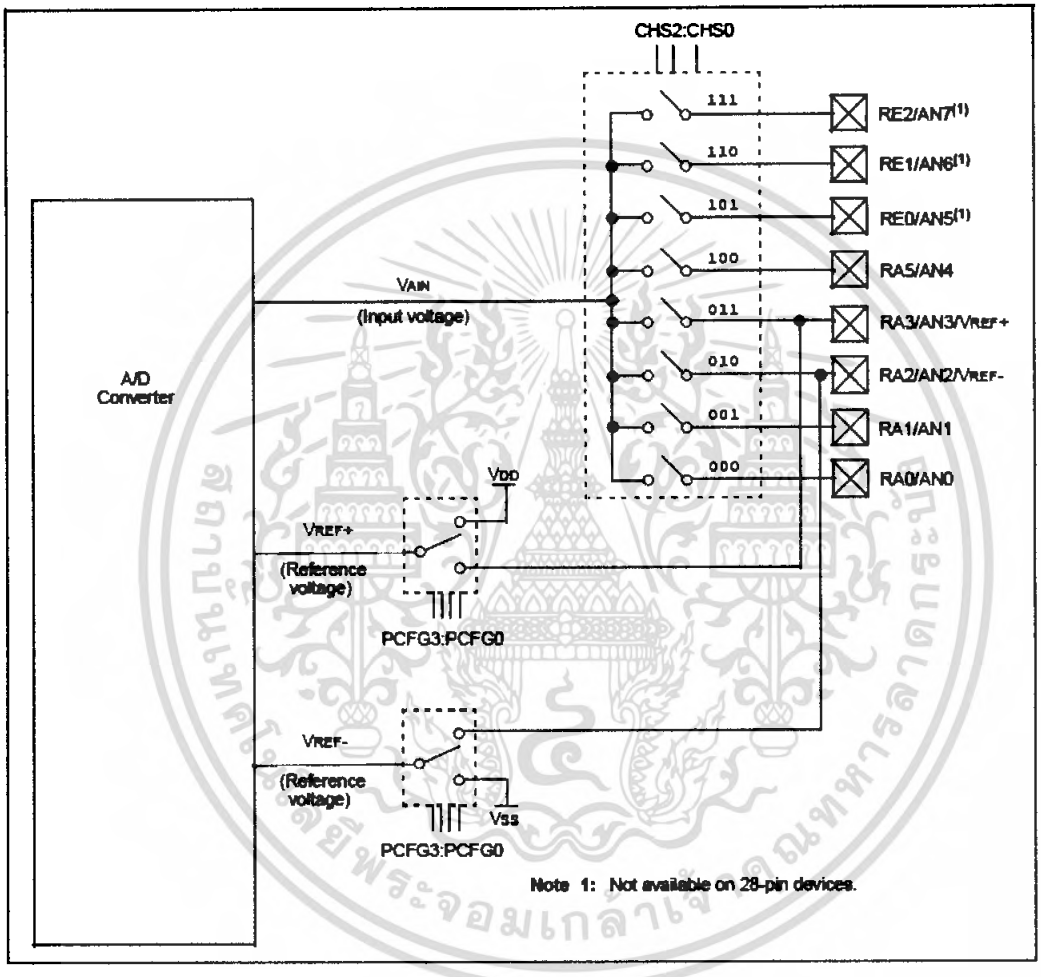
Note 1: These channels are not available on the 28-pin devices.

2: This column indicates the number of analog channels available as A/D inputs and the numer of analog channels used as voltage reference inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการแปลงสัญญาณของโมดูล A/D

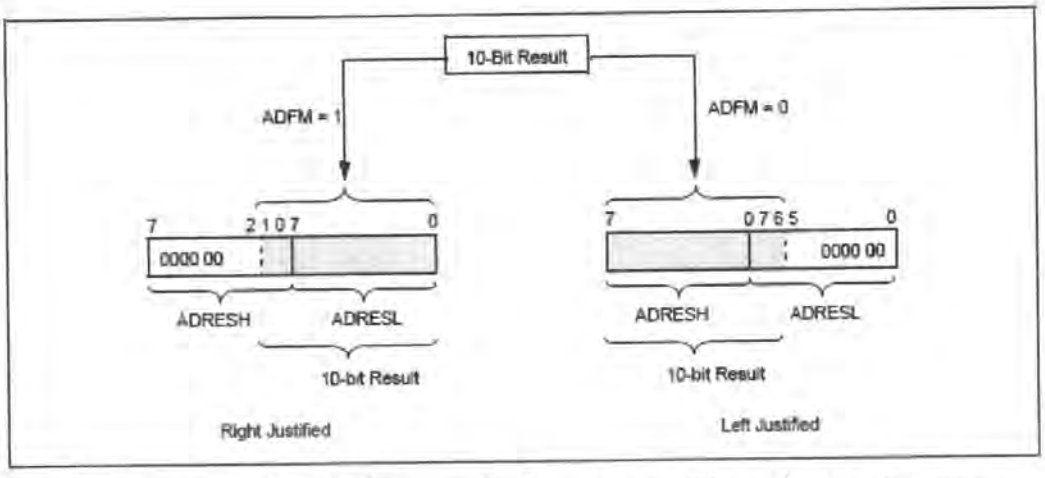
เมื่อผ่านกระบวนการในการแปลงสัญญาณในโมดูล A/D (Analog to Digital) แล้วผลลัพธ์ที่ได้ขนาด 10 บิตจะถูกเก็บในรีจิสเตอร์ ADRESH: ADRESL และหลังจากการแปลงสัญญาณเสร็จสิ้นบิต GO/DONE (ADCON0<2>) จะถูกเคลียร์เป็น "0" โดยอัตโนมัติดังนั้นเราสามารถที่จะทราบว่าการแปลงสัญญาณเสร็จสิ้นแล้วหรือไม่ โดยการตรวจสอบที่บิตนี้ หรือ จะใช้วิธีการอินเตอร์รัพท์เนื่องจากเมื่อมีการแปลงข้อมูลเสร็จ บิต ADIF จะถูกเซตเป็น "1" และจะมีการอินเตอร์รัพท์เกิดขึ้นหากมีการอนุญาตการอินเตอร์รัพท์ประเภทนี้ไว้



รูปที่ 2.10 บล็อกไดอะแกรมภายในของโมดูล A/D

การเก็บค่าผลลัพธ์จากการแปลงสัญญาณของรีจิสเตอร์ ADRESH: ADRESL นั้นสามารถทำได้ 2 รูปแบบคือ เก็บผลข้อมูลแบบชิดขวา (Right justified) และ แบบเก็บผลลัพธ์ข้อมูลแบบชิดซ้าย (Left justified) ดังจะเห็นได้ดังรูปด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงการเก็บผลลัพธ์จากการแปลงข้อมูล A/D ของรีจิสเตอร์ ADRESH: ADRESL

การทำงานของ A/D ในโหมดประหยัดพลังงาน (Sleep Mode)

โมดูล A/D นี้สามารถทำงานได้แม้จะอยู่ในโหมดประหยัดพลังงาน (Sleep Mode) แต่ต้องเลือกแหล่งสัญญาณนาฬิกาที่ใช้แปลงสัญญาณ A/D เป็นโหมด RC กำหนดที่บิต ADCS1: ADCS0 ให้เป็น "11" แต่หากเลือกแหล่งสัญญาณนาฬิกาเป็นโหมดอื่น ๆ เมื่อเข้าสู่โหมด Sleep จะทำให้การแปลงสัญญาณนั้นเกิดความผิดพลาด หรือ ล้มเหลว และ โมดูล A/D จะถูกปิดการทำงานลง แม้ว่าบิต ADON จะยังเป็น "1" อยู่ก็ตาม ดังนั้นหากต้องการใช้งาน A/D โหมดประหยัดพลังงานนี้จะต้องกำหนดให้ A/D ทำงานในโหมด RC (ADCS1: ADCS0 = 11) เท่านั้น

โดยเมื่ออยู่ในโหมด Sleep โมดูล A/D นี้จะสามารถทำให้เกิด Wake-up ขึ้นได้จากการอินเตอร์รัพท์ของโมดูล A/D เฉพาะในกรณีที่เราได้ทำการ Enable อินเตอร์รัพท์นี้ไว้เท่านั้น แต่หากไม่มีการ Enable การอินเตอร์รัพท์ การทำงานของโมดูล A/D ก็จะถูกปิดการทำงานลงเช่นกัน ถึงแม่ว่า บิต ADON จะยังคงเซตเป็น "1" อยู่ก็ตาม

ผลกระทบต่อโมดูล A/D จากการเกิดรีเซต

เมื่อเกิดการรีเซตของ CPU จะมีผลกระทบต่อการทำงานของโมดูล A/D คือ โมดูล A/D จะหยุดการทำงานและ ขาสัญญาณ I/O Ports ที่มีการทำงานในส่วนของโมดูล A/D คือ ขา AN7-AN0 จะถูกกำหนดให้เป็นขาสัญญาณอินพุตอนาลอก ดังนั้นหากต้องการนำขาสัญญาณเหล่านี้ไปใช้งานเป็นขาสัญญาณ Digital I/O จะต้องระวังในจุดนี้ด้วย ส่วนค่าในรีจิสเตอร์ ADRESH: ADRESL จะไม่เปลี่ยนแปลงโดยจะคงสถานะของค่าเดิมไว้ แต่หากเกิดรีเซตจาก Power-on Reset ค่าในรีจีสเตอร์ทั้ง 2 นี้จะอยู่ในสถานะใด ๆ (unknown data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.5 รีจีสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการใช้งาน A/D

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	$\overline{MCLR}$, WDT
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.
Note 1: These registers/bits are not available on the 28-pin devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารอนุกรม USART

การสื่อสารอนุกรมUSART (Universal Synchronous Asynchronous Receiver Transmitter) เป็นโมดูลการสื่อสารที่มีอยู่ภายในตัวไมโครคอนโทรลเลอร์ มีโหมดการทำงานอยู่ 2 โหมด คือ โหมดการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous Mode) และโหมดการสื่อสารอนุกรมแบบอะซิงโครนัส (Asynchronous Mode) ซึ่งการใช้งานจะใช้งานได้เพียงโหมดใดโหมดหนึ่งเท่านั้น ไม่สามารถใช้งานพร้อมกันได้ 2 โหมดได้ โดยสามารถนำไปใช้งานติดต่อกับอุปกรณ์ต่างๆ เช่น ติดต่อกับไมโครคอนโทรลเลอร์ด้วยกันหรือเครื่อง PC และอุปกรณ์ที่มีรูปแบบการสื่อสารเหมือนกัน โดยสามารถแบ่งการทำงานของโมดูลการสื่อสาร USART ออกเป็นโหมดต่างๆดังนี้

- โหมดการสื่อสาร Asynchronous แบบ Full Duplex
- โหมดการสื่อสาร Synchronous- Master แบบ Half Duplex
- โหมดการสื่อสาร Synchronous- Slave แบบ Half Duplex

2.3.1 รีจิสเตอร์ควบคุมการรับส่งข้อมูล

ในโมดูลการรับส่งข้อมูลของ USART จะมีรีจิสเตอร์ที่ควบคุมการรับส่งข้อมูลหลักๆอยู่ 2 ตัวดังนี้ คือ

TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit7							bit0

Bit 7 CSRC: เป็นบิตที่ใช้เลือกแหล่งสัญญาณนาฬิกาในโหมดการสื่อสารแบบซิงโครนัส (Synchronous)

- โหมด อะซิงโครนัส (Asynchronous Mode)

ไม่มีผล(Don't care)

- โหมด ซิงโครนัส (Synchronous Mode)

1 = เป็น Master Mode สัญญาณนาฬิกามาจากภายในCPU (จากรีจิสเตอร์ BRG)

0 = เป็น Slave Mode สัญญาณนาฬิกาจะมาจากภายนอก

Bit 6 TX9: เป็นบิตที่ใช้กำหนดรูปแบบการส่งข้อมูล

1 = กำหนดการส่งข้อมูลแบบ 9 บิต

0 = กำหนดการส่งข้อมูลแบบ 8 บิต

Bit 5 TXEN: เป็นบิตควบคุมการส่งข้อมูล

1 = Enable การส่งข้อมูล

0 = Disable การส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 4 SYNC: บิตเลือกการทำงานของ USART

1 = กำหนดการทำงานเป็นโหมด ซิงโครนัส (Synchronous Mode)

0 = กำหนดการทำงานเป็นโหมด อะซิงโครนัส (Asynchronous Mode)

Bit 3 Unimplemented: ไม่ใช้งานอ่านค่าได้เป็น "0"

Bit 2 BRGH: บิตเลือกโหมดความเร็วของ Baud Rate

- โหมด อะซิงโครนัส (Asynchronous Mode)

1 = โหมดความเร็วสูง (High Speed)

0 = โหมดความเร็วต่ำ (Low Speed)

- โหมด ซิงโครนัส (Synchronous Mode)

ไม่มีผล(Don't Care)

Bit 1 TRMT: บิตแสดงสถานะของ Shift Registerที่ใช้ส่งข้อมูล (TSR)

1 = รีจีสเตอร์ TSR ว่างไม่มีข้อมูล (Empty)

0 = รีจีสเตอร์ TSR มีข้อมูลอยู่ (Full)

Bit 0 TX9D: เป็นบิตที่ 9 สำหรับใช้ในการส่งข้อมูลแบบ 9 บิต

RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS: 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit0

Bit 7 SPEN: บิต Enable พอร์ตการสื่อสารอนุกรม

1 = Enable พอร์ตการสื่อสารอนุกรม (ขา RC7/RX/DT และ RC6/TX/CX จะถูกกำหนดให้ทำหน้าที่เป็นขาสัญญาณที่ใช้ในการสื่อสารอนุกรม)

0 = Disable พอร์ตการสื่อสารอนุกรม

Bit 6 RX9: เป็นบิตที่ใช้กำหนดการรับข้อมูล

1 = กำหนดการรับข้อมูลแบบ 9 บิต

0 = กำหนดการรับข้อมูลแบบ 8 บิต

Bit 5 SREN: บิตเลือกการรับข้อมูลแบบครั้งเดียว (Single receiver) ของการสื่อสาร โหมดซิงโครนัส

- โหมด อะซิงโครนัส (Asynchronous Mode)

ไม่มีผล (Don't care)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมด ซิงโครนัส- มาสเตอร์ (Synchronous Mode - master)

  1 = Enable การรับข้อมูลแบบครั้งเดียว

  0 = Disable การรับข้อมูลแบบครั้งเดียว

- โหมด ซิงโครนัส-สเลฟ (Synchronous Mode-slave)

  ไม่มีผล(Don't care)

Bit 4 CREN: บิตเลือกการรับข้อมูลแบบต่อเนื่อง (Continuous Receiver Enable bit)

- โหมด อะซิงโครนัส (Asynchronous Mode)

  1 = Enable การรับข้อมูลแบบต่อเนื่อง

  0 = Disable การรับข้อมูลแบบต่อเนื่อง

- โหมด ซิงโครนัส (Synchronous Mode)

  1 = Enable การรับข้อมูลแบบต่อเนื่อง

  0 = Disable การรับข้อมูลแบบต่อเนื่อง

Bit 3 ADDEN: บิตเลือกการตรวจจับแอดเดรสในโหมด อะซิงโครนัส แบบ 9 บิต

- Enable การตรวจจับแอดเดรส, Enable การอินเตอร์รัพท์ และมีการโหลดข้อมูลไปยังบัฟเฟอร์เมื่อบิตRSR<8>เซต
- Disable การตรวจจับแอดเดรสจะรับข้อมูลเข้ามาทั้งหมดและในบิตที่ 9 สามารถใช้เป็น Parity bit ได้

Bit 2 FERR: บิตแสดงสถานะความผิดพลาดของเฟรมข้อมูล (Framing error bit)

1 = เฟรมข้อมูลเกิดผิดพลาด (Framing error)

0 = ไม่เกิดข้อผิดพลาด

Bit 1 OERR: บิตแสดงสถานการณ์ชนกันของข้อมูลของข้อมูลในบัฟเฟอร์

1 = เกิดการชนกันของข้อมูล (สามารถเคลียร์ได้ที่บิตCREN)

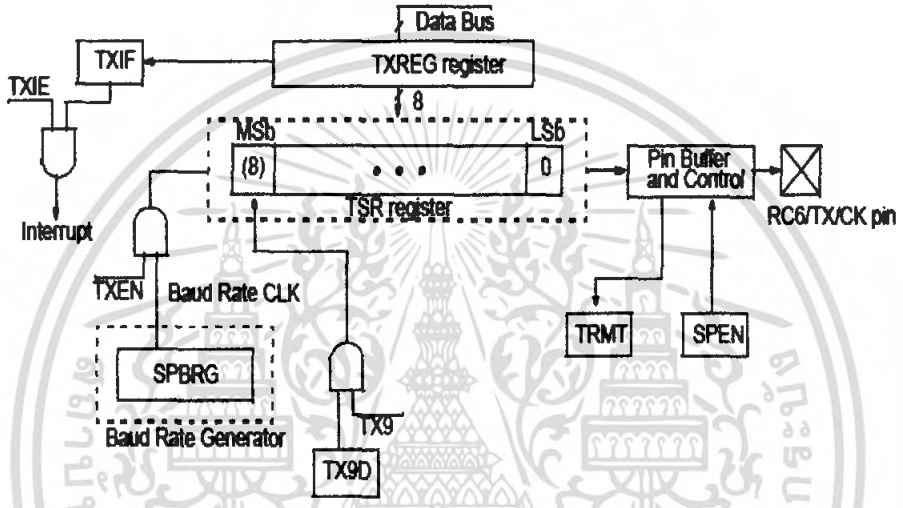
0 = ไม่มีการชนกันของข้อมูล

Bit 0 RX9D: เป็นบิตที่ 9 สำหรับข้อมูลในโหมด 9 บิต

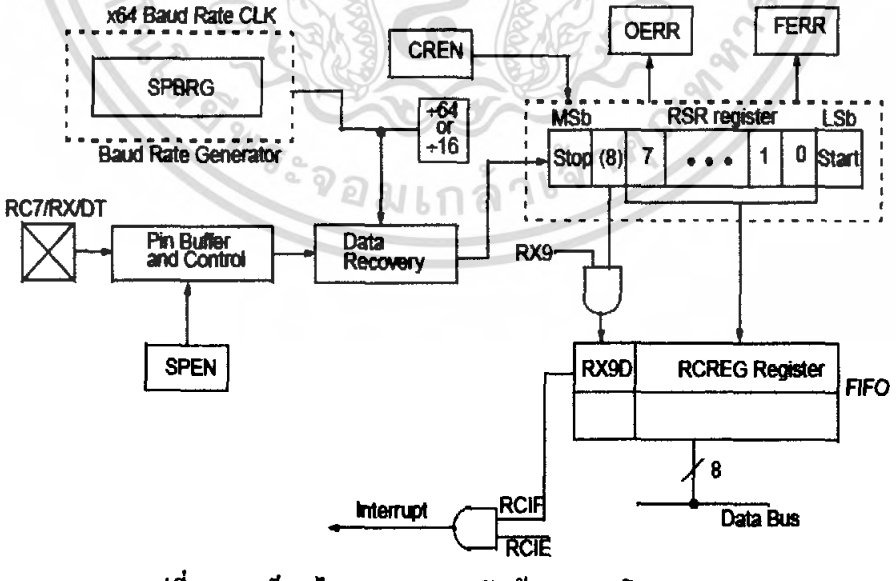
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารในโหมด อะซิงโครนัส (Asynchronous Mode)

การสื่อสารในโหมดนี้จะเป็นลักษณะของการสื่อสารแบบ NRZ (non-return-to-zero) ในการส่งข้อมูลใน 1 เฟรมจะประกอบไปด้วย บิตstart จำนวน 1 บิตและบิตข้อมูล 8 บิตหรือ 9 บิต ปิดด้วยบิตstop อีกหนึ่งบิต ซึ่งรวมแล้วหากเป็นข้อมูลแบบ 8 บิต การส่งข้อมูลใน 1 เฟรม จะมีข้อมูล 10 บิต ส่วนการส่งข้อมูลแบบ 9 บิตเฟรมจะมีขนาด 11 บิต โดยในการส่งข้อมูลแบบ 9 บิตนี้ส่วนใหญ่ จะใช้บิตที่ 9 เป็นบิต พาริตี้(Parity bit) และเมื่ออยู่ในโหมดประหยัดพลังงาน(Sleep Mode) การสื่อสารนี้จะหยุดลง คือไม่สามารถทำงานในโหมดSleep ได้ ส่วนการกำหนดการทำงานเป็นโหมดอะซิงโครนัส นั้นทำได้โดยการเคลียร์บิต SYNC



รูปที่ 2.12 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART



รูปที่ 2.13 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารแบบซิงโครนัส (Synchronous Mode)

การสื่อสารในโหมดซิงโครนัสนี้เป็นการสื่อสารแบบ ฮาล์ฟดูเพล็กซ์ (Half-duplex) ทำให้ไม่สามารถทำการส่งหรือรับข้อมูลในเวลาเดียวกันได้ ซึ่งก็คือเมื่อตัวใดตัวหนึ่งทำการส่งข้อมูล อีกตัวหนึ่งจะต้องทำการรับข้อมูล โดยในการสื่อสารแบบซิงโครนัสนี้จะประกอบด้วย ขาสัญญาณข้อมูล(DT:data) และขาสัญญาณนาฬิกา(CK:clock) การรับข้อมูลจะสัมพันธ์กับสัญญาณนาฬิกาทั้งตัวรับและตัวส่ง ทำให้การสื่อสารแบบนี้มีประสิทธิภาพมากกว่าการสื่อสารแบบอะซิงโครนัส แต่ในเรื่องของความเร็วในการส่งข้อมูลแบบนี้จะมีความเร็วต่ำกว่าแบบอะซิงโครนัส ซึ่งในการสื่อสารแบบซิงโครนัสนี้ จะแบ่งการทำงานออกเป็น 2 โหมด คือ

3. โหมดมาสเตอร์(Master Mode)

การสื่อสารในโหมดมาสเตอร์นี้ อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์จะเป็นที่ตัวกำหนดสัญญาณนาฬิกาในการสื่อสารข้อมูลเข้าสู่การทำงานในโหมดนี้จะต้องเซตบิต SYNC (TXSTA<4>) และทำการ Enable การสื่อสารข้อมูลโดยการเซตบิต SPEN เป็น “1” จะทำให้ขาสัญญาณ RC6/TX/CX และ RC7/RX/CK ทำหน้าที่เป็นขาสัญญาณนาฬิกา (CK:clock) และขาสัญญาณข้อมูล (DT:data) จากนั้นกำหนดการทำงานให้อยู่ในโหมดมาสเตอร์โดยการเซตบิตCSRC(TXSTA<7>) อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์นี้จะเป็นตัวหลักที่ทำหน้าที่ควบคุมการรับส่งข้อมูลและการกำเนิดสัญญาณนาฬิกา(Clock) ที่ใช้ในระบบ

4. โหมดสเลฟ(Slave Mode)

การสื่อสารแบบซิงโครนัสในโหมดสเลฟนี้จะต่างจากการสื่อสารในโหมดมาสเตอร์ในเรื่องของแหล่งกำเนิดสัญญาณนาฬิกาในการสื่อสารข้อมูล โดยในโหมดสเลฟนี้จะอาศัยสัญญาณนาฬิกาในการรับส่งข้อมูลจากภายนอกที่ขาสัญญาณ RC6/TX/CK ซึ่งในโหมดมาสเตอร์จะใช้สัญญาณนาฬิกาภายใน โดยสัญญาณนาฬิกาที่ส่งให้ขา RC6/TX/CK มาจากอุปกรณ์ที่เป็นมาสเตอร์นั้นเอง การรับส่งข้อมูลในโหมดสเลฟนี้สามารถเกิดขึ้นได้แม้ว่า CPU จะอยู่ในโหมดประหยัดพลังงาน (Sleep Mode) การกำหนดการทำงานในโหมดสเลฟทำได้โดยการเคลียร์บิต CSRC (TXSTA<7>) ซึ่งจะเป็นการเลือกรับสัญญาณนาฬิกาจากภายนอกนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Pulse Width Modulation (PWM)

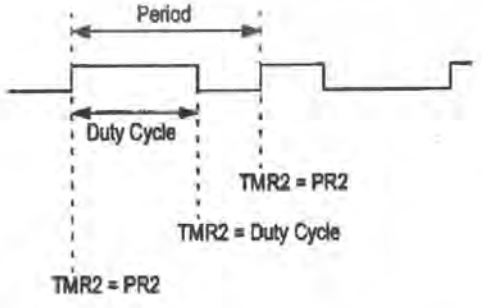
การมอดดูเลชั่นทางความกว้างของพัลส์(PWM) Pulse Width Modulation เป็นการปรับเปลี่ยนที่สัดส่วนและความกว้างของสัญญาณพัลส์ โดยความถี่ของพัลส์นั้นไม่เปลี่ยนแปลงหรือการเปลี่ยนแปลงค่าดิวตี้ไซเคิล (duty cycle) ซึ่งค่าดิวตี้ไซเคิล คือ ช่วงความกว้างของพัลส์มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของทั้งหมด เช่น ถ้าค่าดิวตี้ไซเคิลเท่ากับ 50 % หมายถึง ใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่อยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่งดังรูปที่1 และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามากก็แสดงว่าความกว้างของพัลส์ที่มีสถานะลอจิกสูงจะมีค่ามากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% แสดงว่าไม่มีสถานะลอจิกต่ำเลยซึ่งค่าดิวตี้ไซเคิลสามารถหาได้จากความสัมพันธ์ดังนี้

ค่าดิวตี้ไซเคิล = (ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ) X 100%

รูปที่ 2.14 แสดงความกว้างของพัลส์ขนาดต่างๆและค่าดิวตี้ไซเคิลของช่วงพัลส์ที่มีค่าคงที่

การสร้างสัญญาณ PWM

สัญญาณ PWM ประกอบด้วย 2 ส่วนคือ คาบเวลา (Period) และดิวตี้ไซเคิล (Duty Cycle) โดยค่าของคาบเวลา (Period) จะถูกกำหนดโดยค่ารีจิสเตอร์ PRx ส่วนค่าของดิวตี้ไซเคิลจะมีความละเอียด 10 บิต โดย 8 บิตบน จะกำหนดที่รีจิสเตอร์ CCPR1L ส่วนอีก 2 บิตล่างจะอยู่ในรีจิสเตอร์ CCP1CON ในบิตที่ 5 และ 4 (CCP1CON<5:4>)



รูปที่ 2.15 สัญญาณเอาต์พุตของ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.9 กระบวนการสร้าง PWM จะมีกระบวนการและขั้นตอนต่างๆ ดังนี้

1. เมื่อเริ่มทำงานค่าของ TMR2 จะมีค่าเพิ่มขึ้นเรื่อยๆในระหว่างที่เพิ่มขึ้นนี้ก็จะมีการเปรียบค่ากับรีจิสเตอร์ PR2 ด้วย ซึ่งเมื่อค่าของ TMR2 เพิ่มขึ้นจนเท่ากับ PR2 แล้วจะเกิดการทำงาน 3 ส่วน

- รีจิสเตอร์ TMR2 จะถูกเคลียร์เป็นศูนย์
- ที่ขาสัญญาณเอาท์พุต CCPx จะเซตเป็น '1' (ถ้าค่า Duty Cycle= 0% จะไม่ถูกเซต)
- ค่า Duty Cycle ที่อยู่ในรีจิสเตอร์ CCPR1L:CCP1CON<5:4> ขนาด 10 บิต จะถ่ายโอนไปยังรีจิสเตอร์ CCPR1H ร่วมกับอีก 2 บิต ในหน่วยความจำพิเศษ

2. สภาวะที่ขาสัญญาณเอาท์พุต CCPx ยังคงสภาวะ '1' อยู่ในขณะที่ค่าในรีจิสเตอร์TMR2 ร่วมกับอีก 2 บิต ใน Prescaler รวมเป็น 10 บิต จะเริ่มเพิ่มขึ้นจากศูนย์อีกครั้งโดยค่าที่เพิ่มขึ้นจะถูกนำไปเปรียบเทียบกับรีจิสเตอร์ CCPR1H ร่วมกับหน่วยความจำพิเศษอีก 2 บิตเป็น 10 บิตเช่นกัน ซึ่งเมื่อค่าทั้งสองนี้ เพิ่มขึ้นจนเท่ากันจะทำให้เกิดการทำงานต่างๆ ดังนี้

- รีจิสเตอร์ TMR2 จะถูกเคลียร์เป็นศูนย์
- ที่ขาสัญญาณเอาท์พุต CCPx จะถูกเคลียร์เป็น '0'

3. ค่าของ TMR2 จะเพิ่มขึ้นจากศูนย์อีกครั้งและจะถูกนำไปเปรียบเทียบกับรีจิสเตอร์ PR2 เมื่อเท่ากันก็จะทำให้ขาสัญญาณ CCPx เซตเป็น '1' เหมือนกับการทำงานใน Step ที่ 1 การทำงานจะวนรอบในลักษณะนี้ไปเรื่อยๆ จนกว่าจะมีการปิดการทำงานของโหมดนี้และในขณะที่โมดูล PWM ยังทำงานอยู่นั้น เราสามารถทำการเปลี่ยนแปลงค่า Duty Cycle ได้ตลอดเวลาโดยการกำหนดค่าใหม่ให้กับรีจิสเตอร์ CCPR1L:CCP1CON<5:4> ส่วนค่า Period หรือค่าความถี่จะคงที่ตลอดเวลานอกจากจะมีการปิดการทำงานของโหมดนี้แล้วกำหนดค่า Period ใหม่ให้กับรีจิสเตอร์ PR2

การคำนวณหาค่า Period ที่จะกำหนดให้กับรีจิสเตอร์ PR2 นั้นสามารถหาได้จากสมการดังนี้

$$\text{PWM period} = [(PR2) + 1] * 4 * Tosc * (\text{TMR2 prescaler value})$$

หรือ

$$PR2 = [(\text{PWM period}) / (4 * Tosc * (\text{TMR2 prescaler})] - 1$$

โดยที่ **PR2** คือ รีจิสเตอร์ที่ใช้กำหนดคาบเวลาของสัญญาณ PWM มีค่าอยู่ระหว่าง 00h-FFh

PWM period คือ คาบเวลาของสัญญาณ PWM ที่ต้องการมีค่าเท่ากับ 1/Fpwm

Fpwm คือ ความถี่ของสัญญาณ PWM ที่ต้องการ

Tosc คือ คาบเวลาของสัญญาณนาฬิกาหลักมีค่าเท่ากับ 1/Fosc

Fosc คือ ความถี่คริสตอลออสซิลเลเตอร์ที่นำมาต่อให้กับ CPU

TMR2 prescaler คือ ค่าอัตราส่วน Prescaler ของไทเมอร์ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ไทเมอร์ (TIMER)

ไทเมอร์/เคาน์เตอร์ หลักใน dsPIC มี 5 ตัว คือ ไทเมอร์ 1 ถึงไทเมอร์ 5 โดยแต่ละตัวมีขนาด 16 บิต สำหรับไทเมอร์ 2 และ3 กับไทเมอร์4 และ 5 สามารถทำงานร่วมกันเป็นไทเมอร์ ขนาด 32 บิต เมื่อไทเมอร์แต่ละตัวทำงานแยกกันสามารถกำหนดการทำงานได้อีก 3 แบบ ตามลักษณะของฐานเวลา คือ ฐานเวลาแบบ A, B และ C

คุณสมบัติของไทเมอร์

ใน dsPIC30F2020 มีไทเมอร์/เคาน์เตอร์ ขนาด 16 บิต ให้ใช้งานรวม 3 ตัว คือ ไทเมอร์ 1 ไทเมอร์ 2 และไทเมอร์ 3

คุณสมบัติของไทเมอร์ 1

- รีจิสเตอร์ตัวนับความละเอียด 16 บิต
- ทำงานได้ทั้งแบบซิงโครนัสและอะซิงโครนัสเคาน์เตอร์
- มีปรีสเกลเลอร์สำหรับหารความถี่การนับ
- สามารถกำหนดการอินเทอร์รัปต์จากการนับหรือจากการตรวจพบสัญญาณขอบขาลงที่ขาอินพุตของไทเมอร์

คุณสมบัติของไทเมอร์ 2 และ 3

- ไทเมอร์ 2 และ3 เมื่อทำงานแยกอิสระต่อกัน มีคุณสมบัติคล้ายกับไทเมอร์ 1
- เมื่อนำไทเมอร์ 2 และ 3 มาทำงานร่วมกัน รีจิสเตอร์ตัวนับมีความละเอียดเพิ่มเป็น 32 บิต
- ทำงานร่วมกับขาอินพุทประจำตัวไทเมอร์ได้(ขา TxCK)
- มีปรีสเกลเลอร์สำหรับหารความถี่การนับ
- สามารถกำหนดการอินเทอร์รัปต์จากการนับหรือจากการตรวจพบสัญญาณขอบขาลงที่ขาอินพุตของไทเมอร์
- สามารถกำเนิดสัญญาณกระตุ้นการทำงานไปยังโมดูล ADC ได้

การทำงานของไทเมอร์ตามลักษณะของฐานเวลา

ไทเมอร์ที่ทำงานด้วยฐานเวลามี 3 แบบ ดังนี้

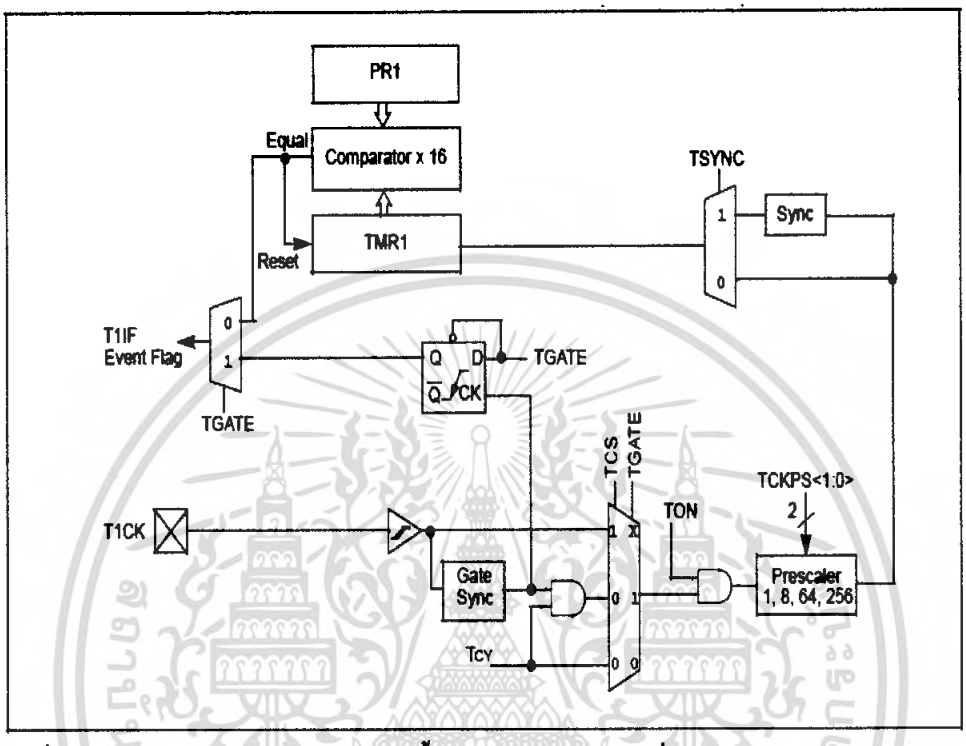
1. **ฐานเวลาแบบ A**

การทำงานในแบบมาตรฐานของ dsPIC โดยไทเมอร์ที่ทำงานในแบบนี้คือ ไทเมอร์ 1 ซึ่งมีคุณสมบัติพิเศษเพิ่มเติมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กำหนดให้ทำงานกับแหล่งกำเนิดสัญญาณนาฬิกากำลังงานต่ำ 32 kHz จากภายนอกได้
- กำหนดทำงานในโหมดแบบอะซิงโครนัสโดยรับสัญญาณนาฬิกาจากภายนอกได้

ดังนั้น ไทเมอร์ 1 จึงเหมาะอย่างยิ่งในการนำมาใช้สร้างฐานเวลานาฬิกาจริง (RTC)



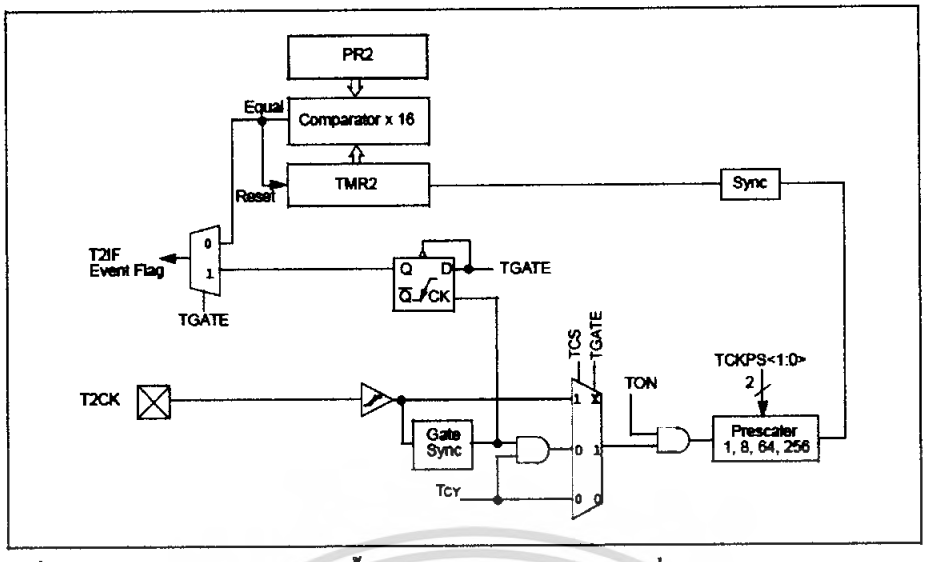
รูปที่ 2.16 แสดงไดอะแกรมการทำงานพื้นฐานของไทเมอร์ 1 ที่ทำงานด้วยฐานเวลาแบบ A

2. ฐานเวลาแบบ B

ไทเมอร์ที่สามารถทำงานในลักษณะนี้ได้มี 2 ตัว คือ ไทเมอร์ 2 และ ไทเมอร์ 4 (โดยเฉพาะไทเมอร์ 4 ไม่มีใน dsPIC30F2010) คุณสมบัติที่เพิ่มเติมเข้ามาสำหรับฐานเวลาแบบ B คือ

- สามารถต่อกับไทเมอร์ฐานเวลาแบบ C เพื่อใช้งานเป็นไทเมอร์ขนาด 32 บิต ได้โดยการกำหนดสถานะที่บิต T32 ในรีจีสเตอร์ TxCON เมื่อทำงานด้วยฐานเวลาแบบ B ให้เป็นลอจิก "1"
- มีวงจรกำหนดจังหวะของสัญญาณนาฬิกาให้สัมพันธ์กัน(ซิงโครไนซ์) หลังจากผ่านการปรับอัตราลดทอนความถี่ด้วยปรีสเกลเลอร์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

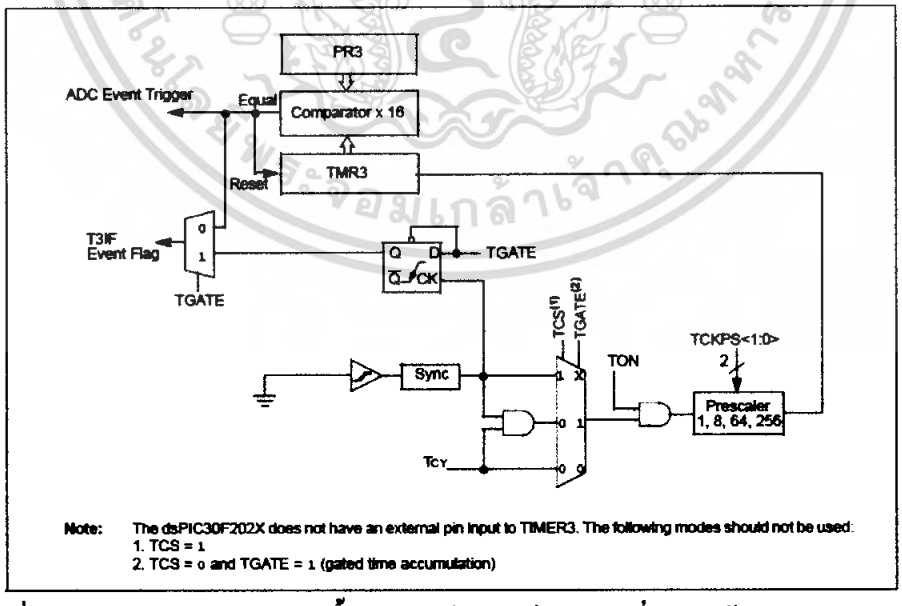


รูปที่ 2.17 ไดอะแกรมการทำงานพื้นฐานของไทเมอร์ 2 และ4 ที่ทำงานด้วยฐานเวลาแบบ B

3. ฐานเวลาแบบ C

ไทเมอร์ที่สามารถทำงานในลักษณะนี้ได้มี 2 ตัว คือ ไทเมอร์ 3 ไทเมอร์ 5 (เฉพาะไทเมอร์ 5 ไม่มีใน dsPIC30F2010) คุณสมบัติที่เพิ่มเติมเข้ามาสำหรับฐานเวลาแบบ C คือ

- สามารถต่อกับไทเมอร์ฐานเวลาแบบ B เพื่อใช้งานเป็นไทเมอร์ 32 บิตได้ โดยการกำหนดสถานะที่บิต T32 ในรีจิสเตอร์ TxCON เมื่อทำงานด้วยฐานเวลาแบบ B ให้เป็นลอจิก “1”
- สามารถสร้างสัญญาณกระตุ้นการทำงานของโมดูลADCได้ ซึ่งทำให้ไทเมอร์และโมดูลแปลงADCทำงานสัมพันธ์กัน



รูปที่ 2.18 ไดอะแกรมการทำงานพื้นฐานของไทเมอร์ 3 และ5 ที่ทำงานด้วยฐานเวลาแบบ C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดให้ทำงานเป็นไทเมอร์ ขนาด 32 บิต

ไทเมอร์ขนาด 32 บิต เกิดจากการนำไทเมอร์ 16 บิต ที่ทำงานด้วยฐานเวลาแบบ B และ C มาทำงานร่วมกัน โดยไทเมอร์ฐานเวลาแบบ C (ไทเมอร์ 3 และ 5) ใช้เก็บข้อมูล 16 บิตบนหรือเวิร์ดสูง (MS word) ส่วนไทเมอร์ฐานเวลาแบบ B (ไทเมอร์ 2 และ 4) ใช้เก็บข้อมูล 16 บิตล่างหรือเวิร์ดต่ำ (LS word)

เมื่อทำงานเป็นไทเมอร์ขนาด 32 บิต การควบคุมการทำงานของไทเมอร์จะกำหนดผ่านไทเมอร์ฐานเวลา B นั่นคือ กำหนดค่าผ่านรีจิสเตอร์ T2CON หรือ T4CON แล้วแต่กรณี สำหรับ dsPIC 30F2020 จะใช้ T2CON ส่วนในกรณีอินเตอร์รัปต์จะไปกำหนดค่าที่ไทเมอร์ฐานเวลา C ที่รีจิสเตอร์ T3CON หรือ T5CON

ยกตัวอย่างการกำหนดให้ไทเมอร์ใน dsPIC30F2020 ทำงานแบบเป็น 32 บิต คือใช้ไทเมอร์ 2 และ 3 โดยการควบคุมปกติจะกระทำที่รีจิสเตอร์ T2CON ส่วนการจัดการอินเตอร์รัปต์กระทำผ่านรีจิสเตอร์ T3CON มีขั้นตอนโดยสรุปดังนี้

1. เซตบิต TON (บิต 15 ในรีจิสเตอร์ T2CON) เพื่อเปิดการทำงานของไทเมอร์
2. เซตบิต T32 (บิต 3 ในรีจิสเตอร์ T2CON) เพื่อเลือกการทำงานแบบ 32 บิต
3. กำหนดอัตราปรีสเกลเลอร์(ถ้าต้องการ) ที่บิต TCKPS1 และ TCKPS0 (บิต 5 และ4 ของรีจิสเตอร์ T2CON)
4. รีจิสเตอร์ PR3 และ PR2 ถูกใช้เก็บค่าคาบเวลาขนาด 32 บิต สำหรับเปรียบเทียบกับค่าในรีจิสเตอร์ไทเมอร์ TMR3 และ TMR2
5. บิต T3IE (บิต 7 ในรีจิสเตอร์ IEC0) ถูกใช้ในการ Enable อินเตอร์รัปต์ของไทเมอร์ 32 บิต
6. บิต T3IF (บิต 7 ในรีจิสเตอร์ IFS0) ถูกใช้แสดงสถานการณ์อินเตอร์รัปต์ของไทเมอร์ 32 บิต
7. กำหนดค่าลงในบิต T3IP2 ถึง P3IP0 (บิต 14 ถึง 12 ของรีจิสเตอร์ IPC1) เพื่อเลือกระดับความสำคัญของการตอบสนองอินเตอร์รัปต์ของไทเมอร์ 32 บิต
8. ไม่สนใจการทำงานของรีจิสเตอร์ T3CON

ส่วนการทำงานในแต่ละโหมดของไทเมอร์ 32 บิต จะเหมือนกับแบบ 16 บิต แต่จำนวนของข้อมูลที่ทำงานด้วยมีค่าเพิ่มขึ้น

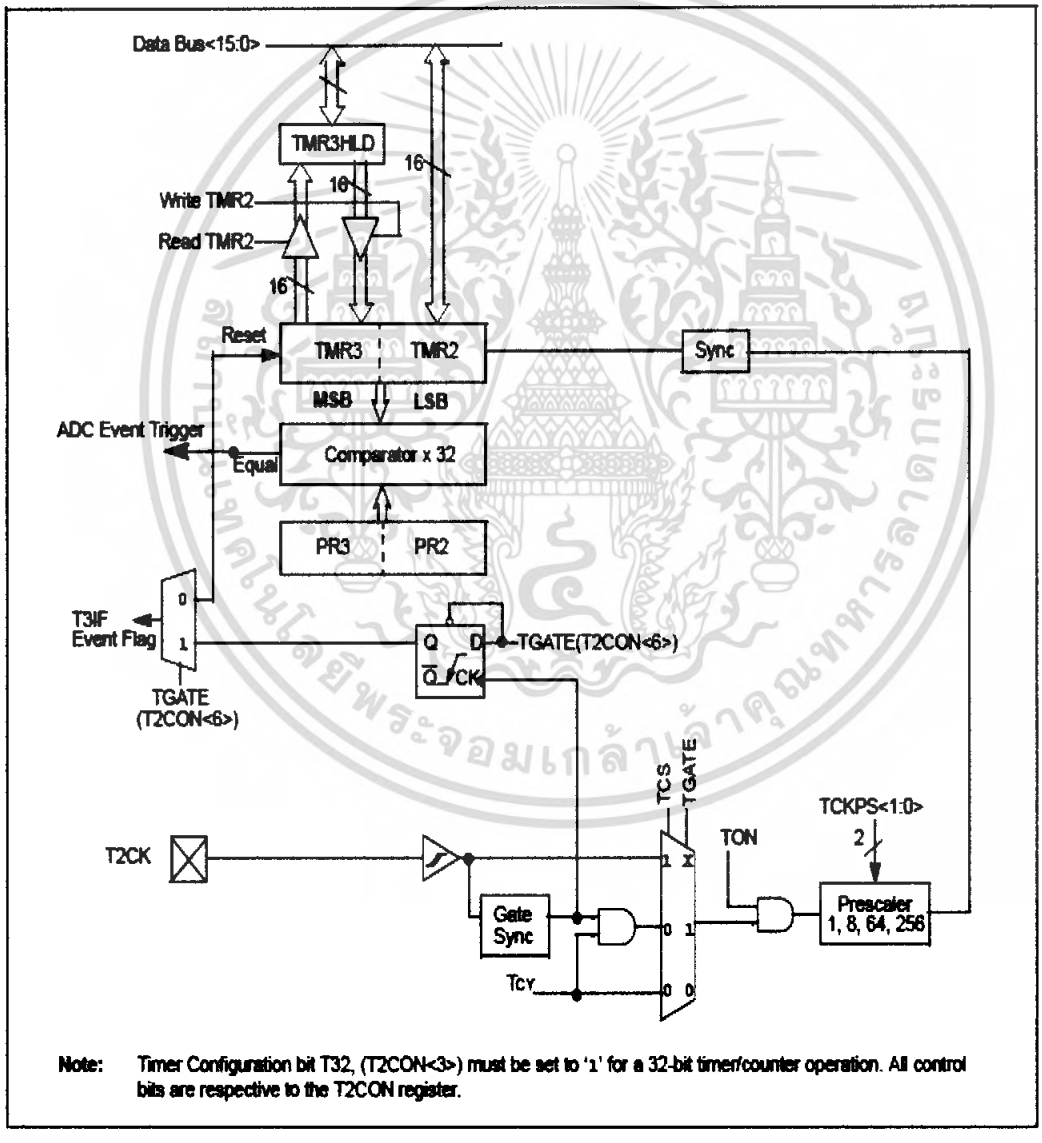
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านและเขียนข้อมูลในไทเมอร์ 32 บิต

การอ่านและเขียนข้อมูลในไทเมอร์ 32 บิต จะต้องกำหนดให้สัมพันธ์กันระหว่างข้อมูลในเวิร์ดสูงและเวิร์ดต่ำร่วมกับรีจิสเตอร์ที่ใช้พักค่าของรีจิสเตอร์ที่ทำงานด้วยฐานเวลาแบบ C ซึ่งใน dsPIC30F2020 คือ ไทเมอร์ 3 จากรูปที่ 6-6 คือรีจิสเตอร์ TMR3HLD

ในการอ่านค่าจะต้องเริ่มอ่านจากเวิร์ดต่ำคือ ค่าของรีจิสเตอร์ TMR2 ในจังหวะเดียวกันนั้น ค่าของรีจิสเตอร์ TMR3 จะถูกโหลดไปยังรีจิสเตอร์ TMR3HLD โดยอัตโนมัติ จึงสามารถอ่านค่าเวิร์ดสูงของไทเมอร์ 32 บิต ได้จากรีจิสเตอร์ TMR3HLD ได้ทันที

ส่วนการเขียนข้อมูลนั้นจะต้องเริ่มต้นการเขียนข้อมูลเวิร์ดสูงไปยังรีจิสเตอร์ TMR3HLD ก่อน จากนั้นจึงเขียนข้อมูลเวิร์ดต่ำไปยังรีจิสเตอร์ TMR2 ข้อมูลจากรีจิสเตอร์ TMR3HLD จะถูกถ่ายทอดไปยังรีจิสเตอร์ TMR3 โดยอัตโนมัติ ทำให้การเขียนข้อมูลของไทเมอร์ 32 บิตเสร็จสิ้นลง



รูปที่ 2.19 ไดอะแกรมของไทเมอร์ 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 แบตเตอรี่

เป็นส่วนที่จำเป็นอีกส่วนหนึ่ง ลักษณะโครงสร้างจะเห็นว่าองค์ประกอบคือ แผ่นอาโนด และแผ่นแคโทดวางสลับกันจุ่มลงในอิเล็กทรอไลต์ที่ทำมาจากสารละลายกรดกำมะถัน แผ่นเพลทจะวางสลับกันเพื่อจะได้มีพื้นที่สัมผัสกับอิเล็กทรอไลต์มากที่สุด ในขณะที่รักษาปริมาตรให้น้อยที่สุดเท่าที่จะเป็นไปได้ การที่มีพื้นที่ผิวสัมผัสระหว่างแผ่นอิเล็กทรอล์มากเท่าไร ปฏิกิริยาทางเคมีเกิดขึ้นมากเท่านั้น นอกจากนี้ค่าความต้านทานภายในเซลล์จะยิ่งน้อยลงด้วย ดังนั้นในการค้นคว้าจึงมีความมุ่งมั่นทางด้านการเพิ่มพื้นที่ผิวสัมผัส วิธีที่นิยมใช้กันคือใช้แผ่นเพลทบางๆคั่นด้วยฉนวนแบบมีรูพรุน

อิเล็กทรอไลต์ที่เป็นอาโนดจะสร้างจากตะกั่วบริสุทธิ์ ขณะที่คาโทดจะสร้างจากส่วนผสมของตะกั่วและตะกั่วเปอร์ออกไซด์ ในขณะที่เซลล์คายประจุให้กระแสไฟฟ้าออกมานั้นอะตอมของตะกั่วจากแผ่นอาโนดแตกตัวเป็นไอออนมีประจุบวกเข้าไปอยู่ในอิเล็กทรอไลต์และทิ้งอิเล็กตรอนให้ไหลเข้าสู่วงจรที่ต่ออยู่ภายนอก

ที่แคโทด ตะกั่วเปอร์ออกไซด์จะแตกตัวเป็นไอออนของตะกั่วซึ่งมีประจุบวกสูง และเป็นไอออนที่มีประจุลบสูง ไอออนของตะกั่วที่มีประจุจะดึงเอาอิเล็กตรอนจากวงจรที่ต่ออยู่ภายนอกเพื่อรวมตัวกลายเป็นอิออนตะกั่วที่มีประจุบวก ซึ่งเป็นชนิดเดียวกับที่อาโนดทำให้เกิดกระแสไหลจากคาโทดผ่านไปยังวงจรภายนอก

ไอออนของตะกั่วจากแผ่นอิเล็คโทดทั้งสองจะทำปฏิกิริยากับกรดกำมะถันซึ่งเป็นอิเล็คทรอไลต์กลายเป็นตะกั่วซัลเฟต(จะเป็นตะกอนสีขาวอยู่ที่อิเล็คโทดทั้งสอง) และก๊าซไฮโดรเจน (ซึ่งจะรวมกับอิออนของออกซิเจนจากแคโทดกลายเป็นน้ำ) เราสามารถเขียนสูตรสำหรับปฏิกิริยาเคมีที่เกิดขึ้นได้ดังนี้

$$PbO_2 + Pb + 2H_2SO_4 \longrightarrow 2PbSO_4 + 2H_2O$$

สูตรทางเคมีแสดงให้เห็นว่าสารละลายอิเล็กทรอไลต์จะเจือจางโดยโมเลกุลของน้ำที่เกิดขึ้นขณะเดียวกันกับที่เซลล์คายประจุ ทำให้เราสามารถใช้เป็นวิธีการหาสถานการณ์ประจุและการคายประจุของเซลล์ได้ โดยการวัดความถ่วงจำเพาะของสารละลายอิเล็กทรอไลต์ ซึ่งเป็นตัวบ่งบอกว่าเซลล์ใกล้จะถึงสถานะคายประจุหมดหรือยัง เพื่อประจุไฟหลับเข้าไปใหม่ โดยค่าความถ่วงจำเพาะที่ประจุเต็มมีค่าประมาณ 1.25 และค่าความถ่วงจำเพาะของเซลล์ที่มีการคายประจุหมดจะมีค่าเท่ากับ 1.2 (ใช้ไฮโดรมิเตอร์ในการวัด) แรงดันของเซลล์ปกติจะมีค่าเท่ากับ 2 โวลต์

การประจุเซลล์แบบตะกั่ว-กรดนั้นสามารถทำได้โดยง่าย โดยการป้อนกระแสกลับทางเข้าแบตเตอรี่เพื่อบังคับให้ปฏิกิริยาเคมีเกิดขึ้นเกิดจากทางขวามือไปทางซ้ายมือ ซึ่งจะเปลี่ยนตะกั่วซัลเฟตให้กลับเป็นตะกั่วและกรดกำมะถันดังเดิม วิธีการง่ายที่สุดในการป้อนกระแสกลับทางโดยต่อขั้วแคโทด (ขั้วบวก) และขั้วอาโนด (ขั้วลบ) เข้ากับขั้วบวกและขั้วลบของแหล่งจ่ายไฟภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันของแหล่งจ่ายไฟภายนอกนี้จะอยู่ในช่วง 1.1 ถึง 1.25 เท่าของแรงดันเซลล์โดยปกติ ดังนั้น เซลล์แบบนี้เซลล์เดียวซึ่งมีแรงดันปกติ 2 โวลท์ สามารถประจุเข้าด้วยแรงดันระหว่าง 2.2-2.5 โวลต์ ตัวอย่างที่พบง่ายๆคือ แบตเตอรี่คือ แบตเตอรี่รถยนต์(ซึ่งแรงดันปกติเท่ากับ12 โวลต์) จะถูกประจุโดยเครื่องกำเนิดไฟฟ้าหรือไดนาโม

ดังนั้น กระแสที่ป้อนเข้าไปเมื่อประจุใหม่นั้นขึ้นอยู่กับแรงดันที่ป้อนให้เข้าไป นอกจากนั้นยังขึ้นอยู่กับสถานะของเซลล์ว่าคายประจุหมดหรือไม่ ถ้าแรงดันที่ป้อนเข้าไปมีแรงดันสูงและคายประจุหมดเต็มที่แล้ว จะทำให้กระแสที่ไหลเข้าประจุเซลล์จะมีค่ามากตามไปด้วย แต่ถ้าเซลล์ถูกใช้งานขณะประจุไฟฟ้าตลอดเวลาเช่นในแบตเตอรี่รถยนต์แล้วกระแสที่ใช้ในการประจุก็จะมีค่าต่ำ ซึ่งเหตุผลที่ว่าทำไมแบตเตอรี่ขนาด12 โวลต์ จึงใช้แรงดันในการประจุเพียง 14 โวลต์ ซึ่งประมาณ 1.17 เท่าของแรงดันปกติของแบตเตอรี่

2.7 GPS

2.7.1 ความเป็นมาของระบบ GPS

ดาวเทียมด้าน GPS ถูกปล่อยสู่อวกาศครั้งแรกในปี 1983 ซึ่งใช้เป็นเพียงระบบนำร่องในขั้นต้นให้กับเครื่องบิน เมื่อกลุ่มดาวเทียม GPS มีการขยายตัวมากขึ้น พื้นการครอบคลุมก็มากขึ้นและมีการพัฒนาอย่างต่อเนื่องในการตรวจสอบแบบ 2 มิติ (ละติจูด, ลองติจูดและระดับความสูง) ซึ่งระบบสมบูรณ์จริงๆจะต้องใช้กลุ่มดาวเทียมปฏิบัติงานจริงถึง 21 ดวง และสำหรับสำรองบนวงโคจรอีก 3 ดวง

2.7.2 ภาพรวมของระบบ GPS

ดังที่กล่าวแล้วว่า ระบบ GPS นี้อาศัยกลุ่มดาวเทียม 21 ดวงและดาวเทียมสำรองอีก 3 ดวง ในระนาบทั้ง 6 ระนาบ ซึ่งสามารถใช้ในการนำร่องทั้งทางภาคพื้นดิน ในทะเลและในอากาศระบบ GPS จะทำการปรับค่าตัวเลขบอกตำแหน่งของวัตถุใน 3 มิติ อย่างสม่ำเสมอ (ละติจูด, ลองติจูดและระดับความสูง)

ระบบ GPS ประกอบด้วย 3 ส่วน คือ

1. Space Segment

ประกอบไปด้วยดาวเทียมที่ห่างจากโลกประมาณ 11,000 ไมล์ สำหรับประสานงานกับดาวเทียม 4 ดวง หรือมากกว่าที่สามารถถูกเห็นได้อย่างชัดเจนในบริเวณใดๆ บนพื้นผิวโลกขึ้นไป ดาวเทียมจะต้องลอยตัวอยู่สูงเพียงพอที่สัญญาณจากระบบภาคพื้นดินไม่สามารถรบกวนได้ เช่น ระบบบนภาคพื้นดิน LORAN-C และ OMEGA

2. Control Segment

ประกอบไปด้วย สถานีควนคุมหลัก(Master control Monitor), 5 สถานีแจ้งผล (Moni

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

tor - station) และ 3 เสาอากาศภาคพื้นดิน (Ground Antennas) ซึ่งจะถูกจัดวางอยู่ทั่วโลก สถานีแจ้งผลจะใช้เครื่องรับ GPS ในการติดตามดาวเทียมทั้งหมดที่สามารถมองเห็นได้และรวบรวมข้อมูลของระยะทางจากการถ่ายทอดของดาวเทียม โดยสถานีแจ้งผลจะส่งข้อมูลที่รวบรวมจากดาวเทียมไปยังสถานีควบคุมหลัก ซึ่งจะคำนวณวงโคจรของดาวเทียมอย่างแม่นยำ ข้อมูลจะถูกจัดเป็นรูปแบบเข้ากับข้อมูลนำร่อง (Navigation message) ที่ถูกปรับเปลี่ยนไปสำหรับดาวเทียมแต่ละดวง ข้อมูลนี้จะถูกส่งไปยังดาวเทียมแต่ละดวงโดยผ่านเสาอากาศภาคพื้นดิน

3. User Segment

ประกอบไปด้วย เครื่องรับ ส่วนประมวลผลและจานสายอากาศ ซึ่งทำให้ผู้ปฏิบัติงานที่อยู่ทางภาคพื้นดิน ในทะเลหรือในอากาศสามารถที่จะรับการถ่ายทอดของกลุ่มดาวเทียม GPS และคำนวณตำแหน่ง ความเร็วและเวลาได้อย่างแม่นยำ

2.7.3 การเข้ารหัสและความสามารถเลือกหาได้ (Precise code and Selective availability)

ระบบการนำร่อง (Navigation system) ซึ่งใช้เครื่องรับแบบอย่างเดียวที่เป็นแบบพาสซีฟ (Passive radio navigation) จะถูกนำมาใช้ประโยชน์สำหรับการปฏิบัติงานที่เป็นความลับ เช่น เครื่องบินโจรกรรม จรวด เป็นต้น นอกจากนี้เพื่อเป็นการป้องกันการใช้งาน GPS ในทางที่ไม่ถูกต้อง จึงต้องมีการเข้ารหัสป้องกัน ซึ่งมีอยู่ 2 ลักษณะ คือ P-code และ C/A Code

การเข้ารหัสแบบ P-code (Pre-cise) เป็นการเข้ารหัสใช้ได้เฉพาะผู้ที่เป็นรัฐบาลสหรัฐเท่านั้น ส่วนการเข้ารหัสแบบ C/A Code (coarse-acquisition) เป็นการเข้ารหัสที่พอหาได้เพื่อป้องกันการใช้งานสำหรับบุคคลทั่วไป

การเข้ารหัสแบบ P-Code ใช้งานใช้เครื่องรับทางทหารที่มีลักษณะเฉพาะพิเศษสำหรับการรับรวมทั้งมีรหัสผ่าน (Password) ส่วนสัญญาณที่เข้ารหัสแบบ C/A Code ซึ่งพอหาใช้งานได้ทั่วไป ความเที่ยงตรงของการบอกตำแหน่งสำหรับผู้ใช้ทั่วไปค่อนข้างมากเกือบเท่ากับทางทหาร

2.7.4 จีโอเดติค ดาตัม

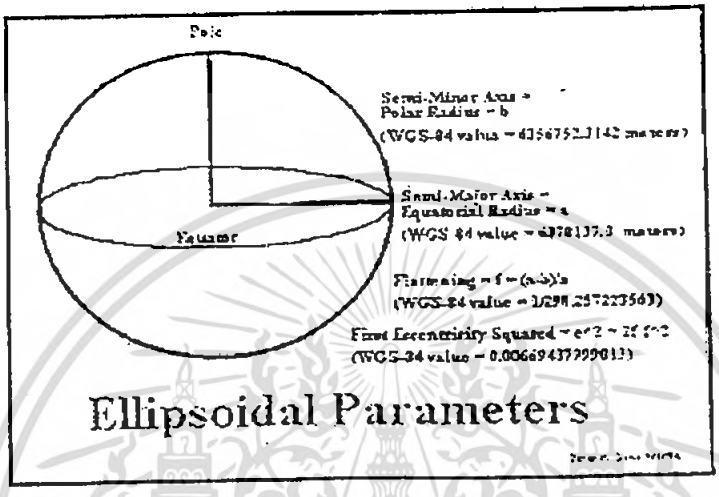
เนื่องจากพิกัดที่ได้จาก GPS มักอยู่ในรูปของละติจูด ลองติจูด หรือค่าตัวแปร XYZ การกำหนดค่าดาตัมผิด จะทำให้เกิดความผิดพลาดได้

จีโอเดติค ดาตัม คือ การกำหนดระดับอ้างอิงที่ใช้อธิบายขนาดและรูปร่างของโลกว่าควรมีลักษณะอย่างไร ในสมัยโบราณถือว่าโลกแบน ดังนั้นระนาบอ้างอิงจึงเป็นเพลนระนาบ ต่อมาพบว่าโลกเป็นทรงกลม ระบบอ้างอิงจึงเปลี่ยนเป็นทรงกลมตามไปด้วย จนภายหลังพบว่ารูปร่างของโลกใกล้เคียงกับความเป็นจริงมากที่สุดเป็น แบบเอลลิพซอยด์(Ellipsoid) และใช้มาจนถึงปัจจุบันนี้ เมื่อผนวกเข้ากับการกำหนดระบบพิกัดอ้างอิงก็จะทำให้สามารถกำหนดตำแหน่งบนพื้นโลกได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.4.1 การอ้างอิงโดยเอลลิพซอยด์ (Referance Ellipsoid)

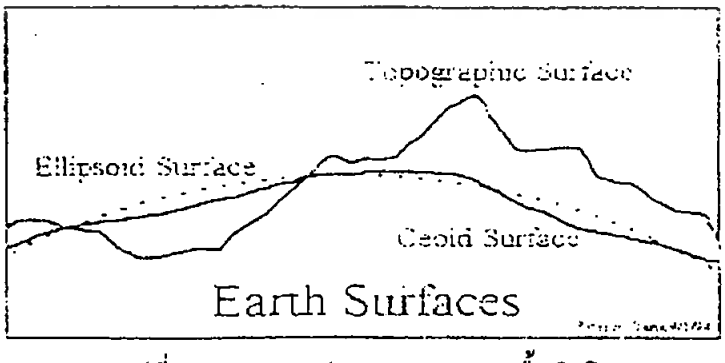
รูปแบบจำลองแบบเอลลิพซอยด์(Ellipsoid) ถือว่าพื้นผิวโลกราบเรียบและมีโครงสร้างเกือบเป็นทรงกลม (ป่องกลางคล้ายผลส้ม แต่เพื่อความสะดวกจะวาดวงกลมแทน) การอ้างอิงใดๆ บนพื้นผิวจะทำผ่านสองแกนคือ Semi-Major Axis คือ เส้นที่ลากจากจุดศูนย์กลางของโลกมายังเส้นอีเควเตอร์ และ Semi-Major Axis ซึ่งก็คือ เส้นที่ลากจากจุดศูนย์กลางของโลกมายังขั้ว



รูปที่ 2.20 ตัวแปรของเอลลิพซอยด์ที่กำหนดมาตรฐาน

2.7.4.2 รูปแบบจำลองของพื้นผิวโลก (Earth Surface Model)

รูปแบบจำลองพื้นผิวโลกที่ถูกต้องมีความจำเป็นอย่างยิ่งสำหรับระบบนำร่อง การสำรวจและการทำแผนที่ แต่เนื่องจากพื้นผิวโลกมีความสูงต่ำไม่เท่ากัน นอกจากนั้นยังมีการเปลี่ยนแปลงจากเดิมตลอดเวลา ดังนั้นรูปแบบจำลองจึงมีหลายแบบด้วยกัน รูปแบบจำลองสภาพทางภูมิประเทศและระดับน้ำทะเลมีไว้เพื่อแสดงการเปลี่ยนแปลงทางกายภาพของผิวโลก รูปแบบจำลองทางด้านแรงโน้มถ่วงและจีออยด์ (Geoids) มีไว้เพื่อแสดงการเปลี่ยนแปลงแรงโน้มถ่วงและระดับน้ำทะเลเฉพาะส่วนของผิวโลก รูปแบบจำลองต่างๆเหล่านี้ถูกสร้างเปรียบเทียบหรืออ้างอิงร่วมกับเอลลิพซอยด์



รูปที่ 2.21 แสดงรูปแบบจำลองของพื้นผิวโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองพื้นผิวโลกแบบโทโพกราฟิค(Topographic Surface) ของโลกจะแสดงพื้นผิวจริงของแผ่นดินและทะเล ในช่วงเวลาหนึ่ง ๆ ส่วนการจำลองพื้นผิวแบบจีออยด์นั้นจะเป็นการจำลองพื้นผิวที่ถูกแสดงค่าแรงโน้มถ่วงของโลกที่ทัดเทียมระดับความสูงของน้ำทะเลเฉลี่ย (Mean sea level) จีออยด์จะเป็นพื้นผิวอ้างอิงหลักในการกำหนดความสูง

รูปที่ 2.22 แสดงระดับของจีออยด์

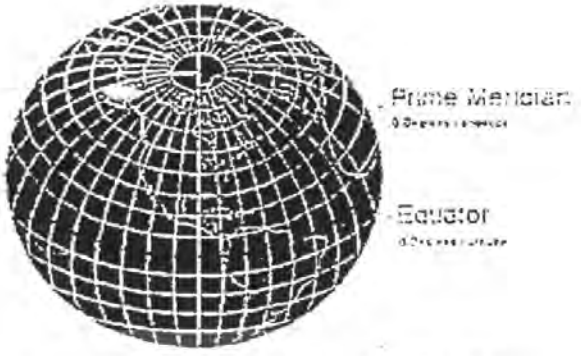
2.7.5 ระบบพิกัดอ้างอิงที่ใช้งานทั่วไป (Global Coordinate System)

เมื่อเรามีรูปแบบจำลองทางคณิตศาสตร์ของโลกแล้ว ต่อไปจะต้องกำหนดพิกัดอ้างอิงขึ้นมาเพื่อใช้กำหนดตำแหน่งบนพื้นผิวโลก มิฉะนั้นเราจะไม่ทราบเราอยู่ตรงจุดใดของโลก ในทำนองเดียวกัน ถ้าระนาบอ้างอิงแตกต่างกันไปพิกัดที่ได้จะแตกต่างกันออกไป ระบบที่ใช้ในการอ้างอิงมีหลายแบบด้วยกัน ในที่นี้จะกล่าวเฉพาะที่เกี่ยวข้องกับระบบ GPS เท่านั้น

2.7.5.1 Latitude, Longitude, Height

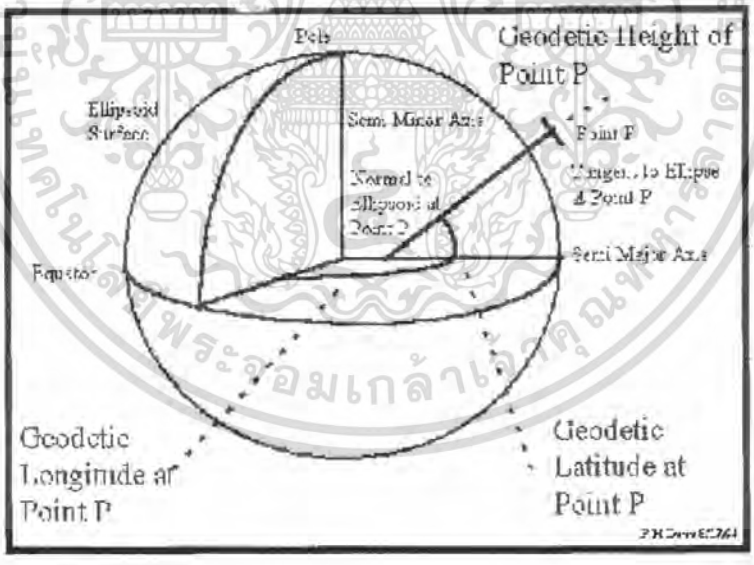
ระบบนี้เป็นระบบที่ใช้ร่วมกันมากที่สุดในโลก โดยกำหนดให้ระนาบอ้างอิง(Reference-plane) ได้มาจาก ระนาบอีเควเตอร์และไพรเมอริเดียนที่วางตั้งฉากกันดังรูป 2.23 เราจินตนาการได้ว่าโลกเสมือนแตงโมที่ว่างอยู่กับพื้นแล้วหันขั้วของแตงโมขึ้นฟ้า จากนั้นให้นึกถึงว่าที่กึ่งกลางลูกแตงโมมีเส้นๆหนึ่ง ลากยาวรอบลูกแตงโมโดยลากขนานกับพื้นที่แตงโมว่างอยู่ เส้นนี้คือ เส้นอีเควเตอร์ ถ้าใช้มีดผ่าแตงโมตามแนวเส้นนี้แล้วยกแตงโมครึ่งบนออก สิ่งที่มองเห็นคือเนื้อแตงโมที่เป็นพื้นผิวแบนราบ นั่นคือ ระนาบอีเควเตอร์นั่นเอง ทำนองเดียวกัน ถ้าเราผ่าแตงโมในทิศทางตั้งฉากกับระนาบของอีเควเตอร์ระนาบนั่นจะเรียกว่า ระนาบไพรม์เมอริเดียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 การกำหนดอีเควเตอร์และไพรม์เมอริเดียน

จากรูปที่แสดงไว้ สมมติว่าเครื่องรับสัญญาณอยู่ที่จุด P เมื่อลากเส้นมาตั้งฉากกับเส้นสัมผัสกับพื้นผิวเอลลิพซอยด์ แล้วต่อเส้นนี้ไปตัดกับระนาบอีเควเตอร์ มุมที่เกิดจากเส้นที่ลากมากับระนาบอีเควเตอร์ เรียกว่า Geodetic Latitude และถ้าถือว่าเส้นที่ลากมาจากจุด P เป็นระนาบๆ หนึ่งที่ตั้งฉากกับระนาบอีเควเตอร์ มุมที่ระนาบกระทำกับระนาบอ้างอิงไพรม์เมอริเดียน เรียก Geodetic Longitude ส่วนความยาวของเส้น h เรียก Geodetic Height



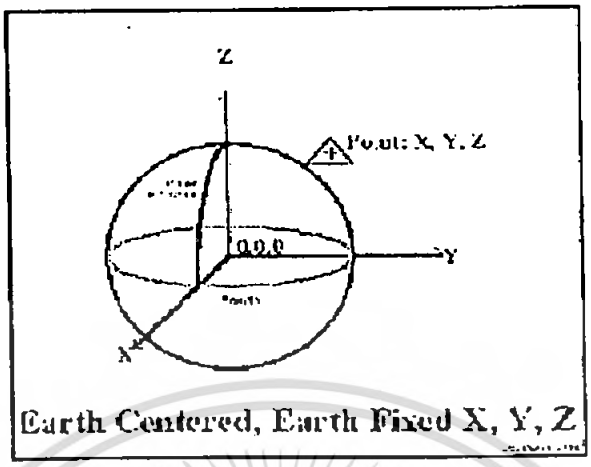
รูปที่ 2.24 แสดงการกำหนดพิกัด ละติจูด ลองติจูดและความสูง

2.7.5.2 Earth Centered และ Earth X, Y, Z

ระบบนี้เป็นการกำหนดพิกัดแบบสามมิติเช่นกัน โดยถือว่าจุดศูนย์กลางมวลของโลกเป็นจุดศูนย์กลางของเอลลิพซอยด์ จากจุดนี้แกน Z จะชี้มายังขั้วโลกเหนือ (ตามแนวแกนหมุนของโลก) แกน X จะชี้มายังจุดตัดกันของเส้นไพรม์เมอริเดียนกับเส้นอีเควเตอร์และแกน Y จะตั้งฉากกับแกน X ดังนั้นแต่ละจุดที่กำหนดขึ้นมาจะต้องมีองค์ประกอบของค่า X, Y, Z อยู่ด้วยเสมอ ระบบนี้จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิงในระบบGPS เป็นหลัก แล้วจึงทำการคำนวณจากค่า X, Y, Z มาเป็นค่า Latitude, Longitude, Height ในภายหลัง



รูปที่ 2.25 แสดงระบบ ECEF X, Y, Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

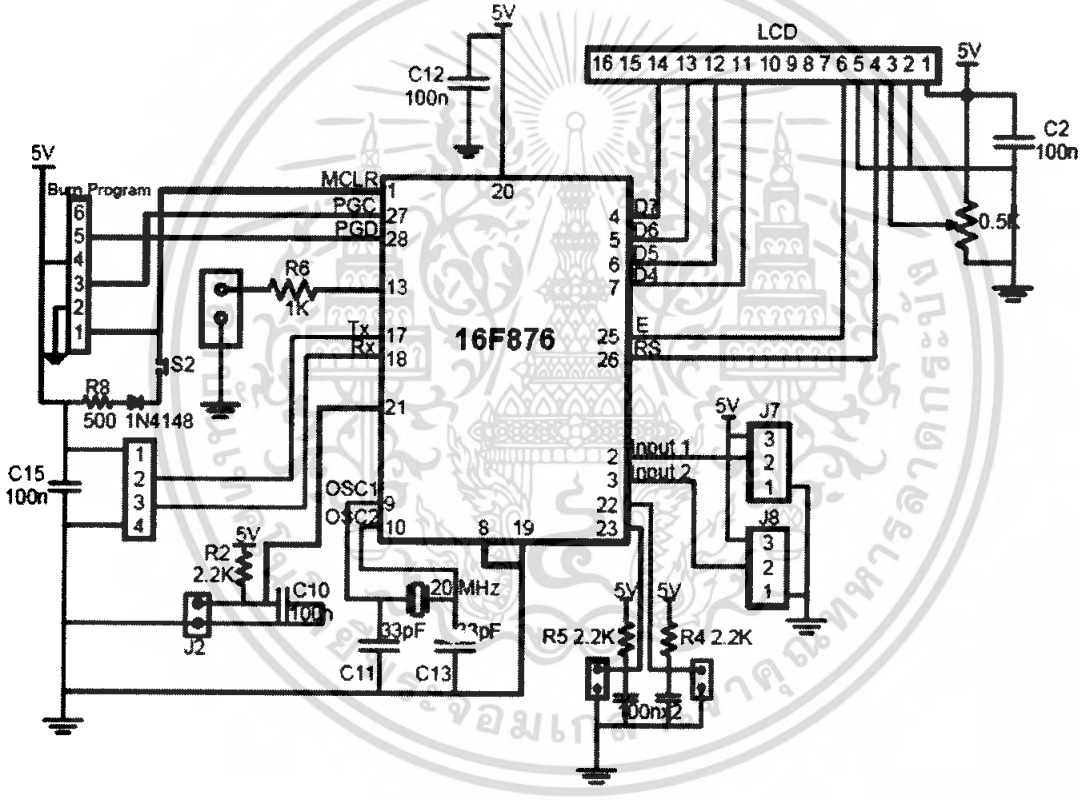
บทที่ 3
ส่วนประกอบและหลักการทำงานของวงจร

การออกแบบวงจรจะแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนของภาคส่งสัญญาณ
2. ส่วนของภาครับสัญญาณ

3.1 ภาคส่งสัญญาณ

ทำหน้าที่แปลงสัญญาณอนาลอกจากจอยสติกโดยใช้โมดูลภายในไมโครคอนโทรลเลอร์ชนิด PIC และส่งสัญญาณไปกับคลื่นวิทยุ ซึ่งภาคส่งสัญญาณทั้งหมดได้แสดงดังนี้



รูปที่ 3.1 วงจรภาคส่งสัญญาณ

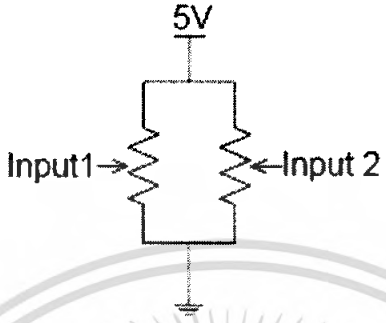
ภาคส่งสัญญาณมีส่วนประกอบต่างๆ ดังนี้

3.1.1 จอยสติก (Joystick)

โครงสร้างภายนอกของจอยสติกประกอบด้วย คันโยก ปุ่มเคลื่อนปรับความต้านทานในแนวแกน X และปุ่มเคลื่อนปรับความต้านทานในแนวแกน Y เมื่อโยกจอยสติกไปในทิศทางต่างๆ ทำให้เกิดการเปลี่ยนค่าความต้านทานในแนวแกน X และการเปลี่ยนค่าความต้านทานแนวแกน Y โดยค่าความความต้านทานในแนวแกน Y จะเกิดการเปลี่ยนแปลงเมื่อโยกจอยสติกไปข้างหน้าหรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

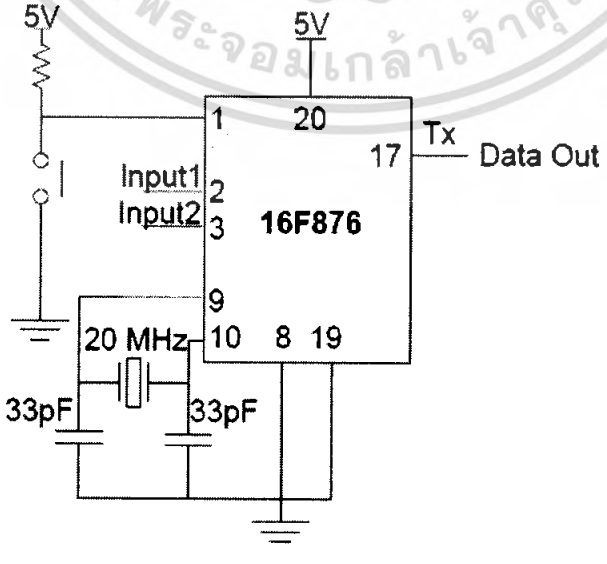
ข้างหลัง ส่วนความต้านทานในแนวแกน X นั้นจะเกิดการเปลี่ยนแปลงเมื่อโยกจอยสติกไปทางขวาหรือซ้าย จุดกึ่งกลางเป็นจุดที่ค่าความต้านทานของแกน X และแกน Y อยู่กึ่งกลางการเปลี่ยนแปลงค่าความต้านทาน การใช้งานจอยสติกจะต้องศึกษาถึงวงจรภายในของจอยสติกที่ทำหน้าที่เปลี่ยนการโยกซึ่งเป็นงานทางกลให้เป็นค่าทางไฟฟ้าโดยแสดงวงจรภายใน



รูปที่ 3.2 วงจรภายในจอยสติก

3.1.2 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ที่ใช้เป็นชนิด PIC16F876 เมื่อรับสัญญาณจากจอยสติก (Joystick) ซึ่งเป็นสัญญาณอนาลอก ซึ่งการใช้งานไมโครคอนโทรลเลอร์จะใช้สัญญาณที่เป็นดิจิตอล จึงต้องมีการแปลงสัญญาณจากอนาลอกเป็นดิจิตอล โดยที่ภายในไมโครคอนโทรลเลอร์มีโมดูล Analog to Digital Converter ภายใน PIC16F876 อยู่แล้วจึงสามารถเขียนโปรแกรมเรียกใช้ได้ เมื่อเขียนโปรแกรมในส่วนของการแปลง A/D และกำหนดการส่งสัญญาณโดยใช้การส่งสัญญาณแบบอนุกรมแบบ UART ออกที่ขา 17 (RC6/TX/CX) และส่งต่อไปที่โมดูล RF Module ส่วนของวงจรไมโครคอนโทรลเลอร์นั้นจะแสดงดังรูป

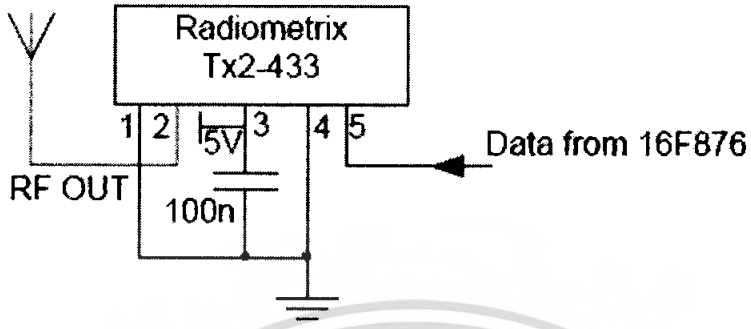


รูปที่ 3.3 วงจรไมโครคอนโทรลเลอร์ PIC16F876

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 RF Module Radiometrix Tx2-433

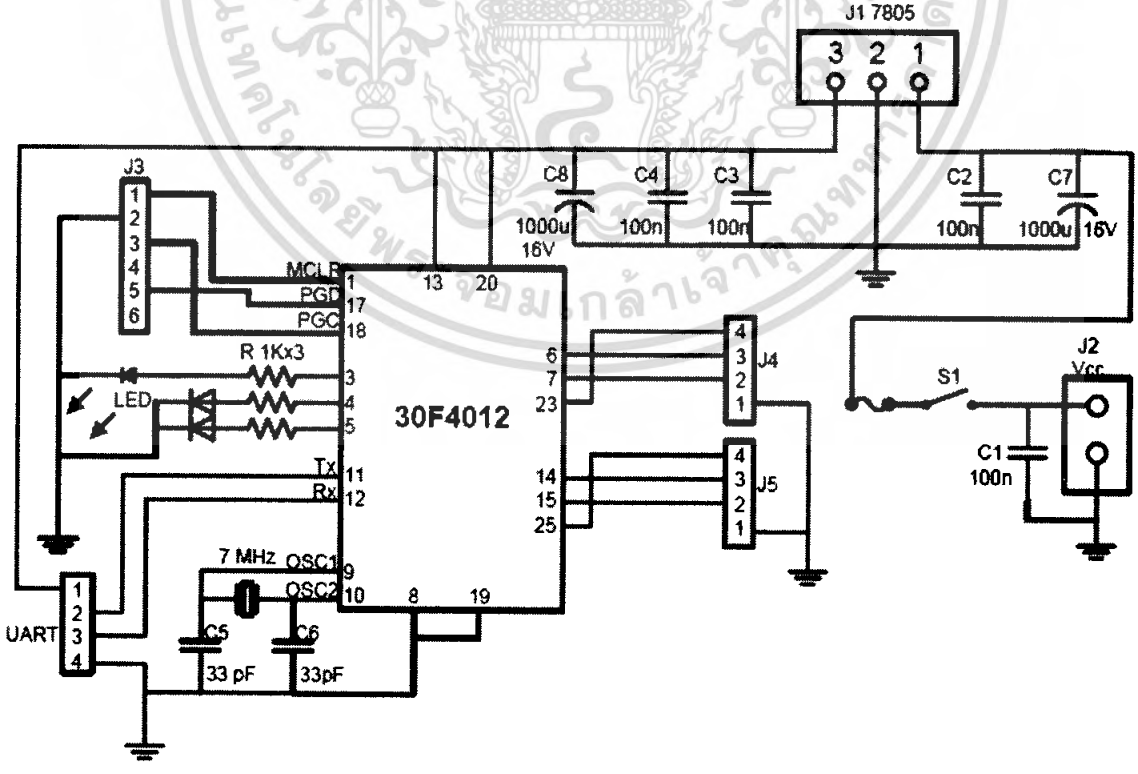
เมื่อรับข้อมูลจากขา 17 (RC6/TX/CX) ของไมโครคอนโทรลเลอร์ PIC16F876 แล้วจะทำการส่งข้อมูลออกไปด้วยคลื่นวิทยุความถี่ 433.92 MHz วงจรRF Module จะแสดงดังรูป



รูปที่ 3.4 RF Module Radiometrix Tx2-433

3.2 ภาครับสัญญาณ

ทำหน้าที่รับสัญญาณที่ส่งมาจากภาคส่งสัญญาณ แล้วใช้ไมโครคอนโทรลเลอร์ตระกูล PIC ในการประมวลผลเพื่อนำไปขับมอเตอร์ทั้ง 2 ตัวในการขับเคลื่อนรถ โดยมีวงจรทั้งหมดของภาครับสัญญาณดังนี้



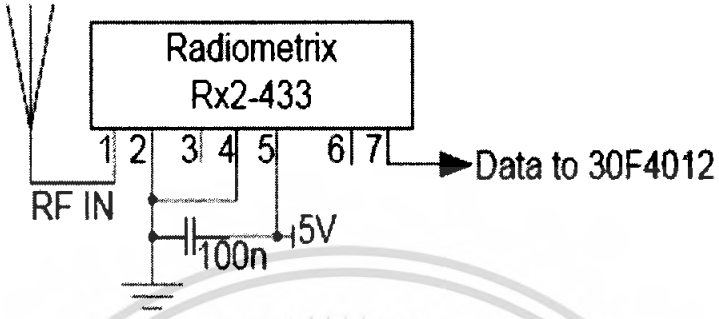
รูปที่ 3.5 วงจรรวมของภาครับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรภาครับสัญญาณมีส่วนประกอบต่างๆ ดังนี้

3.2.1 RF Module Radiometrix Rx2-433

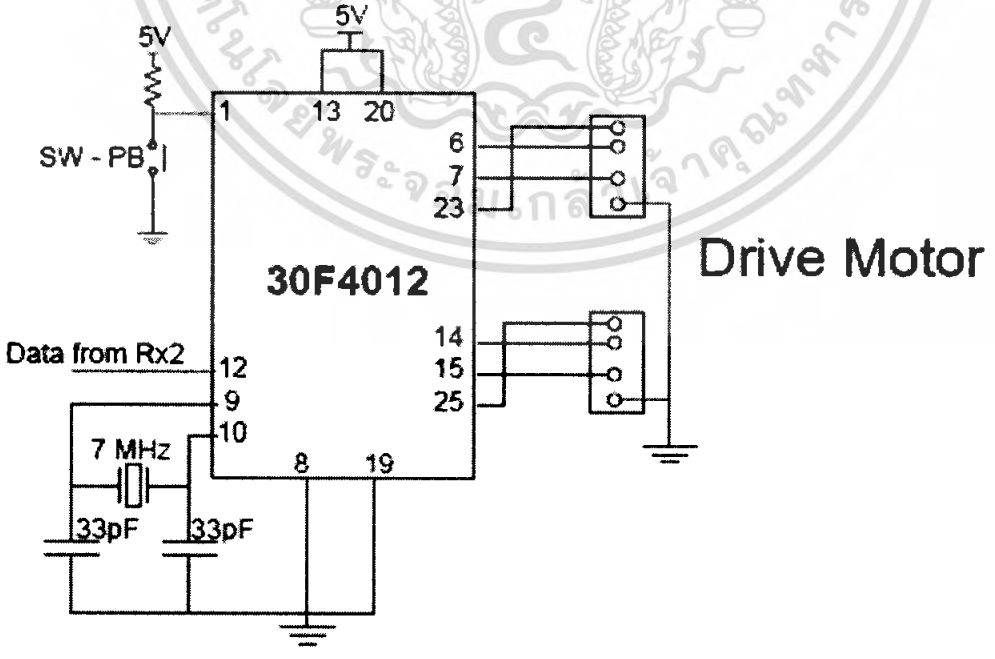
ทำหน้าที่ดีมอดดูเลทสัญญาณคำสั่งออกจากคลื่นพาห์ และส่งข้อมูลที่ดีมอดดูเลทแล้วไปที่ไมโครคอนโทรลเลอร์ dsPIC30F4012 ในการประมวลผลต่อไป วงจรRF Module จะแสดงดังรูป



รูปที่ 3.6 RF Module Radiometrix Rx2-433

3.2.2 ไมโครคอนโทรลเลอร์

เมื่อไมโครคอนโทรลเลอร์ตระกูล dsPIC30F4012 รับสัญญาณจากวงจรRF Module เข้ามาที่ขา 12 (UA1/Rx/CN0) ไมโครคอนโทรลเลอร์ตระกูล dsPIC30F4012 ซึ่งจะทำการประมวลผลของโปรแกรมเพื่อนำไปขับมอเตอร์ทั้ง 2 ตัวในการขับเคลื่อนรถต่อไป วงจรไมโครคอนโทรลเลอร์นั้นจะแสดงดังรูป

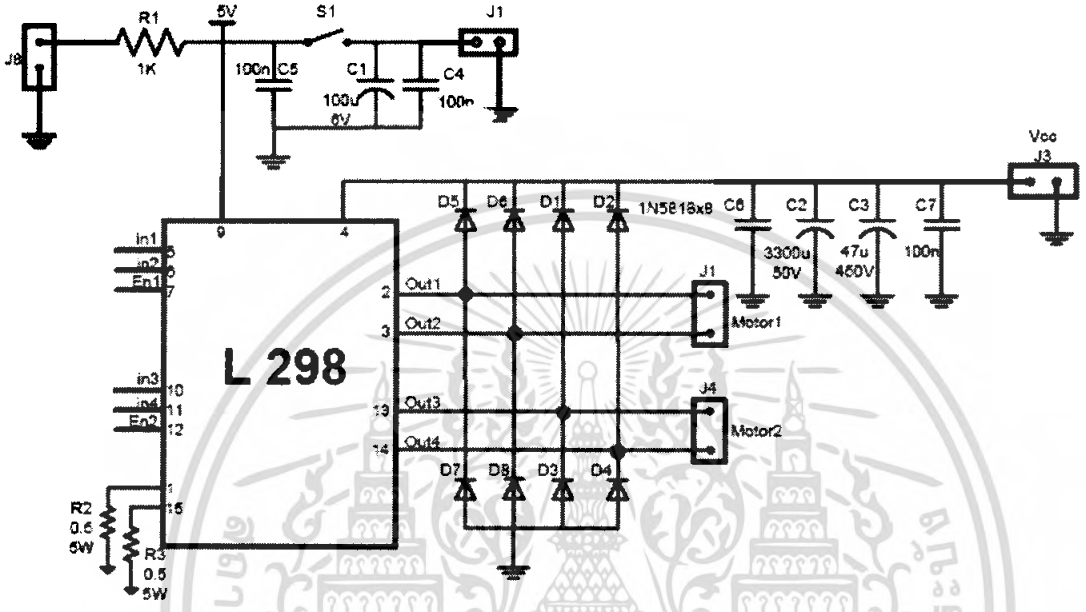


รูปที่ 3.7 วงจรไมโครคอนโทรลเลอร์ dsPIC30F4012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 วงจรขับมอเตอร์

เมื่อได้ข้อมูลจากไมโครคอนโทรลเลอร์ dsPIC30F4012 จะนำข้อมูลที่ได้ไปต่อเป็นอินพุทของวงจรขับกระแสที่ใช้ IC L298 ซึ่งทำการขับมอเตอร์แบบ full-bridge ซึ่งวงจรภายในของ IC L298 แสดงดังรูป3.8



รูปที่ 3.8 วงจรขับมอเตอร์

3.2.4 มอเตอร์

ใช้มอเตอร์กระแสตรง(DC Motor) ซึ่งใช้แรงดันการทำงาน 12 โวลต์ ซึ่งมอเตอร์ที่ใช้นี้ไปมอเตอร์ที่ติดมากับตัวรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

การทดลองที่ 1 การแปลงค่าแรงดันของจอยสติกจากอนาลอกเป็นดิจิตอลแล้วอ่านค่าดิจิตอลจากคอมพิวเตอร์ผ่านทางพอร์ทอนุกรม

ตาราง 4.1 (ค่าแรงดันที่ใช้ = 5 Volts)

ค่าแรงดันที่อ่านจากตัวต้านทานปรับค่าได้(Volts)	ค่าแรงดันที่แปลงเป็นDigital	ค่าแรงดันที่แปลงจากค่าDigital (Volts)
0	0	0
0.001	0	0
0.504	103	0.507
1.003	205	1.005
1.518	311	1.518
2.020	413	2.018
2.511	514	2.509
3.010	615	3.014
3.502	716	3.498
4.010	820	4.000
4.530	926	4.525
5.000	1023	5.000

หมายเหตุ

โมดูล A/D ของ PIC16F876 มีความละเอียด 10 บิต ฉะนั้นค่าสูงสุดของค่า Digital ที่แปลงจากแรงดันมีค่า 1023 (2^8)

การทดลองที่ 2 ทดสอบสมรรถนะของตัวรถ

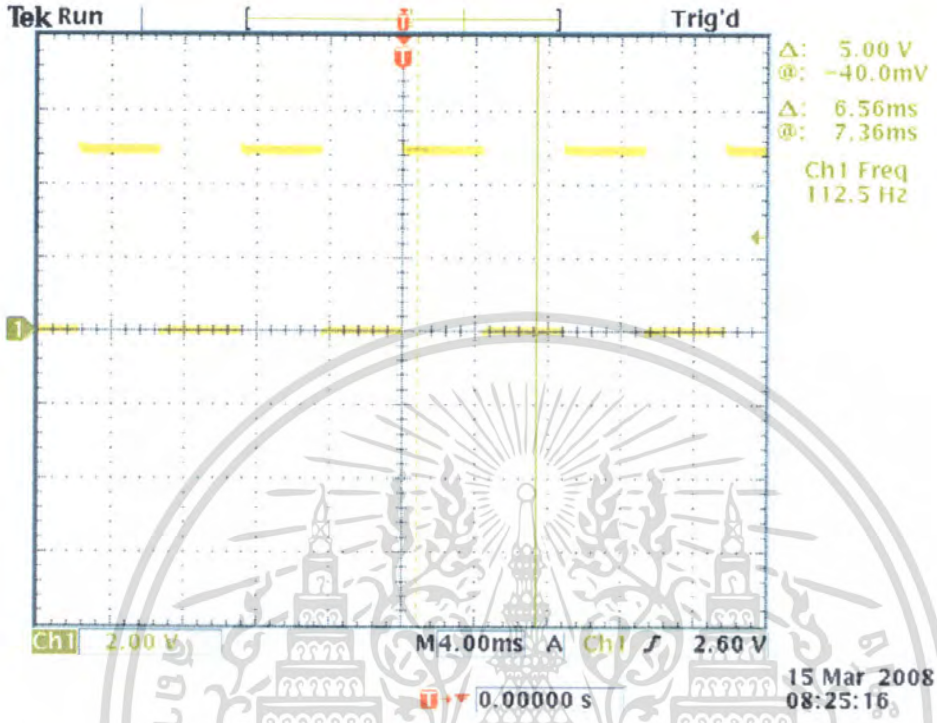
มุมเลี้ยว	=	360	องศา
ระยะไกลสุดที่ควบคุมได้	=	60	เมตร
ระยะเวลาในการใช้งาน	=	30	นาที
น้ำหนักที่สามารถรับได้	=	500	กรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

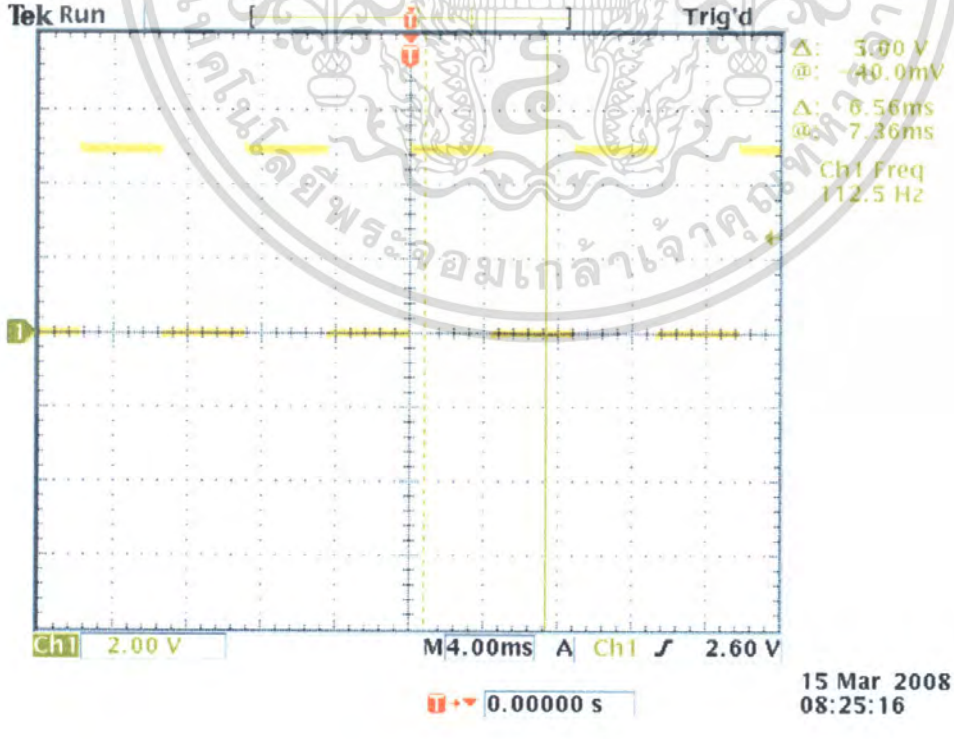
การทดลองที่ 3 วัดสัญญาณ PWM ที่วงจร Drive Motor

FORWARD (duty cycle = 50%)

WHEEL1



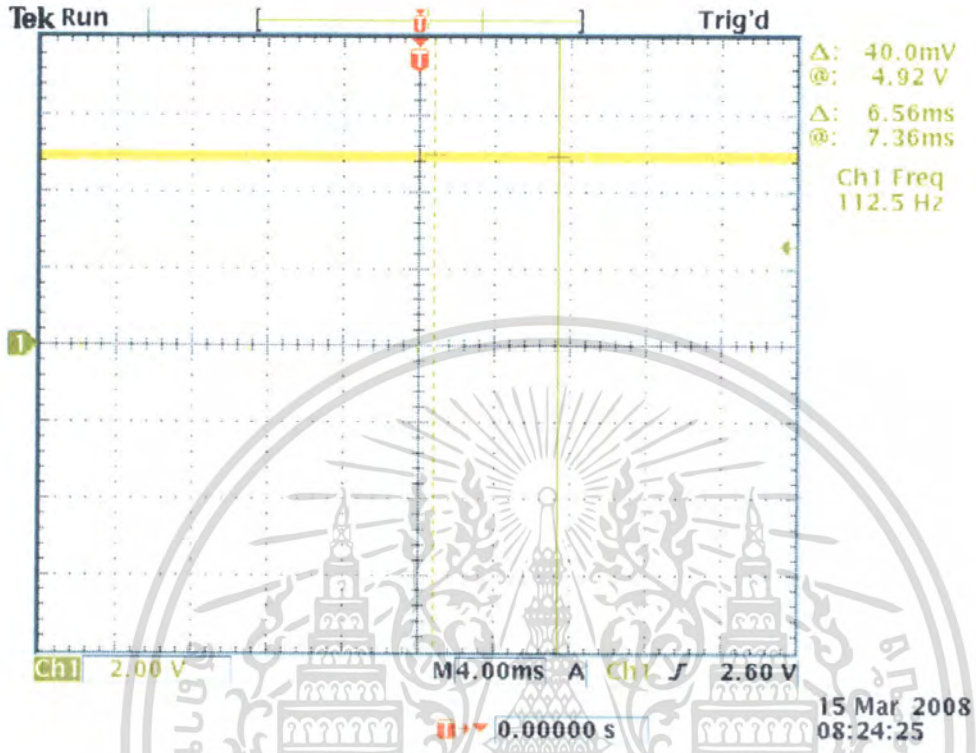
WHEEL2



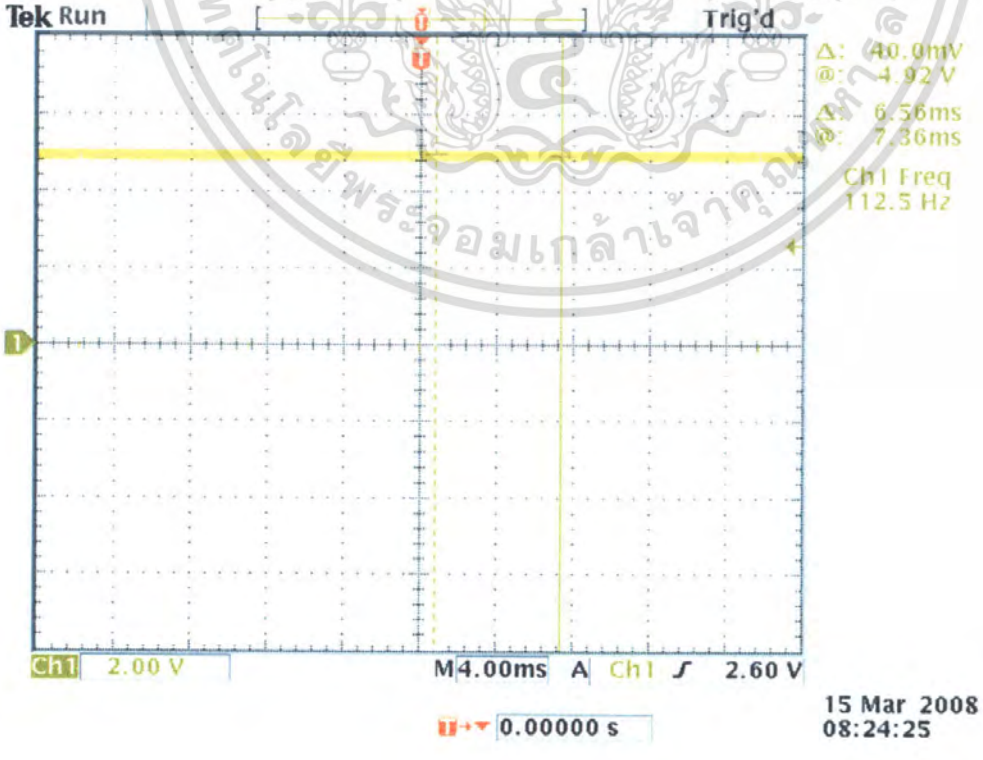
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FORWARD (duty cycle =100%)

WHEEL1



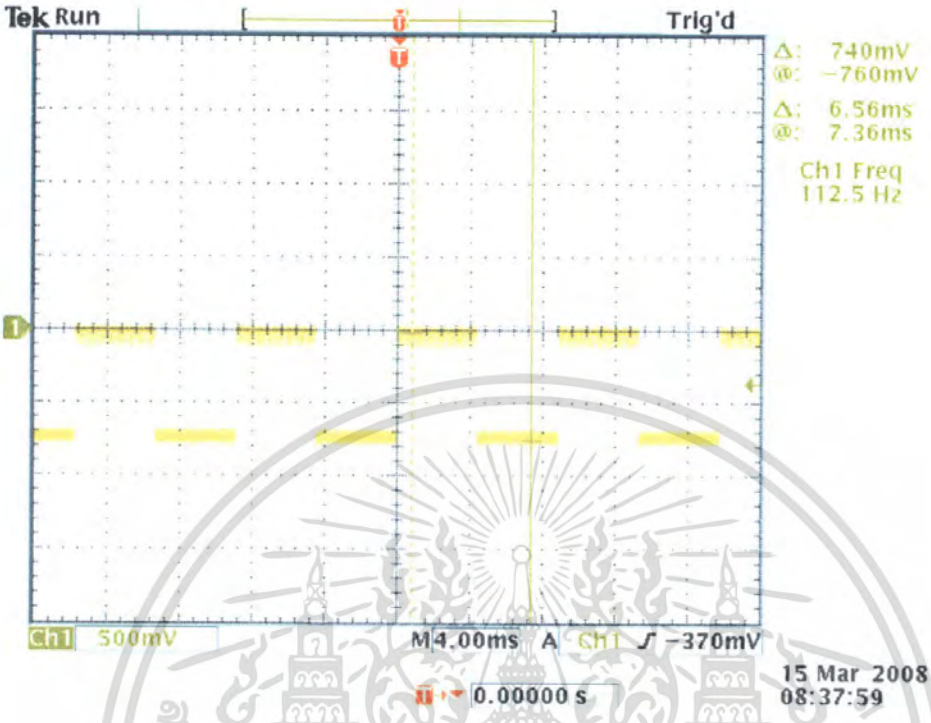
WHEEL2



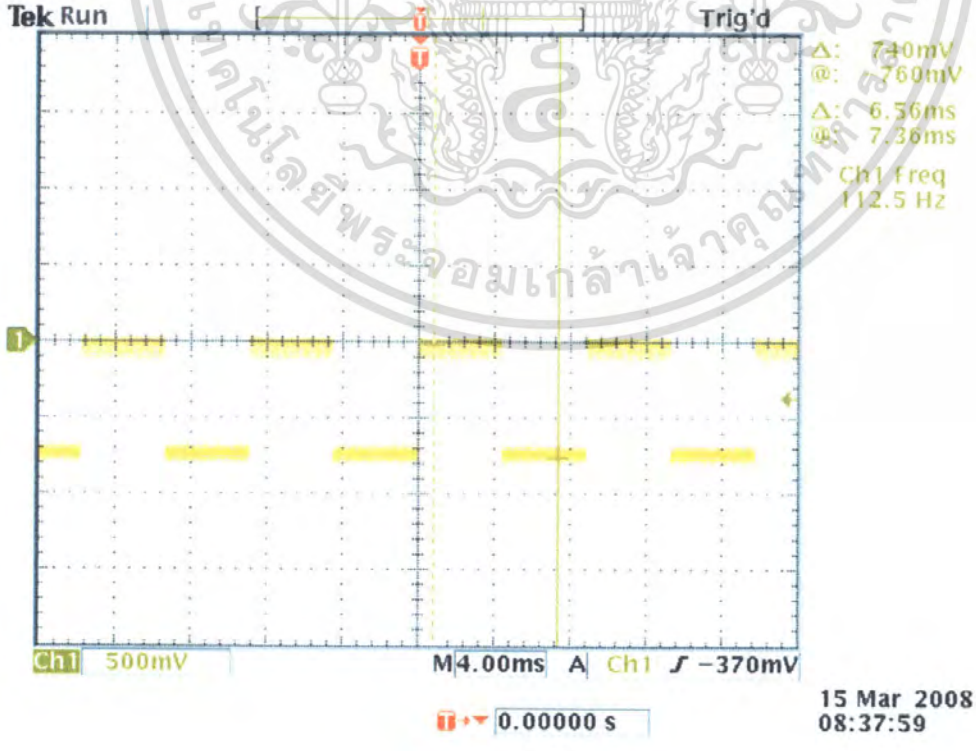
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BACKWARD (duty cycle = 50%)

WHEEL1



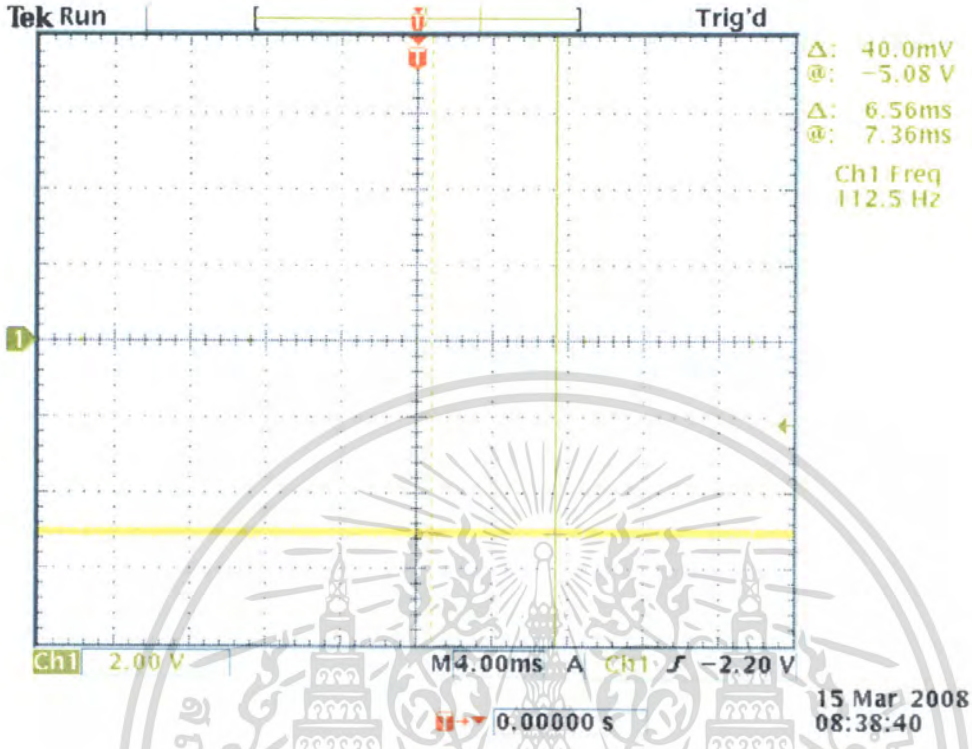
WHEEL2



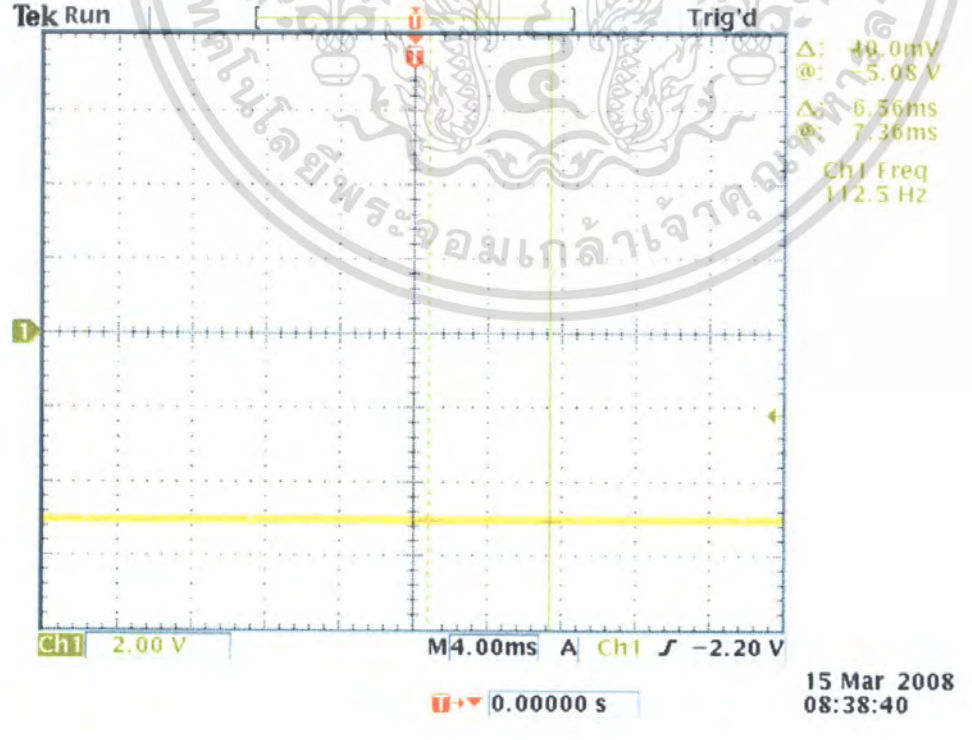
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BACKWARD (duty cycle = 100%)

WHEEL1



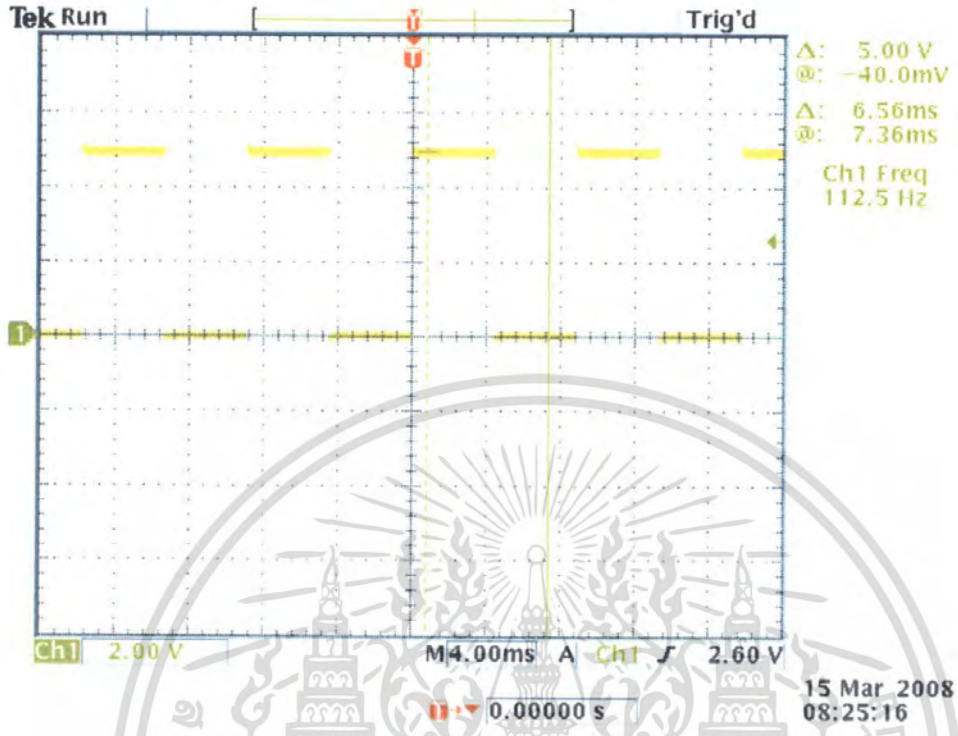
WHEEL2



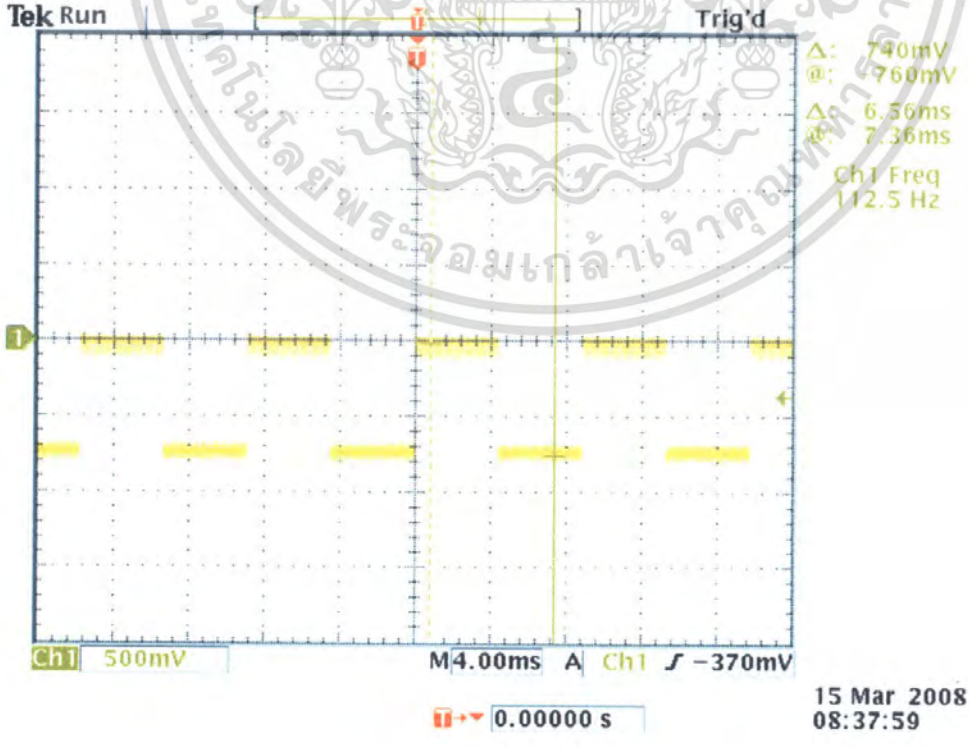
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TURN RIGHT (duty cycle = 50%)

WHEEL1



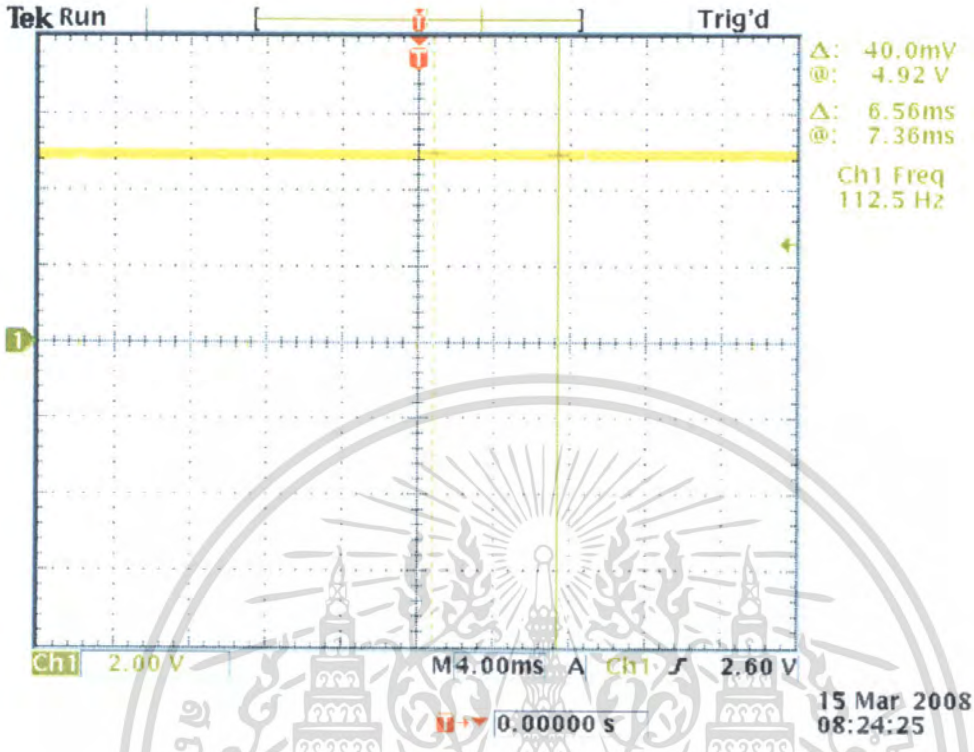
WHEEL2



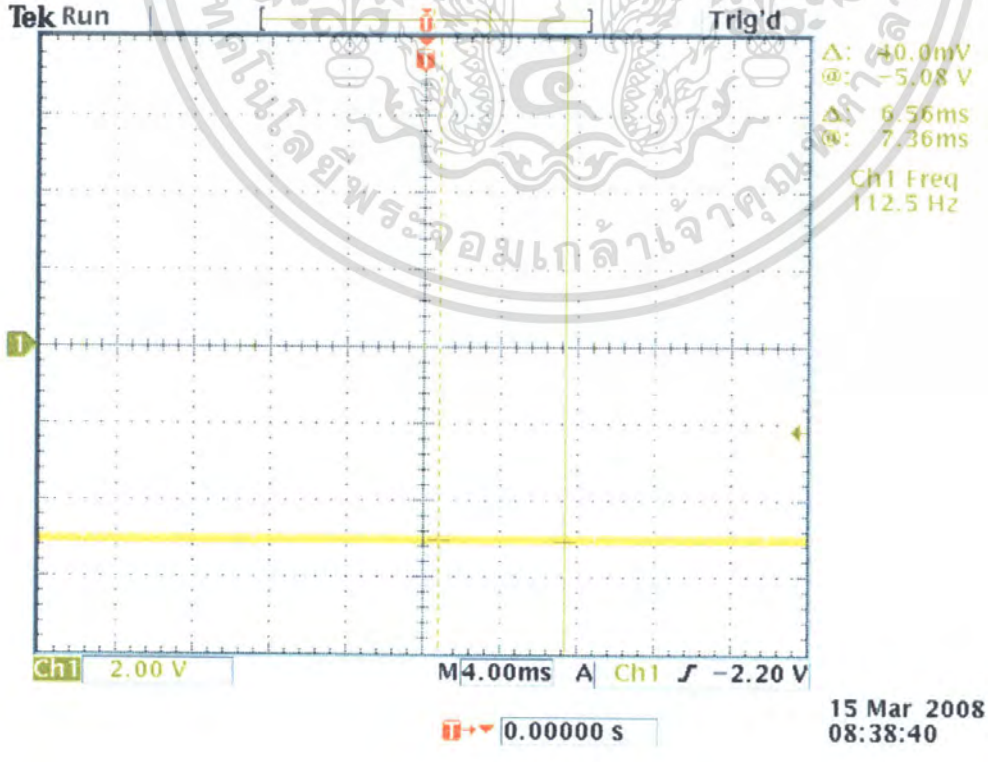
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TURN RIGHT (duty cycle =100%)

WHEEL1



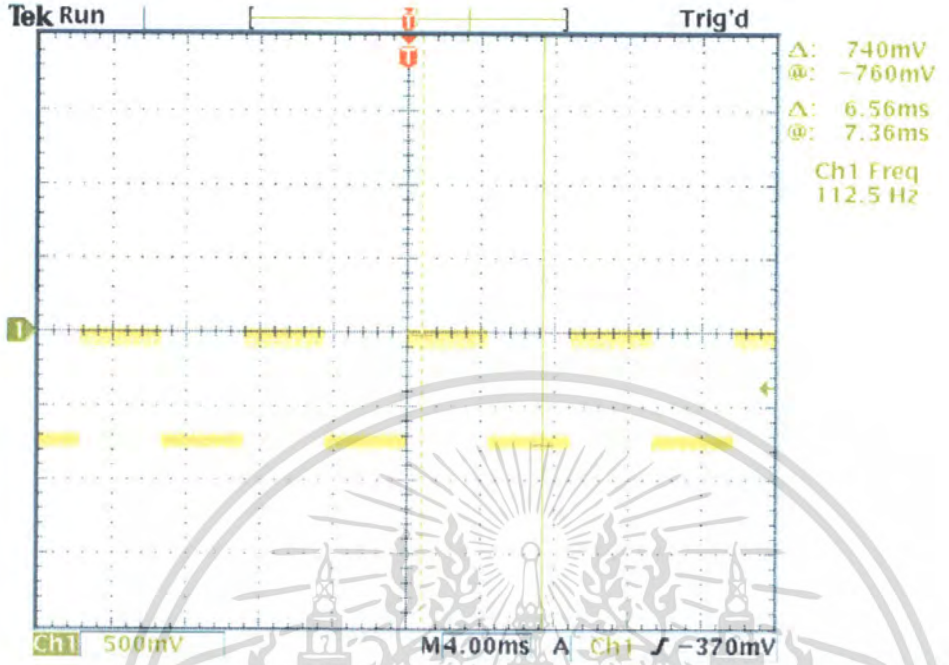
WHEEL2



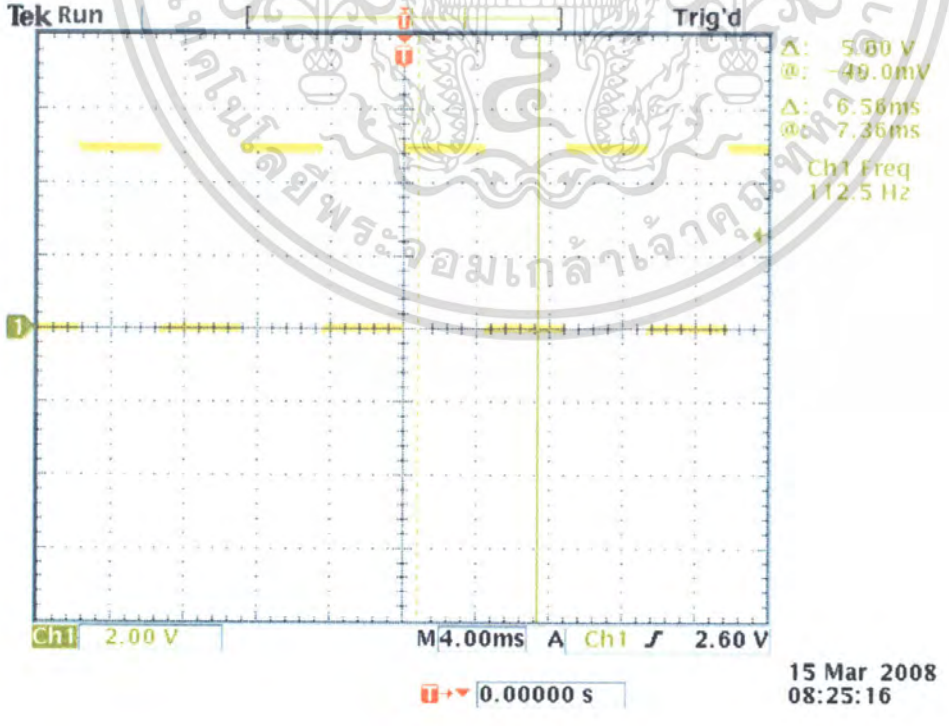
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TURN LEFT (duty cycle = 50%)

WHEEL1



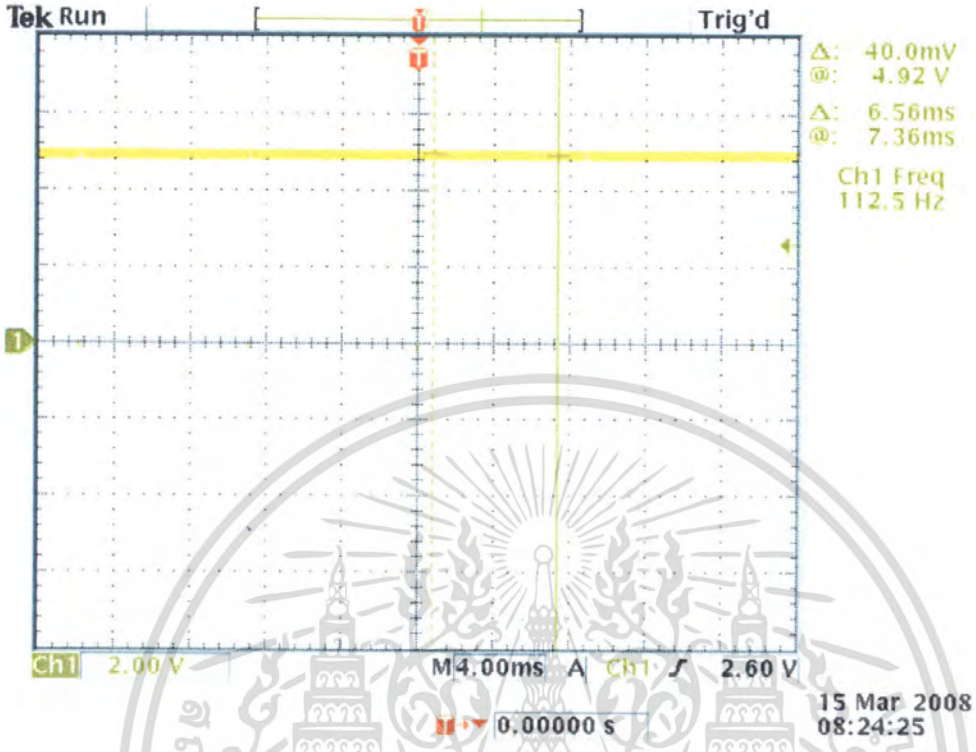
WHEEL2



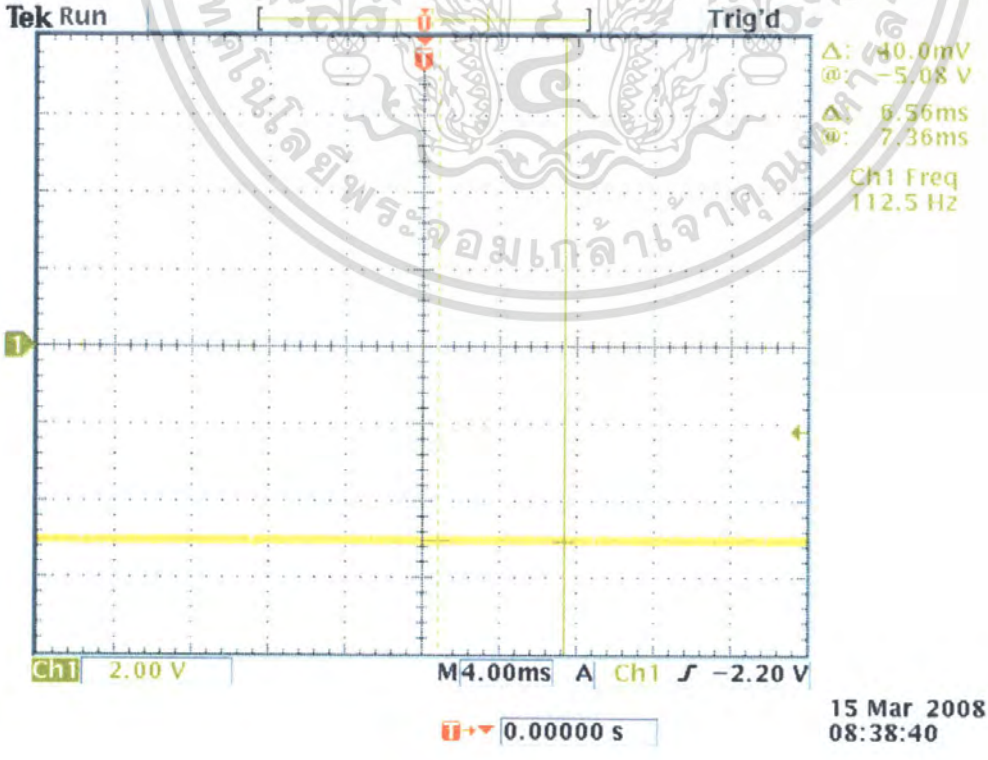
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TURN LEFT (duty cycle = 100%)

WHEEL1



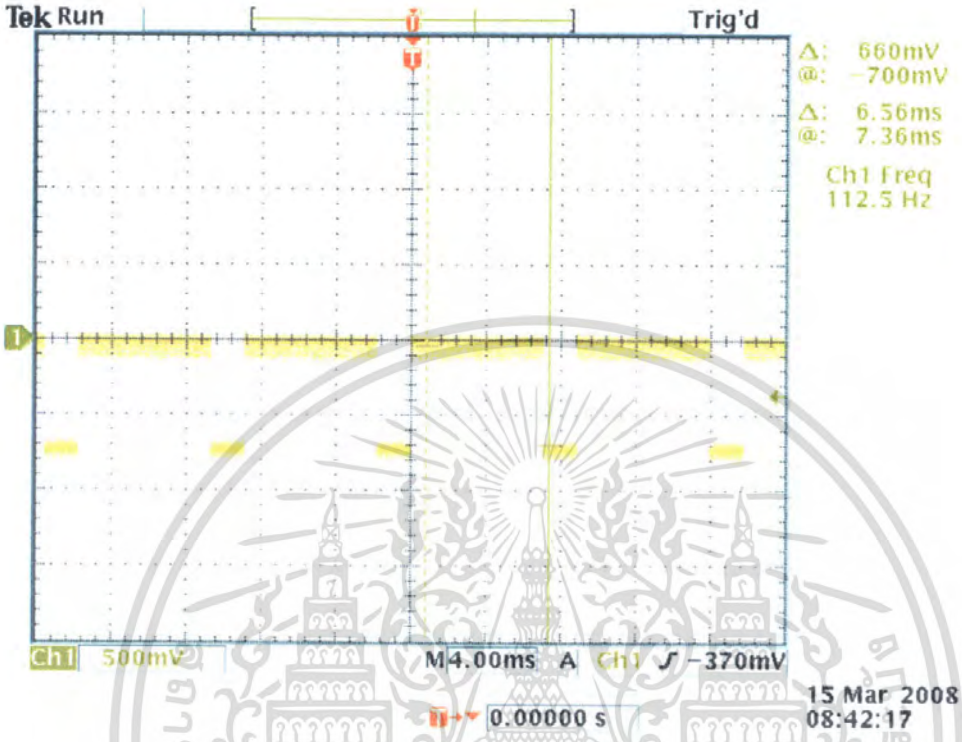
WHEEL2



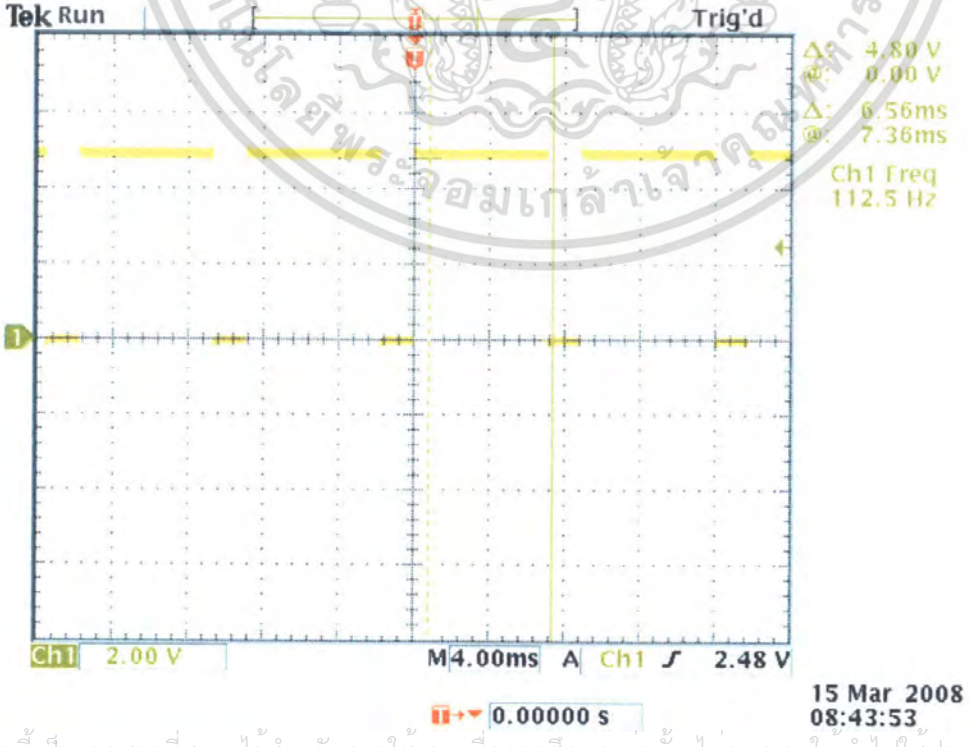
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดินหน้าเฉียง 25 องศา

WHEEL1



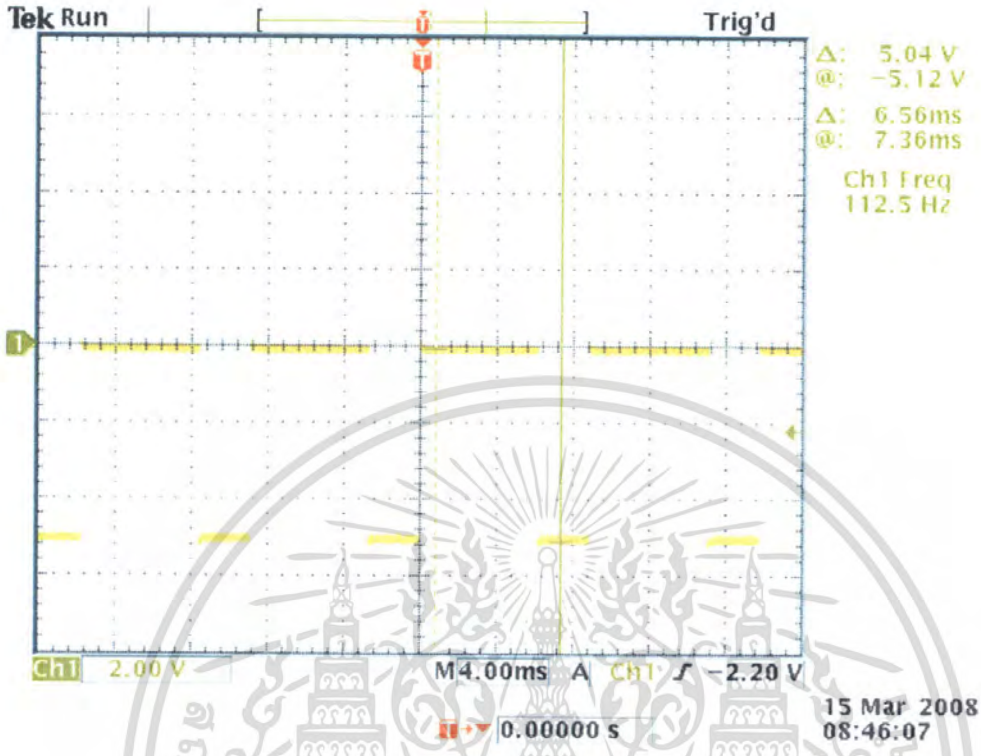
WHEEL2



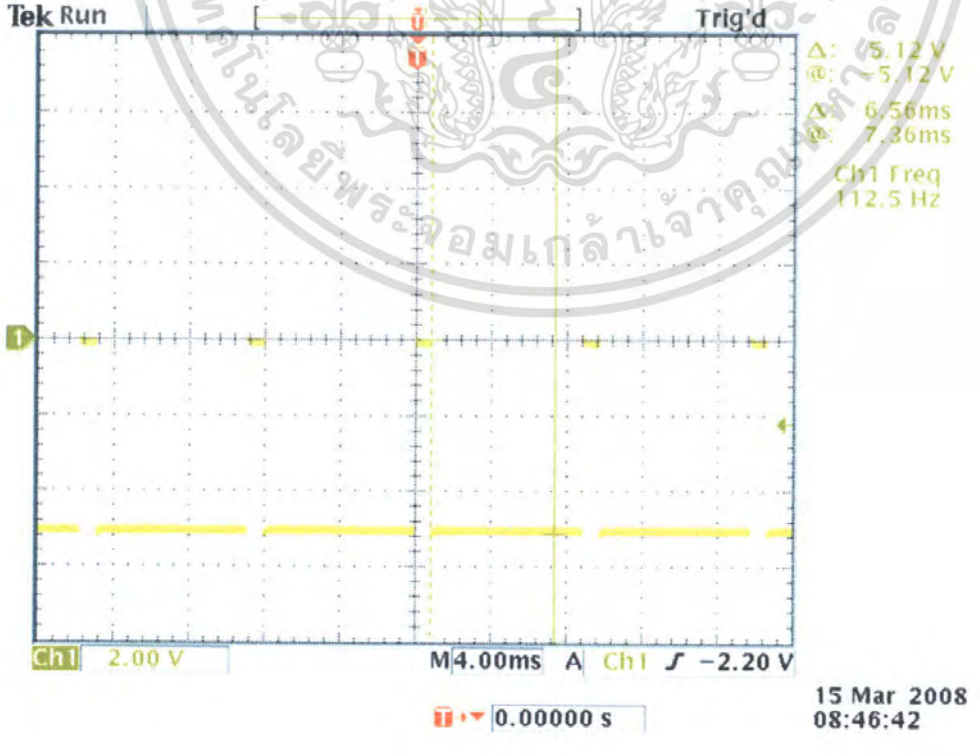
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอยหลังเฉียง 25 องศา

WHEEL1



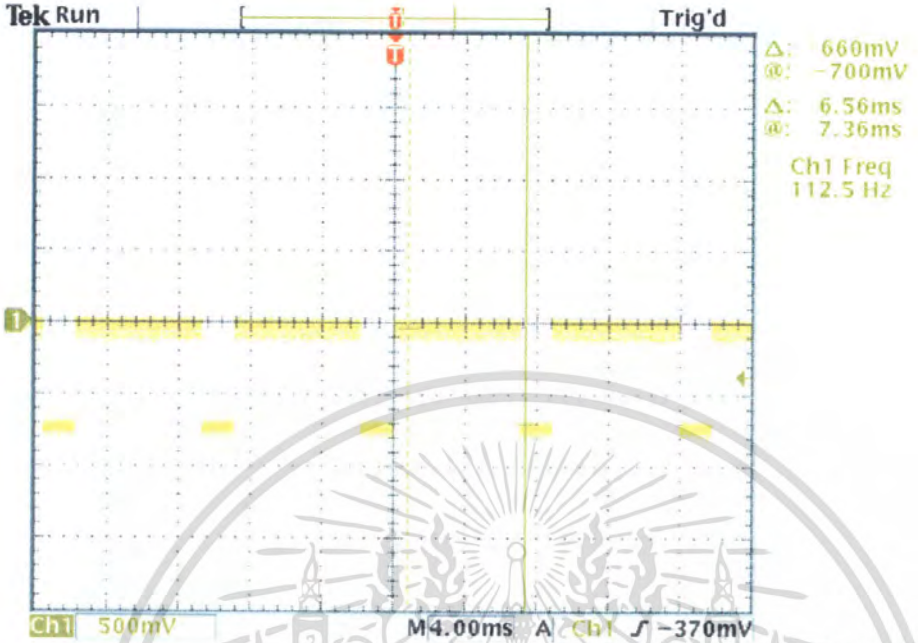
WHEEL2



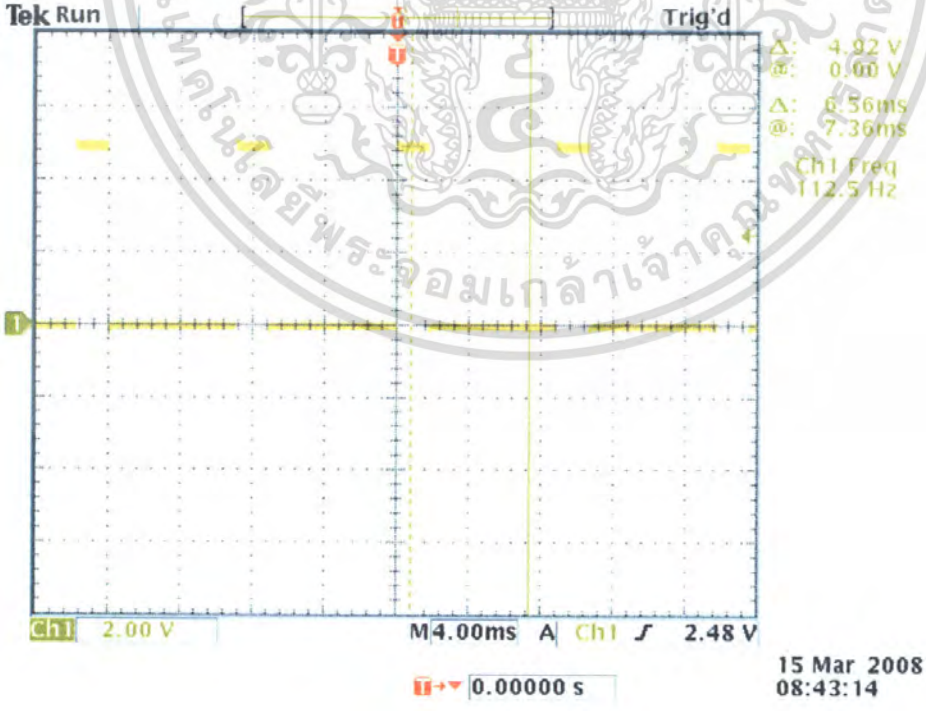
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดินหน้าเฉียง 60 องศา

WHEEL1



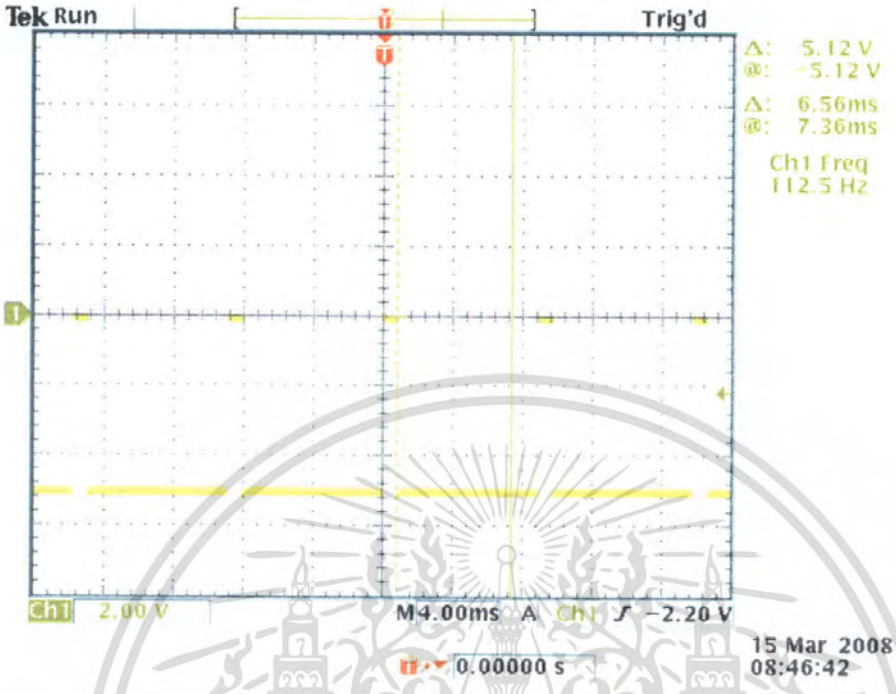
WHEEL2



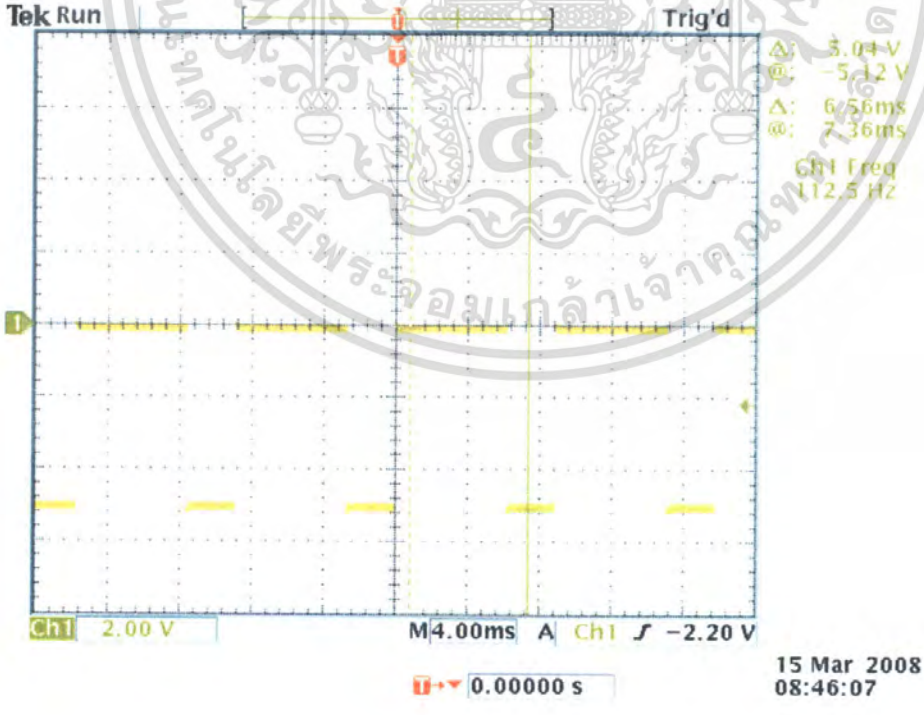
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอยหลังเฉียง 60 องศา

WHEEL1



WHEEL2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุป

จากการทำโครงงานนี้ขั้นแรกได้ทำการศึกษาถึงรายละเอียดการเขียนโปรแกรมภาษา C ในการใช้ไมโครคอนโทรลเลอร์ PIC16F876 และ dsPIC30F4012 ในการประมวลผล การเขียนภาษา C นั้นใช้คอมไพเลอร์ CCS ในการเขียน และใช้ MPLAB ในการ Burn โปรแกรม ซึ่งสิ่งที่กล่าวมาข้างต้นนี้เป็นสิ่งที่ต้องเริ่มศึกษาใหม่เพราะไม่เคยศึกษาการเขียนภาษา C เลย

ในส่วนของจอยสติกนั้นเราใช้นั้นต้องนำมาดัดแปลงแก้ไขให้ใช้งานเป็นอย่างที่ต้องการคือนำจอยสติกมาศึกษาว่าวงจรของจอยสติกเป็นอย่างไร เอาท์พุทที่ได้เป็นอย่างไร เมื่อเราโยกจอยสติกแล้วค่าของเอาท์พุทเป็นอย่างไร จะต้องศึกษาการใช้งานโมดูล A/D ในการแปลงค่าสัญญาณอนาลอกจากจอยสติกเป็นสัญญาณดิจิตอลเพื่อนำไปให้ไมโครคอนโทรลเลอร์ PIC16F877 ในการประมวลผลต่อไป ส่วนของฮาร์ทแวร์นั้นตัวรถได้ซื้อสำเร็จ ขนาดรูปร่างของรถไม่เหมาะสมอย่างที่ควรซึ่งจะต้องปรับปรุงให้ดีขึ้น

แนวทางที่จะพัฒนาต่อไปจะทำการติดแผ่นโซลาร์เซลล์ให้กับตัวรถลาดตระเวนเพื่อชาร์จแบตเตอร์รี่เองเมื่อแบตเตอร์รี่ของรถลาดตระเวนใกล้จะหมด

5.2 ปัญหาและแนวทางแก้ไข

ภาษาC เป็นภาษาที่ไม่เคยศึกษา จึงต้องใช้เวลาในการศึกษาภาษา C และศึกษาโมดูลต่างๆ จึงเสียเวลาในการศึกษาค่อนข้างมาก ส่วนวงจรที่ใช้ในติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ PIC16F876 และ dsPIC30F4012 กับ คอมพิวเตอร์เพื่อใช้ในการศึกษาการแปลงค่าอนาลอกเป็นดิจิตอลนั้นไม่สามารถติดต่อสื่อสารกันได้ แก้ไขโดยการศึกษาวงจรจากหนังสือหลายๆเล่ม เพื่อศึกษาว่ามีการต่อวงจรผิดพลาดหรือไม่

5.3 ประโยชน์ที่ได้รับ

ในปัจจุบันการสื่อสารโดยอาศัยคลื่นวิทยุมีการใช้งานอย่างแพร่หลาย จากการศึกษาการรับส่งคลื่นวิทยุและการได้ศึกษาภาษา C ทำให้โครงงานนี้เป็นการเพิ่มพูนความรู้อย่างยิ่ง

อีกทั้งในการทำโครงงานนี้ยังช่วยเพิ่มประสบการณ์ในการทำงาน ทำให้ได้ความรู้จากการศึกษาวงจรและอุปกรณ์ต่างๆ ซึ่งจะเป็นประโยชน์ในอนาคตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

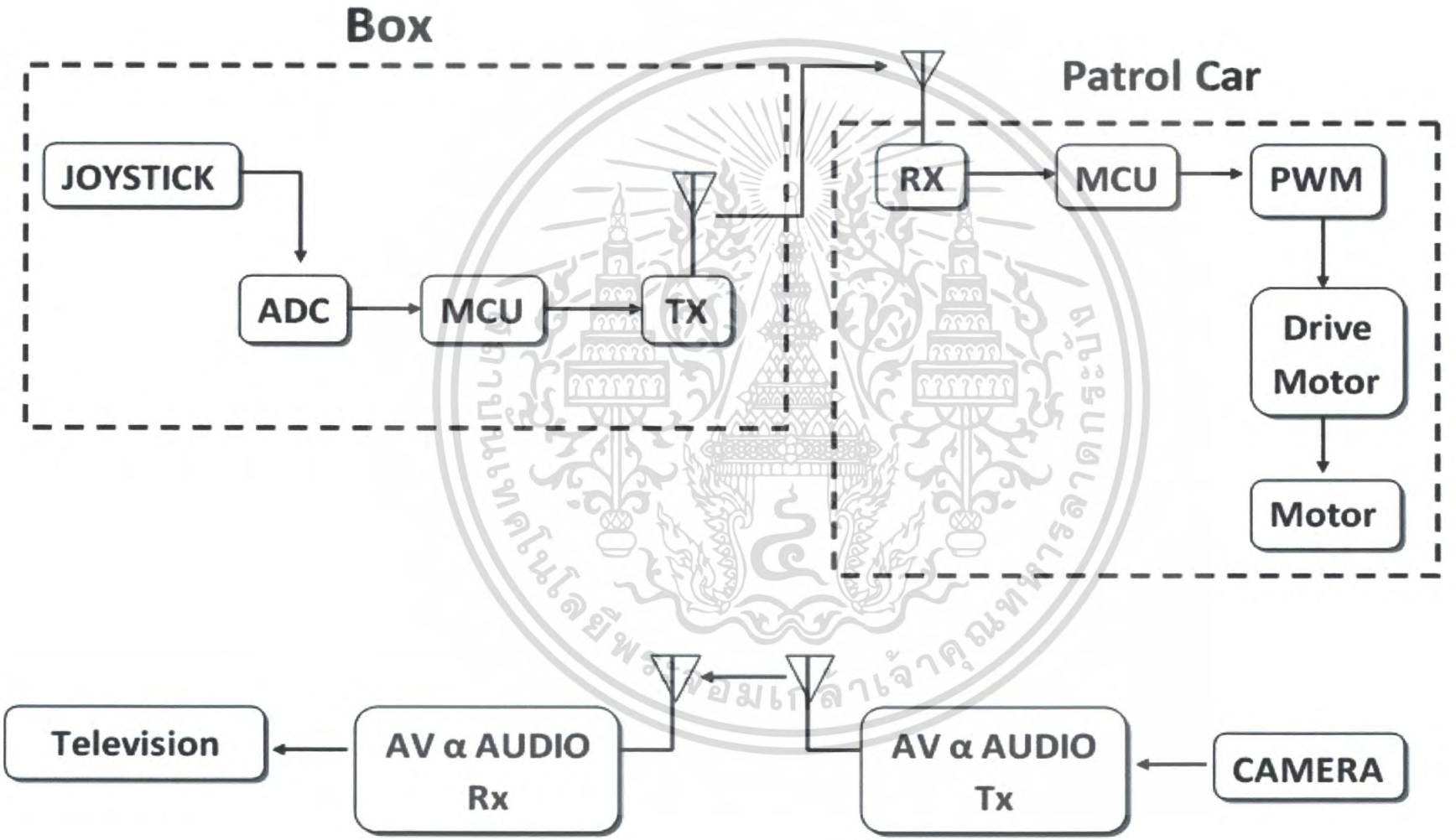
1. ประจิน พลังสันติกุล " PIC Works Examples and C Source Code" บริษัทแอพซอฟเทค จำกัด พ.ศ. 2537
2. http://www.appsofttech.com
3. วัชรินทร์ เคารพ "การเรียนรู้และเข้าใจสถาปัตยกรรมไมโครคอนโทรลเลอร์PIC16F877" บริษัท อีทีที จำกัด พ.ศ. 2537
4. ประจิน พลังสันติกุล " PIC microcontroller programming with CCS C complier " บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด พ.ศ. 2521

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

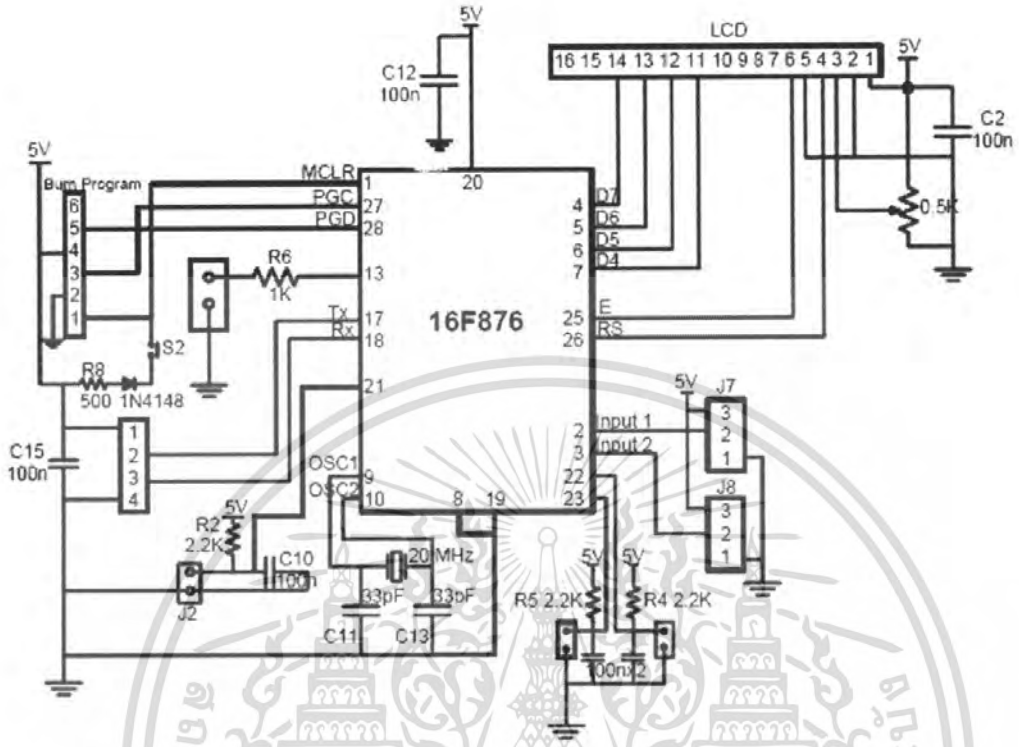
ภาคผนวก

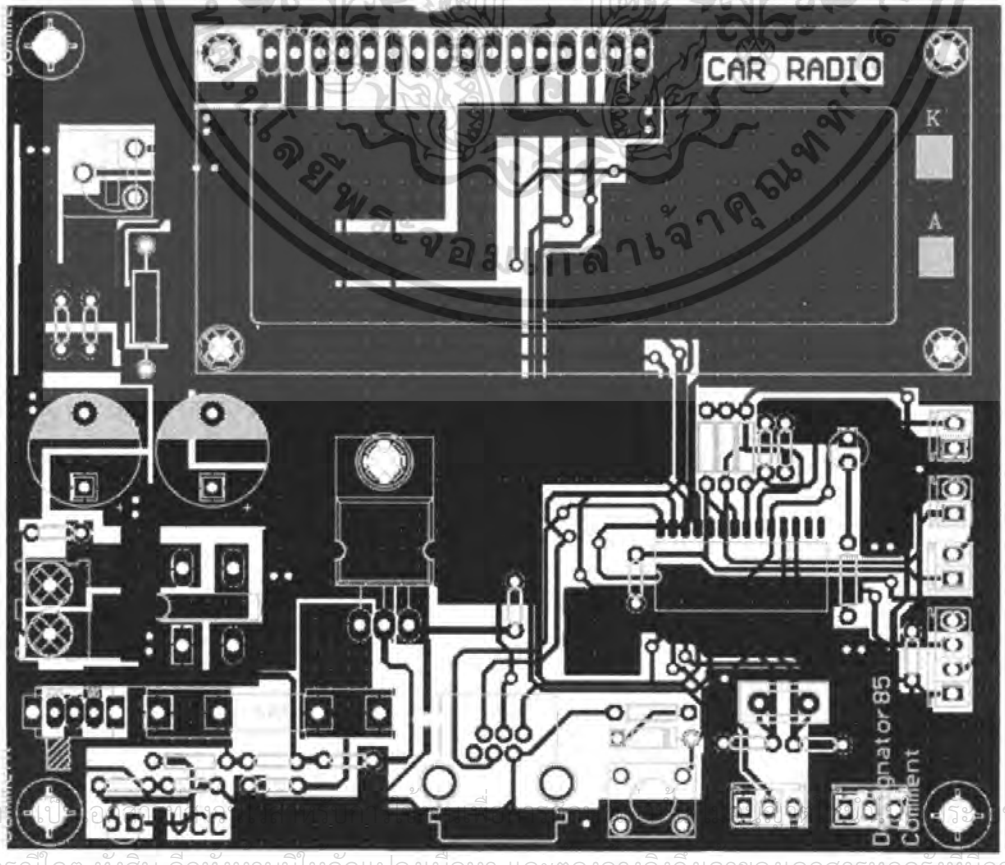
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM



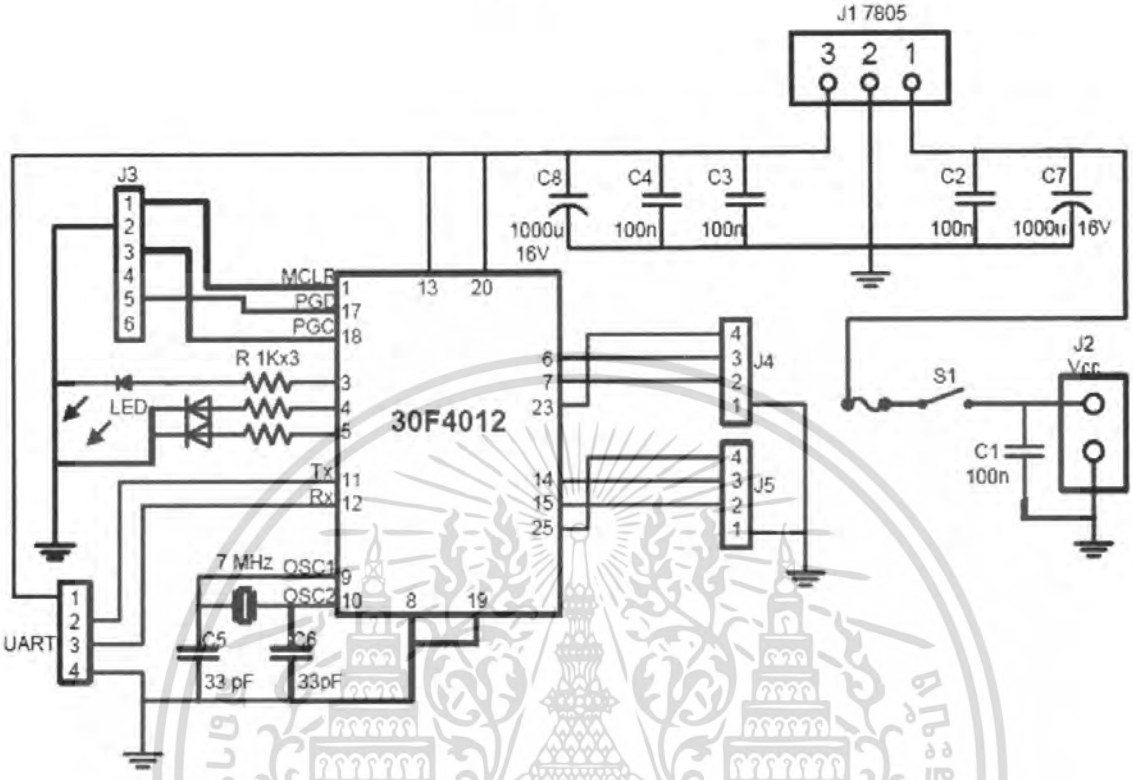
ภาคส่ง

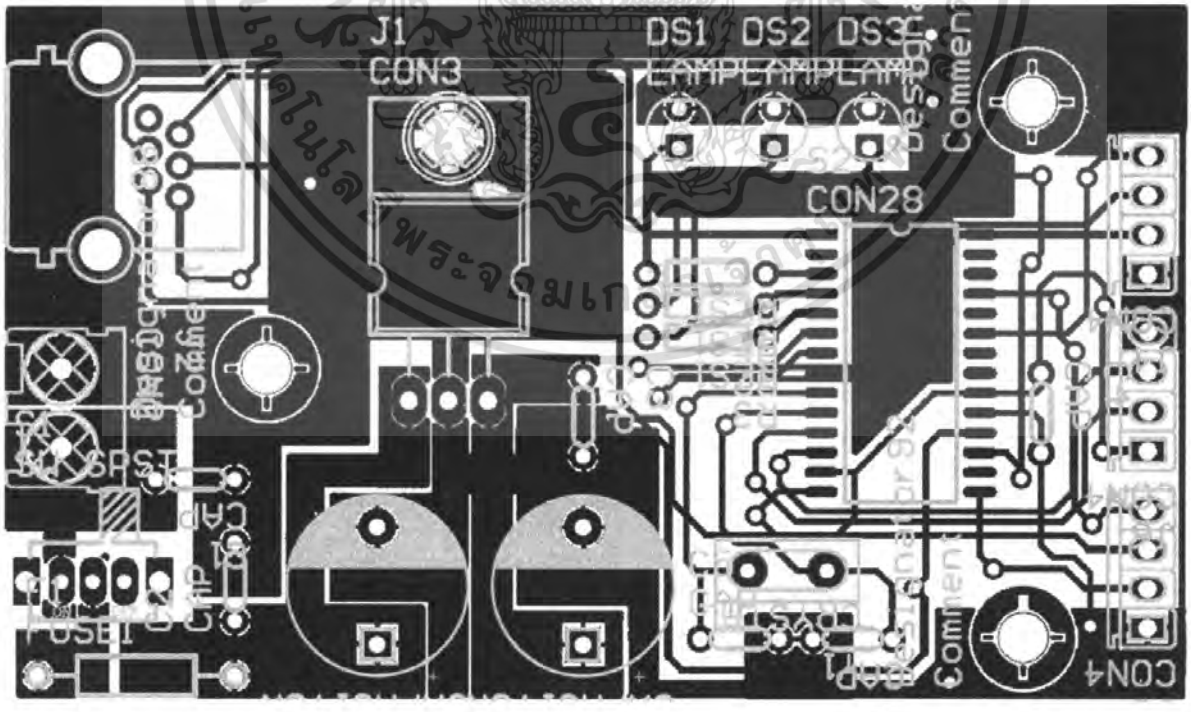




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

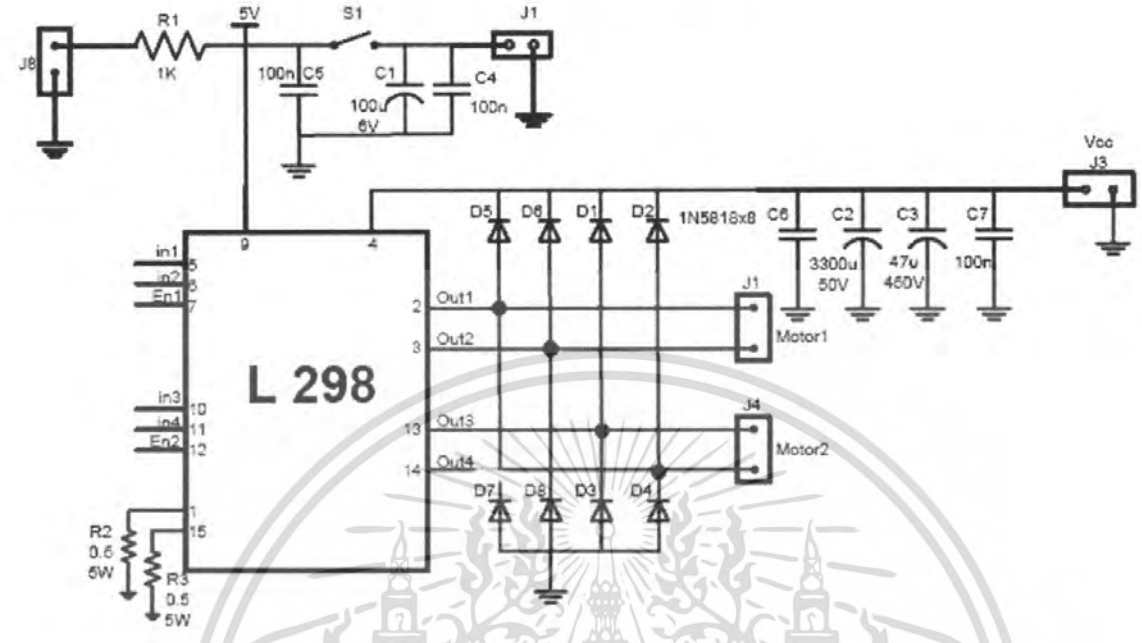
ภาครับ

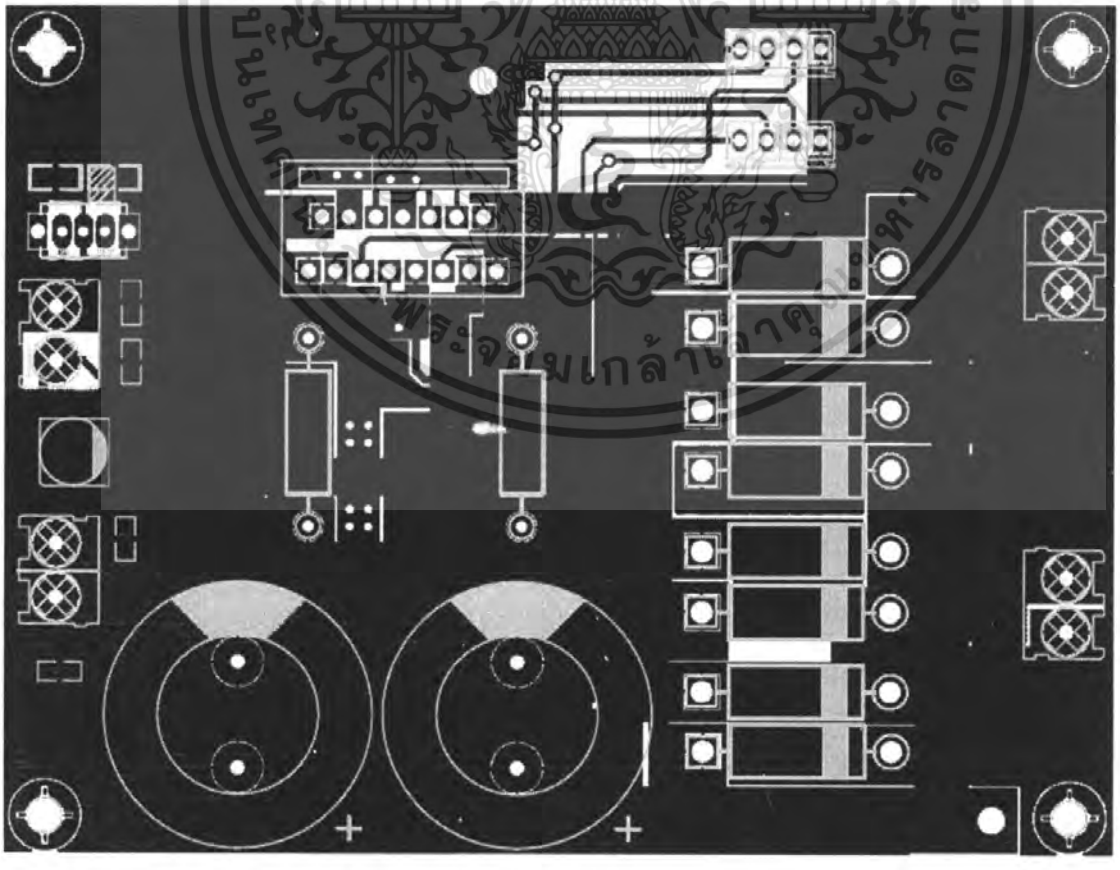




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคขับมอเตอร์





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาคส่ง

```
#include <16F876A.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
#include <string.h>
#include "lcd-uart.c"

int value1=0,value2=0;
int x=0,y=0;
int var1=0,var2=0;
char data=0x00;
////////////////////////////a/d_check//////////////////////////////
void direction (void)
{
setup_port_a( ALL_ANALOG );
setup_adc( ADC_CLOCK_INTERNAL );
set_adc_channel( 0 );
delay_ms(10);
value1 = read_adc();

set_adc_channel( 1 );
delay_ms(10);
value2 = read_adc();

setup_port_a( NO_ANALOGS );
setup_adc( ADC_OFF );
}

#INT_TBE
void isr_tx(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
putc(0xb00110011);// ID byte
putc(data1);     // WH1
putc(data2);     // WH2
putc(data3);     // OPTION
}
////////////////////////////////////////////////////////////////
void main(void)
{
set_tris_C(0b00000000);
set_tris_A(0b00000011);
set_tris_B(0x00);

enable_interrupts(GLOBAL);
enable_interrupts(INT_TBE);

  while(TRUE)
  {
    set_tris_A(0b00000011);
    delay_ms(10);
    direction();
    data_confirm();

    set_tris_A(0b00000000);
    delay_ms(10);
    lcd_init ();

    printf (lcd_putc,"\f %u   %u",value1,value2);

    output_high(PIN_C2);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาครับ

```
#include <p30f4012.h>   // Header file for dsPIC30F2010
#include <pwm.h>        // Module function for PWM
#include<uart.h>      // Module function for uart
#define PERIOD 0xFFFF      // Define Period
#define POW 50   // Fix power 60%



//-------------------------------------------------------------------------------------------------------//
//------------------------------------ Function delay ---------------------------------------------------//
//-------------------------------------------------------------------------------------------------------//
void delay(unsigned int ms)
{
  unsigned int x,a;      // Keep for counter loop
  for(x=0;x<ms;x++)
  {
    for(a=0;a<816;a++);   // Loop for delay 1 millisec per unit
  }
}
//-------------------------------------------------------------------------------------------------------//
//------------------------------------ Interrupt service routine for PWM--------------------------------//
//-------------------------------------------------------------------------------------------------------//
void _ISR _PWMInterrupt(void)
{
  IFS2bits.PWMIF = 0;   // Clear PWM interrupt flag
}


void _ISR _U1TXInterrupt(void)
{
  IFS0bits.U1TXIF = 0;   // Clear TX interrupt flag
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-------------------------------------------------------------------------------------------------------------//
//---------------------------- Interrupt service routine RX UART1 ------------------------------------//
//-------------------------------------------------------------------------------------------------------------//
void _ISR _U1RXInterrupt(void)
{
  IFS0bits.U1RXIF = 0;  // Clear RX interrupt flag
}

//-------------------------------------------------------------------------------------------------------------//
//---------------------------------------- Function initial PWM ---------------------------------------//
//-------------------------------------------------------------------------------------------------------------//
void pwm_init()
{
  unsigned int config;  // Holds the PWM interrupt configuration value
  unsigned int period;  // Holds the value to be loaded into dutycycle register
  unsigned int sptime;  // Holds the value to be loaded into special event compare register
  unsigned int config1;  // Holds PWM configuration value
  unsigned int config2;  // Holds the value be loaded into PWMCON1 register
  unsigned int config3;  // Holds the value to configure the special event trigger
            // postscale and dutycycle
  unsigned int dutycyclereg;  // The value of 'dutycyclereg' determines the duty cycle
  unsigned int dutycycle;     // register(PDCx) to be written
  unsigned char updatedisable;  // Configure pwm interrupt enable/disable and set interrupt
// priorties
  _TRISD0 = 0;
  _TRISD1 = 0;
  _TRISB4 = 0;
  _TRISB5 = 0;
    // Configuration for bit control direction DC Motor

  //------------------------------ Configuration for FLTA control ----------------------------------//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ConfigIntMCPWM(PWM_INT_EN &      // Enable interrupt for PWM
        PWM_FLTA_DIS_INT &      // Enable interrupt for FLTA
        PWM_INT_PR7);       // FLTA interrupt priority level 7

  SetMCPWMFaultA(PWM_OVA1H_ACTIVE &       // FLTA override 1H active
         PWM_OVA2H_ACTIVE &     // FLTA override 2H active
        PWM_FLTA_MODE_LATCH&   // FLTA latch mode
        PWM_FLTA1_DIS&          // Enable FLTA CH1
        PWM_FLTA2_DIS);         // Enable FLTA CH1
  sptime = 0x0;

  config1   =   (PWM_EN &          // Enable PWM module
       PWM_OP_SCALE1   &      // Output post scaler select 1:1
       PWM_IPCLK_SCALE1 &     // Input prescaler select 1:1
       PWM_MOD_DBL);       // PWM double update mode

  config2   =   (PWM_MOD1_COMP &      // 1th channel in complimentary mode
       PWM_MOD2_COMP &     // 2th channel in complimentary mode
       PWM_PEN2H &         // H of channel 2 works as PWM
       PWM_PEN1H  &        // H of channel 1 works as PWM
       PWM_PEN2L &         // L of channel 2 works as PWM
       PWM_PEN1L);       // L of channel 1 works as PWM

  config3   =   (PWM_SEVOPS1 &        // Special event post scaler 1:1
       PWM_OSYNC_PWM&       // over ride syncronised with PWM clk
       PWM_UEN);         // Update of PDCs and PTPER enabled

  OpenMCPWM(PERIOD,sptime,config1,config2,config3);   // Setup parameter for PWM
module

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-------------------------------------------------------------------------------------------------------//
//------------------------------ Function drive DC motor Forward ----------------------------------------//
//-------------------------------------------------------------------------------------------------------//
void forward(char chanel,char pow)
{
  unsigned int dutycycle,updatedisable;

  dutycycle = 100 - (pow*(PERIOD/100)) ;   // Keep dutycycle

  updatedisable = 0;      // Disable update

  if(chanel==1)
  {
    LATDbits.LATD0 = 1;    // Bit1 control direction drive motor forward for module 1
    LATDbits.LATD1 = 0;   // Bit2 control direction drive motor forward for module 1
  }
  else if(chanel==2)
  {
    LATBbits.LATB4 = 1;    // Bit1 control direction drive motor forward for module 2
    LATBbits.LATB5 = 0;   // Bit2 control direction drive motor forward for module 2
  }

  SetDCMCPWM(chanel,dutycycle,updatedisable);
}
//-------------------------------------------------  -------------------------------------------------------//
//------------------------------ Function drive DC motor Backward ---------------------------------------//
//-------------------------------------------------------------------------------------------------------//
void backward(char chanel,char pow)
{
  unsigned int dutycycle,updatedisable;
  dutycycle = 100 - (pow*(PERIOD/100)) ;   // Keep dutycycle
  updatedisable = 0;   // Disable update
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(chanel==1)
  {
    LATDbits.LATD0 = 0;   // Bit1 control direction drive motor backward for module 1
    LATDbits.LATD1 = 1;   // Bit2 control direction drive motor backward for module 1
  }
  else if(chanel==2)
  {
    LATBbits.LATB4 = 0;   // Bit1 control direction drive motor backward for module 2
    LATBbits.LATB5 = 1;   // Bit2 control direction drive motor backward for module 2
  }

  SetDCMCPWM(chanel,dutycycle,updatedisable);
}
//-------------------------------------------------------------------------------------------------//
//------------------------------------ Function stop motor ----------------------------------------//
//-------------------------------------------------------------------------------------------------//
void stop_motor(char chanel)
{
  if(chanel==1)
  {
    LATDbits.LATD0 = 0;   // Bit1 control stop motor  for module 1
    LATDbits.LATD1 = 0;   // Bit2 control stop motor  for module 1
  }
  else if(chanel==2)
  {
    LATBbits.LATB4 = 0;   // Bit1 control stop motor  for module 2
    LATBbits.LATB5 = 0;   // Bit2 control stop motor  for module 2
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-------------------------------------------------------------------------------------------------------------//
//--------------------------------- Function for configuration UART1 ----------------------------------//
//-------------------------------------------------------------------------------------------------------------//
void uart1_init()
{
  unsigned int baudvalue;     // Keep baud rate value for Load into U1BRG
  unsigned int U1MODEvalue;   // Keep value for Load into U1MODE
  unsigned int U1STAvalue;    // Keep value for Load into U1STA

  CloseUART1();       // Disable UART1
  ConfigIntUART1(UART_RX_INT_EN &        // Enable RX interrupt UART1
         UART_RX_INT_PR6 &     // Set RX interrupt Priority ==>6
         UART_TX_INT_EN &      // Enable TX interrupt UART1
         UART_TX_INT_PR2);     // Set RX interrupt Priority ==>2

  baudvalue = 47;  // Baud rate 9600 bps

  U1MODEvalue =   UART_EN &          // Enable UART1
         UART_IDLE_CON &     // UART1 working in idle mode
         UART_RX_TX&         // UART1 normal pin(TX==>RF3 pin,RX==>RF2 pin)
         UART_DIS_WAKE &     // Disable wake-up on start UART
         UART_DIS_LOOPBACK & // Disable loop back mode
         UART_DIS_ABAUD &    // Disable autobaud mode
         UART_NO_PAR_8BIT &  // Set data 8 bit ,no parity bit
         UART_1STOPBIT;      // Set 1 stop bit
  U1STAvalue  =  UART_INT_TX_BUF_EMPTY &  // Interrupt on buffer empty mode
         UART_TX_PIN_NORMAL &    // TX Break bit normal
         UART_TX_ENABLE &        // Enable TX for UART
         UART_INT_RX_3_4_FUL&    // UART interrupt receive mode
         UART_ADR_DETECT_DIS &   // Disable detect address mode
         UART_RX_OVERRUN_CLEAR;  // Reset buffer over run
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  OpenUART1(U1MODEvalue, U1STAvalue, baudvalue);   // Execute configuration for UART1
}

//-------------------------------------------------------------------------------------------------------------//
//---------------------------------------- Main Program ---------------------------------------------//
//-------------------------------------------------------------------------------------------------------------//

int main()
{
  TRISBbits.TRISB1 = 0;
  TRISBbits.TRISB2 = 0;
  char Txdata[] = "\r\nUart test input key...\r\n";   // Table character
  char  dat,data_rx;      // Buffer for receive character
  int buffer_count=0;
  int direct=0;
  uart1_init();     // Initial UART1
  pwm_init();

  while(1)        // Infinite loop
  {

    if(DataRdyUART1())   //   Check data receive
    {
      dat = ReadUART1();   // Load data into buffer variable
      if((dat != (0xF0)) || (dat != (0x0F)) )
        {

////////////////////////////////forward/////////////////////////////////////

          if(dat == (0b00111111))
          {
            forward(1,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  forward(2,1);
  _LATB1 = 1;
  _LATB2 = 0;
 }

 else if(dat == (0b00110110))
 {
  forward(1,40);
  forward(2,40);
  _LATB1 = 1;
  _LATB2 = 0;
 }

 else if(dat == (0b00101101))
 {
  forward(1,80);
  forward(2,80);
  _LATB1 = 1;
  _LATB2 = 0;
 }
////////////////////////////////////////////////////////////////////

////////////////////////////////backward/////////////////////////////////
 else if(dat == (0b00011011))
 {
  backward(1,1);
  backward(2,1);
  _LATB2 = 1;
  _LATB1 = 0;
 }
 else if(dat == (0b00010010))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
          backward(1,40);
          backward(2,40);
          _LATB2 = 1;
          _LATB1 = 0;
        }

        else if(dat == (0b00001001))
        {
          backward(1,80);
          backward(2,80);
          _LATB2 = 1;
          _LATB1 = 0;
        }

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

////////////////////////////left/////////////////////////////////////
        else if(dat == (0b00111011))
        {
          forward(1,1);
          backward(2,1);
          _LATB2 = 1;
          _LATB1 = 0;
        }

        else if(dat == (0b00110010))
        {
          forward(1,40);
          backward(2,40);
          _LATB2 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _LATB1 = 0;
  }

  else if(dat == (0b00101001))
  {
    forward(1,80);
    backward(2,80);
    _LATB2 = 1;
    _LATB1 = 0;
}
////////////////////////////////////////////////////////////////////

////////////////////////////////right/////////////////////////////////
  else if(dat == (0b00011111))
  {
    backward(1,1);
    forward(2,1);
    _LATB2 = 0;
    _LATB1 = 1;
  }

  else if(dat == (0b00010110))
  {
    backward(1,40);
    forward(2,40);
    _LATB2 = 0;
    _LATB1 = 1;
  }

  else if(dat == (0b00001101))
  {
    backward(1,80);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            forward(2,80);
            _LATB2 = 0;
            _LATB1 = 1;
          }

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

/////////////////////////////stop/////////////////////////////////////////
      else
          {
            stop_motor(1);
            stop_motor(2);
            _LATB1 = 0;
            _LATB2 = 0;
          }

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

/////////////////////////////////END/////////////////////////////////////
        }

    else
    {
      _LATB1 = 0;
    }


  }


  }
  return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F4011/4012

dsPIC30F4011/4012 Enhanced Flash 16-Bit Digital Signal Controller

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *"dsPIC30F Family Reference Manual"* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

High-Performance, Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 48 Kbytes on-chip Flash program space (16K instruction words)
- 2 Kbytes of on-chip data RAM
- 1 Kbyte of nonvolatile data EEPROM
- Up to 30 MIPS operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 30 interrupt sources:
  - 3 external interrupt sources
  - 8 user-selectable priority levels for each interrupt source
  - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
- ±16-bit, single-cycle shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
  - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I^2C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 1 CAN module, 2.0B compliant

Motor Control PWM Module Features:

- 6 PWM output channels:
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 3 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-Bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analog Features:

- 10-Bit Analog-to-Digital Converter (A/D) with 4 S/H inputs:
  - 1 Msps conversion rate
  - 9 input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

Special Digital Signal Controller Features:

- Enhanced Flash program memory:
  - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
  - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

Special Digital Signal Controller Features (Cont.):

- Flexible Watchdog Timer (WDT) with on-chip, low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation detects clock failure and switches to on-chip, low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
  - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption

dsPIC30F Motor Control and Power Conversion Family*

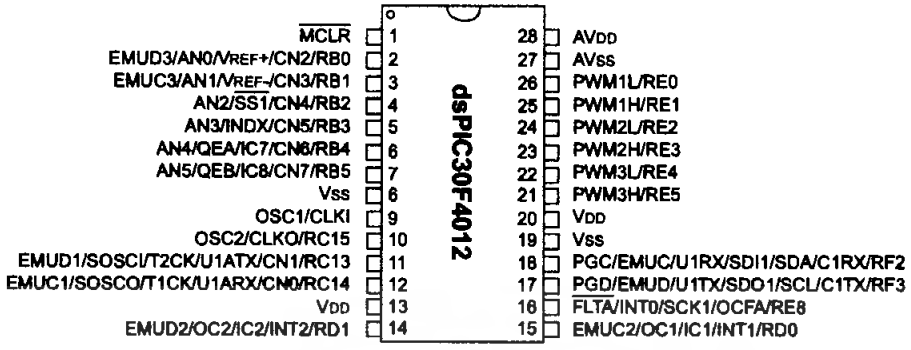
Device	Pins	Program Mem. Bytes/ Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	10-Bit A/D 1 Msps	Quad Enc	UART	SPI	I²C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

* This table provides a summary of the dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

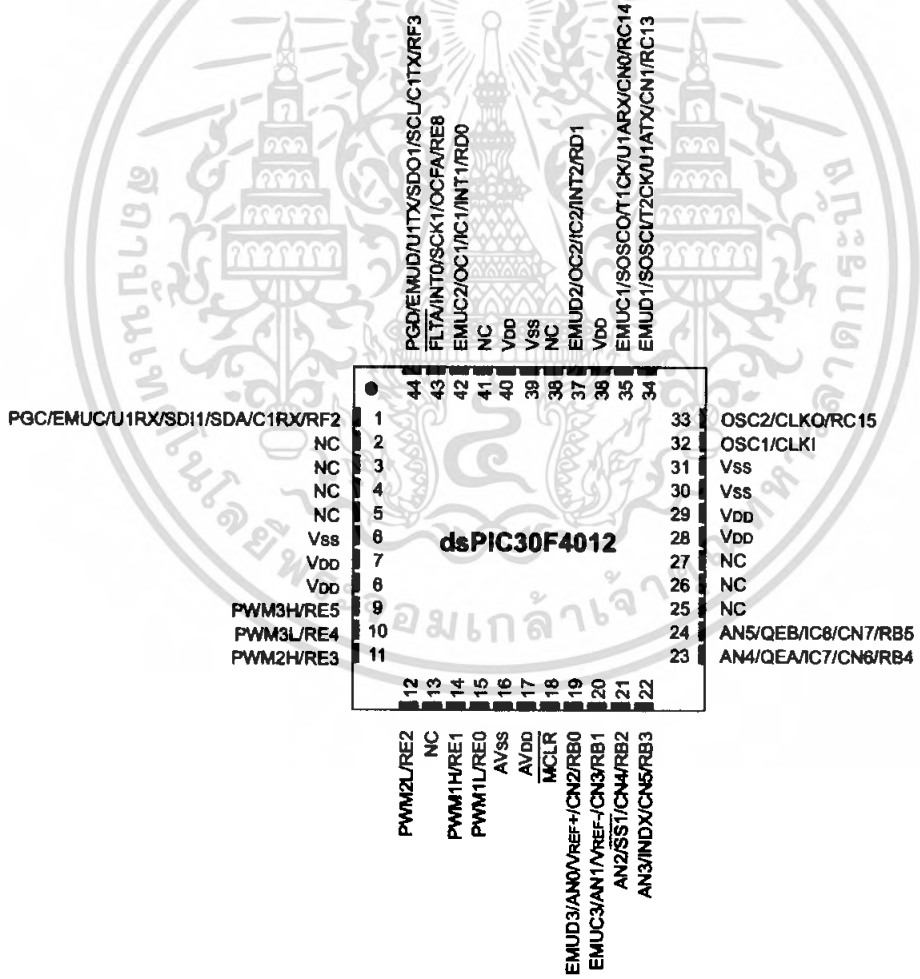
© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagrams (Continued)

28-Pin SPDIP and SOIC



44-Pin QFN



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.0 TIMER1 MODULE

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

This section describes the 16-bit general purpose Timer1 module and associated operational modes. Figure 9-1 depicts the simplified block diagram of the 16-bit Timer1 module.

Note: Timer1 is a 'Type A' timer. Please refer to the specifications for a Type A timer in Section 24.0 "Electrical Characteristics" of this document.

The following sections provide a detailed description, including setup and control registers, along with associated block diagrams for the operational modes of the timers.

The Timer1 module is a 16-bit timer which can serve as the time counter for the Real-Time Clock, or operate as a free-running, interval timer/counter. The 16-bit timer has the following modes:

- 16-bit Timer
- 16-bit Synchronous Counter
- 16-bit Asynchronous Counter

Further, the following operational characteristics are supported:

- Timer gate operation
- Selectable prescaler settings
- Timer operation during CPU Idle and Sleep modes
- Interrupt on 16-bit Period register match or falling edge of external gate signal

These operating modes are determined by setting the appropriate bit(s) in the 16-bit SFR, T1CON. Figure 9-1 presents a block diagram of the 16-bit timer module.

16-bit Timer Mode: In the 16-bit Timer mode, the timer increments on every instruction cycle up to a match value, preloaded into the Period register, PR1, then resets to 0 and continues to count.

When the CPU goes into the Idle mode, the timer will stop incrementing unless the TSIDL (T1CON<13>) bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

16-bit Synchronous Counter Mode: In the 16-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal, which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in PR1, then resets to 0 and continues.

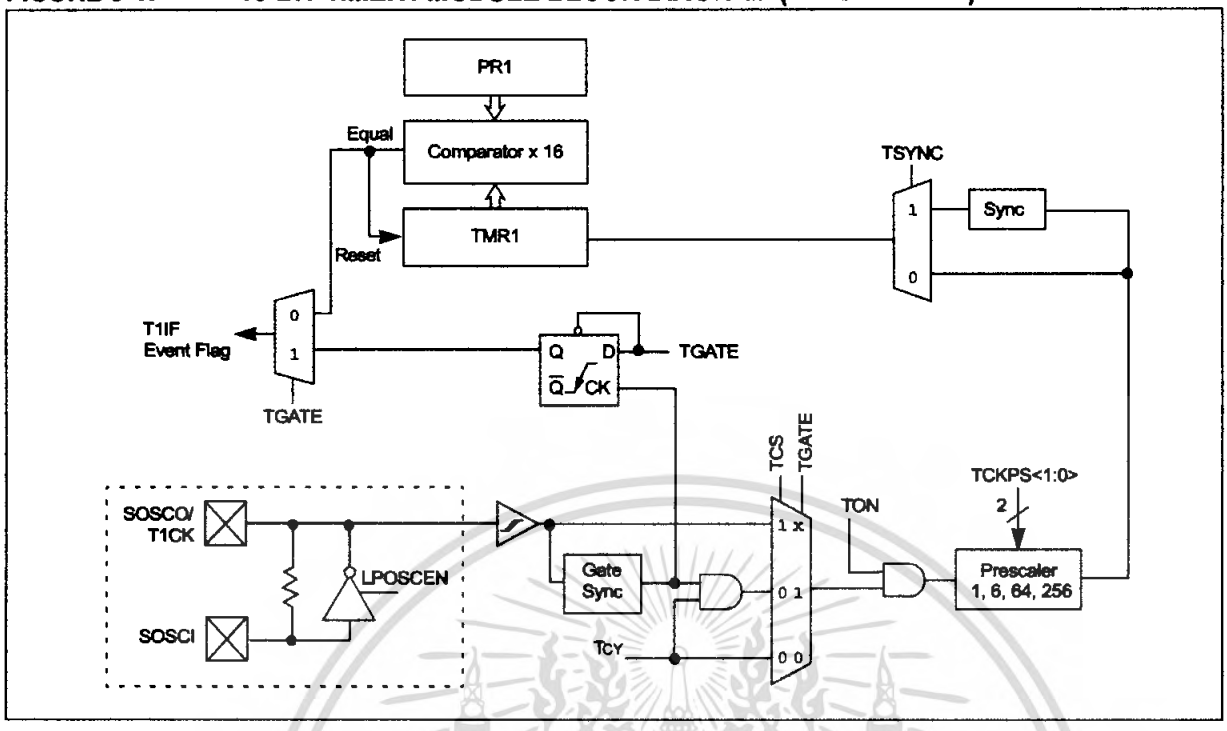
When the CPU goes into the Idle mode, the timer will stop incrementing unless the respective TSIDL bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

16-bit Asynchronous Counter Mode: In the 16-bit Asynchronous Counter mode, the timer increments on every rising edge of the applied external clock signal. The timer counts up to a match value preloaded in PR1, then resets to 0 and continues.

When the timer is configured for the Asynchronous mode of operation, and the CPU goes into the Idle mode, the timer will stop incrementing if TSIDL = 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 9-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM (TYPE A TIMER)



9.1 Timer Gate Operation

The 16-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal TCY to increment the respective timer when the gate input signal (T1CK pin) is asserted high. Control bit, TGATE (T1CON<6>), must be set to enable this mode. The timer must be enabled (TON = 1) and the timer clock source set to internal (TCS = 0).

When the CPU goes into the Idle mode, the timer will stop incrementing unless TSIDL = 0. If TSIDL = 1, the timer will resume the incrementing sequence upon termination of the CPU Idle mode.

9.2 Timer Prescaler

The input clock (FOSC/4 or external clock) to the 16-bit Timer has a prescale option of 1:1, 1:8, 1:64 and 1:256 selected by control bits, TCKPS<1:0> (T1CON<5:4>). The prescaler counter is cleared when any of the following occurs:

- a write to the TMR1 register
- clearing of the TON bit (T1CON<15>)
- device Reset, such as POR and BOR

However, if the timer is disabled (TON = 0), then the timer prescaler cannot be reset since the prescaler clock is halted.

TMR1 is not cleared when T1CON is written. It is cleared by writing to the TMR1 register.

9.3 Timer Operation During Sleep Mode

During CPU Sleep mode, the timer will operate if:

- The timer module is enabled (TON = 1) and
- The timer clock source is selected as external (TCS = 1) and
- The TSYNC bit (T1CON<2>) is asserted to a logic '0', which defines the external clock source as asynchronous

When all three conditions are true, the timer will continue to count up to the Period register and be reset to 0x0000.

When a match between the timer and the Period register occurs, an interrupt can be generated if the respective timer interrupt enable bit is asserted.

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4 Timer Interrupt

The 16-bit timer has the ability to generate an interrupt on period match. When the timer count matches the Period register, the T1IF bit is asserted and an interrupt will be generated, if enabled. The T1IF bit must be cleared in software. The Timer Interrupt Flag, T1IF, is located in the IFS0 control register in the interrupt controller.

When the Gated Time Accumulation mode is enabled, an interrupt will also be generated on the falling edge of the gate signal (at the end of the accumulation cycle).

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T1IE. The timer interrupt enable bit is located in the IEC0 Control register in the interrupt controller.

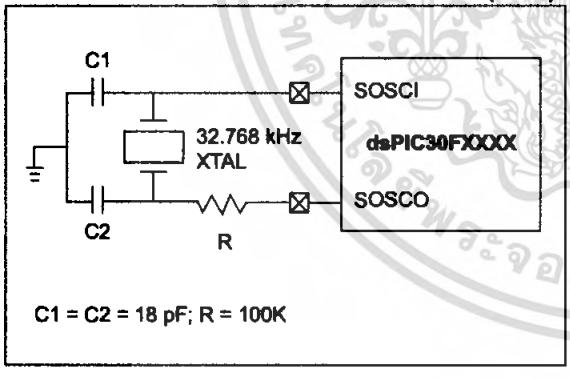
9.5 Real-Time Clock

Timer1, when operating in Real-Time Clock (RTC) mode, provides time-of-day and event time-stamping capabilities. Key operational features of the RTC are:

- Operation from 32 kHz LP oscillator
- 8-bit prescaler
- Low power
- Real-Time Clock interrupts

These operating modes are determined by setting the appropriate bit(s) in the T1CON Control register

FIGURE 9-2: RECOMMENDED COMPONENTS FOR TIMER1 LP OSCILLATOR REAL-TIME CLOCK (RTC)



9.5.1 RTC OSCILLATOR OPERATION

When TON = 1, TCS = 1 and TGATE = 0, the timer increments on the rising edge of the 32 kHz LP oscillator output signal, up to the value specified in the Period register, and is then reset to '0'.

The TSYNC bit must be asserted to a logic '0' (Asynchronous mode) for correct operation.

Enabling LPOSCEN (OSCCON<1>) will disable the normal Timer and Counter modes and enable a timer carry-out wake-up event.

When the CPU enters Sleep mode, the RTC will continue to operate, provided the 32 kHz external crystal oscillator is active and the control bits have not been changed. The TSIDL bit should be cleared to '0' in order for RTC to continue operation in Idle mode.

9.5.2 RTC INTERRUPTS

When an interrupt event occurs, the respective Timer Interrupt Flag, T1IF, is asserted and an interrupt will be generated, if enabled. The T1IF bit must be cleared in software. The respective Timer Interrupt Flag, T1IF, is located in the IFS0 Status register in the interrupt controller.

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T1IE. The Timer Interrupt Enable bit is located in the IEC0 control register in the interrupt controller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 9-1: TIMER1 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TMR1	0100	Timer1 Register																uuuu uuuu uuuu uuuu
PR1	0102	Period Register 1																1111 1111 1111 1111
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit

Note: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

© 2007 Microchip Technology Inc.

10.0 TIMER2/3 MODULE

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

This section describes the 32-bit general purpose timer module (Timer2/3) and associated operational modes. Figure 10-1 depicts the simplified block diagram of the 32-bit Timer2/3 module. Figure 10-2 and Figure 10-3 show Timer2/3 configured as two independent 16-bit timers, Timer2 and Timer3, respectively.

Note: Timer2 is a 'Type B' timer and Timer3 is a 'Type C' timer. Please refer to the appropriate timer type in **Section 24.0 "Electrical Characteristics"** of this document.

The Timer2/3 module is a 32-bit timer, which can be configured as two 16-bit timers, with selectable operating modes. These timers are utilized by other peripheral modules, such as:

- Input Capture
- Output Compare/Simple PWM

The following sections provide a detailed description, including setup and control registers, along with associated block diagrams for the operational modes of the timers.

The 32-bit timer has the following modes:

- Two independent 16-bit timers (Timer2 and Timer3) with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit timer operation
- Single 32-bit synchronous counter

Further, the following operational characteristics are supportad:

- ADC Event Trigger
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During Idle and Sleep Modes
- Interrupt on a 32-bit Period Register Match

These operating modes are determined by setting the appropriate bit(s) in the 16-bit T2CON and T3CON SFRs.

For 32-bit timer/counter operation, Timer2 is the least significant word and Timer3 is the most significant word of the 32-bit timer.

Note: For 32-bit timer operation, T3CON control bits are ignored. Only T2CON control bits are used for setup and control. Timer2 clock and gate inputs are utilized for the 32-bit timer module, but an interrupt is generated with the Timer3 Interrupt Flag (T3IF) and the interrupt is enabled with the Timer3 Interrupt Enable bit (T3IE).

16-bit Mode: In the 16-bit mode, Timer2 and Timer3 can be configured as two independent 16-bit timers. Each timer can be set up in either 16-bit Timer mode or 16-bit Synchronous Counter mode. See **Section 9.0 "Timer1 Module"**, Timer1 Module, for details on these two operating modes.

The only functional difference between Timer2 and Timer3 is that Timer2 provides synchronization of the clock prescaler output. This is useful for high-frequency external clock inputs.

32-bit Timer Mode: In the 32-bit Timer mode, the timer increments on every instruction cycle up to a match value, preloaded into the combined 32-bit Period register, PR3/PR2, then resets to 0 and continues to count.

For synchronous 32-bit reads of the Timer2/Timer3 pair, reading the least significant word (TMR2 register) will cause the msw to be read and latched into a 16-bit holding register, termed TMR3HLD.

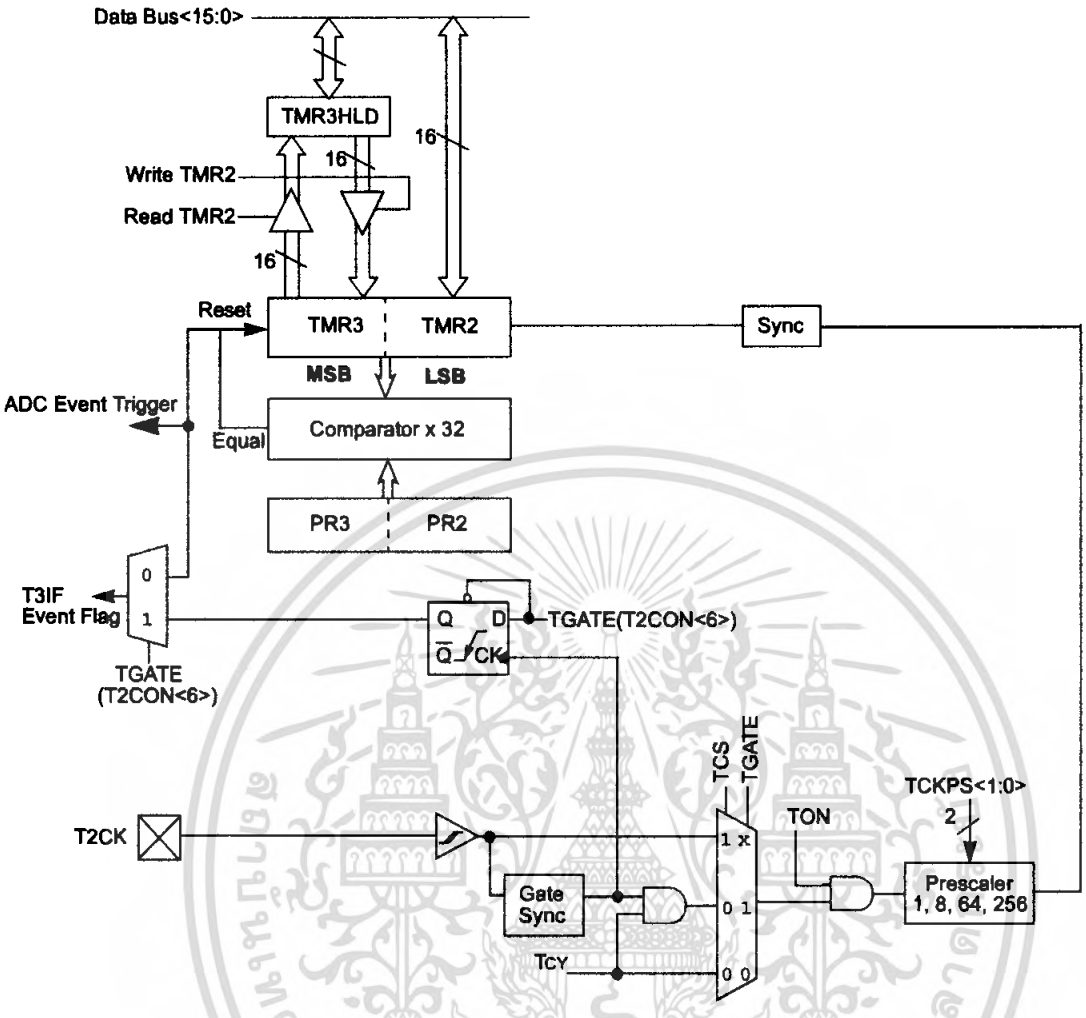
For synchronous 32-bit writes, the holding register (TMR3HLD) must first be written to. When followed by a write to the TMR2 register, the contents of TMR3HLD will be transferred and latched into the MSB of the 32-bit timer (TMR3).

32-bit Synchronous Counter Mode: In the 32-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal, which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in the combined 32-bit period register, PR3/PR2, then resets to 0 and continues.

When the timer is configured for the Synchronous Counter mode of operation and the CPU goes into the Idle mode, the timer will stop incrementing unless the TSIDL (T2CON<13>) bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 10-1: 32-BIT TIMER2/3 BLOCK DIAGRAM



Note: Timer configuration bit, T32 (T2CON<3>), must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 10-2: 16-BIT TIMER2 BLOCK DIAGRAM

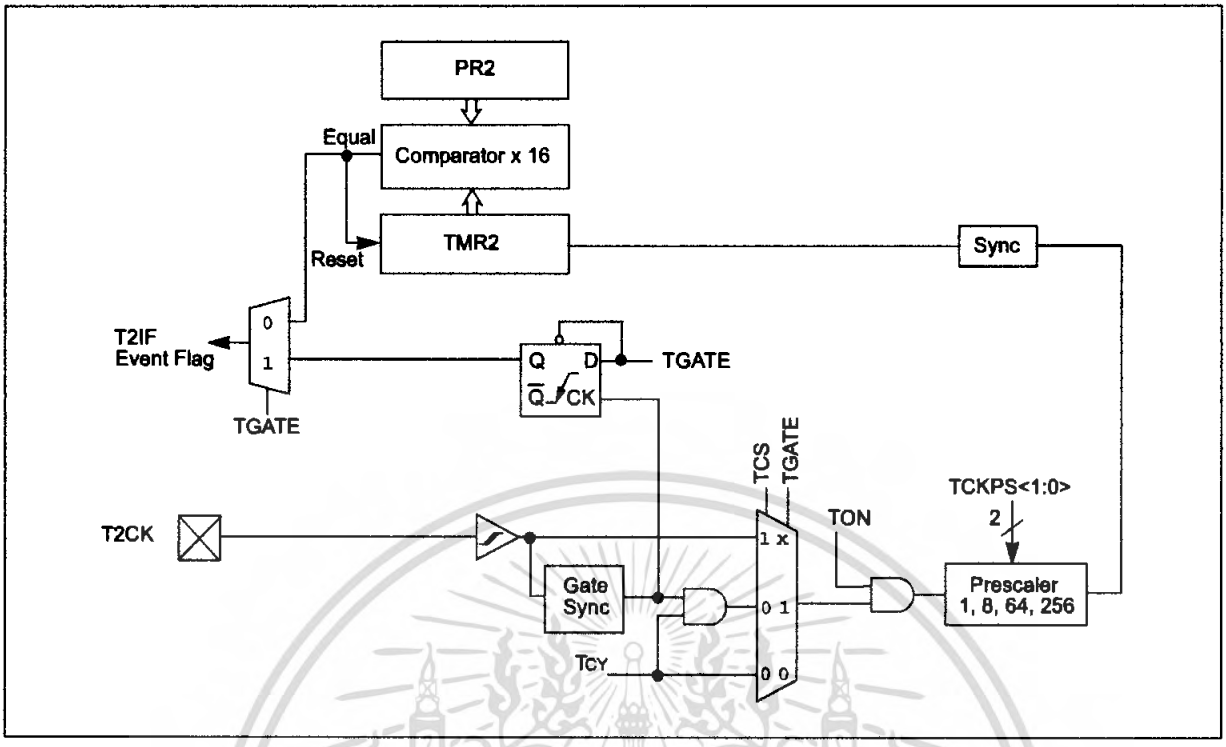
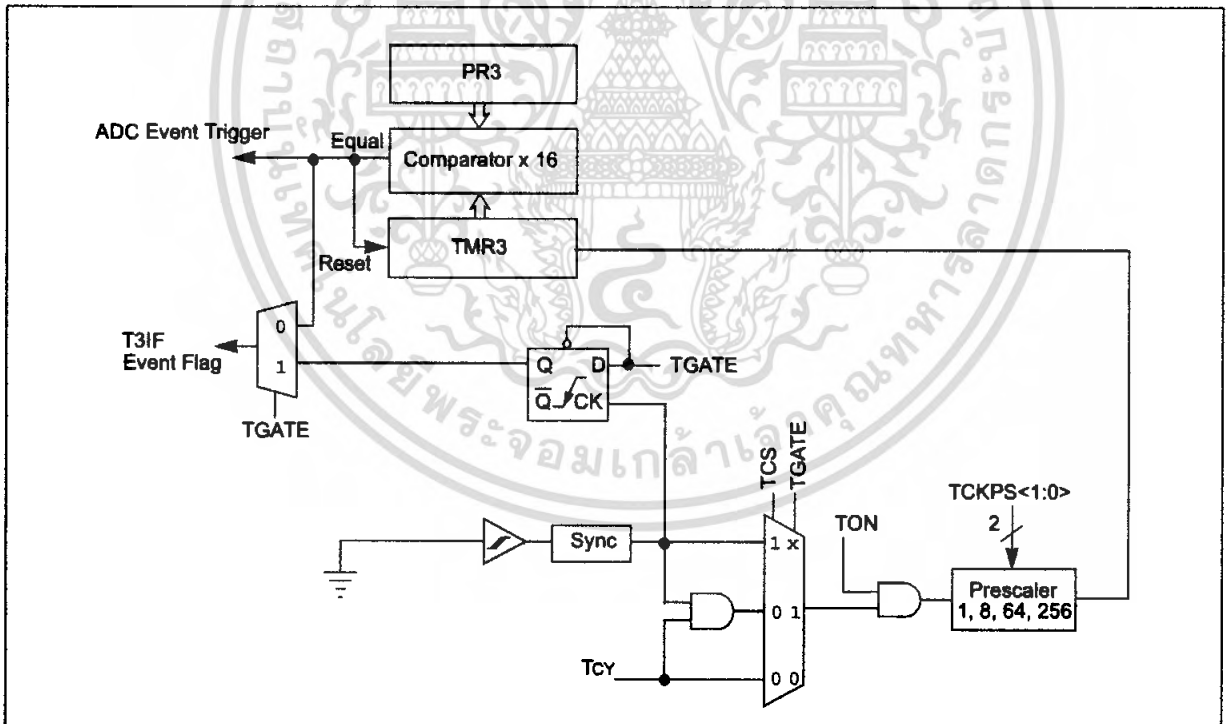


FIGURE 10-3: 16-BIT TIMER3 BLOCK DIAGRAM



Note: The dsPIC30F4011/4012 devices do not have external pin inputs to Timer3. In these devices, the following modes should not be used:
1. TCS = 1.
2. TCS = 0 and TGATE = 1 (gated time accumulation).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.1 Timer Gate Operation

The 32-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal TCY to increment the respective timer when the gate input signal (T2CK pin) is asserted high. Control bit, TGATE (T2CON<6>), must be set to enable this mode. When in this mode, Timer2 is the originating clock source. The TGATE setting is ignored for Timer3. The timer must be enabled (TON = 1) and the timer clock source set to internal (TCS = 0).

The falling edge of the external signal terminates the count operation but does not reset the timer. The user must reset the timer in order to start counting from zero.

10.2 ADC Event Trigger

When a match occurs between the 32-bit timer (TMR3/TMR2) and the 32-bit combined Period register (PR3/PR2), a special ADC trigger event signal is generated by Timer3.

10.3 Timer Prescaler

The input clock (FOSC/4 or external clock) to the timer has a prescale option of 1:1, 1:8, 1:64 and 1:256, selected by control bits, TCKPS<1:0> (T2CON<5:4> and T3CON<5:4>). For the 32-bit timer operation, the originating clock source is Timer2. The prescaler operation for Timer3 is not applicable in this mode. The prescaler counter is cleared when any of the following occurs:

- a write to the TMR2/TMR3 register
- clearing either of the TON (T2CON<15> or T3CON<15>) bits to '0'
- device Reset, such as POR and BOR

However, if the timer is disabled (TON = 0), then the Timer 2 prescaler cannot be reset, since the prescaler clock is halted.

TMR2/TMR3 is not cleared when T2CON/T3CON is written.

10.4 Timer Operation During Sleep Mode

During CPU Sleep mode, the timer will not operate because the internal clocks are disabled.

10.5 Timer Interrupt

The 32-bit timer module can generate an interrupt on period match, or on the falling edge of the external gate signal. When the 32-bit timer count matches the respective 32-bit Period register, or the falling edge of the external "gate" signal is detected, the T3IF bit (IFS0<7>) is asserted and an interrupt will be generated, if enabled. In this mode, the T3IF interrupt flag is used as the source of the interrupt. The T3IF bit must be cleared in software.

Enabling an interrupt is accomplished via the respective Timer Interrupt Enable bit, T3IE (IEC0<7>).

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 10-1: TIMER2/3 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TMR2	0106	Timer2 Register																uuuu uuuu uuuu uuuu
TMR3HLD	0108	Timer3 Holding Register (For 32-bit timer operations only)																uuuu uuuu uuuu uuuu
TMR3	010A	Timer3 Register																uuuu uuuu uuuu uuuu
PR2	010C	Period Register 2																1111 1111 1111 1111
PR3	010E	Period Register 3																1111 1111 1111 1111
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit

Note: Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

© 2007 Microchip Technology Inc.

18.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART) MODULE

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

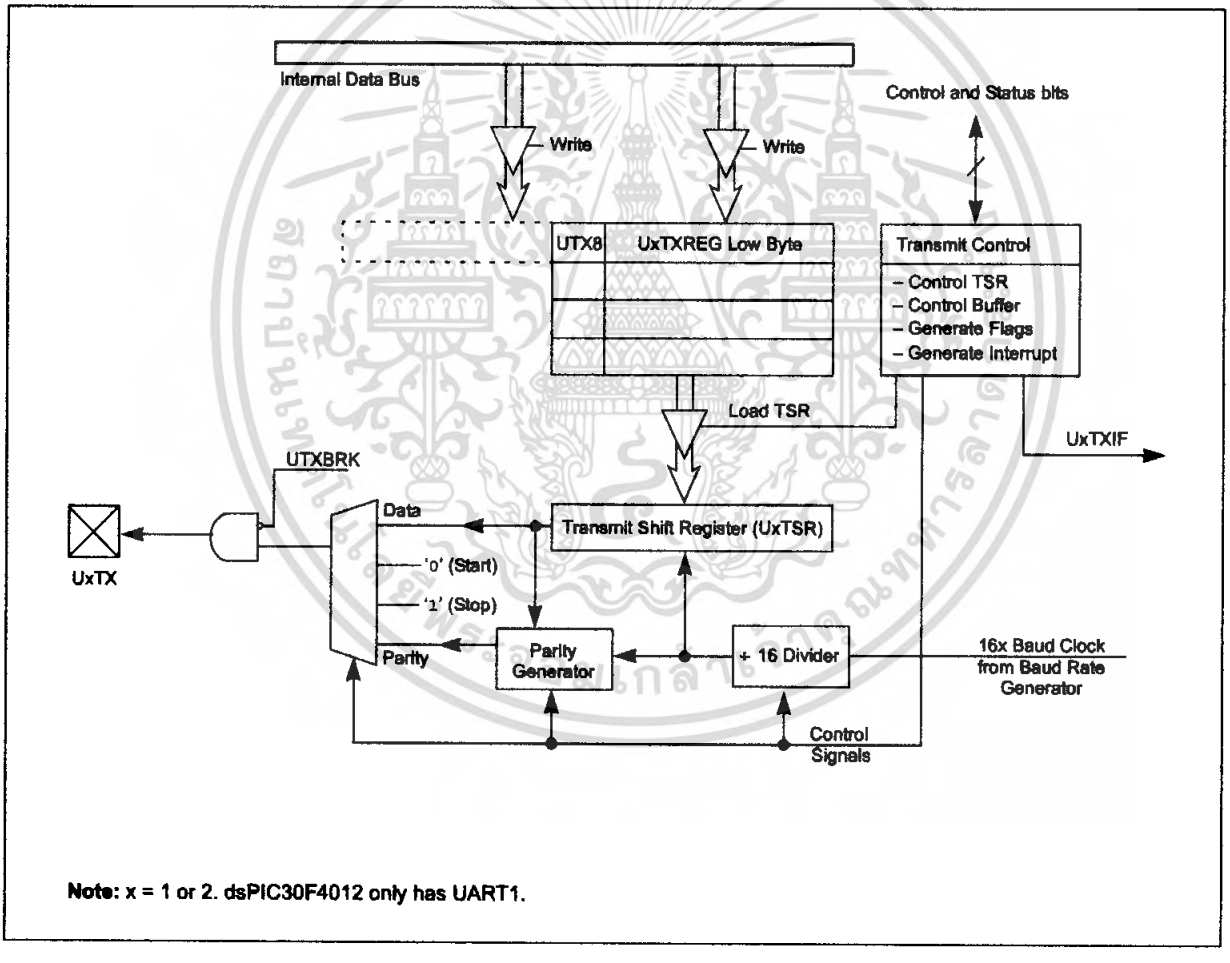
This section describes the Universal Asynchronous Receiver/Transmitter communications module.

18.1 UART Module Overview

The key features of the UART module are:

- Full-Duplex, 8 or 9-bit Data Communication
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Fully Integrated Baud Rate Generator with 16-bit Prescaler
- Baud Rates ranging from 38 bps to 1.875 Mbps at a 30 MHz Instruction Rate
- 4-Word Deep Transmit Data Buffer
- 4-Word Deep Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for Interrupt Only on Address Detect (9th bit = 1)
- Separate Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support

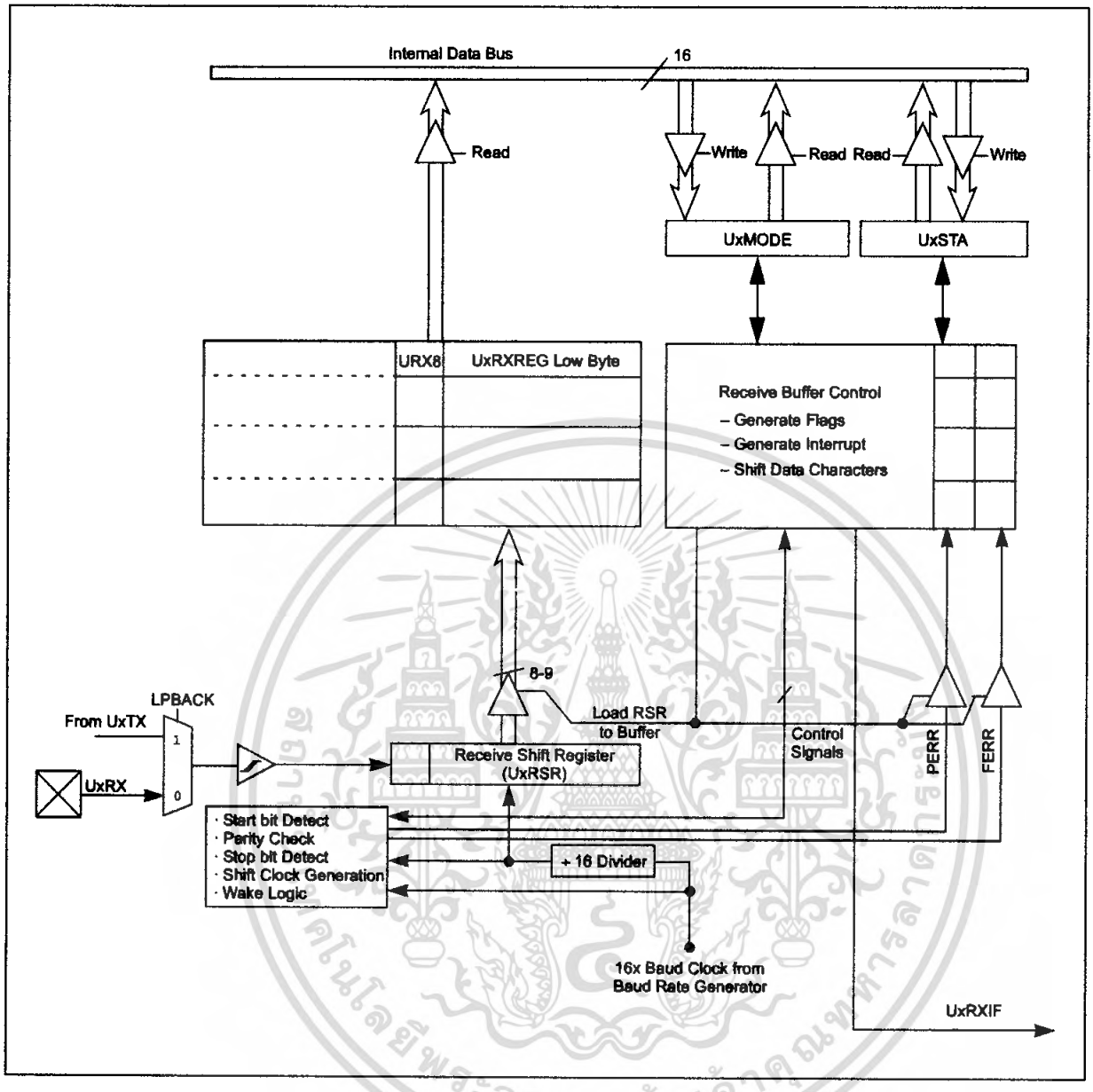
FIGURE 18-1: UART TRANSMITTER BLOCK DIAGRAM



Note: x = 1 or 2. dsPIC30F4012 only has UART1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 18-2: UART RECEIVER BLOCK DIAGRAM



© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18.2 Enabling and Setting Up UART

18.2.1 ENABLING THE UART

The UART module is enabled by setting the UARTEN bit in the UxMODE register (where x = 1 or 2). Once enabled, the UxTX and UxRX pins are configured as an output and an input, respectively, overriding the TRIS and LATCH register bit settings for the corresponding I/O port pins. The UxTX pin is at logic '1' when no transmission is taking place.

18.2.2 DISABLING THE UART

The UART module is disabled by clearing the UARTEN bit in the UxMODE register. This is the default state after any Reset. If the UART is disabled, all I/O pins operate as port pins under the control of the LATCH and TRIS bits of the corresponding port pins.

Disabling the UART module resets the buffers to empty states. Any data characters in the buffers are lost and the baud rate counter is reset.

All error and status flags associated with the UART module are reset when the module is disabled. The URXDA, OERR, FERR, PERR, UTXEN, UTXBRK and UTXBF bits are cleared, whereas RIDLE and TRMT are set. Other control bits, including ADDEN, URXISEL<1:0>, UTXISEL, as well as the UxMODE and UxBRG registers, are not affected.

Clearing the UARTEN bit while the UART is active will abort all pending transmissions and receptions and reset the module as defined above. Re-enabling the UART will restart the UART in the same configuration.

18.2.3 ALTERNATE I/O

The alternate I/O function is enabled by setting the ALTIO bit (UxMODE<10>). If ALTIO = 1, the UxATX and UxARX pins (alternate transmit and alternate receive pins, respectively) are used by the UART module instead of the UxTX and UxRX pins. If ALTIO = 0, the UxTX and UxRX pins are used by the UART module.

18.2.4 SETTING UP DATA, PARITY AND STOP BIT SELECTIONS

Control bits, PDSEL<1:0> in the UxMODE register, are used to select the data length and parity used in the transmission. The data length may either be 8 bits with even, odd or no parity, or 9-bits with no parity.

The STSEL bit determines whether one or two Stop bits will be used during data transmission.

The default (power-on) setting of the UART is 8 bits, no parity and 1 Stop bit (typically represented as 8, N, 1).

18.3 Transmitting Data

18.3.1 TRANSMITTING IN 8-BIT DATA MODE

The following steps must be performed in order to transmit 8-bit data:

1. Set up the UART:
   First, the data length, parity and number of Stop bits must be selected. Then, the transmit and receive interrupt enable and priority bits are set up in the UxMODE and UxSTA registers. Also, the appropriate baud rate value must be written to the UxBRG register.
2. Enable the UART by setting the UARTEN bit (UxMODE<15>).
3. Set the UTXEN bit (UxSTA<10>), thereby enabling a transmission.
4. Write the byte to be transmitted to the lower byte of UxTXREG. The value will be transferred to the Transmit Shift register (UxTSR) immediately and the serial bit stream will start shifting out during the next rising edge of the baud clock. Alternatively, the data byte may be written while UTXEN = 0, following which, the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
5. A transmit interrupt will be generated depending on the value of the Interrupt control bit UTXISEL (UxSTA<15>).

18.3.2 TRANSMITTING IN 9-BIT DATA MODE

The sequence of steps involved in the transmission of 9-bit data is similar to 8-bit transmission, except that a 16-bit data word (of which the upper 7 bits are always clear) must be written to the UxTXREG register.

18.3.3 TRANSMIT BUFFER (UxTXB)

The transmit buffer is 9 bits wide and 4 characters deep. Including the Transmit Shift register (UxTSR), the user effectively has a 5-deep FIFO (First In First Out) buffer. The UTXBF Status bit (UxSTA<9>) indicates whether the transmit buffer is full.

If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO, and no data shift will occur within the buffer. This enables recovery from a buffer overrun condition.

The FIFO is reset during any device Reset, but is not affected when the device enters or wakes up from a power-saving mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18.3.4 TRANSMIT INTERRUPT

The Transmit Interrupt Flag (U1TXIF or U2TXIF) is located in the corresponding interrupt flag register.

The transmitter generates an edge to set the UxTXIF bit. The condition for generating the interrupt depends on UTXISEL control bit:

a) If UTXISEL = 0, an interrupt is generated when a word is transferred from the transmit buffer to the Transmit Shift register (UxTSR). This implies that the transmit buffer has at least one empty word.

b) If UTXISEL = 1, an interrupt is generated when a word is transferred from the transmit buffer to the Transmit Shift register (UxTSR) and the transmit buffer is empty.

Switching between the two interrupt modes during operation is possible and sometimes offers more flexibility.

18.3.5 TRANSMIT BREAK

Setting the UTXBRK bit (UxSTA<11>) will cause the UxTX line to be driven to logic '0'. The UTXBRK bit overrides all transmission activity. Therefore, the user should generally wait for the transmitter to be Idle before setting UTXBRK.

To send a Break character, the UTXBRK bit must be set by software and must remain set for a minimum of 13 baud clock cycles. The UTXBRK bit is then cleared by software to generate Stop bits. The user must wait for a duration of at least one or two baud clock cycles in order to ensure a valid Stop bit(s) before reloading the UxTXB or starting other transmitter activity. Transmission of a Break character does not generate a transmit interrupt.

18.4 Receiving Data

18.4.1 RECEIVING IN 8-BIT OR 9-BIT DATA MODE

The following steps must be performed while receiving 8-bit or 9-bit data:

1. Set up the UART (see **Section 18.3.1 "Transmitting in 8-bit Data Mode"**).
2. Enable the UART (see **Section 18.3.1 "Transmitting in 8-bit Data Mode"**).
3. A receive interrupt will be generated when one or more data words have been received, depending on the receive interrupt settings specified by the URXISEL bits (UxSTA<7:6>).
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read the received data from UxRXREG. The act of reading UxRXREG will move the next word to the top of the receive FIFO and the PERR and FERR values will be updated.

18.4.2 RECEIVE BUFFER (UxRXB)

The receive buffer is 4 words deep. Including the Receive Shift register (UxRSR), the user effectively has a 5-word deep FIFO buffer.

URXDA (UxSTA<0>) = 1 indicates that the receive buffer has data available. URXDA = 0 implies that the buffer is empty. If a user attempts to read an empty buffer, the old values in the buffer will be read and no data shift will occur within the FIFO.

The FIFO is reset during any device Reset. It is not affected when the device enters or wakes up from a power-saving mode.

18.4.3 RECEIVE INTERRUPT

The Receive Interrupt Flag (U1RXIF or U2RXIF) can be read from the corresponding interrupt flag register. The interrupt flag is set by an edge generated by the receiver. The condition for setting the receive interrupt flag depends on the settings specified by the URXISEL<1:0> (UxSTA<7:6>) control bits.

a) If URXISEL<1:0> = 00 or 01, an interrupt is generated every time a data word is transferred from the Receive Shift register (UxRSR) to the receive buffer. There may be one or more characters in the receive buffer.

b) If URXISEL<1:0> = 10, an interrupt is generated when a word is transferred from the Receive Shift register (UxRSR) to the receive buffer which, as a result of the transfer, contains 3 characters.

c) If URXISEL<1:0> = 11, an interrupt is set when a word is transferred from the Receive Shift register (UxRSR) to the receive buffer which, as a result of the transfer, contains 4 characters (i.e., becomes full).

Switching between the interrupt modes during operation is possible, though generally not advisable during normal operation.

18.5 Reception Error Handling

18.5.1 RECEIVE BUFFER OVERRUN ERROR (OERR BIT)

The OERR bit (UxSTA<1>) is set if all of the following conditions occur:

a) The receive buffer is full.

b) The Receive Shift register is full, but unable to transfer the character to the receive buffer.

c) The Stop bit of the character in the UxRSR is detected, indicating that the UxRSR needs to transfer the character to the buffer.

Once OERR is set, no further data is shifted in UxRSR (until the OERR bit is cleared in software or a Reset occurs). The data held in UxRSR and UxRXREG remains valid.

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18.5.2 FRAMING ERROR (FERR)

The FERR bit (UxSTA<2>) is set if a '0' is detected instead of a Stop bit. If two Stop bits are selected, both Stop bits must be '1', otherwise FERR will be set. The read-only FERR bit is buffered along with the received data; it is cleared on any Reset.

18.5.3 PARITY ERROR (PERR)

The PERR bit (UxSTA<3>) is set if the parity of the received word is incorrect. This error bit is applicable only if a Parity mode (odd or even) is selected. The read-only PERR bit is buffered along with the received data bytes; it is cleared on any Reset.

18.5.4 IDLE STATUS

When the receiver is active (i.e., between the initial detection of the Start bit and the completion of the Stop bit), the RIDLE bit (UxSTA<4>) is '0'. Between the completion of the Stop bit and detection of the next Start bit, the RIDLE bit is '1', indicating that the UART is Idle.

18.5.5 RECEIVE BREAK

The receiver will count and expect a certain number of bit times based on the values programmed in the PDSEL<1:0> (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.

If the Break is longer than 13 bit times, the reception is considered complete after the number of bit times specified by PDSEL and STSEL. The URXDA bit is set, FERR is set, zeros are loaded into the receive FIFO, interrupts are generated, if appropriate and the RIDLE bit is set.

When the module receives a long Break signal and the receiver has detected the Start bit, the data bits and the invalid Stop bit (which sets the FERR), the receiver must wait for a valid Stop bit before looking for the next Start bit. It cannot assume that the Break condition on the line is the next Start bit.

Break is regarded as a character containing all '0's, with the FERR bit set. The Break character is loaded into the buffer. No further reception can occur until a Stop bit is received. Note that RIDLE goes high when the Stop bit has not been received yet.

18.6 Address Detect Mode

Setting the ADDEN bit (UxSTA<5>) enables this special mode in which a 9th bit (URX8) value of '1' identifies the received word as an address, rather than data. This mode is only applicable for 9-bit data communication. The URXISELx control bit does not have any impact on interrupt generation in this mode, since an interrupt (if enabled) will be generated every time the received word has the 9th bit set.

18.7 Loopback Mode

Setting the LPBACK bit enables this special mode in which the UxTX pin is internally connected to the UxRX pin. When configured for the Loopback mode, the UxRX pin is disconnected from the internal UART receive logic. However, the UxTX pin still functions as in a normal operation.

To select this mode:

a) Configure UART for desired mode of operation.
b) Set LPBACK = 1 to enable Loopback mode.
c) Enable transmission as defined in **Section 18.3 "Transmitting Data"**.

18.8 Baud Rate Generator

The UART has a 16-bit Baud Rate Generator to allow maximum flexibility in baud rate generation. The Baud Rate Generator register (UxBRG) is readable and writable. The baud rate is computed as follows:

BRG = 16-bit value held in UxBRG register (0 through 65535)

FCY = Instruction Clock Rate (1/TCY)

The baud rate is given by Equation 18-1.

EQUATION 18-1: BAUD RATE

$$\text{Baud Rate} = F_{CY}/(16 * (BRG + 1))$$

Therefore, maximum baud rate possible is:

FCY/16 (if BRG = 0),

and the minimum baud rate possible is:

FCY/(16 * 65536).

With a full, 16-bit Baud Rate Generator, at 30 MIPS operation, the minimum baud rate achievable is 28.5 bps.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18.9 Auto-Baud Support

To allow the system to determine baud rates of received characters, the input can be optionally linked to a selected capture input (IC1 for UART1, IC2 for UART2). To enable this mode, the user must program the input capture module to detect the falling and rising edges of the Start bit.

18.10 UART Operation During CPU Sleep and Idle Modes

18.10.1 UART OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shut down and stay at logic '0'. If entry into Sleep mode occurs while a transmission is in progress, then the transmission is aborted. The UxTX pin is driven to logic '1'. Similarly, if entry into Sleep mode occurs while a reception is in progress, then the reception is aborted. The UxSTA, UxMODE, UxBRG, transmit and receive registers and buffers, are not affected by Sleep mode.

If the WAKE bit (UxMODE<7>) is set before the device enters Sleep mode, then a falling edge on the UxRX pin will generate a receive interrupt. The Receive Interrupt Select Mode bit (URXISEL) has no effect for this function. If the receive interrupt is enabled, then this will wake-up the device from Sleep. The UARTEN bit must be set in order to generate a wake-up interrupt.

18.10.2 UART OPERATION DURING CPU IDLE MODE

For the UART, the USIDL bit selects if the module will stop operation when the device enters Idle mode, or whether the module will continue on Idle. If USIDL = 0, the module will continue operation during Idle mode. If USIDL = 1, the module will stop on Idle.

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 18-1: UART1 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U1MODE	020C	UARTEN	—	USIDL	—	—	ALTIO	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U1STA	020E	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U1TXREG	0210	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 000u uuuu uuuu
U1RXREG	0212	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U1BRG	0214	Baud Rate Generator Prescaler																0000 0000 0000 0000

Legend: u = uninitialized bit

Note: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

TABLE 18-2: UART2 REGISTER MAP (NOT AVAILABLE ON dsPIC30F4012)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U2MODE	0216	UARTEN	—	USIDL	—	—	—	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U2STA	0218	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U2TXREG	021A	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 000u uuuu uuuu
U2RXREG	021C	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U2BRG	021E	Baud Rate Generator Prescaler																0000 0000 0000 0000

Legend: u = uninitialized bit

Note: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

© 2007 Microchip Technology Inc.

20.0 10-BIT, HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

The 10-bit, high-speed Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit digital number. This module is based on a Successive Approximation Register (SAR) architecture and provides a maximum sampling rate of 1 Msps. The ADC module has 16 analog inputs which are multiplexed into four sample and hold amplifiers. The output of the sample and hold is the input into the converter which generates the result. The analog reference voltages are software selectable to either the device supply voltage (AVDD/AVSS) or the voltage level on the (VREF+/VREF-) pins. The ADC module has a unique feature of being able to operate while the device is in Sleep mode.

The ADC module has six, 16-bit registers:

- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)
- A/D Control Register 3 (ADCON3)
- A/D Input Select Register (ADCHS)
- A/D Port Configuration Register (ADPCFG)
- A/D Input Scan Selection Register (ADCSSL)

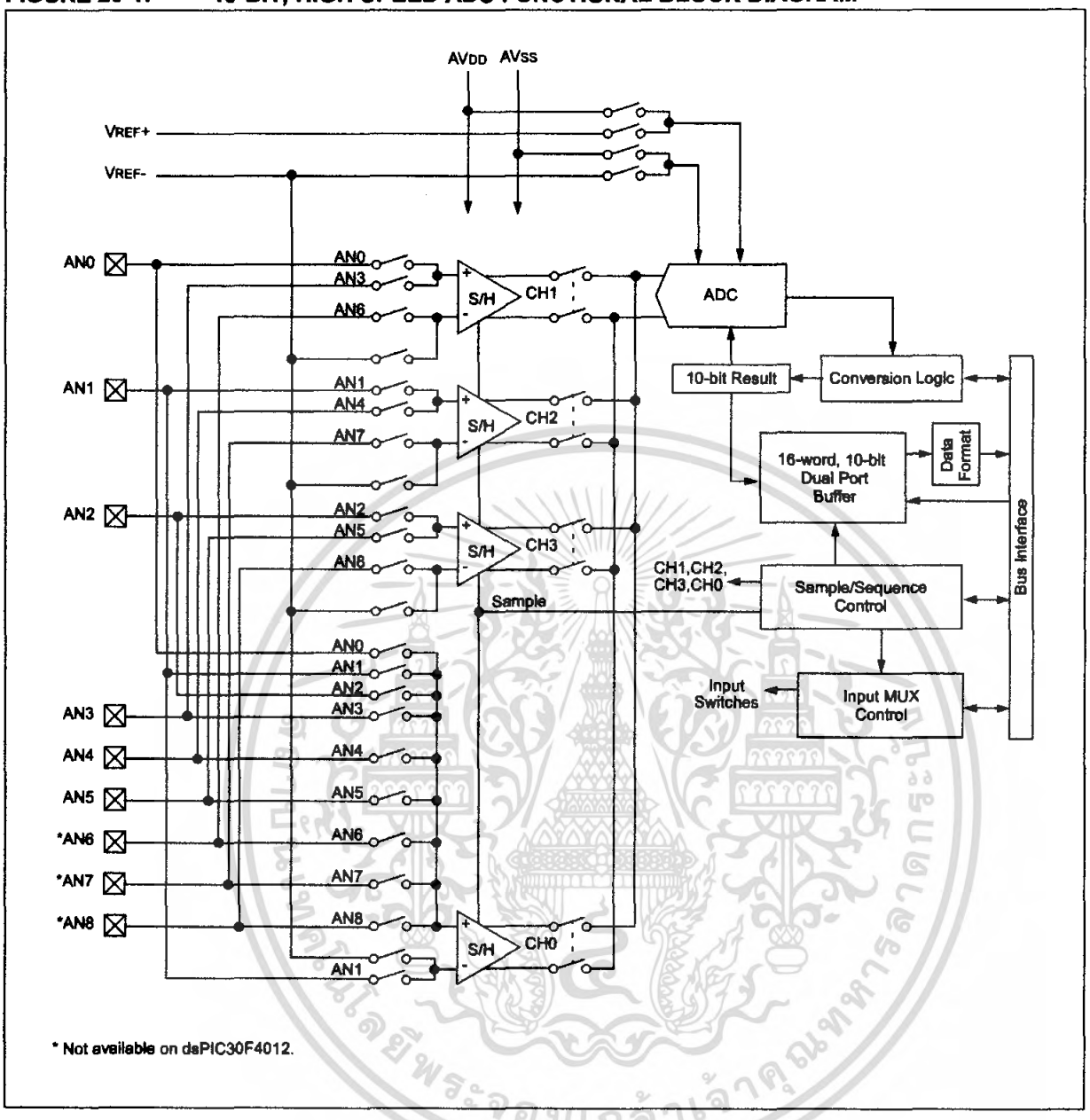
The ADCON1, ADCON2 and ADCON3 registers control the operation of the ADC module. The ADCHS register selects the input channels to be converted. The ADPCFG register configures the port pins as analog inputs or as digital I/O. The ADCSSL register selects inputs for scanning.

Note: The SSRC<2:0>, ASAM, SIMSAM, SMPI<3:0>, BUFM and ALTS bits, as well as the ADCON3 and ADCSSL registers, must not be written to while ADON = 1. This would lead to indeterminate results.

The block diagram of the ADC module is shown in Figure 20-1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 20-1: 10-BIT, HIGH-SPEED ADC FUNCTIONAL BLOCK DIAGRAM



© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20.1 A/D Result Buffer

The module contains a 16-word, dual port, read-only buffer, called ADCBUF0...ADCBUFF, to buffer the A/D results. The RAM is 10 bits wide, but is read into different format 16-bit words. The contents of the sixteen A/D Conversion Result Buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

20.2 Conversion Operation

After the ADC module has been configured, the sample acquisition is started by setting the SAMP bit. Various sources, such as a programmable bit, timer time-outs and external events, terminate acquisition and start a conversion. When the A/D conversion is complete, the result is loaded into ADCBUF0...ADCBUFF, and the A/D Interrupt Flag, ADIF, and the DONE bit are set after the number of samples specified by the SMPI<3:0> bits.

The following steps should be followed for doing an A/D conversion:

1. Configure the ADC module:
   - Configure analog pins, voltage reference and digital I/O
   - Select A/D input channels
   - Select A/D conversion clock
   - Select A/D conversion trigger
   - Turn on ADC module
2. Configure the A/D interrupt (if required):
   - Clear ADIF bit
   - Select A/D interrupt priority
3. Start sampling.
4. Wait the required acquisition time.
5. Trigger acquisition end, start conversion.
6. Wait for A/D conversion to complete by either:
   - Waiting for the A/D interrupt
   - Waiting for the DONE bit to get set
7. Read A/D result buffer, clear ADIF if required.

20.3 Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the A/D connects inputs to the sample/hold channels, converts channels, writes the buffer memory and generates interrupts. The sequence is controlled by the sampling clocks.

The SIMSAM bit controls the acquire/convert sequence for multiple channels. If the SIMSAM bit is '0', the two or four selected channels are acquired and converted sequentially with two or four sample clocks. If the SIMSAM bit is '1', two or four selected channels are acquired simultaneously with one sample clock. The channels are then converted sequentially. Obviously, if there is only 1 channel selected, the SIMSAM bit is not applicable.

The CHPS<1:0> bits select how many channels are sampled. This selection can vary from 1, 2 or 4 channels. If the CHPS bits select 1 channel, the CH0 channel is sampled at the sample clock and converted. The result is stored in the buffer. If the CHPS bits select 2 channels, the CH0 and CH1 channels are sampled and converted. If the CHPS bits select 4 channels, the CH0, CH1, CH2 and CH3 channels are sampled and converted.

The SMPI<3:0> bits select the number of acquisition/conversion sequences that would be performed before an interrupt occurs. This can vary from 1 sample per interrupt to 16 samples per interrupt.

The user cannot program a combination of CHPS and SMPI bits that specifies more than 16 conversions per interrupt, or 8 conversions per interrupt, depending on the BUFM bit. The BUFM bit, when set, splits the 16-word results buffer (ADCBUF0...ADCBUFF) into two, 8-word groups. Writing to the 8-word buffers is alternated on each interrupt event. Use of the BUFM bit depends on how much time is available for moving data out of the buffers after the interrupt, as determined by the application.

If the processor can quickly unload a full buffer within the time it takes to acquire and convert one channel, the BUFM bit can be '0' and up to 16 conversions may be done per interrupt. The processor has one sample-and-conversion time to move the sixteen conversions.

If the processor cannot unload the buffer within the acquisition and conversion time, the BUFM bit should be '1'. For example, if SMPI<3:0> (ADCON2<5:2>) = 0111, then eight conversions are loaded into half of the buffer, following which an interrupt occurs. The next eight conversions are loaded into the other half of the buffer. The processor has the entire time between interrupts to move the eight conversions.

The ALTS bit can be used to alternate the inputs selected during the sampling sequence. The input multiplexer has two sets of sample inputs: MUX A and MUX B. If the ALTS bit is '0', only the MUX A inputs are selected for sampling. If the ALTS bit is '1' and SMPI<3:0> = 0000, on the first sample/convert sequence, the MUX A inputs are selected, and on the next acquire/convert sequence, the MUX B inputs are selected.

The CSCNA bit (ADCON2<10>) allows the CH0 channel inputs to be alternately scanned across a selected number of analog inputs for the MUX A group. The inputs are selected by the ADCSSL register. If a particular bit in the ADCSSL register is '1', the corresponding input is selected. The inputs are always scanned from lower to higher numbered inputs, starting after each interrupt. If the number of inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs are unused.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20.4 Programming the Start of the Conversion Trigger

The conversion trigger terminates acquisition and starts the requested conversions.

The SSRC<2:0> bits select the source of the conversion trigger.

The SSRC bits provide for up to 5 alternate sources of conversion trigger.

When SSRC<2:0> = 000, the conversion trigger is under software control. Clearing the SAMP bit causes the conversion trigger.

When SSRC<2:0> = 111 (Auto-Start mode), the conversion trigger is under A/D clock control. The SAMC bits select the number of A/D clocks between the start of acquisition and the start of conversion. This provides the fastest conversion rates on multiple channels. SAMC must always be at least 1 clock cycle.

Other trigger sources can come from timer modules, motor control PWM module or external interrupts.

Note: To operate the ADC at the maximum specified conversion speed, the auto-convert trigger option should be selected (SSRC = 111) and the auto-sample time bits should be set to '1' TAD (SAMC = 00001). This configuration gives a total conversion period (sample + convert) of 13 TAD.

The use of any other conversion trigger results in additional TAD cycles to synchronize the external event to the ADC.

20.5 Aborting a Conversion

Clearing the ADON bit during a conversion aborts the current conversion and stops the sampling sequencing. The ADCBUFx is not updated with the partially completed A/D conversion sample. That is, the ADCBUFx will continue to contain the value of the last completed conversion (or the last value written to the ADCBUFx register).

If the clearing of the ADON bit coincides with an auto-start, the clearing has a higher priority.

After the A/D conversion is aborted, a 2 TAD wait is required before the next sampling may be started by setting the SAMP bit.

If sequential sampling is specified, the A/D continues at the next sample pulse, which corresponds with the next channel converted. If simultaneous sampling is specified, the A/D continues with the next multichannel group conversion sequence.

20.6 Selecting the A/D Conversion Clock

The A/D conversion requires 12 TAD. The source of the A/D conversion clock is software selected using a 6-bit counter. There are 64 possible options for TAD.

EQUATION 20-1: A/D CONVERSION CLOCK

$$T_{AD} = T_{CY} * (0.5 * (ADCS<5:0> + 1))$$

$$ADCS<5:0> = 2\ \frac{T_{AD}}{T_{CY}} - 1$$

The internal RC oscillator is selected by setting the ADRC bit.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 83.33 nsec (for VDD = 5V). Refer to **Section 24.0 "Electrical Characteristics"** for minimum TAD under other operating conditions.

Example 20-1 shows a sample calculation for the ADCS<5:0> bits, assuming a device operating speed of 30 MIPS.

EXAMPLE 20-1: A/D CONVERSION CLOCK CALCULATION

$T_{AD} = 154$ nsec
$T_{CY} = 33$ nsec (30 MIPS)

$$ADCS<5:0> = 2\ \frac{T_{AD}}{T_{CY}} - 1$$

$$= 2 \cdot \frac{154\ \text{nsec}}{33\ \text{nsec}} - 1$$

$$= 8.33$$

Therefore,
Set ADCS<5:0> = 9

$$\text{Actual } T_{AD} = \frac{T_{CY}}{2}\ (ADCS<5:0> + 1)$$

$$= \frac{33\ \text{nsec}}{2}\ (9 + 1)$$

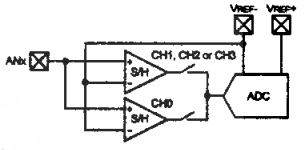
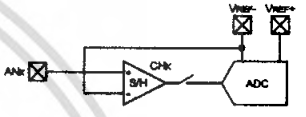
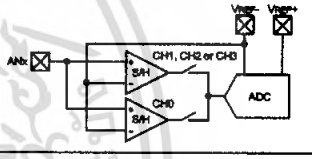
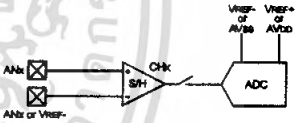
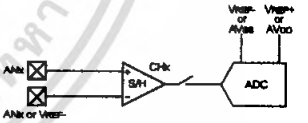
$$= 165\ \text{nsec}$$

© 2007 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20.7 A/D Conversion Speeds

The dsPIC30F 10-bit ADC specifications permit a maximum 1 Msps sampling rate. Table 20-1 summarizes the conversion speeds for the dsPIC30F 10-bit ADC and the required operating conditions.

TABLE 20-1: 10-BIT A/D CONVERSION RATE PARAMETERS

dsPIC30F 10-bit A/D Converter Conversion Rates						
A/D Speed	**TAD Minimum**	**Sampling Time Min.**	**Rs Max.**	**VDD**	**Temperature**	**A/D Channels Configuration**
Up to 1 Msps(1)	83.33 ns	12 TAD	500Ω	4.5V to 5.5V	-40°C to +85°C	
Up to 750 ksps(1)	95.24 ns	2 TAD	500Ω	4.5V to 5.5V	-40°C to +85°C	
Up to 600 ksps(1)	138.89 ns	12 TAD	500Ω	3.0V to 5.5V	-40°C to +125°C	
Up to 500 ksps	153.85 ns	1 TAD	5.0 kΩ	4.5V to 5.5V	-40°C to +125°C	
Up to 300 ksps	256.41 ns	1 TAD	5.0 kΩ	3.0V to 5.5V	-40°C to +125°C	

Note 1: External VREF- and VREF+ pins must be used for correct operation. See Figure 20-2 for recommended circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 20-2: ADC REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
ADCBUF0	0280	—	—	—	—	—	—	ADC Data Buffer 0										0000 00uu uuuu uuuu
ADCBUF1	0282	—	—	—	—	—	—	ADC Data Buffer 1										0000 00uu uuuu uuuu
ADCBUF2	0284	—	—	—	—	—	—	ADC Data Buffer 2										0000 00uu uuuu uuuu
ADCBUF3	0286	—	—	—	—	—	—	ADC Data Buffer 3										0000 00uu uuuu uuuu
ADCBUF4	0288	—	—	—	—	—	—	ADC Data Buffer 4										0000 00uu uuuu uuuu
ADCBUF5	028A	—	—	—	—	—	—	ADC Data Buffer 5										0000 00uu uuuu uuuu
ADCBUF6	028C	—	—	—	—	—	—	ADC Data Buffer 6										0000 00uu uuuu uuuu
ADCBUF7	028E	—	—	—	—	—	—	ADC Data Buffer 7										0000 00uu uuuu uuuu
ADCBUF8	0290	—	—	—	—	—	—	ADC Data Buffer 8										0000 00uu uuuu uuuu
ADCBUF9	0292	—	—	—	—	—	—	ADC Data Buffer 9										0000 00uu uuuu uuuu
ADCBUFA	0294	—	—	—	—	—	—	ADC Data Buffer 10										0000 00uu uuuu uuuu
ADCBUFB	0296	—	—	—	—	—	—	ADC Data Buffer 11										0000 00uu uuuu uuuu
ADCBUFC	0298	—	—	—	—	—	—	ADC Data Buffer 12										0000 00uu uuuu uuuu
ADCBUFD	029A	—	—	—	—	—	—	ADC Data Buffer 13										0000 00uu uuuu uuuu
ADCBUFE	029C	—	—	—	—	—	—	ADC Data Buffer 14										0000 00uu uuuu uuuu
ADCBUFF	029E	—	—	—	—	—	—	ADC Data Buffer 15										0000 00uu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>					ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>				CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	—	—	—	—	—	—	—	PCFG8*	PCFG7*	PCFG6*	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	—	—	—	—	—	—	—	CSSL8*	CSSL7*	CSSL6*	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

Legend: u = uninitialized bit

* Not available on dsPIC30F4012.

Note: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

© 2007 Microchip Technology Inc.

10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI). The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **CSRC:** Clock Source Select bit
Asynchronous mode
Don't care
Synchronous mode
1 = Master mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)

bit 6: **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5: **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4: **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3: **Unimplemented:** Read as '0'

bit 2: **BRGH**: High Baud Rate Select bit
Asynchronous mode
1 = High speed
0 = Low speed
Synchronous mode
Unused in this mode

bit 1: **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0: **TX9D:** 9th bit of transmit data. Can be parity bit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **SPEN:** Serial Port Enable bit
1 = Serial port enabled (Configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6: **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit
Asynchronous mode
Don't care
Synchronous mode - master
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave
Unused in this mode

bit 4: **CREN**: Continuous Receive Enable bit
Asynchronous mode
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3: **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1)
1 = Enables address detection, enable interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit

bit 2: **FERR**: Framing Error bit
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1: **OERR**: Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error

bit 0: **RX9D:** 9th bit of received data (Can be parity bit)

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.1 USART Baud Rate Generator (BRG)

The BRG supports both the asynchronous and synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the Fosc/(16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used by the BRG.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.301	0.33	185
1.2	1.202	0.17	51	1.216	1.33	46
2.4	2.404	0.17	25	2.432	1.33	22
9.6	8.929	6.99	6	9.322	2.90	5
19.2	20.833	8.51	2	18.643	2.90	2
28.8	31.250	8.51	1	-	-	-
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	55.930	2.90	0
HIGH	0.244	-	255	0.218	-	255
LOW	62.500	-	0	55.930	-	0

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000		0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.203	0.25	185
2.4	2.404	0.17	103	2.406	0.25	92
9.6	9.615	0.16	25	9.727	1.32	22
19.2	19.231	0.16	12	18.643	2.90	11
28.8	27.798	3.55	8	27.965	2.90	7
33.6	35.714	6.29	6	31.960	4.88	6
57.6	62.500	8.51	3	55.930	2.90	3
HIGH	0.977	-	255	0.874	-	255
LOW	250.000	-	0	273.722	-	0

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2 USART Asynchronous Mode

In this mode, the USART uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). The most common data format is 8 bits. An on-chip, dedicated, 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator produces a clock either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. Status bit TRMT is a read only bit, which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

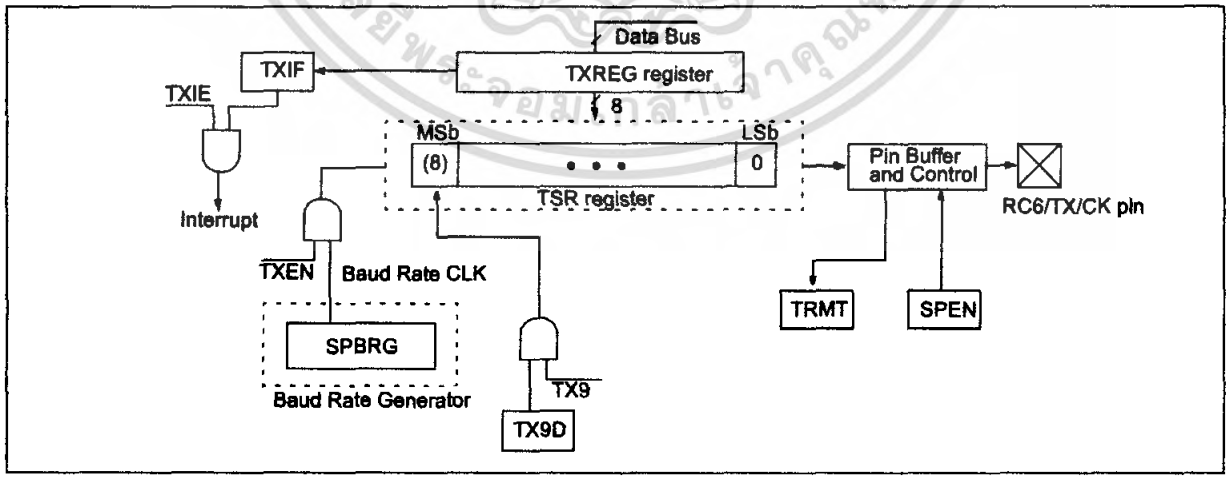
Note 1: The TSR register is not mapped in data memory, so it is not available to the user.

2: Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 10-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 10-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 10.1)
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION

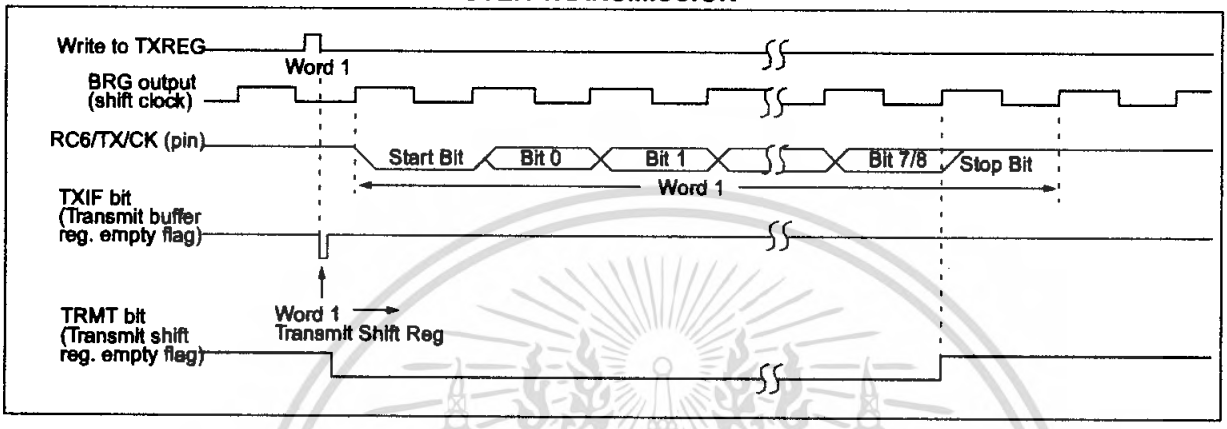
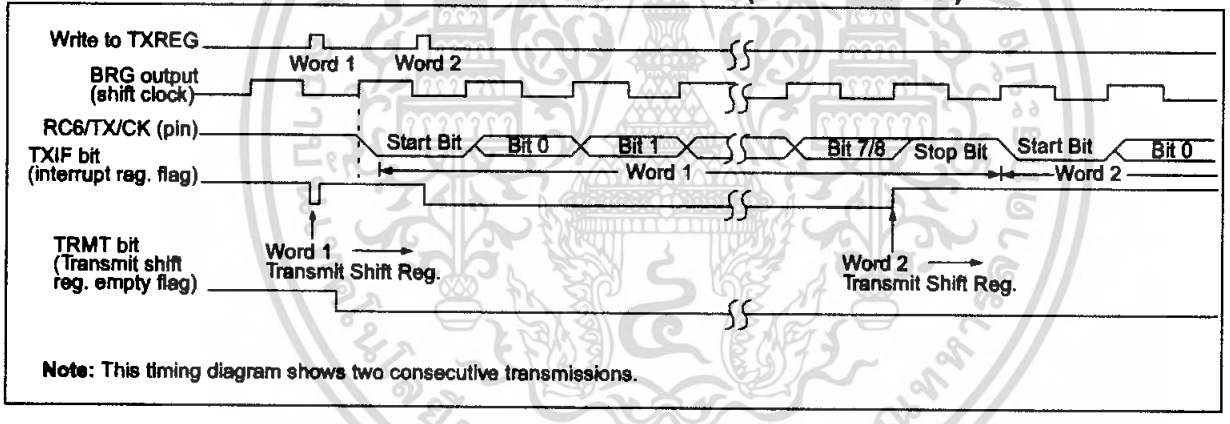


FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e. it is a two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full, the overrun error bit OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited, so it is essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTA<2>) is set if a stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore it is essential for the user to read the RCSTA register before reading RCREG register in order not to lose the old FERR and RX9D information.

FIGURE 10-4: USART RECEVE BLOCK DIAGRAM

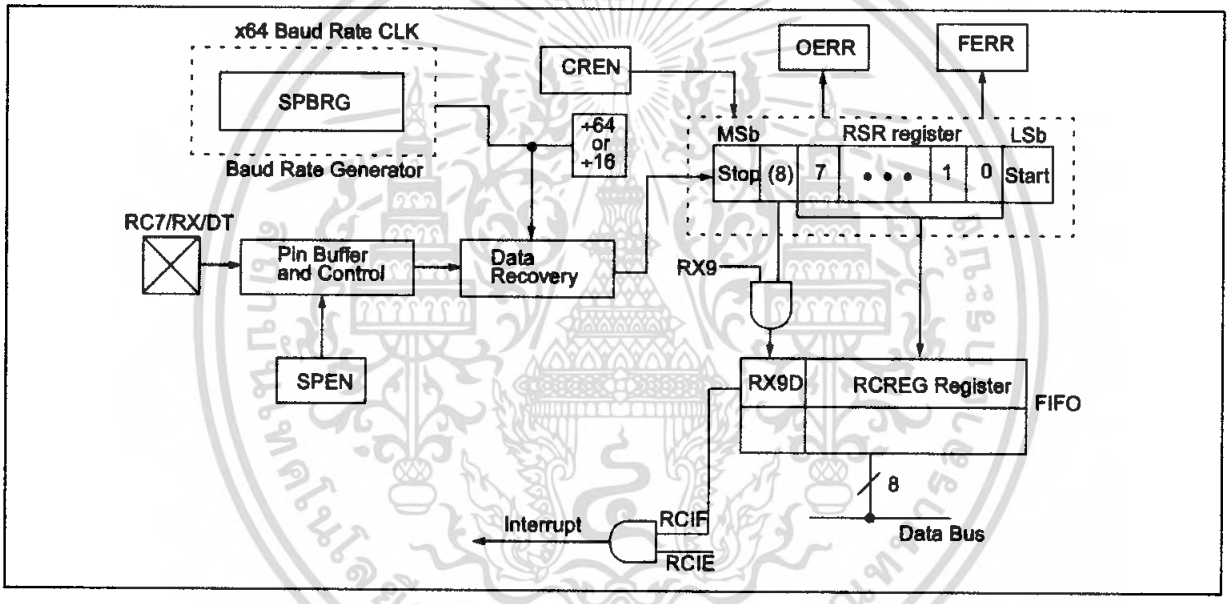
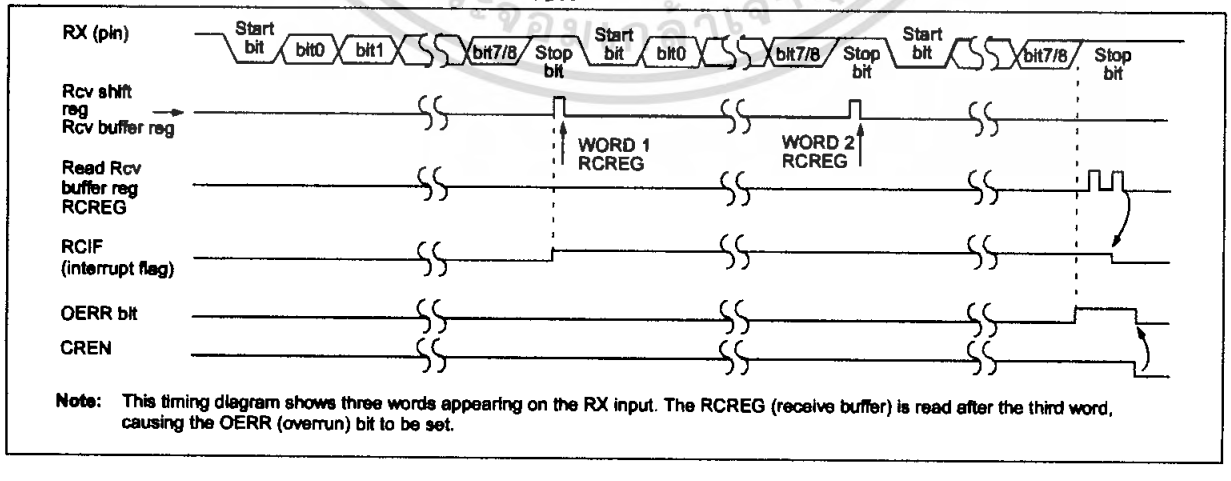


FIGURE 10-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 10.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.

TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.
Note 1: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

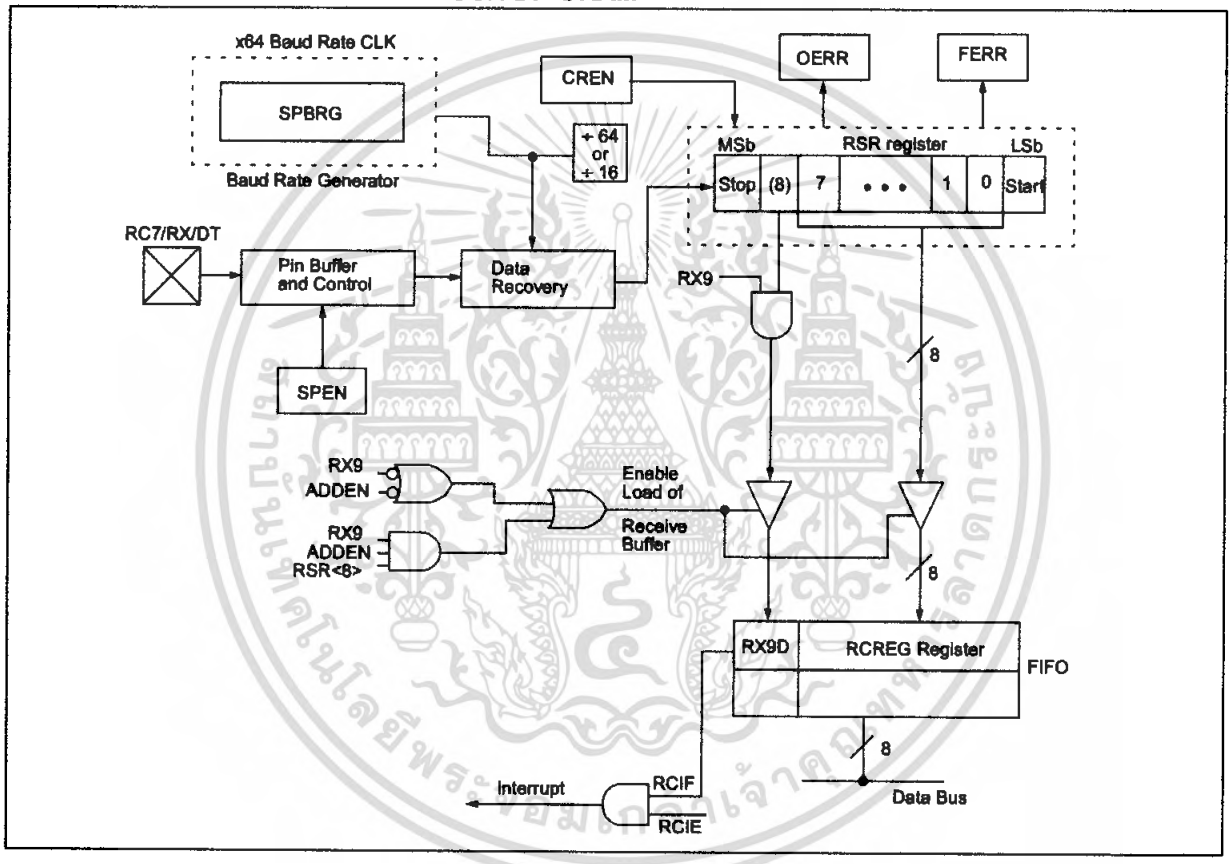
© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

Steps to follow when setting up an Asynchronous Reception with Address Detect Enabled:

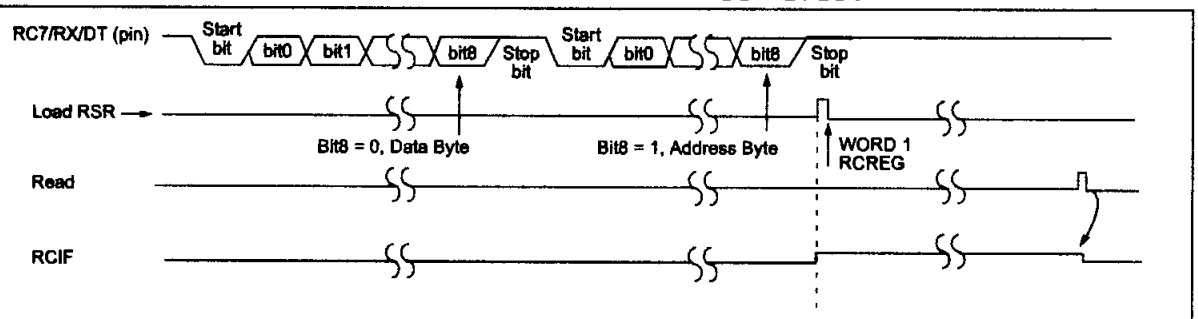
- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
- Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- If interrupts are desired, then set enable bit RCIE.
- Set bit RX9 to enable 9-bit reception.
- Set ADDEN to enable address detect.
- Enable the reception by setting enable bit CREN.
- Flag bit RCIF will be set when reception is complete, and an interrupt will be generated if enable bit RCIE was set.
- Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG register, to determine if the device is being addressed.
- If any error occurred, clear the error by clearing enable bit CREN.
- If the device has been addressed, clear the ADDEN bit to allow data bytes and address bytes to be read into the receive buffer, and interrupt the CPU.

FIGURE 10-6: USART RECEIVE BLOCK DIAGRAM



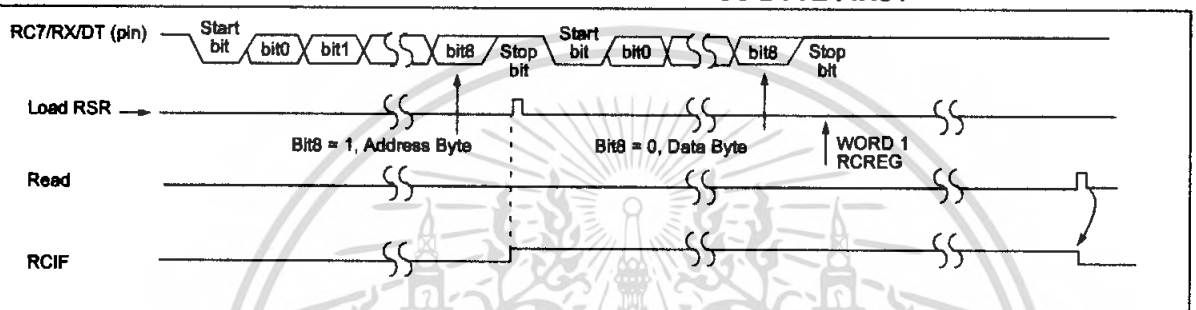
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 10-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (receive buffer) because ADDEN = 1.

FIGURE 10-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (receive buffer) because ADDEN was not updated and still = 0.

TABLE 10-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.
Note 1: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in sleep, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference) or as digital I/O.

Additional information on using the A/D module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7-6: **ADCS1:ADCS0: A/D Conversion Clock Select bits**
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from an RC oscillation)

bit 5-3: **CHS2:CHS0**: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)(1)
110 = channel 6, (RE1/AN6)(1)
111 = channel 7, (RE2/AN7)(1)

bit 2: **GO/$\overline{\text{DONE}}$**: A/D Conversion Status bit
If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1: **Unimplemented**: Read as '0'

bit 0: **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shutoff and consumes no operating current

Note 1: These channels are not available on the 28-pin devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset

bit 7: **ADFM: A/D Result format select**
1 = Right Justified. 6 most significant bits of ADRESH are read as '0'.
0 = Left Justified. 6 least significant bits of ADRESL are read as '0'.

bit 6-4: **Unimplemented: Read as '0'**

bit 3-0: **PCFG3:PCFG0**: A/D Port Configuration Control bits

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / Refs(2)
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input

D = Digital I/O

Note 1: These channels are not available on the 28-pin devices.

2: This column indicates the number of analog channels available as A/D inputs and the numer of analog channels used as voltage reference inputs.

© 1999 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

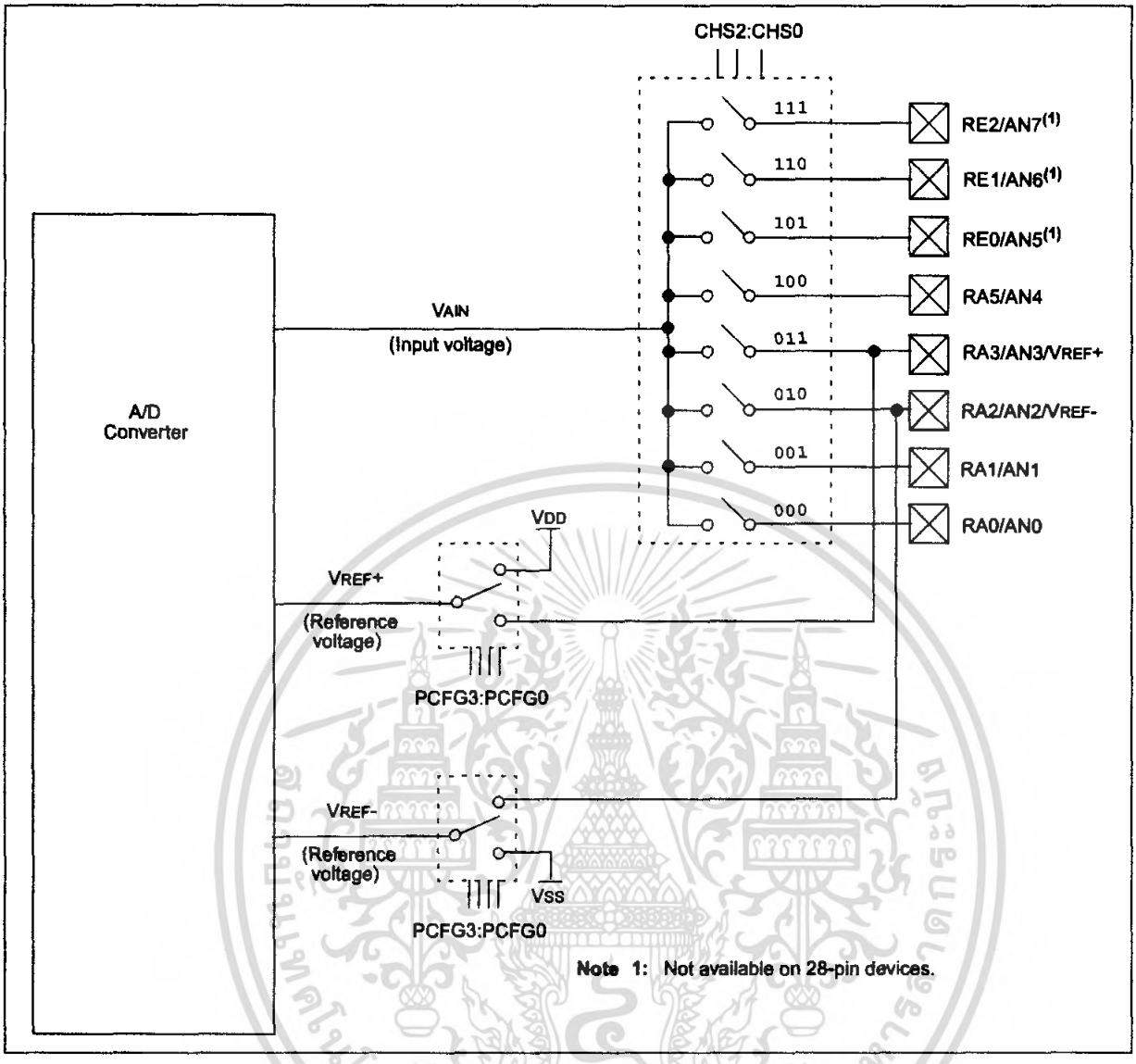
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine sample time, see Section 11.1. After this acquisition time has elapsed, the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
   - Configure analog pins / voltage reference / and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before next acquisition starts.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 11-1: A/D BLOCK DIAGRAM



11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), Figure 11-2. **The maximum recommended impedance for analog sources is 10 kΩ.** As the impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the PICmicro™ Mid-Range Reference Manual (DS33023).

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EQUATION 11-1: ACQUISITION TIME

T_{ACQ}	=	Amplifier Settling Time +
		Hold Capacitor Charging Time +
		Temperature Coefficient
	=	$T_{AMP} + T_C + T_{COFF}$
	=	$2\mu S + T_C + [(\text{Temperature} - 25°C)(0.05\mu S/°C)]$
T_C	=	$C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/2047)$
	=	$-120pF (1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)$
	=	$16.47\mu S$
T_{ACQ}	=	$2\mu S + 16.47\mu S + [(50°C - 25\times C)(0.05\mu S/\times C)$
	=	$19.72\mu S$

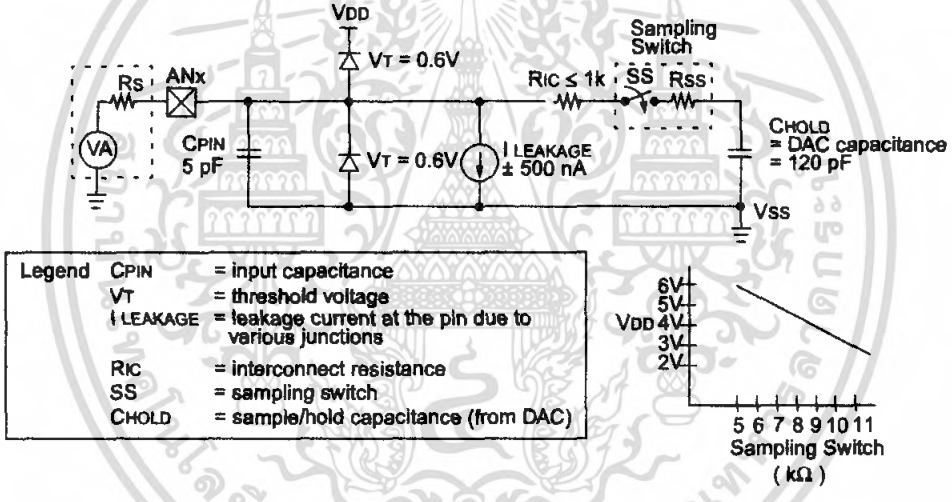
Note 1: The reference voltage (V_{REF}) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (C_{HOLD}) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

4: After a conversion has completed, a 2.0T_{AD} delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

FIGURE 11-2: ANALOG INPUT MODEL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 µs.

Table 11-1shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
2Tosc	00	1.25 MHz
8Tosc	01	5 MHz
32Tosc	10	20 MHz
RC(1, 2, 3)	11	Note 1

Note 1: The RC source has a typical TAD time of 4 µs but can vary between 2-6 µs.
2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for sleep operation.
3: For extended voltage devices (LC), please refer to the Electrical Specifications section.

11.3 Configuring Analog Port Pins

The ADCON1, and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

Note 1: When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the device specifications.

© 1999 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

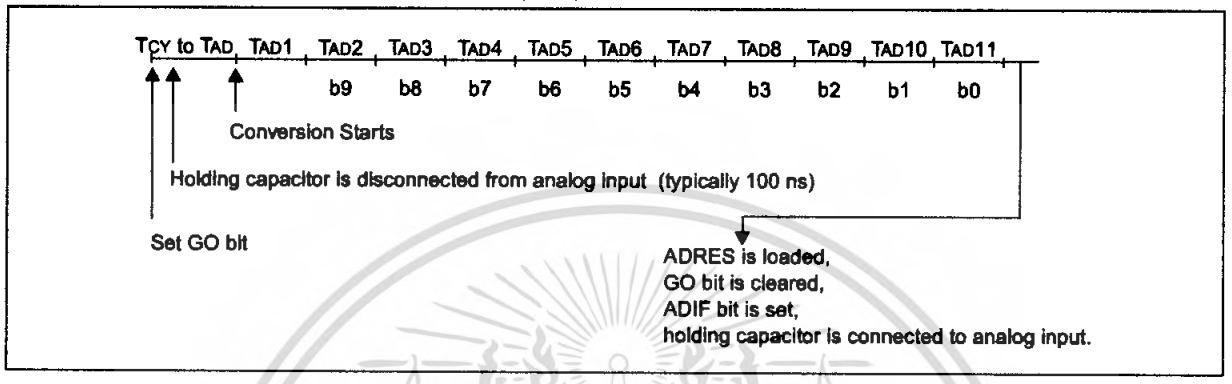
11.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started.

In Figure 11-3, after the GO bit is set, the first time segmant has a minimum of TCY and a maximum of TAD.

Note: The GO/$\overline{\text{DONE}}$ bit should NOT be set in the same instruction that turns on the A/D.

FIGURE 11-3: A/D CONVERSION TAD CYCLES



© 1999 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

11.5 A/D Operation During Sleep

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed the GO/$\overline{\text{DONE}}$ bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

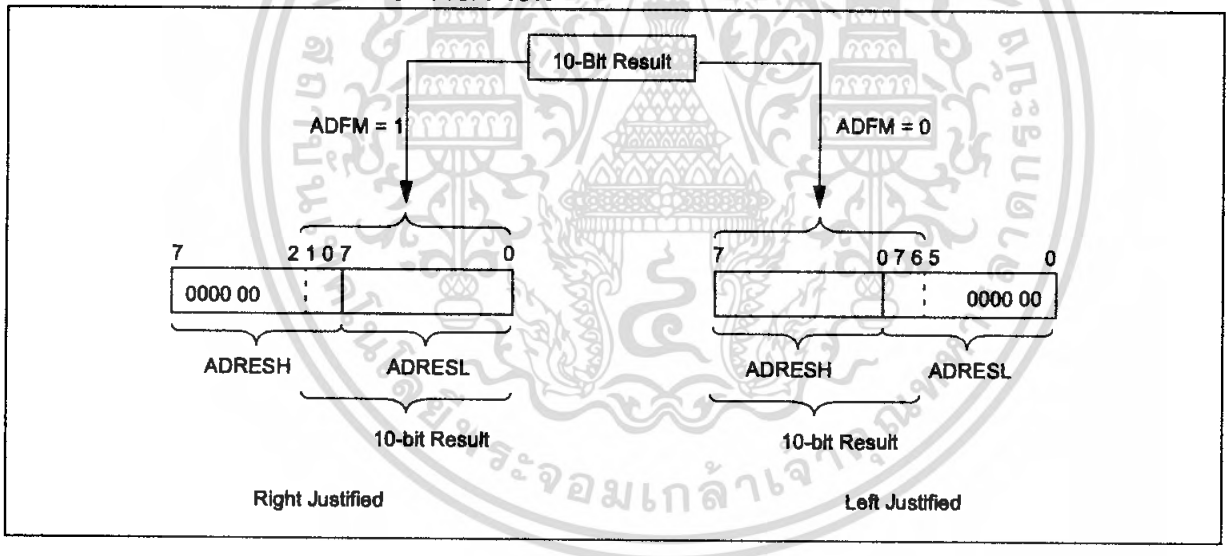
Note:	**For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/$\overline{\text{DONE}}$ bit.**

11.6 Effects of a Reset

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion is aborted.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

FIGURE 11-4: A/D RESULT JUSTIFICATION



© 1999 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR	$\overline{MCLR}$, WDT
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.
Note 1: These registers/bits are not available on the 28-pin devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชนด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Radiometrix

Hartcran House, 231 Kenton Lane, Harrow, HA3 8RP, England

Tel: +44 (0) 20 8909 9595, Fax: +44 (0) 20 8909 2233

Issue 2, 01 November 2004 **TX2 & RX2**

UHF FM Data Transmitter and Receiver Modules

The TX2 and RX2 data link modules are a miniature PCB mounting UHF radio transmitter and receiver pair which enable the simple implementation of a data link at upto 160 kbit/s at distances upto 75 metres in-building and 300 metres open ground.

left: TX2 transmitter *right:RX2 receiver*

Features:

- CE certified by independent Notified Body
- Verified to comply with harmonised radio standard EN 300 220-3 by accredited Test Laboratory
- Verified to comply with harmonised EMC standard EN 301 489-3 by accredited Test Laboratory
- Data rates up to 160 kbps
- Usable range up to 300 m
- 433.92 MHz versions.
- Fully screened

Available for operation at 433.92 MHz in Europe (including UK), both modules combine full screening with extensive internal filtering to ensure EMC compliance by minimising spurious radiations and susceptibilities. The TX2 and RX2 modules will suit one-to-one and multi-node wireless links in applications including car and building security, EPOS and inventory tracking, remote industrial process monitoring and computer networking. Because of their small size and low power requirements, both modules are ideal for use in portable, battery-powered applications such as hand-held terminals.

Transmitter - TX2

- 2 stage SAW controlled, FM modulated at up to 160 kbps
- Operation from 2.2V to 6V
- +9dBm on 433.92MHz
- High efficiency, >15%, DC RF
- Improved frequency and deviation accuracy
- 2nd harmonic, < -60dBc

Receiver - RX2

- Double conversion FM superhet
- SAW front end filter, image rejection 50dB
- Supply 3.0V to 6.0V @ 13mA
- -96 dBm sensitivity @1ppm BER with 160kbps version
- -100 dBm sensitivity @ 1ppm BER with 40kbps-version
- -107 dBm sensitivity @ 1ppm BER with 14kbps-version
- LO leakage, < -60 dBm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional description:

The TX2 transmitter module is a two stage, SAW controlled FM transmitter operating between 2V and 6V and is available in 433.92MHz. The TX2 module is type approved to EN 300 220-3 for European use and delivers nominally +9dBm from a 5V supply at 12mA. The module measures 32 x 12 x 3.8 mm.

The RX2 module is a double conversion FM superhet receiver capable of handling date rates of up to 160kbps. The SIL style RX2 receiver measures 48 x 17.5 x 4.5 mm. It will operate from a supply of 3-6V and draws 14mA when receiving. A fast-acting carrier detect and a power-up enable time of less than 1ms. This allows effective duty cycle power saving and a -107 dBm sensitivity. This combined with a SAW front-end filter results in an excellent RF performance and EMC conformance.

TX2 transmitter:

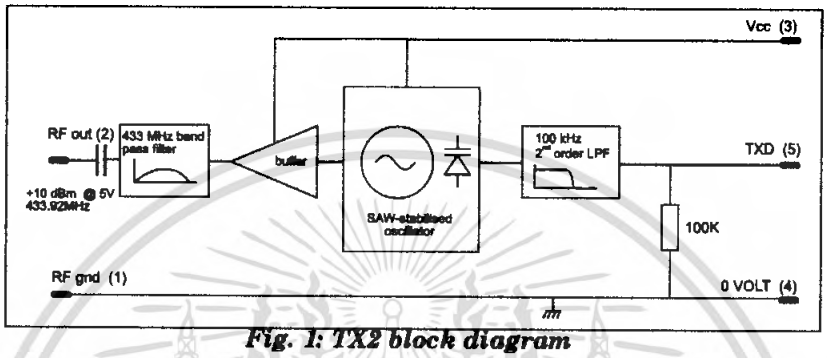


Fig. 1: TX2 block diagram

Pin description:

RF GND *(pin 1)*
RF ground pin, internally connected to the module screen and pin 4 (0V). This pin should be connected to the RF return path (e.g. coax braid, main PCB ground plane etc.)

RF out *(pin 2)*
50Ω RF output to the antenna, it is DC isolated internally. (see antenna section for suggested antenna/feeds).

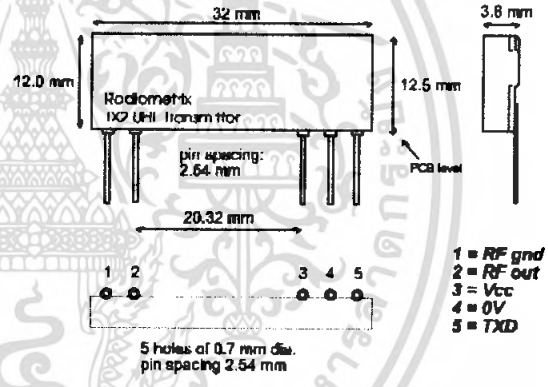


Fig.2: physical dimensions

Vcc *(pin 3)*
+ve supply pin. The module will generate RF when the Vcc supply is present. Maximum ripple content 0.1V_{P-P}. A 100nF de-coupling ceramic capacitor is suggested.

0V *(pin 4)*
Supply ground connection, connected to pin 1 and screen.

TXD *(pin 5)*
This DC coupled modulation input will accept either serial digital data (0V to Vcc levels) or high level linear signals. Input impedance is 100kΩ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RX2 receiver:

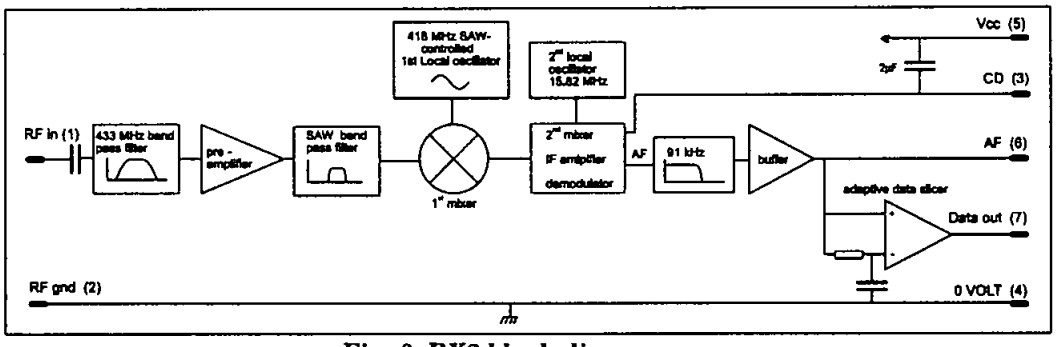


Fig. 3: RX2 block diagram

Pin description

RF in *(pin 1)*
50Ω RF input from the antenna, it is DC isolate internally. (see antenna section for suggested antenna/feeds).

RF GND *(pin 2)*
RF ground pin, internally connected to the module screen and pin 4 (0V). This pin should be connected to the RF return path (e.g. coax braid, main PCB ground plane etc.)

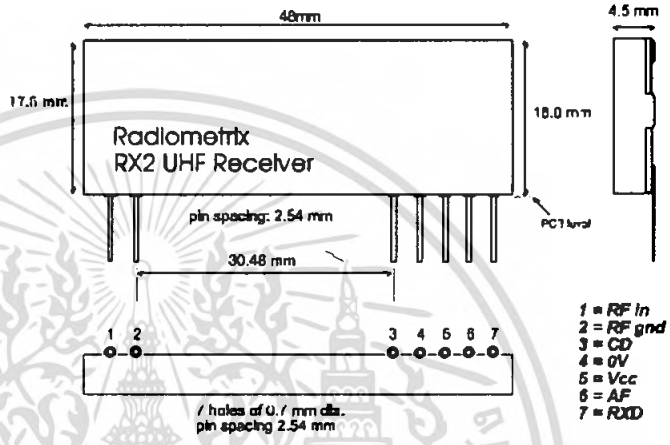


Fig. 4: physical dimensions

CD *(pin 3)*
The Carrier Detect may be used to drive an external PNP transistor to obtain a logic level carrier detect signal, see test circuit. If not required it should be connected to pin 5 (Vcc).

0 Volt *(pin 4)*
Supply ground connection, connected to pin 1 and screen.

Vcc *(pin 5)*
+ve supply pin. +3.0V to +6.0V @ <17mA . The supply must be clean < $2mV_{P-P}$ ripple. A 10µF de-coupling capacitor and 10Ω series resistor is recommended if a clean supply is not available.

AF *(pin 6)*
This is a buffered and filtered analogue output from the FM demodulator. It has a standing DC bias of 1.2V and $400mV_{P-P}$ base band signal. It is useful as a test point or to drive linear decoders. Load impedance should be > 2kΩ and < 100pF.

RXD *(pin 7)*
This digital output from the internal data slicer is a squared version of the signal on pin 6 (AF). It may be used to drive external decoders. The data is true data, i.e. as fed to the transmitter. Load impedance should be > 1kΩ and < 1nF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Survival Maximums:

Operating temperature:	-10°C to +55°C
Storage temperature:	-40°C to +100°C

TX2, all variants

Vcc (pin 3)	-0.1V to +10.0V
Data input (pin 5)	-0.1V to +10.0V
RF out (pin 2)	±50V @ < 10MHz , +20dBm @ > 10MHz

RX2, all variants

Vcc (pin 5)	-0.1V to +10.0V
Data, CD & AF (pin 7,3,6)	-0.1V to + Vcc V
RF input (pin 1)	±50V @ < 10MHz , +13dBm @ > 10MHz

note: *Operation of the TX2 above 6V may cause the module to exceed the licensed power level.*

Electrical Performance: TX2 transmitter

	pin	min.	typ.	max.	units	notes
DC LEVELS						
supply voltage	3	2.2	3.0	4.0	V	-3V version
supply voltage	3	4.0	5.0	6.0	V	-5V version
CURRENT & RF POWER						
TX2-433-3V						
supply current @ Vcc = 3V	3	4	6	10	mA	
RF power @ Vcc = 3V	2	0	+4	+6	dBm	1
TX2-433-5V						
supply current @ Vcc = 5V	3	7	10	14	mA	
RF power @ Vcc = 5V	2	+6	+9	+12	dBm	1
RF						
2nd harmonic	2	-	-65	-54	dBc	1
harmonics @ > 1GHz	2	-	-50	-40	dBc	1
initial frequency accuracy	-	-30	0	+30	kHz	
overall frequency accuracy	-	-70	-	+70	kHz	
FM deviation (+/-)	-	20	25	30	kHz	
modulation bandwidth @ -3dB	-	DC	-	20	kHz	
modulation bandwidth @ -3dB	-	DC	-	100	kHz	2
modulation distortion (THD)	-	-	5	10	%	
power up time to full RF	-	-	-	100	µs	

note:
1. Measured into a 50Ω load.
2. For 160kbps version

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Performance: RX2 Receiver

Unless otherwise noted: Figures apply to 5V versions unless noted otherwise
Vcc = 5.0V, temperature 20°C unless noted otherwise

	pin	min.	typ.	Max.	units	notes
DC LEVELS						
supply voltage, Vcc standard	5	4.0	5.0	6.0	V	
supply voltage, Vcc, 3V version	5	3.3	3.5	4.0	V	
supply current	5	11	13	17	mA	
supply ripple	5	-	-	2	mV_{P-P}	1
data output high, 100µA source	7	-	Vcc-0.6	-	V	
data output low, 100µA sink	7	-	0.4	-	V	
load capacitance on AF / Data	6,7	-	-	100	pF	
RF						
RF sensitivity for 10dB (S+N)/N	1,6	-107	-113	-	dBm	14kbps version
RF sensitivity for 10dB (S+N)/N	1,6	-100	-107	-	dBm	40kbps version
RF sensitivity for 10dB (S+N)/N	1,6	-	-96	-	dBm	160kbps version
RF sensitivity for 1ppm BER	1,6	-100	-107	-	dBm	14kbps version
RF sensitivity for 1ppm BER	1,6	-93	-100	-	dBm	40kbps version
RF sensitivity for 1ppm BER	1,6	-	-90	-	dBm	160kbps version
CD threshold	1,3	-97	-107	-	dBm	14kbps version
CD threshold	1,3	-90	-100	-	dBm	40kbps version
CD threshold	1,3	-	-104	-	dBm	160kbps version
IF band width	-	-	250	-	kHz	
initial frequency accuracy	1	-30	0	+30	kHz	
E.M.C.						
image rejection ($2*F_{IF}$)	1	-	-50	-	dB	2
spurious responses upto 1GHz	1	-	-70	-	dB	2
LO leakage, conducted	1	-	-65	-	dBm	
LO leakage, radiated	-	-	-70	-	dBm	
AF BASE BAND						
baseband bandwidth @ -3dB	6	0.006	-	7	kHz	14kbps version
baseband bandwidth @ -3dB	6	0.060	-	20	kHz	40kbps version
baseband bandwidth @ -3dB	6	0.250	-	91	kHz	160kbps version
AF level	6	-	450	-	mV_{P-P}	
DC offset on AF	6	0.8	1.2	1.6	V	
distortion on recovered AF	6	-	0.5	1	%	
ultimate (S+N)/N	6	35	45	-	dB	
ultimate (S+N)/N	6	-	28	-	dB	160kbps version
DYNAMIC TIMING						
Power up with signal present						
power up to valid CD, T_{PU-CD}	-	-	1	-	ms	
power up to stable data, T_{PU-DAT}	-	-	20	-	ms	14kbps version
power up to stable data, T_{PU-DAT}	-	-	5	-	ms	40kbps version
power up to stable data, T_{PU-DAT}	-	-	3	-	ms	160kbps version
power up to stable AF, T_{PU-AF}	-	-	0.6	-	ms	160kbps version
power up to valid CD, T_{PU-CD}	-	-	0.7	-	ms	160kbps version

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	pin	min.	typ.	Max.	units	notes
Signal applied with supply on						
signal to valid CD, T_{SIG-CD}	-	-	0.5	-	ms	
signal to stable data, $T_{SIG-DAT}$	-	-	15	-	ms	14kbps version
signal to stable data, $T_{SIG-DAT}$	-	-	3	-	ms	40kbps version
signal to stable data, $T_{SIG-DAT}$	-	-	0.75	-	ms	160kbps version
signal to stable AF, T_{SIG-AF}	-	-	0.1	-	ms	160kbps version
signal to valid CD, T_{SIG-CD}	-	-	0.2	-	ms	160kbps version
time between data transitions	7	0.07	-	15	ms	4, 14kbps version
time between data transitions	7	0.025	-	1.5	ms	4, 40kbps version
time between data transitions	7	0.00625	-	0.5	ms	4, 160kbps version
mark:space ratio	-	20	50	80	%	3

notes:

1. For 6dB (S+N)/N degradation on wanted -100dBm signal
2. Receiver spurious responses are at $F_{RF} \pm (n \times 15.92MHz)$, n=1,2,3 etc.
3. Average over 30ms (14kbps), 3ms (40kbps), 1ms (160kbs) at maximum bit rate.
4. Values for 50:50 mark to space (i.e. square wave)

Module test circuits

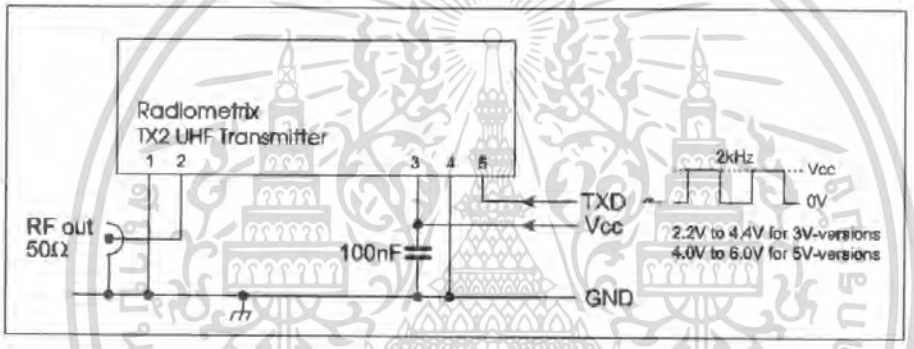


Fig.5: TX2 test circuit

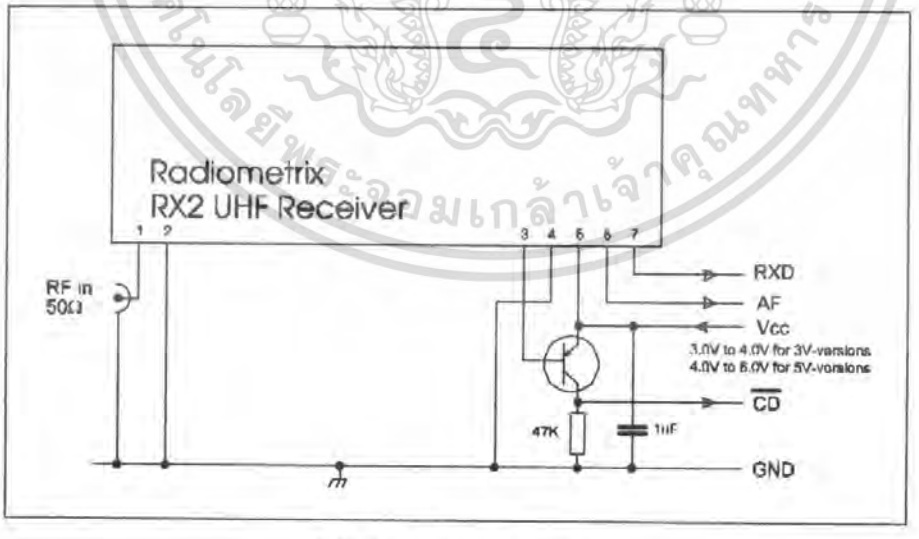


Fig.6: RX2 test circuit

* The PNP transistor (e.g. BC558B) enables a CMOS compatible Carrier Detect signal to be derived from pin 3. If no CD signal required pin 3 should be connected directly to pin 5 (Vcc).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Module mounting considerations

The modules may be mounted horizontally or vertically on an area of ground plane preferably close to the antenna to minimise feed length. The receiver and it's antenna should be kept away from sources of interference (micro's, SMPS etc.). The modules may be potted if required in a viscous compound which can not enter the screen can..

Warning: *Do NOT wash the modules. They are not hermetically sealed.*

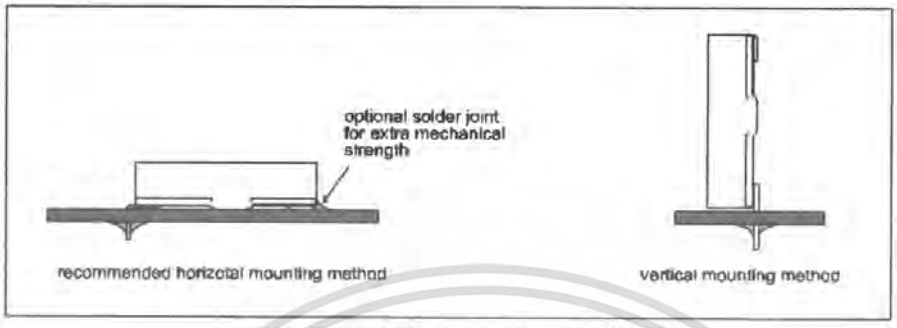


Fig. 7: module mounting options

Antenna requirements

Three types of integral antenna are recommended and approved for use with the module:

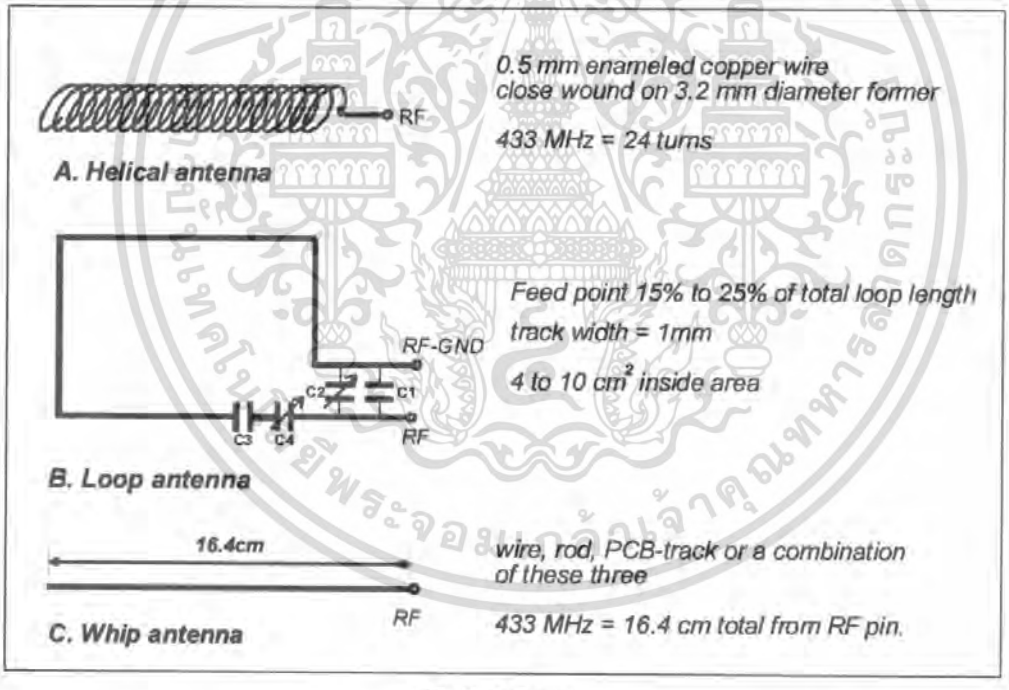


Fig.8: Antenna

	A ***helical***	B ***loop***	C ***whip***
Ultimate performance	**	*	***
Easy of design set-up	**	*	***
Size	***	**	*
Immunity proximity effects	**	***	*
Range open ground to similar antenna (for TX2-433-14-5V & RX2-433-14-5V)	200m	100m	300m

Antenna selection chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A) *Helical* Wire coil, connected directly to pin 2, open circuit at other end. This antenna is very efficient given it's small size (20mm x 4mm dia.). The helical is a high Q antenna, trim the wire length or expand the coil for optimum results. The helical de-tunes badly with proximity to other conductive objects.

B) *Loop* A loop of PCB track tuned by a fixed or variable capacitor to ground at the 'hot' end and fed from pin 2 at a point 20% from the ground end. Loops have high immunity to proximity de-tuning.

C) *Whip* This is a wire, rod, PCB track or combination connected directly to pin 2 of the module. Optimum total length is 15.5cm (1/4 wave @ 433MHz). Keep the open circuit (hot) end well away from metal components to prevent serious de-tuning. Whips are ground plane sensitive and will benefit from internal 1/4 wave earthed radial(s) if the product is small and plastic cased

The antenna choice and position directly controls the system range. Keep it clear of other metal in the system, particularly the 'hot' end. The best position by far, is sticking out the top of the product. This is often not desirable for practical/ergonomic reasons thus a compromise may need to be reached. If an internal antenna must be used try to keep it away from other metal components, particularly large ones like transformers, batteries and PCB tracks/earth plane. The space around the antenna is as important as the antenna itself

Duty Cycle requirements

The duty cycle is defined as the ratio, expressed as a percentage, of the maximum transmitter "on" time on one or more carrier frequencies, relative to a one hour period. Where an acknowledgement message is required, the additional transmitter "on" time shall be included.

There is a 10% duty cycle restriction on 433.050-434.790 MHz band in most of the EU member states.

The TX2-433 is a RF module intended to be incorporated into a wide variety of applications and finished products, Radiometrix has no control over the end use of the TX2-433.The harmonised band 433.050 to 434.790 MHz as detailed in Annex 1 Band E of CEPT/ERC Recommendation 70-03 (which can be downloaded at http://www.ero.dk/scripts/docmanag98/dm.dll/QueryDoc?Cat=Recommendation) has list of countries where Duty Cycle restriction apply.

Module users should, therefore, ensure that they comply with the stated Duty Cycle requirements of the version of CEPT/ERC Recommendation 70-03 in place at the time of incorporation of the TX2-433 into their product. It should be noted that the stated Duty Cycle must not be exceeded otherwise any approval granted for the TX2-433 will be invalidated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

Multiwatt15

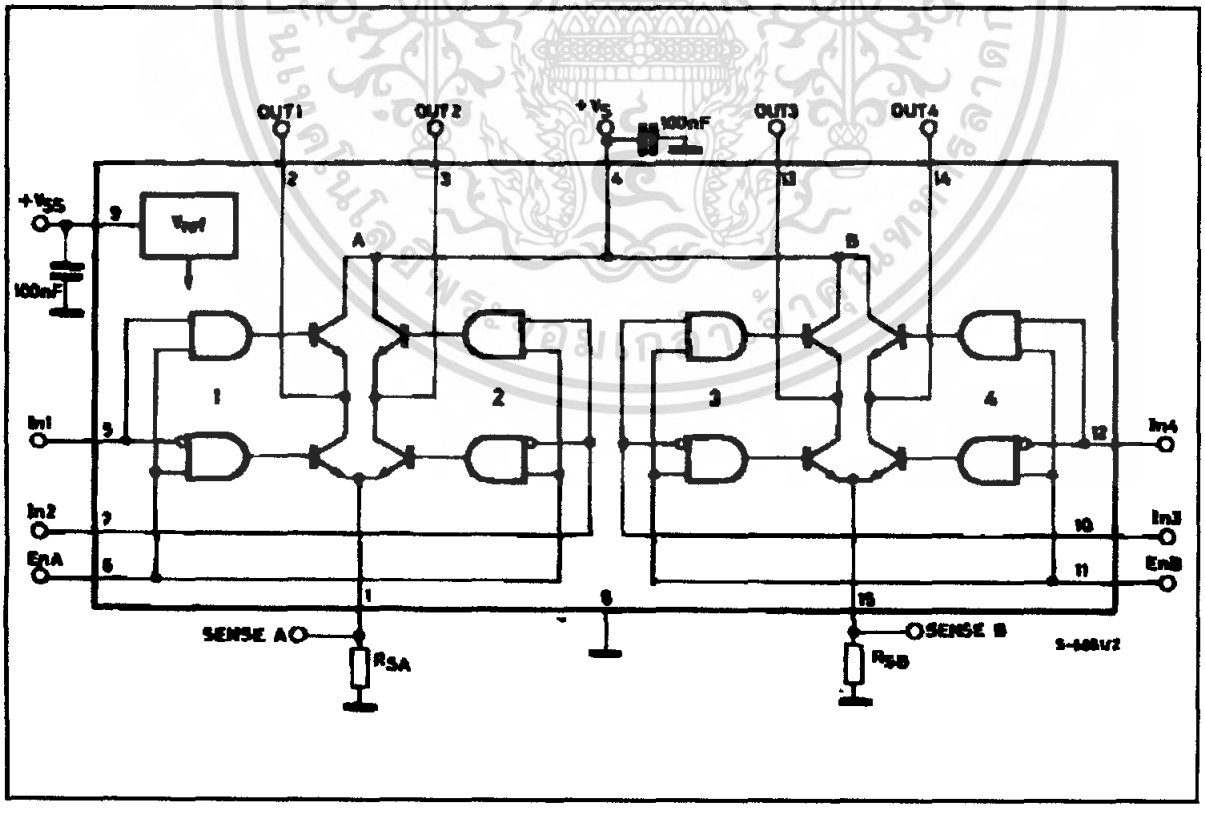
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

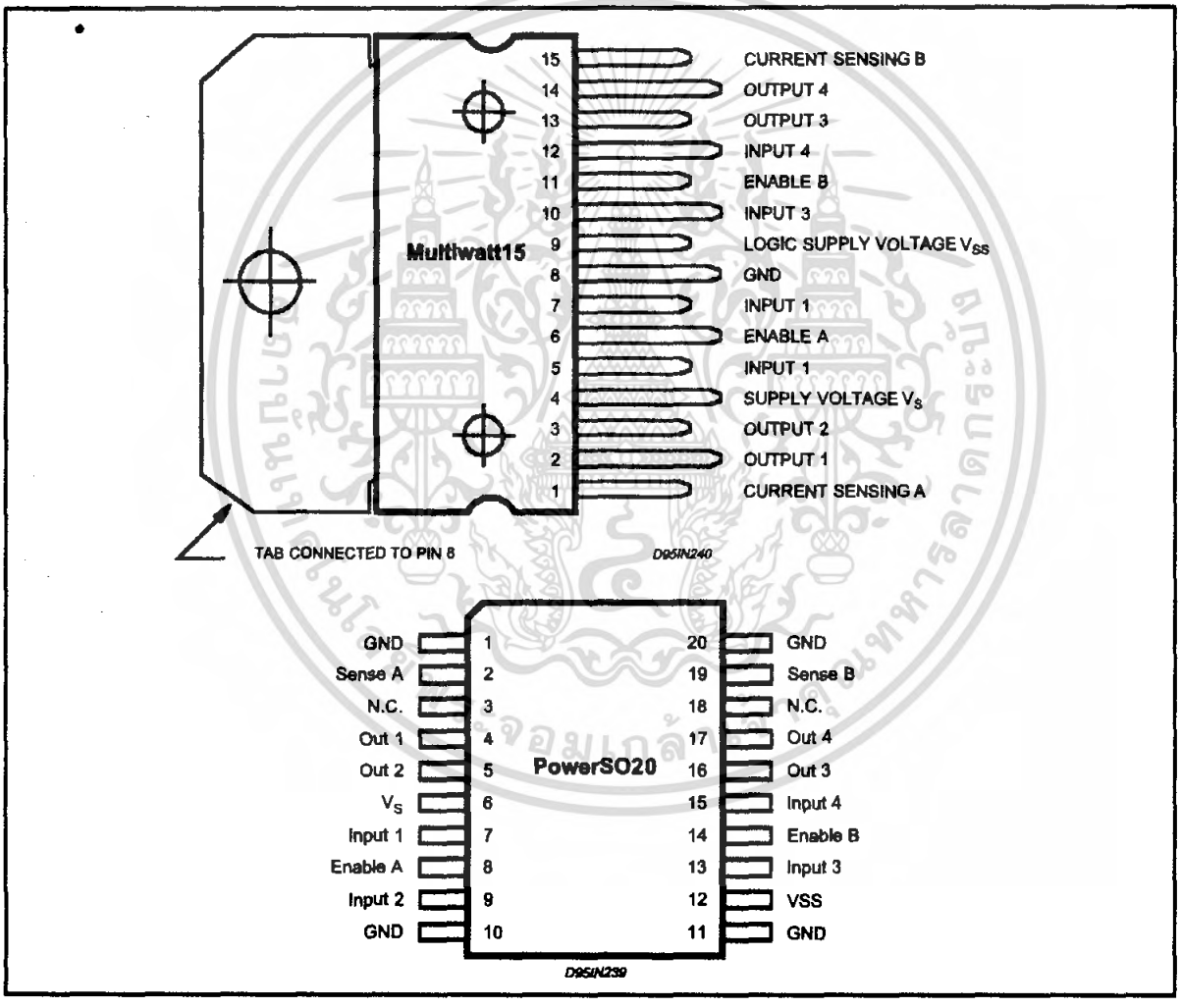


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

SGS-THOMSON MICROELECTRONICS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition		V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)			4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0	V_i = L		13	22	mA
			V_i = H		50	70	mA
		V_{en} = L	V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0	V_i = L		24	36	mA
			V_i = H		7	12	mA
		V_{en} = L	V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)			–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)			2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L				–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V			30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)			–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)			2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L				–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V			30	100	µA
$V_{CEsat(H)}$	Source Saturation Voltage	I_L = 1A			1.35	1.7	V
		I_L = 2A			2	2.7	V
$V_{CEsat(L)}$	Sink Saturation Voltage	I_L = 1A (5)			1.2	1.6	V
		I_L = 2A (5)			1.7	2.3	V
V_{CEsat}	Total Drop	I_L = 1A (5)				3.2	V
		I_L = 2A (5)				4.9	V
V_{sens}	Sensing Voltage (pins 1, 15)			–1 (1)		2	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs
fc (V_{en})	Commutation Frequency	I_L = 2A		1		KHz

1) 1)Sensing voltage can be –1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ – 0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.
5) PIN 1 and PIN 15 connected to GND.

Figure 1 : Typical Saturation Voltage vs. Output Current.

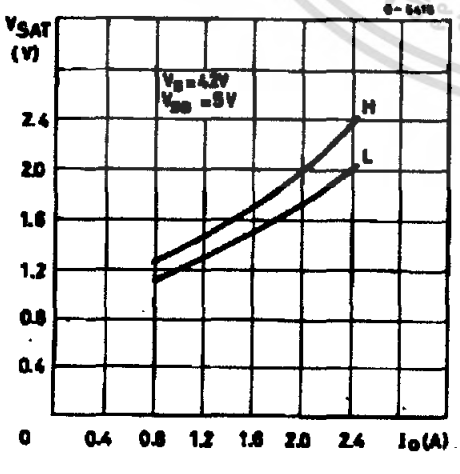
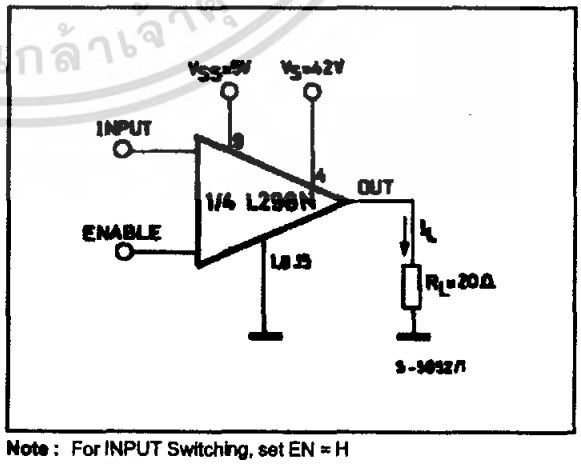


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

SGS-THOMSON MICROELECTRONICS
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

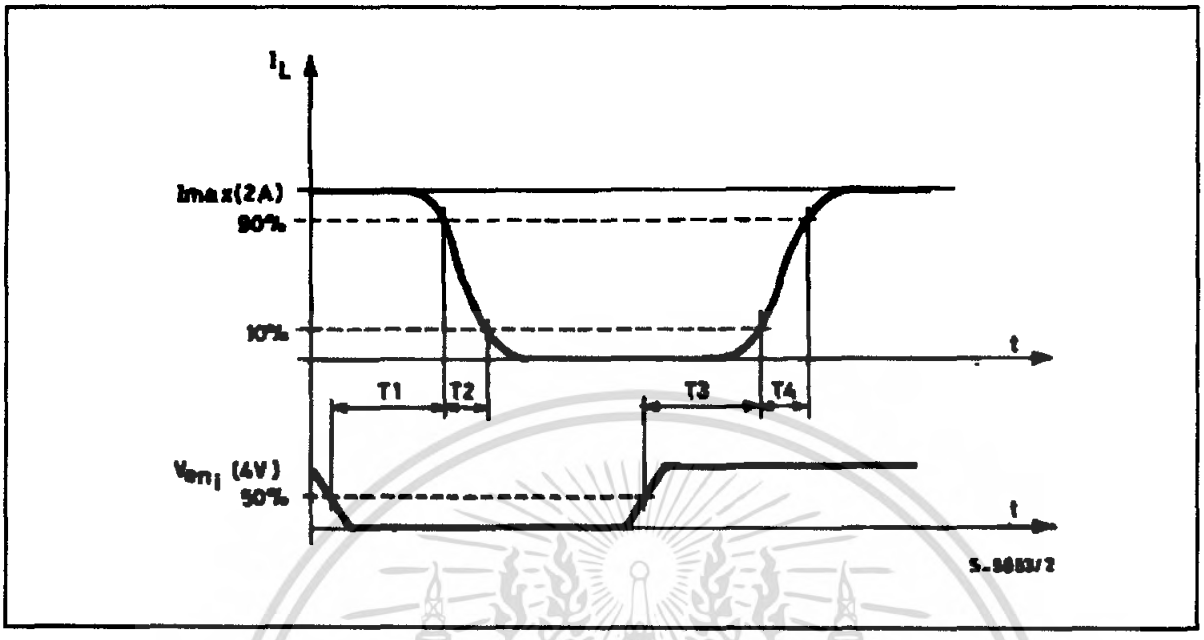
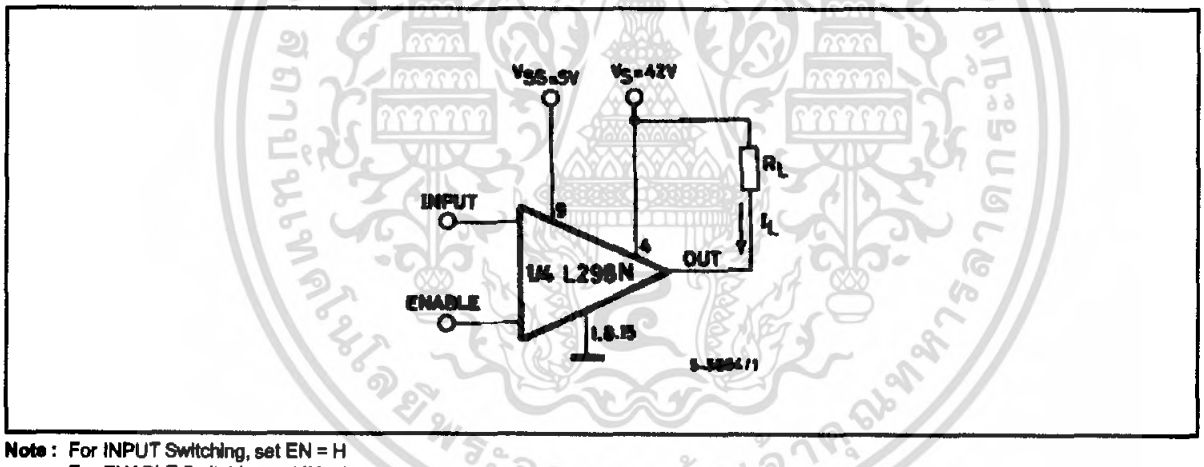


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

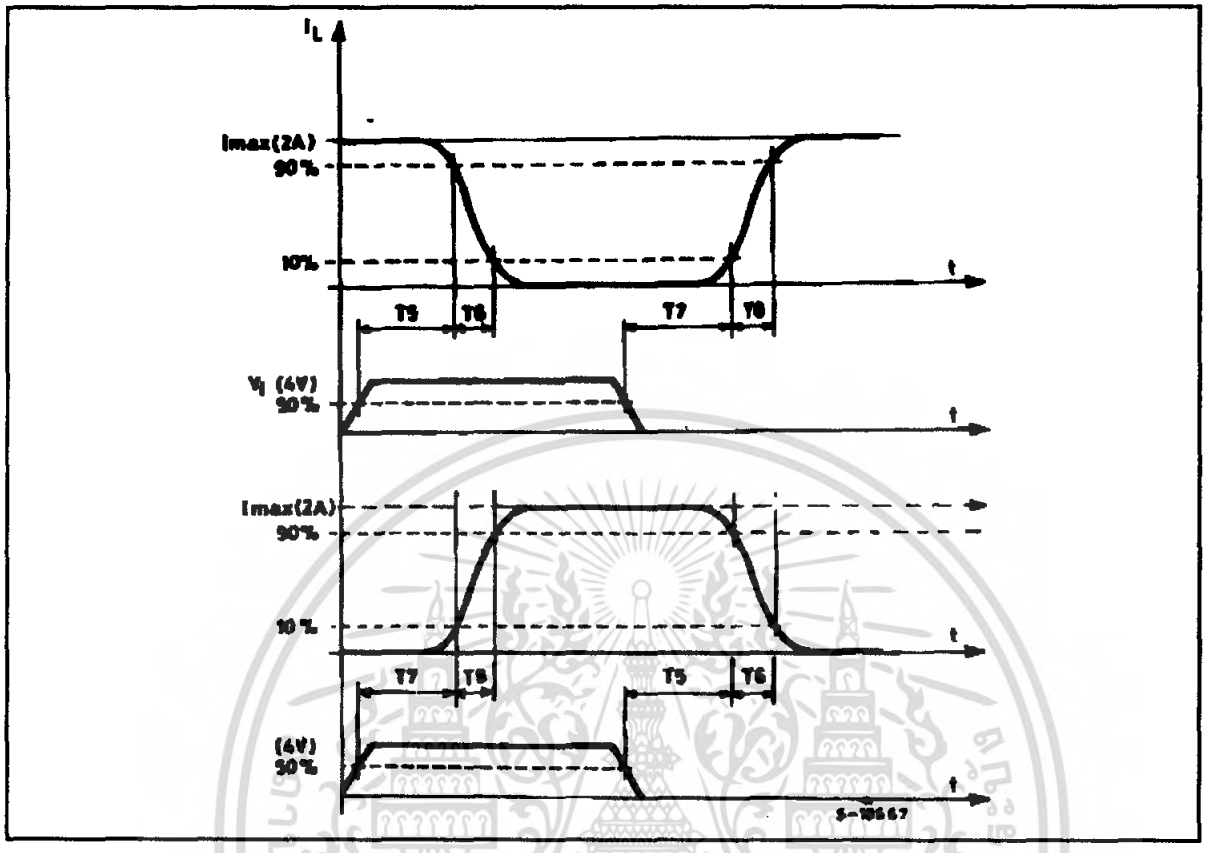
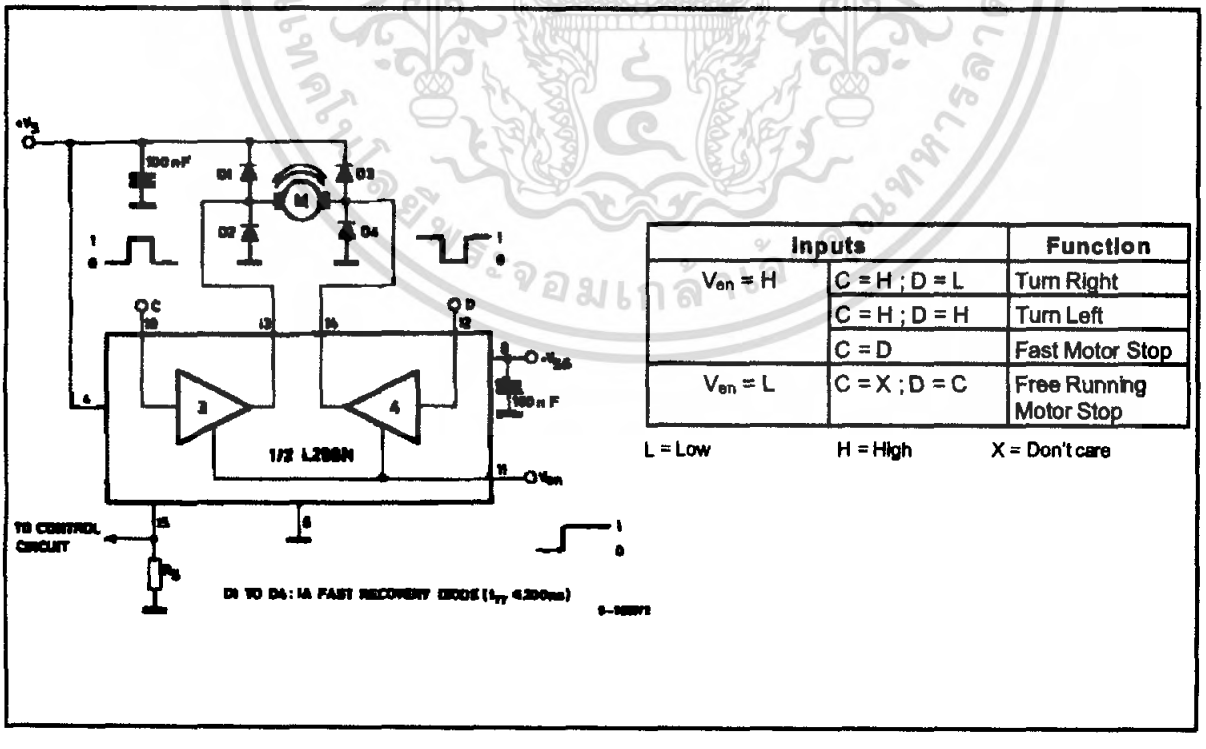


Figure 6 : Bidirectional DC Motor Control.



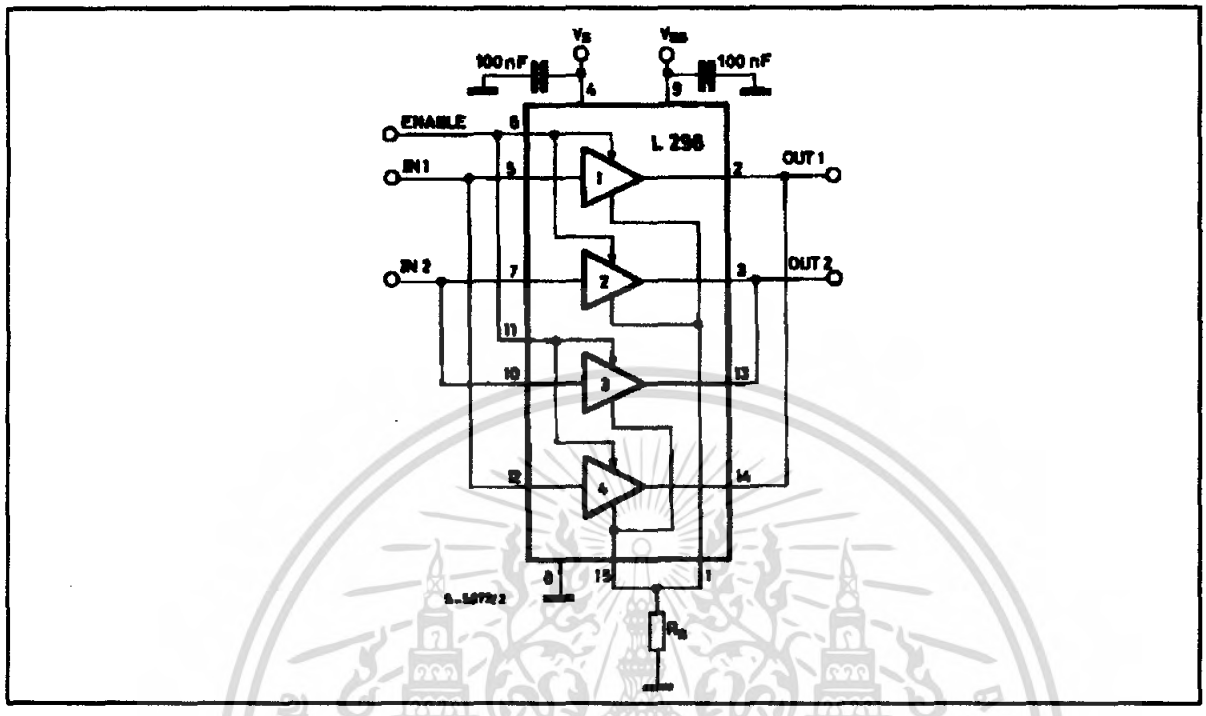
Inputs		Function
V_{en} = H	C = H ; D = L	Turn Right
	C = H ; D = H	Turn Left
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = C	Free Running Motor Stop

L = Low H = High X = Don't care

SGS-THOMSON MICROELECTRONICS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

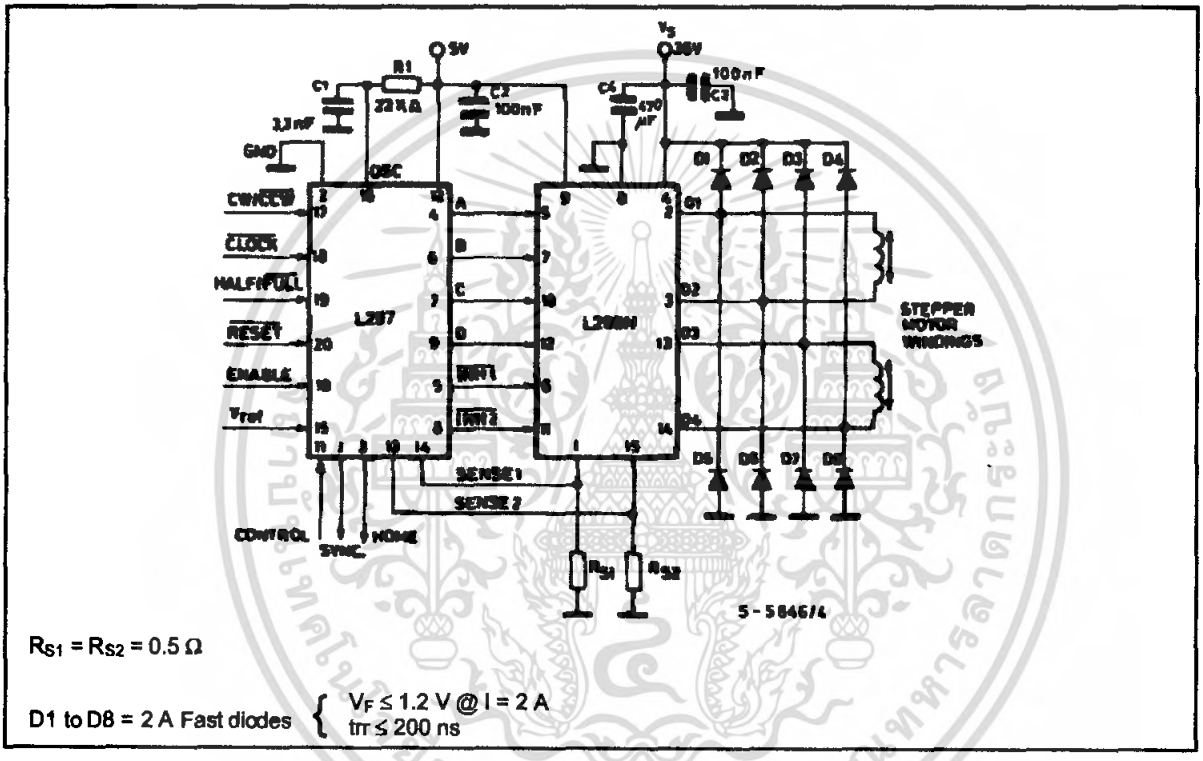
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\ \Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2$ V @ I = 2 A ; $t_{rr} \leq 200$ ns

SGS-THOMSON MICROELECTRONICS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้