สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวิเคราะห์สเปคตรัม

นายจักรพันธ์ การุณกิตติสาร

วพ.
ว2250ค
2550

เลขหมู่.....
เลขทะเบียน 81596
วัน,เดือน,ปี 19 มี.ย. 2551

b 11193224
i.....

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา พ.ศ. 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Spectrum Analyzer

Mr. Jakaphun Karoonkittisarn

A Special Project Submitted in Partial Fulfillment of Requirement for the Degree of

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabung

Academic Year 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง	เครื่องวิเคราะห์สเปคตรัม
นักศึกษา	นายจักรพันธ์ การุณกิตติสาร
ภาควิชา	ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**
สาขาวิชา	ฟิสิกส์ประยุกต์ – เครื่องมือวิทยาศาสตร์และอุตสาหกรรม
อาจารย์ที่ปรึกษา	รองศาสตราจารย์วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ		ลายมือชื่อ
ประธานกรรมการ	ผศ. สาหร่าย เล็กชะอุ่ม	
กรรมการ	ดร. ประธาน บุรณะศิริ	
กรรมการ	อ. กาจปัญญา สุวรรณสุโข	
กรรมการที่ปรึกษา	รศ. วิชิต ศิริโชติ	

.....

(รองศาสตราจารย์ วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง เครื่องวิเคราะห์สเปคตรัม
นักศึกษา นายจักรพันธ์ การุณกิตติสาร
ภาควิชา ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**
สาขาวิชา ฟิสิกส์ประยุกต์ – เครื่องมือวิทยาศาสตร์และอุตสาหกรรม
อาจารย์ที่ปรึกษา รองศาสตราจารย์วิชิต ศิริโชติ

บทคัดย่อ

การพัฒนาเครื่องมือสำหรับวิเคราะห์สเปคตรัม วงจรสร้างขึ้นด้วยตัวประมวลสัญญาณดิจิตอลของบริษัทไมโครชิฟเบอร์ dsPIC30F6015 และแอลซีดีแสดงผลกราฟิก NOKIA 3310 อัลกอริทึมที่ใช้วิเคราะห์สเปคตรัมเป็นแบบการแปลงฟูริเยร์ทรานส์ฟอร์มจำนวน 256 จุด โปรแกรมที่ใช้จับสัญญาณและคำนวณ FFT ได้พัฒนาขึ้นด้วยคอมไพเลอร์ Mikro-C dsPIC V 3.0 อัตราการสุ่มสัญญาณอินพุตค่าเท่ากับ 3400 จุดต่อวินาที สัญญาณที่ใช้สำหรับทดสอบเป็นคลื่นรูปไซน์และรูปสี่เหลี่ยม มีความถี่ตั้งแต่ 20Hz – 100Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Spectrum Analyzer
Name	Mr. Jakaphun Karoonkittisarn
Department	Applied Physics **Faculty of Science**
Program	Applied Physics – Science and Industry Instrumentation
Academic Year	2007
Special Project Advisor	Assoc. Prof. Wichit Sirichot

ABSTRACT

A device used for spectral analysis has been developed. The circuit is built with the Microchip Digital Signal Controller, dsPIC30F6015 and the Nokia 3310 graphic LCD display. The spectral analysis algorithm is based on a 256-point Fast Fourier Transform. The firmware for signal capturing and FFT computing has been developed using the Mikro-C compiler for dsPIC V3.0. The sampling rate of input signal is 3400 sample per second. The test signals are sine and square waves with frequency of 20-100Hz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานพิเศษนี้ได้สำเร็จลุล่วงไปได้ด้วยดี ต้องขอขอบคุณ รองศาสตราจารย์วิชิต ศิริโชติ อาจารย์ที่ปรึกษาโครงงานพิเศษนี้ โดยให้ทั้งความรู้ คำปรึกษาวิธีการแก้ปัญหาต่างๆเรื่อยมา และขอบคุณอาจารย์ทุกท่านและรุ่นพี่และเพื่อนๆทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้ความรู้แก่ข้าพเจ้า สะสมเพิ่มพูนความรู้จนสามารถนำมาประยุกต์ทำโครงงานพิเศษนี้ขึ้นมาได้ และสุดท้ายขอขอบพระคุณ คุณพ่อ คุณแม่ ที่ให้การสนับสนุนดูแลห่วงใยข้าพเจ้าตั้งแต่เริ่มต้นเรื่อยมา จนโครงงานพิเศษนี้สำเร็จ

นายจักรพันธ์ การุณกิตติสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการวิจัย	1
1.4 ขั้นตอนการวิจัยและดำเนินการ	1
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 dsPIC	3
2.1.1 คุณสมบัติของซีพียู	3
2.1.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล	4
2.1.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ	4
2.2 LCD NOKIA 3310	5
2.2.1 หลักการทำงานของ LCD NOKIA 3310	6
2.2.2 การระบุตำแหน่งบนหน้าจอแสดงผลของ LCD NOKIA 3310	6
2.3 การแปลงฟูริเยร์ (fourier Transform)	7
2.4 Discrete Fourier Transform	7
2.5 Fast Fourier Transform	9
2.6 หน้าต่างกรองความถี่	13
บทที่ 3 วิธีการดำเนินงานวิจัย	16
3.1 ฮาร์ดแวร์	16
3.2ซอฟแวร์	18
3.3การทดลองทางทฤษฎี	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่4 ผลการทดลองและอภิปรายผล	22
4.1 การทดลองแปลงฟูริเยร์แบบรวดเร็วโดยใช้dsPIC30F6015	22
บทที่5 สรุปและข้อเสนอแนะ	32
5.1 ข้อเสนอแนะและแนวทางการพัฒนา	32
บรรณานุกรม	33
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงขาสัญญาณของจอ LCD 3310 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 ชิปไมโครคอนโทรลเลอร์ dsPIC30F6015 3
รูปที่ 2.2 ขนาดของ LCD NOKIA 3310 5
รูปที่ 2.3 ขาสัญญาณที่ใช้ทั้งหมด 5
รูปที่ 2.4 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 8 จุด เป็นการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด 11
รูปที่ 2.5 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 8 จุด เป็นการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด หลังจากใช้คุณสมบัติของความสมมาตร 12
รูปที่ 2.6 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด เป็นการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 2 จุด 12
รูปที่ 2.7 แผนภาพรวมของการคำนวณการแปลงฟูริเยร์แบบรวดเร็ว 8 จุด 13
รูปที่ 2.8 ลักษณะหน้าต่างชนิดต่างๆ 15
รูปที่ 3.1 แผนผังการออกแบบฮาร์ดแวร์ 16
รูปที่ 3.2 วงจรยกระดับแรงดันไฟฟ้า 16
รูปที่ 3.3 รูปบอร์ดทดลอง dsPIC30F4011 17
รูปที่ 3.4 รูปบอร์ดทดลอง dsPIC30F6015 17
รูปที่ 3.5 การทำงานของโปรแกรม 18
รูปที่ 3.6 ลักษณะการกรองสัญญาณ 18
รูปที่3.7 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 1 คาบ 19
รูปที่3.8 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 2 คาบ 19
รูปที่3.9 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 5 คาบ 19
รูปที่3.10 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz ที่เก็บข้อมูลได้ไม่ครบคาบเป็นจำนวนเต็ม 20
รูปที่3.11 การแปลงฟูริเยร์ของสัญญาณจากรูปที่3.4 เมื่อนำมาผ่านหน้าต่างกรองความถี่ 20
รูปที่3.12 การแปลงฟูริเยร์ของคลื่นรูปไซน์ 100Hz 300Hz 500Hz 700Hz มาบวกกัน 20
รูปที่3.13 การแปลงฟูริเยร์ของคลื่นรูปสี่เหลี่ยม 100 Hz จำนวน 2 คาบ 21
รูปที่3.14 การแปลงฟูริเยร์ของคลื่นรูปสี่เหลี่ยม100 Hz ที่เก็บข้อมูลได้ไม่ครบคาบเป็นจำนวนเต็ม 21
รูปที่3.15 การแปลงฟูริเยร์ของสัญญาณจากรูปที่3.8 เมื่อนำมาผ่านหน้าต่างกรองความถี่ 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่4.1 แสดงลักษณะแผงวงจร dsPIC30F6015 22
รูปที่4.2 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 15Hz 30Hz และ 45 Hz รวมกัน 23
รูปที่4.3 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 23
รูปที่4.4 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 20 Hz 23
รูปที่4.5 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 20 Hz โดยซอฟแวร์ 24
รูปที่4.6 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 30 Hz 24
รูปที่4.7 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 30 Hz โดยซอฟแวร์ 24
รูปที่4.8 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 40 Hz 25
รูปที่4.9 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 40 Hz โดยซอฟแวร์ 25
รูปที่4.10 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 50 Hz 25
รูปที่4.11 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 50 Hz โดยซอฟแวร์ 26
รูปที่4.12 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 80 Hz 26
รูปที่4.13 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 80 Hz โดยซอฟแวร์ 26
รูปที่4.14 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 20 Hz 27
รูปที่4.15 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 20 Hz โดยซอฟแวร์ 27
รูปที่4.16 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 30 Hz 27
รูปที่4.17 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 30 Hz โดยซอฟแวร์ 28
รูปที่4.18 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 40 Hz 28
รูปที่4.19 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 40 Hz โดยซอฟแวร์ 28
รูปที่4.20 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 50 Hz 29
รูปที่4.21 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 50 Hz โดยซอฟแวร์ 29
รูปที่4.22 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 80 Hz 29
รูปที่4.23 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 80 Hz โดยซอฟแวร์ 30
รูปที่4.24 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ในเครื่องออสซิลโลสโคป 30
รูปที่4.25 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยมในเครื่องออสซิลโลสโคป 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

สัญญาณทางไฟฟ้านั้น ไม่สามารถมองเห็นได้ด้วยตาเปล่าต้องอาศัยเครื่องมือในการสังเกตและวิเคราะห์สัญญาณ เพื่อที่จะนำไปศึกษาวิจัย ในปัจจุบันเครื่องมือที่ใช้ดูและวิเคราะห์ก็จะเป็นเครื่องมือสำเร็จรูปหรือโปรมแกรมสำเร็จรูปบนระบบปฏิบัติการคอมพิวเตอร์

ในการวัดสัญญาณและนำสัญญาณมาวิเคราะห์นั้นค่อนข้างทำได้ยุ่งยาก ซึ่งปกติแล้วจะใช้คอมพิวเตอร์หรือออสซิลโลสโคป ในการวัดความถี่ของสัญญาณ และวิเคราะห์สัญญาณที่เข้ามาแต่มีปัญหาอยู่ตรงที่ข้อมูลที่จะให้คอมพิวเตอร์คำนวณนั้นมีความยุ่งยากที่จะส่งเข้าไปและคอมพิวเตอร์กับออสซิลโลสโคป เองก็มีขนาดใหญ่ไม่สะดวกในการเคลื่อนย้าย

จึงมีความคิดที่จะนำไมโครคอนโทรลเลอร์มาประยุกต์ใช้ สร้างเป็นเครื่องมือสำหรับวัดหาความถี่ของสัญญาณและวิเคราะห์สัญญาณจากนั้นก็แสดงผลออกมาทางหน้าจอ LCD NOKIA 3310

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการประยุกต์นำเอาไมโครคอนโทรลเลอร์ dsPIC ไปใช้งานได้จริง

1.2.2 เพื่อศึกษาการทำงานของ LCD 3310

1.2.3 เพื่อศึกษาการเขียนโปรแกรมสำหรับ dsPIC

1.2.4 เพื่อศึกษาการแปลงฟูริเยร์แบบรวดเร็ว

1.2.5 ฝึกการคิดและแก้ปัญหาอย่างมีระบบ

1.2.6 เพื่อสร้างอุปกรณ์ในการวิเคราะห์สัญญาณให้มีขนาดเล็กและราคาถูก

1.3 ขอบเขตการวิจัย

1.3.1 สร้างแผงวงจรไมโครคอนโทรลเลอร์ dsPIC30F6015 เชื่อมต่อกับ LCD 3310

1.3.2 สามารถวิเคราะห์สัญญาณความถี่ต่ำ

1.4 ขั้นตอนการวิจัยและดำเนินการ

1.4.1 ศึกษาเกี่ยวกับ dsPIC

1.4.2 ศึกษาการทำงานของ LCD NOKIA 3310

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.3 ออกแบบวงจรเบื้องต้นสำหรับ dsPIC

1.4.4 ออกแบบโปรแกรมให้ dsPIC สามารถที่จะเก็บสัญญาณและจัดรูปแบบข้อมูล

1.4.5 ศึกษาเกี่ยวกับการแปลงฟูริเยร์แบบรวดเร็ว

1.4.6 ออกแบบโปรแกรมติดต่อกับ LCD 3310

1.4.7 ทำการวิเคราะห์สัญญาณด้วยวิธีการแปลงฟูริเยร์แบบรวดเร็วโดยใช้ dsPIC

1.4.8 ออกแบบเครื่องมือและสร้างเครื่องมือสำหรับวิเคราะห์สัญญาณ

1.4.9 ทดสอบวงจรทั้งหมด

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ความรู้เกี่ยวกับการออกแบบและประยุกต์ใช้งานdsPIC

1.5.2 เข้าใจถึงการเขียนโปรแกรมสำหรับdsPIC

1.5.3 ได้เครื่องมือสำหรับวิเคราะห์สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและหลักการ

2.1 dsPIC

dsPIC คือชื่อของไมโครคอนโทรลเลอร์ 16 บิตจาก Microchip Technology Inc. ไมโครคอนโทรลเลอร์อนุกรมนี้เรียกว่า Digital Signal Controller หรือ DSC นั่นหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิตอล สำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

รูปที่ 2.1 ชิปไมโครคอนโทรลเลอร์ dsPIC30F6015

2.1.1 คุณสมบัติของซีพียู

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูแบบ RISC
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- มี 84 คำสั่งภาษาแอสเซมบลีมาตรฐาน รองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง สามารถป้องกันการอ่านได้ และสามารถโปรแกรมตัวเอง โดยใช้กระบวนการทางซอฟต์แวร์
- มีหน่วยความจำข้อมูลอีอีพรอมที่สามารถลบและเขียนใหม่ได้มากกว่า 1,000,000 ครั้ง
- มีอินเตอร์รัปต์เวกเตอร์จำนวนมาก จึงรองรับการตอบสนองสัญญาณอินเตอร์รัปต์ได้ดี
- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดโปรแกรมได้
- มีวอตช์ด็อกไทเมอร์แบบโปรแกรมได้
- มีวงจรตรวจสอบการทำงานของวงจรกำเนิดสัญญาณนาฬิกา
- รองรับการโปรแกรมในวงจรแบบอนุกรม (ICSP : In – Circuit Programming)

2.1.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล

- มีเอกคิวมูเลเตอร์ขนาด 40 บิต 2 ตัว รองรับการประมวลผลทางคณิตศาสตร์ซึ่งทำให้ประมวลผลได้รวดเร็ว
- มีหน่วยประมวลผลด้านการคูณและหารเลข 17 บิตในรูปของฮาร์ดแวร์ จึงทำให้สามารถคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลข 16 บิตได้ภายในสัญญาณนาฬิกาเพียง 1 ไซเกิล
- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่จำนวนบิตมาก ๆ สามารถทำได้อย่างรวดเร็ว
- มีวงจรเฟตซ์ข้อมูลคู่ จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

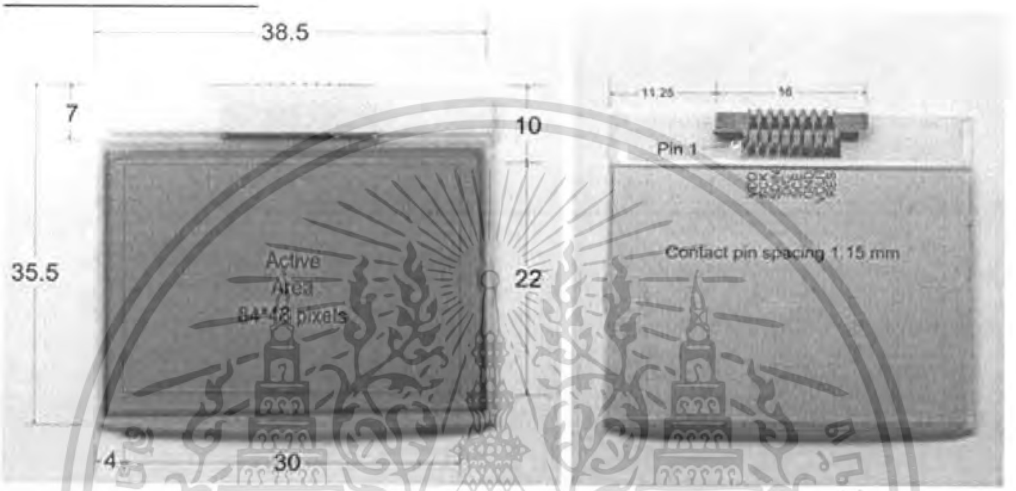
2.1.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ตได้ 25mA ทั้งแบบกระแสซิงก์และซอร์ส
- ไทเมอร์/เคาน์เตอร์มีขนาด 16 บิตไม่น้อยกว่า 3 ตัวต่อใช้งานร่วมกันเป็นไทเมอร์ 32บิต
- มีโมดูลตรวจจับและเปรียบเทียบสัญญาณดิจิตอล
- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมทั้งแบบ SPI และผ่านระบบบัส 12C
- มีโมดูลสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO
- มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ความละเอียด 10บิต
- มีโมดูลสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

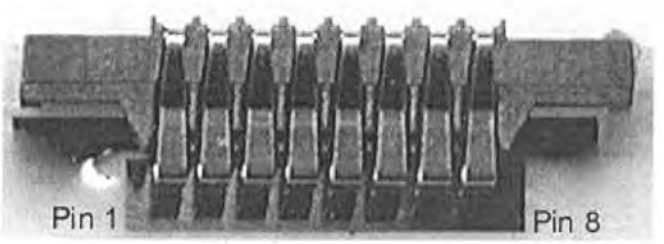
2.2 LCD NOKIA 3310

LCD NOKIA 3310 นี้เรียกตามบริษัทผู้ผลิต (Manufacturer Part Number) คือ LPH7779 ซึ่งผลิตโดย Philips Semiconductor โดยมีไอซี PCD8544 รวมอยู่แล้วในตัวเดียว การติดต่อจะใช้รูปแบบของ SPI communication ซึ่งย่อมาจาก Serial Peripheral Interface



รูปที่ 2.2 ขนาดของ LCD NOKIA 3310

ตามรูปที่ 2.2 หน่วยในรูปเป็น มิลลิเมตร (mm)
ทั้งชิ้นอุปกรณ์มีความ กว้าง × ยาว = 38.5 mm × 35.5 mm
ตัวหน้าจอแสดงผล กว้าง × ยาว = 30 mm × 22 mm
โดยมีความละเอียดหน้าจอ (resolution) เท่ากับ 84 × 48 pixels
ส่วนขนาดของช่อง Pin มีความยาว 16 mm



รูปที่ 2.3 ขาสัญญาณที่ใช้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงขาสัญญาณของจอ LCD 3310

PIN	Signal	Description	Port
1	VDD	ขาไฟเลี้ยงสำหรับ LCD โดยใช้ voltage ช่วง 2.7 – 7 volt	Power
2	SCLK	ขาสัญญาณ clock ของ LCD ซึ่งกำหนดค่า max สุดคือ 4 Mbits/s	Input
3	SDIN	ย่อมาจาก serial data input เป็นขาที่ใช้ในการรับข้อมูล เพื่อแสดงผลที่หน้าจอ LCD	Input
4	D/C	ขาที่ใช้เลือกควบคุม ระหว่าง ให้รับข้อมูล data(D) / ส่งคำสั่ง command(C) ซึ่ง active low	Input
5	SCE	ขาที่ enable ตัวจอ LCD ให้ทำงาน(active low)	Input
6	GND	ขา ground	Power
7	VOUT	ขาไฟเลี้ยงขาออกของ LCD โดยขานี้ต้องต่อ electrolytic capacitors จาก VOUT ไปยัง GND โดยใช้ค่าได้ตั้งแต่ 1 uF ถึง 10 uF	Power
8	RES	ขา reset (active low)	Input

2.2.1 หลักการทำงานของ LCD NOKIA 3310

แบ่งการทำงานออกเป็น 2 โหมด คือ คำสั่งหมด(C) กับ ข้อมูลโหมด(D) โดยอาศัยขาสัญญาณ D/C เพื่อเลือกโหมด มีรายละเอียดโหมด ดังนี้

1) ข้อมูลโหมด ใช้สำหรับ รับข้อมูล เพื่อนำข้อมูลไปแสดงผลบนหน้าจอ

2) คำสั่งโหมด ใช้สำหรับ รับคำสั่ง (active low) เพื่อกำหนดตำแหน่งของการเริ่มต้นเขียนข้อมูลและ กำหนดค่า config ต่างๆให้แก่ LCD นอกจากนั้นยังควบคุมให้แสดงผลปกติ หรือแบบ Inverse ได้ด้วย

2.2.2 การระบุตำแหน่งบนหน้าจอแสดงผลของ LCD NOKIA 3310

หน้าจอของ LCD จะประกอบไปด้วยจำนวนจุดทั้งหมดเท่ากับ 48 x 84 pixels เป็นจำนวนคอลัมน์ (column)ได้เท่ากับ 84 คอลัมน์ และ จำนวนแถว (row) 48 แถวด้วยกัน โดยแถวทั้ง 48 แถวนั้น จะแสดงผลออกมาได้เป็น 6 บรรทัด เนื่องจากไอซี PCD8544 เก็บไฟล์ทีละ 1 byte หรือ 8 bits มาแสดงผล โดยเก็บข้อมูลจากซ้ายไปขวา บนสุดมาล่างสุด ดังนั้น 48 pixels คือ 48 bits ทำให้เรามีจำนวนบรรทัด สำหรับแสดง ผลเท่ากับ 48 / 8 = 6 บรรทัด เมื่อ LCD 3310 รับข้อมูลเพื่อแสดงผลจะรับแบบต่อเนื่อง จากบรรทัดแรกบนสุดแสดงจนครบทั้ง 84คอลัมน์ จากซ้ายไปขวาก่อน จึงขึ้นแถวที่ 2 แล้วแสดงให้ครบทั้ง 84 คอลัมน์ไปเรื่อย ๆ จนครบ 6 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การแปลงฟูริเยร์ (fourier Transform)

ในระบบสัญญาณต่อเนื่องการวิเคราะห์และการสังเคราะห์นั้นทำได้ยุ่งยาก วิธีการหนึ่งที่ได้รับความนิยมก็คือการแปลงจากโดเมนหนึ่งไปอีกโดเมนหนึ่ง ซึ่งการแปลงฟูริเยร์ก็เป็นอีกวิธีหนึ่งที่รู้จักกันดี

การแปลงฟูริเยร์นิยามโดย

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{i\omega\omega}dt$$

การแปลงฟูริเยร์ คือการแปลงจาก x(t) ซึ่งเป็น time domain ไปเป็น X(ω) ซึ่งเป็น frequency domain ส่วนการแปลงกลับ(inverse Fourier Transform) เป็นการแปลงจาก X(ω)ไปเป็น x(t)

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega$$

x(t) จะเป็นสัญญาณจริง แต่ผลจากการแปลงฟูริเยร์คือ X(ω) จะเป็นจำนวนเชิงซ้อน ประกอบด้วยจำนวนจริงและจำนวนจินตภาพที่เขียนในรูป Rectangular form หรือ Polar form ก็ได้

$$X(\omega) = R(\omega) + jI(\omega) = Ae^{j\phi}$$

โดยที่

R(ω) คือส่วนที่เป็นค่าจริง

I(ω) คือส่วนที่เป็นค่าจินตภาพ

A คือขนาดหรือamplitude

Φ คือphase angle

2.4 Discrete Fourier Transform

การแปลงฟูริเยร์แบบไม่ต่อเนื่องมีความหมายเหมือนกับอนุกรมฟูริเยร์แบบไม่ต่อเนื่อง ซึ่งมีสมการใช้งานเหมือนกัน ที่ต่างกันคือที่มาและความหมายทั้ง 2 การแปลงฟูริเยร์แบบไม่ต่อเนื่อง ก็คือ อนุกรมฟูริเยร์แบบไม่ต่อเนื่องเพียงคาบเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณา x(t) เป็นสัญญาณต่อเนื่องซึ่งมีคาบเวลา T คือ

$x(t) = x(t + mT_0)$ m คือจำนวนเต็มบวกใดๆ

x(t) สามารถเขียนแทนด้วยอนุกรมฟูริเยร์ โดย C(k) คือสัมประสิทธิ์ฟูริเยร์

$$x(t) = \sum_{k=-\infty}^{k=+\infty} x(k)e^{jk\omega_0 t} = \sum_{k=-\infty}^{k=+\infty} x(k)e^{\frac{j2\pi kt}{T_0}}$$

f0=1/T0 คือความถี่มูลฐาน (Fundamental Frequency) ค่าสัมประสิทธิ์ฟูริเยร์ (Fourier Coefficient C(k)) กำหนดโดย

$$C(k) = \frac{1}{T}\int_0^T x(t)e^{\frac{-j2\pi kt}{T_0}}\,dt$$

ใน 1 คาบเวลาหากสัญญาณ x(t) ถูกสุ่มเป็นจำนวน N ครั้ง สัญญาณที่ถูกสุ่มจะเขียนได้เป็น

$$x'(t) = x(t/T) = \sum_{n=0}^{N-1} x(t/T) \bullet \delta(t/T - n)$$

ค่า t/T ถือว่าเป็นnormalize time เมื่อสัญญาณ x(t) เป็นสัญญาณมีคาบเวลา สัญญาณ x'(t) ก็เป็นสัญญาณมีคาบเวลาเหมือนกัน ดังนั้นสัมประสิทธิ์ฟูริเยร์ของ x'(t)

$$X(k) = \frac{1}{N}\int_{k=0}^{N-1} x(t/T)e^{\frac{-2j\pi k}{N}t/T}\,d(t/T)$$

$$X(k) = \frac{1}{N}\int_{k=0}^{N-1}\sum_{n=0}^{N-1} x(t/T)\delta(t/T - n)e^{\frac{-2\pi k}{N}t/T}\,d(t/T)$$

จากคุณสมบัติของฟังก์ชั่น δ(t) ที่ว่า $\int_{-\infty}^{\infty} f(\tau)\delta(\tau - n)d\tau = f(n)$

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{\frac{-2\pi nk}{N}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก x(t) และx(n) ต่างมีคาบ และ X(k) ก็มีคาบเช่นเดียวกับ x(n) คือ N

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{\frac{2j\pi nk}{N}}$$

ให้ $e^{\frac{-j2\pi nk}{N}}=W_N^{nk}$ จึงเขียนได้เป็น

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{\frac{2j\pi nk}{N}}=\sum_{n=0}^{N-1}x(n)W_N^{nk}$$

เขียนในรูปเมตริกได้เป็น

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\\vdots\\X(N-1)\end{bmatrix}=\begin{bmatrix}W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0\\W_N^0 & W_N^{1\times1} & W_N^{1\times2} & \cdots & W_N^{1\times(N-1)}\\W_N^0 & W_N^{2\times1} & W_N^{2\times2} & \cdots & W_N^{2\times(N-1)}\\\vdots & \vdots & \vdots & & \vdots\\W_N^0 & W_N^{(N-1)} & W_N^{(N-1)\times2} & \cdots & W_N^{(N-1)\times(N-1)}\end{bmatrix}\bullet\begin{bmatrix}x(0)\\x(1)\\x(2)\\\vdots\\x(N-1)\end{bmatrix}$$

2.5 Fast Fourier Transform

เนื่องจากการแปลงฟูริเยร์แบบไม่ต่อเนื่องมีประโยชน์มากจึงได้มีการคิดวิธีคำนวณให้เร็วขึ้น เรียกว่าการแปลงฟูริเยร์แบบรวดเร็วซึ่งในการแปลงฟูริเยร์แบบรวดเร็วก็มีหลายวิธีด้วยกันเนื่องจากโครงสร้างการคำนวณทางคณิตศาสตร์ของการแปลงฟูริเยร์แบบรวดเร็ว วิธีระบบฐาน 2 แบบแตกเป็นส่วนย่อยทางฝั่งเวลามีความซับซ้อนไม่มากนัก รวมทั้งตัวแปรต่างๆ มีจำนวนน้อย ทำให้ง่ายต่อการนำมาออกแบบ เมื่อเทียบกับการแปลงฟูริเยร์แบบรวดเร็ววิธีอื่นๆ ด้วยเหตุนี้จึงสามารถออกแบบให้มีโครงสร้างการทำงานที่มีขนาดเล็ก และใช้พลังงานต่ำได้ สามารถอธิบายการคำนวณของการแปลงฟูริเยร์แบบรวดเร็ว วิธีระบบฐาน 2 แบบแตกเป็นส่วนย่อยทางฝั่งเวลา ได้ดังนี้

จากการแปลงฟูริเยร์แบบไม่ต่อเนื่อง

$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn}$ โดยที่ $W_N=e^{-j2\pi/N}$ และ k=0,1,2,…., N-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ N เป็นเลขคู่ ทำให้สามารถกระจาย X(k) ให้อยู่ในรูปของผลบวกของเทอมที่ n เป็นคู่และเทอมที่ n เป็นคี่ได้

$$X(k)= \sum_{n=0}^{N/2-1} x(2n)W_N^{2kn} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k}$$

เทอมคู่ เทอมคี่

$$X(k)= \sum_{n=0}^{N/2-1} x(2n)W_N^{2kn} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{2nk}W_N^{k}$$

พิจารณาเทอม W_N^{ab} ที่มี a และ b เป็นจำนวนใดๆ ที่ไม่เท่ากับ 0 จะพบว่า สามารถย้ายตัวยกกำลังของ W ไปเป็นตัวหารของ N ได้ดังนี้

$$W_N^{ab} = e^{-j\frac{2\pi}{N}ab} = e^{-j\frac{2\pi}{N/b}a} = W_{N/b}^{a}$$

จึงใช้ความจริงข้อนี้แทนค่าเทอม W_N^{2nk} ด้วย $W_{N/2}^{nk}$ ได้จะเห็นได้ว่า X(k) ได้กลายเป็นผลบวกของสองเทอม แต่ละเทอมเป็นรูปแบบของการคำนวณการแปลงฟูริเยร์แบบไม่ต่อเนื่อง N/2 จุด โดยเทอมแรกการกระทำกับสัญญาณ x(0),x(2),…,x(N-2)

และเทอมที่สองกระทำกับสัญญาณ x(1),x(3),…,x(N-1) โดยที่ค่า W_N^{k} จะเป็นค่าสัมประสิทธิ์ที่ใช้สำหรับคูณเทอมคี่

ซึ่งถ้าทำการแตกเทอมของการแปลงฟูริเยร์แบบไม่ต่อเนื่องN/2จุดที่อยู่ในสมการต่อไป แต่ละเทอมก็จะสามารถกระจายให้เป็นผลบวกของการแปลงฟูริเยร์แบบไม่ต่อเนื่อง N/4 จุดสองเทอม และสามารถกระจายเช่นนี้ต่อไปเรื่อยๆจนกระทั่งทุกตัวอยู่ในรูปของการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 2 จุด ซึ่งการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 2 จุดสามารถคำนวณได้ดังนี้

สมมติให้ x(n) ยาว 2 จุด จะได้

$$X(k) = \sum_{n=0}^{1} x(n)W_2^{kn}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

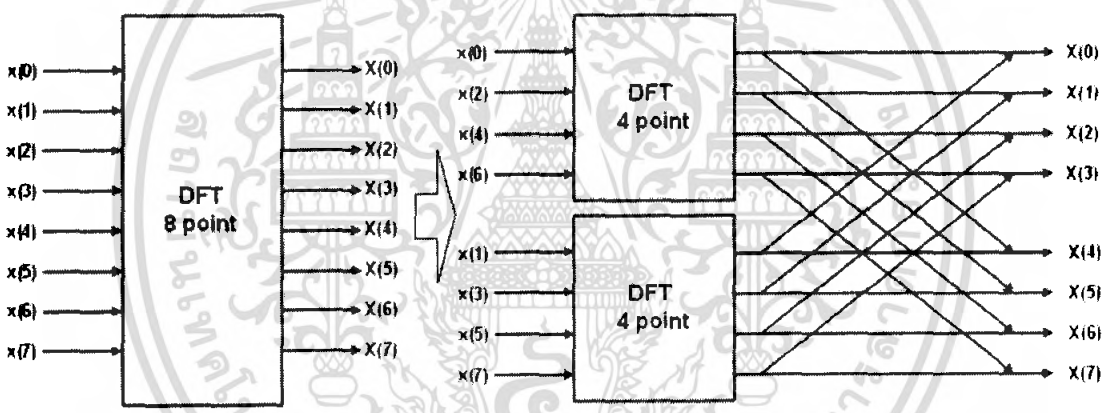
อาศัยความจริงที่ว่า $W_2^0 = 1$ และ $W_2^1 = e^{-j\pi} = -1$ จะได้ว่า

$$X(0) = x(0) + x(1)$$
$$X(1) = x(0) - x(1)$$

ขั้นตอนที่อธิบายทั้งหมดนี้รวมเรียกว่า การแปลงฟูริเยร์แบบรวดเร็ว โดยทั่วไปนิยมเขียนการคำนวณการแปลงฟูริเยร์แบบรวดเร็วโดยใช้แผนภาพที่เรียกว่าแผนภาพผีเสื้อ(butterfly diagram)

แสดงขั้นตอนการคิดแผนภาพผีเสื้อสำหรับการแปลงฟูริเยร์แบบรวดเร็ว เมื่อ N=8

เริ่มจากกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 8 จุด ให้อยู่ในรูปของการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด สองเทอมบวกกัน สามารถเขียนเป็นแผนภาพแสดงในรูปที่ 2.4



รูปที่ 2.4 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 8 จุด เป็นการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด

จากรูปที่2.1 สามารถทำค่าสัมประสิทธิ์ที่คูณอยู่ในแผนภาพให้ง่ายลงได้ โดยใช้คุณสมบัติความสมมาตรของ W_N ดังนี้

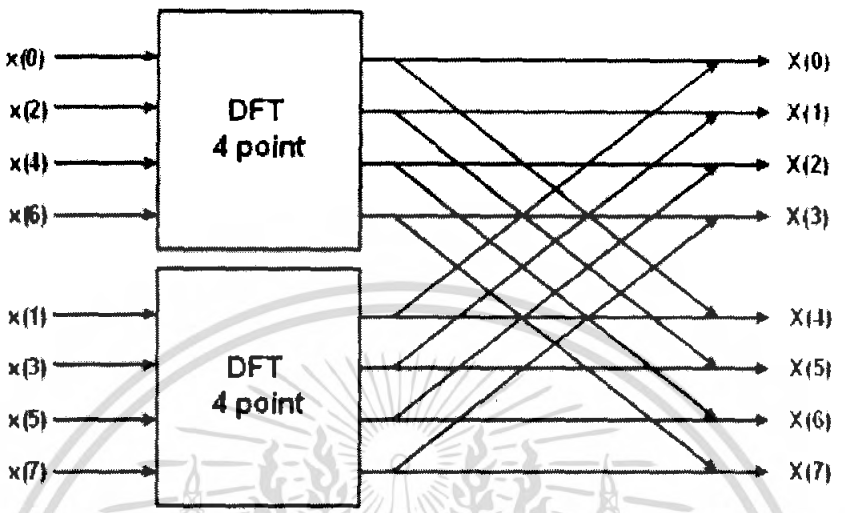
$$W_N^{k+N/2} = W_N^k W_N^{N/2} = W_N^k(-1) = -W_N^k$$

ใช้คุณสมบัติตามสมการข้างต้น จะได้ว่า

$$W_8^4 = -W_8^0, W_8^5 = -W_8^1, W_8^6 = -W_8^2, W_8^7 = W_8^3$$

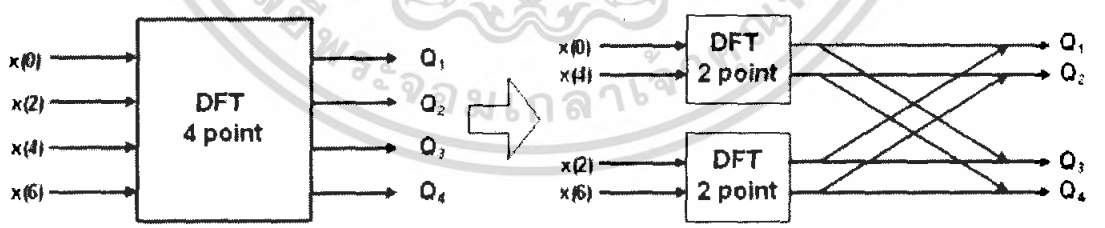
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าทั้งหมดลงในแผนภาพรูปที่ 2.4 จะได้แผนภาพ แสดงในรูปที่ 25



รูปที่ 2.5 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 8 จุด เป็นการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด หลังจากใช้คุณสมบัติของความสมมาตร

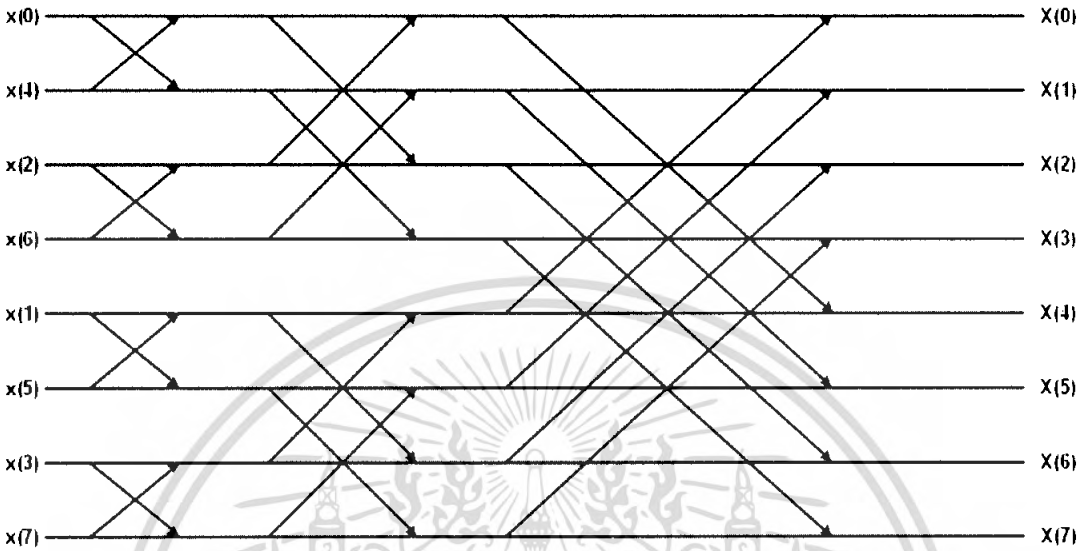
ในทำนองเดียวกันการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด ก็สามารถกระจายเป็นการแปลงฟูริเยรีไม่ต่อเนื่อง 2 จุด ได้ แสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการกระจายการแปลงฟูริเยร์แบบไม่ต่อเนื่อง 4 จุด เป็นการแปลงฟูริเยร์ แบบไม่เนื่อง 2 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรวมผลลัพธ์ของแต่ละส่วนเข้าเป็นแผนภาพเดียวกัน แสดงในรูปที่2.7



รูปที่ 2.7 แผนภาพรวมของการคำนวณการแปลงฟูริเยร์แบบรวดเร็ว 8 จุด

จากรูปที่ 2.7 สามารถนำแผนภาพของการคำนวณการแปลงฟูริเยร์แบบรวดเร็ว 8 จุด ไปใช้เป็นแนวทางในการเขียนแผนภาพสำหรับการแปลงฟูริเยร์แบบรวดเร็ว N จุดใดๆ รวมทั้งใช้เป็นแนวทางในการเขียนโปรแกรมเพื่อคำนวณการแปลงฟูริเยร์แบบรวดเร็ว N จุดใดๆ ได้ด้วยเช่นกัน

2.6 หน้าต่างกรองความถี่

หน้าต่างในการออกแบบวงจรกรองความถี่เชิงเลขตอบสนองตัวอย่างเดี่ยวจำกัดจำนวน (Windowing in designing FIR Digital Filter)

วงจรกรองความถี่ที่มีจำนวนเทอมของสัมประสิทธิ์ที่มากพอ จะให้คุณสมบัติที่ใกล้เคียงกับทฤษฎีมากขึ้น แต่การสร้างวงจรดังกล่าวในทางปฏิบัติเป็นสิ่งที่ทำได้ยากและไม่เหมาะสม เพราะว่าในการคำนวณจะต้องใช้เวลานานและเป็นการสิ้นเปลืองอุปกรณ์อีกด้วย วิธีที่ง่ายในการออกแบบวงจรกรองความถี่เชิงเลขการตอบสนองตัวอย่างเดี่ยวจำกัดจำนวนก็คือการจำกัดจำนวนอิมพัลส์การกระทำดังกล่าวย่อมส่งผลให้วงจรที่ได้มีคุณภาพที่ไม่สมบูรณ์ภายใต้ข้อแลกเปลี่ยนคือวงจรง่ายขึ้นและมีความสิ้นเปลืองน้อยกว่า การปรับค่า h(n) ที่ได้โดยการปรับน้ำหนักใหม่ตามลักษณะการคูณเฉพาะช่วง (Windows function, w(n) สามารถที่จะลดการกระเพื่อม และการปรับปรุงช่วงการตอบสนองเปลี่ยนแปลง (Transition) ได้ ดังนั้นค่าสัมประสิทธิ์ (ค่าการตอบสนองอิมพัลส์) ที่ใช้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\hat{h}(n) = h_d(n) \cdot w(n)$$

โดยค่า hd(n) เป็นค่าสัมประสิทธิ์ที่ได้จากการออกแบบ การคูณจุดต่อจุดในโดเมนเวลาก็คือการคูณประสานกันในโดเมนความถี่ ดังนั้น

$$\hat{H}(e^{j\theta}) = H(e^{j\theta}) * W(e^{j\theta})$$

การไม่คูณ h(n) ด้วยฟังก์ชันใดเลย แท้จริงก็คือการคูณด้วย 1 ซึ่งก็คือการใช้ฟังก์ชันหน้าต่างเป็นแบบสี่เหลี่ยมนั่นเอง

ถ้าหากความชัน (slope) ของฟังก์ชันหน้าต่างมีความกว้างมาก ก็จะทำให้การลาดลงของการตอบสนองความถี่กว้างไปด้วยแต่ถ้าหากความชันของฟังก์ชันหน้าต่างแคบผลการตอบสนองก็จะใกล้เคียง $H(e^{j\theta})$ มากขึ้น อย่างไรก็ดีถ้าให้ความชันมากสิ่งที่ค้านกันก็คือการกระเพื่อม (Ripple) ในช่วงแถบความถี่จำกัด (Stop band) ก็จะมีค่าสูง (ตาม Gibb's Phenomenon)

ฟังก์ชันหน้าต่างที่นิยมใช้กันมากมีดังนี้ Rectangular , Bartlett, Hanning, Hammimg, Blackman, และ Kaiser อันที่จริงแล้วมีมากกว่านี้ขึ้นอยู่กับความเหมาะสมต่อการใช้งาน

1. Rectangular

$$w(n) = \begin{cases} 1; 0 \le n \le N-1 \\ 0; elsewhere \end{cases}$$

2. Bartlett or Triangle

$$w(n) = \begin{cases} 2n/(N-1); 0 \le n \le (N-1) \\ 2 - 2n/(n-1); (N-1)/2 \le n \le N-1 \\ 0; elsewhere \end{cases}$$

3. Hanning

$$w(n) = \begin{cases} \left\{1 - \cos\frac{2\pi n}{N-1}\right\} \Big/ 2; 0 \le n \le N-1 \\ 0; elsewhere \end{cases}$$

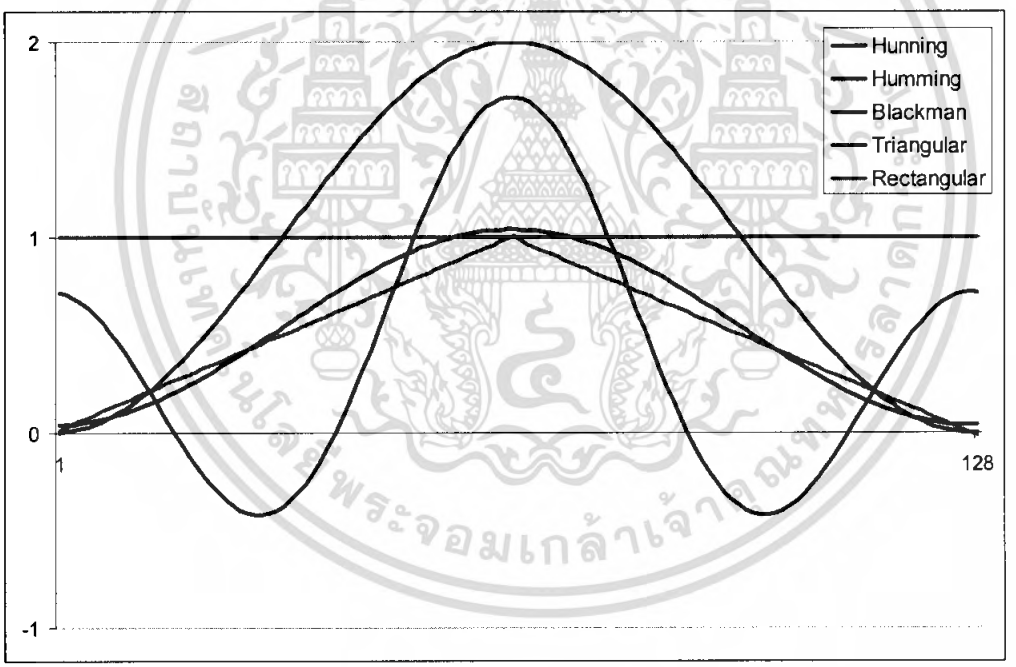
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Hamming

$$w(n)=\begin{cases}0.54-0.5\cos\dfrac{2\pi n}{N-1};0\le n\le N-1\\ 0;elsewhere\end{cases}$$

5. Blackman

$$w(n)=\begin{cases}0.42-0.5\cos\dfrac{2\pi n}{N-1}+0.8\cos\dfrac{2\pi n}{N-1};0\le n\le N-1\\ 0;elsewhere\end{cases}$$



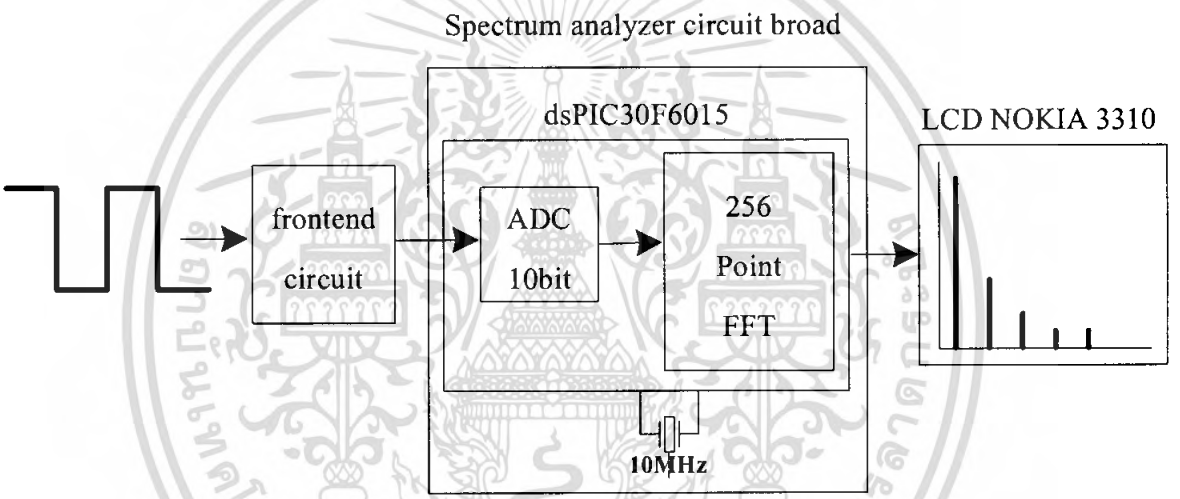
รูปที่ 2.8 ลักษณะหน้าต่างชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

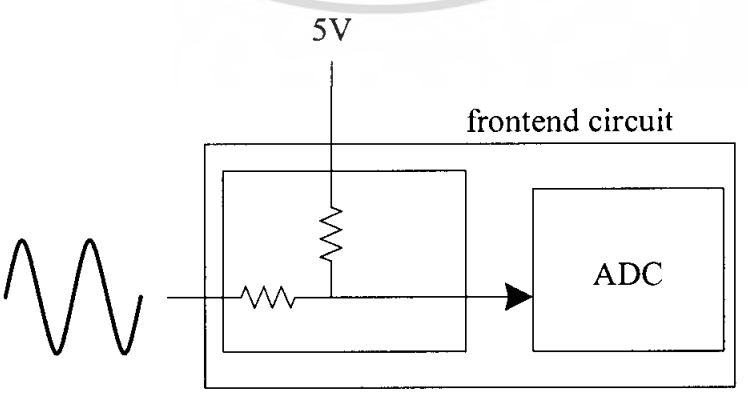
บทที่ 3
วิธีการดำเนินงานวิจัย

การทำงานนี้จะแบ่งเป็น 3 ส่วนหลัก ส่วนแรกก็จะเป็นส่วนของฮาร์ดแวร์ ส่วนที่สองเป็นซอฟแวร์ ส่วนที่สามการทดลองทางทฤษฎี

3.1 ฮาร์ดแวร์



รูปที่ 3.1 แผนผังการออกแบบฮาร์ดแวร์

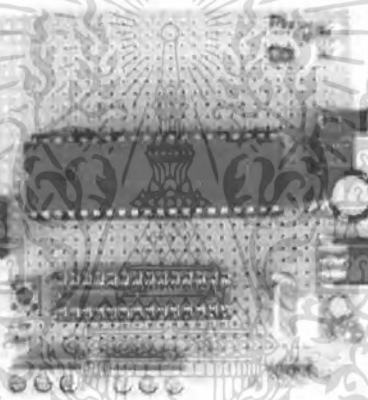


รูปที่ 3.2 วงจรยกระดับแรงดันไฟฟ้า

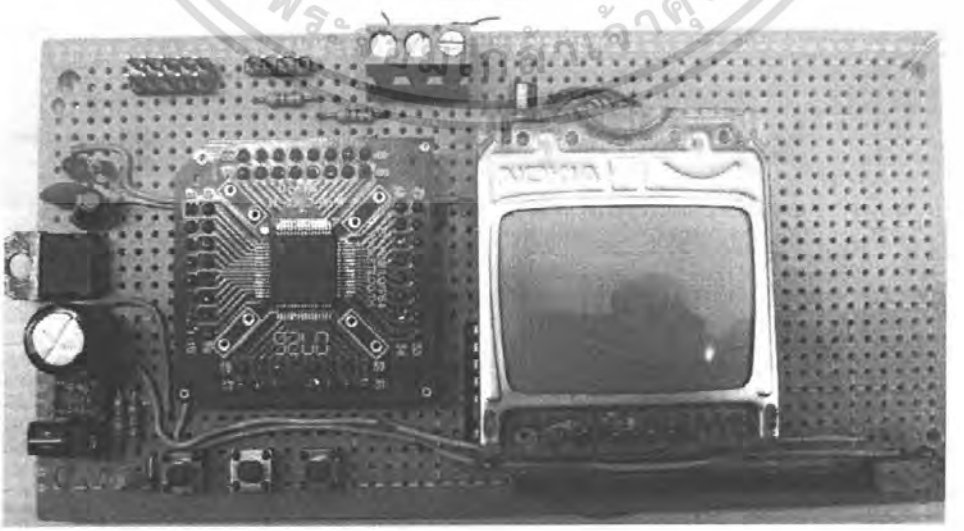
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนฮาร์ดแวร์เริ่มจากวงจรยกระดับแรงดันไฟฟ้า ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลที่มีความละเอียด 10บิต ตัวประมวลสัญญาณดิจิตอล dsPIC30F6015 มีความถี่ 10 Mz จะเป็นตัวอ่านค่าจาก ADC เข้ามาและทำการคำนวณ FFT จากนั้นจะส่งข้อมูลไปแสดงผลทางหน้าจอแสดงผล LCD NOKIA 3310

วงจรทดสอบเบื้องต้นของไมโครคอนโทรลเอร์เบอร์ dsPIC30F4011และ dsPIC30F6015 สร้างขึ้นมาเพื่อทดลองตัวคอมไพเลอร์และบอร์ดสำหรับโปรแกรม และทำการทดสอบพอร์ตอินพุต เอาท์พุต การรับส่งข้อมูลทาง RS232 การใช้ ADC รับค่าความถี่ต่าง การปรับแต่งรีจีสเตอร์ให้ได้ตามต้องการ



รูปที่ 3.3 บอร์ดทดลอง dsPIC30F4011



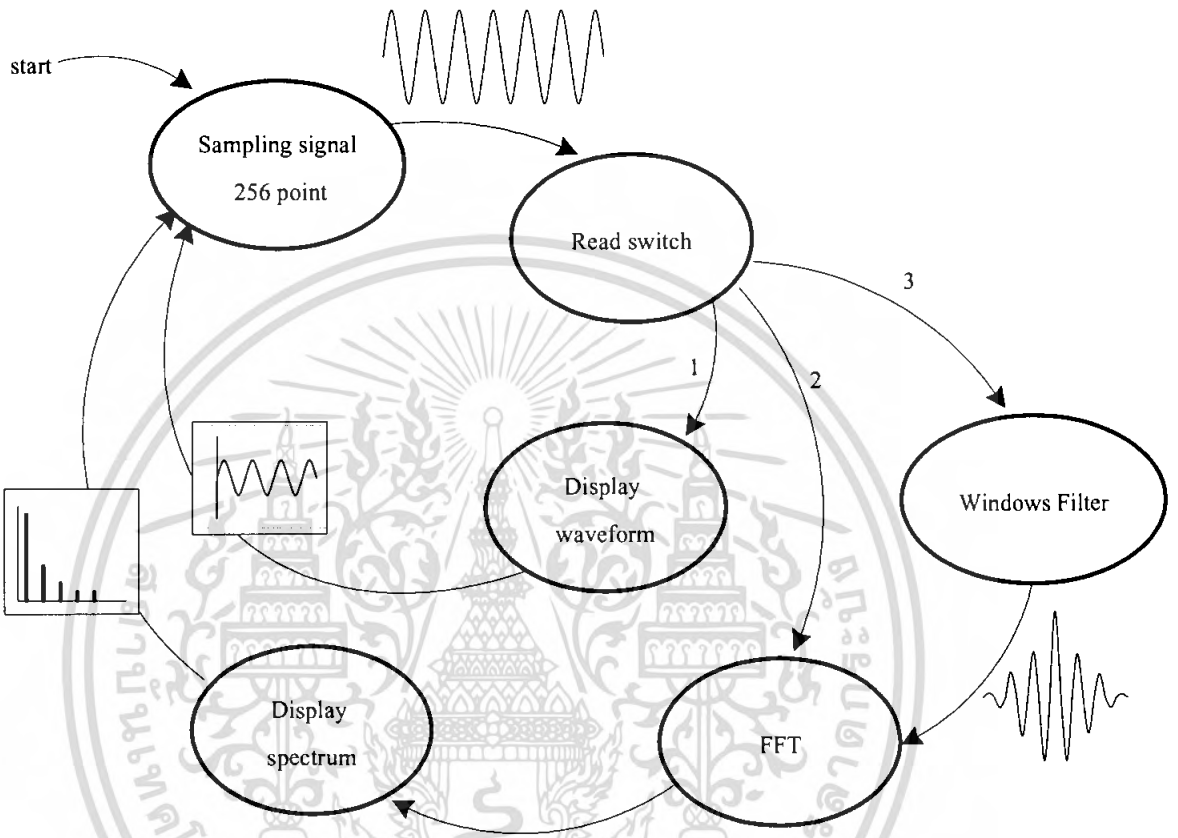
รูปที่ 3.4 รูปบอร์ดทดลองdsPIC30F6015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

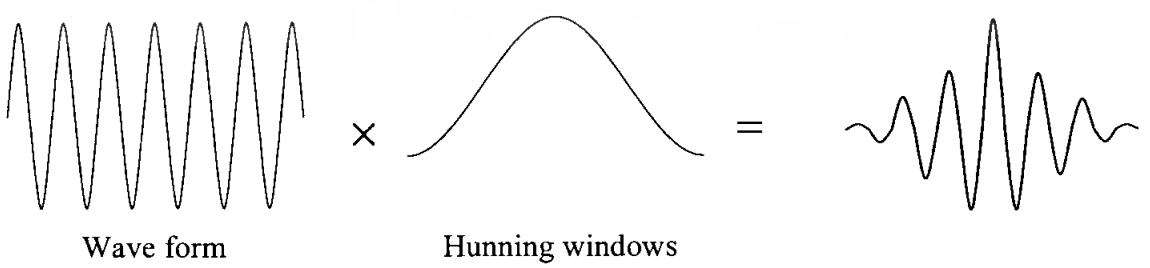
81596

3.2 ซอฟแวร์

ใช้ภาษาซีและคอมไพเลอร์Mikro C for dsPIC V3.0 ในการพัฒนาโปรแกรม



รูปที่ 3.5 การทำงานของโปรแกรม



รูปที่ 3.6 ลักษณะการกรองสัญญาณ

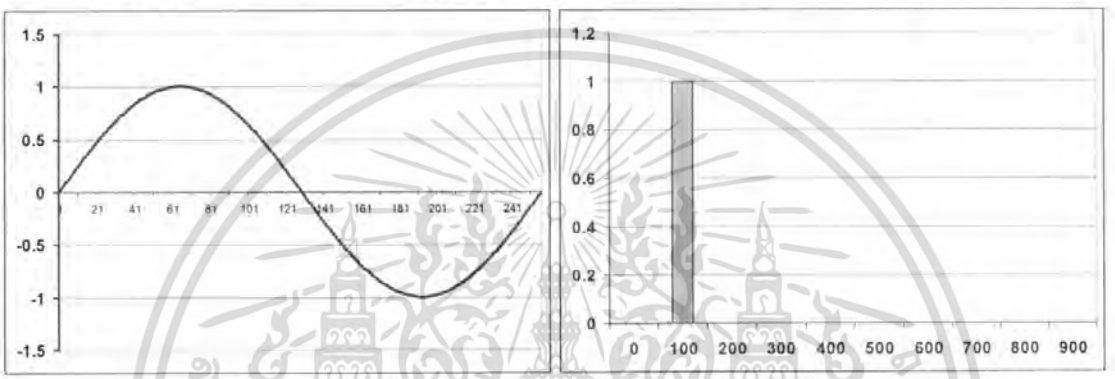
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทดลองทางทฤษฎี

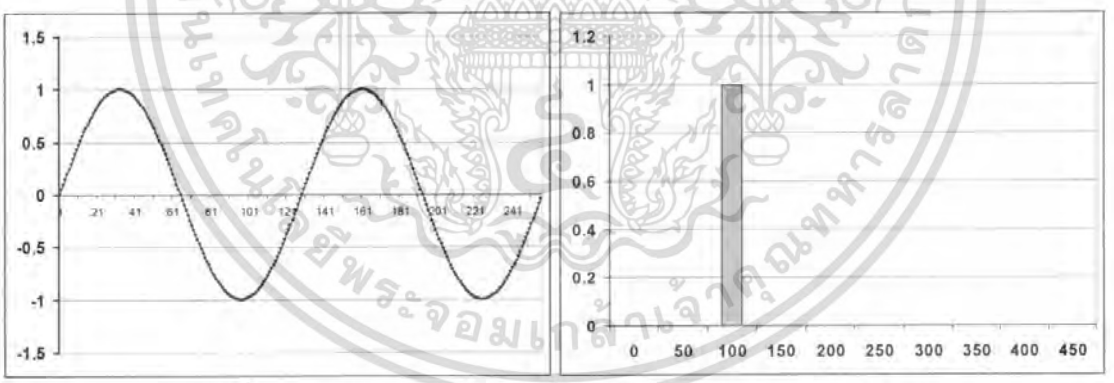
ทำการทดลองเปรียบเทียบข้อผิดพลาดปกติการคำนวณ DFT จะต้องใช้ข้อมูลจำนวน 2^n จุด และได้ 1 คาบเวลาพอดี แต่ถ้าสัญญาณสมมาตรกันก็คิดแค่ครึ่งคาบก็ได้

3.3.1 การวิเคราะห์สัญญาณโดยใช้ซอฟแวร์บนคอมพิวเตอร์

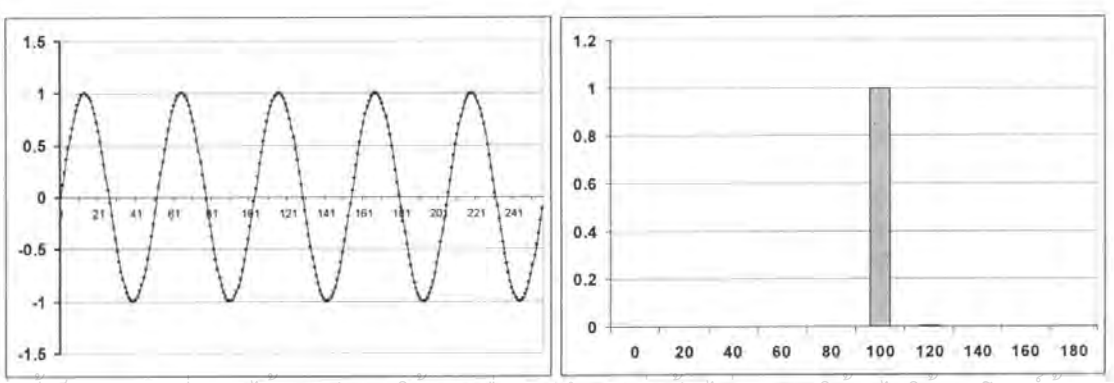
ทำการทดลองโดยใช้ ซอฟแวร์บนเครื่องคอมพิวเตอร์เพื่อทดสอบผลที่จะเกิดขึ้นจากความถี่ในการจับสัญญาณ จำนวนคาบการจับสัญญาณและผลจากการใส่วินโดว์ฟิลเตอร์



รูปที่3.7 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 1 คาบ

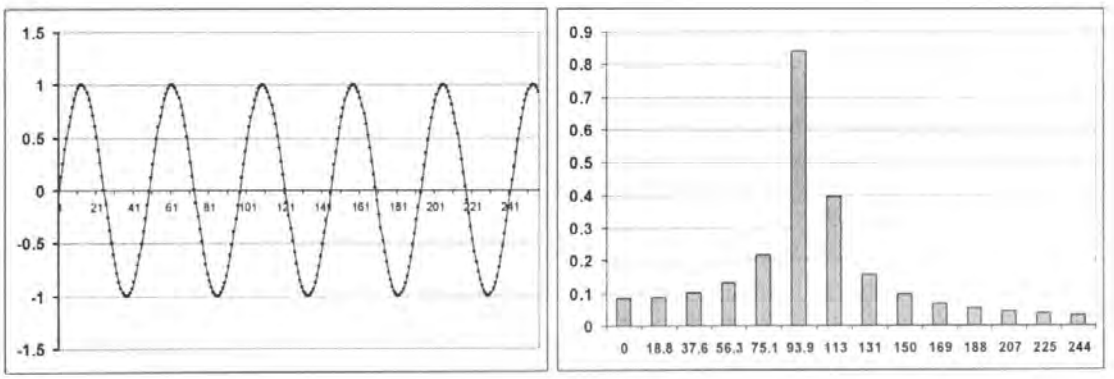


รูปที่3.8 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 2 คาบ

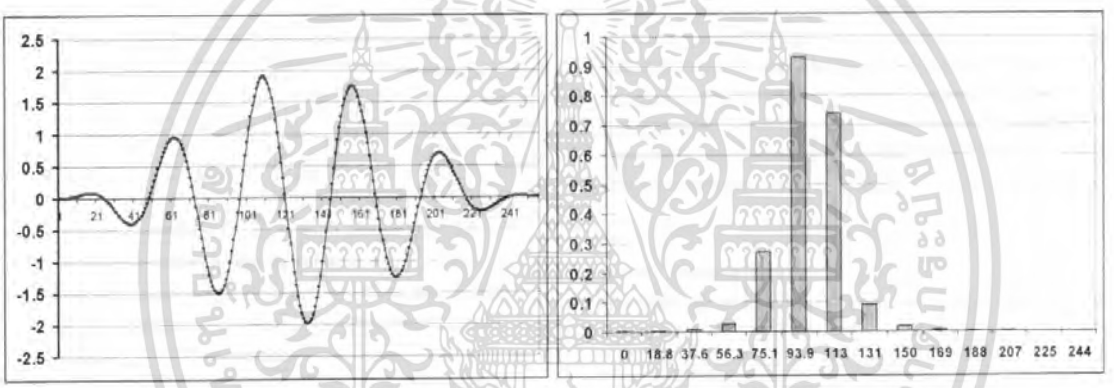


รูปที่3.9 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz จำนวน 5 คาบ

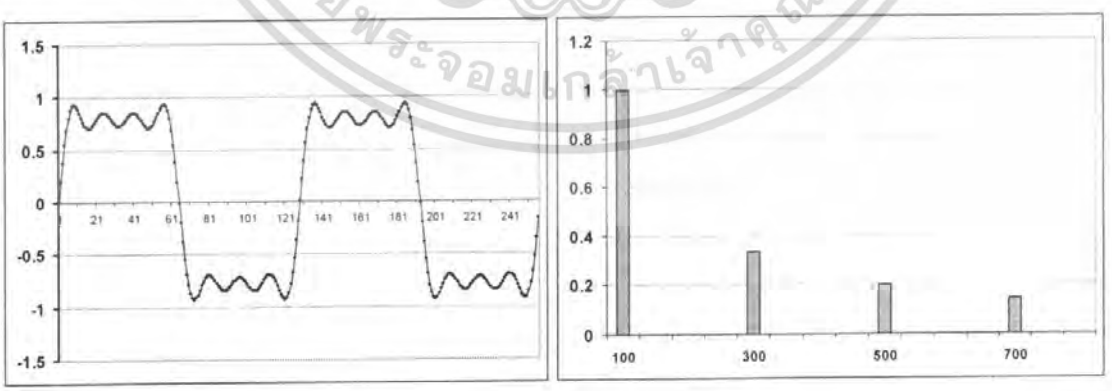
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.8 การแปลงฟูริเยร์ของคลื่นรูปไซน์100 Hz ที่เก็บข้อมูลได้ไม่ครบคาบเป็นจำนวนเต็ม

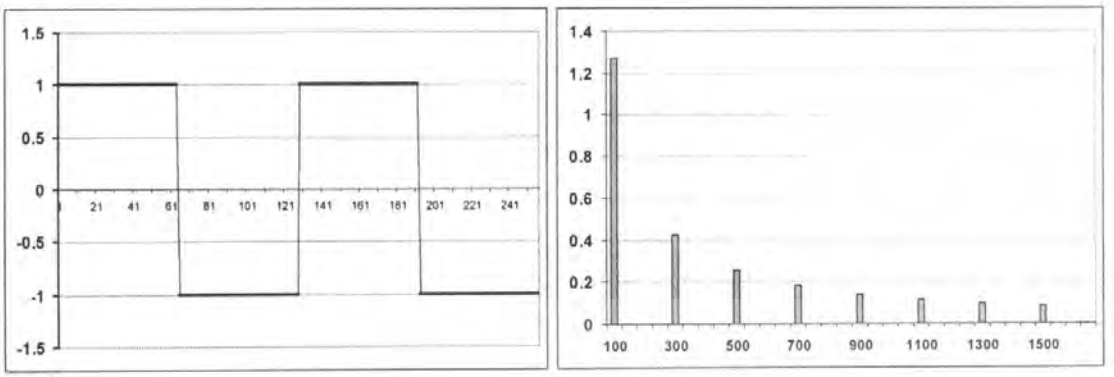


รูปที่3.9 การแปลงฟูริเยร์ของสัญญาณจากรูปที่3.4 เมื่อนำมาผ่านหน้าต่างกรองความถี่

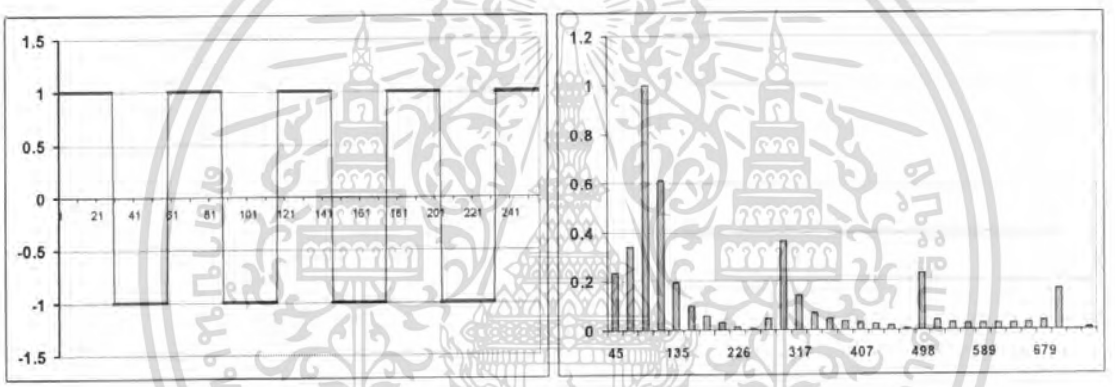


รูปที่3.10 การแปลงฟูริเยร์ของคลื่นรูปไซน์ 100Hz 300Hz 500Hz 700Hz มาบวกกัน

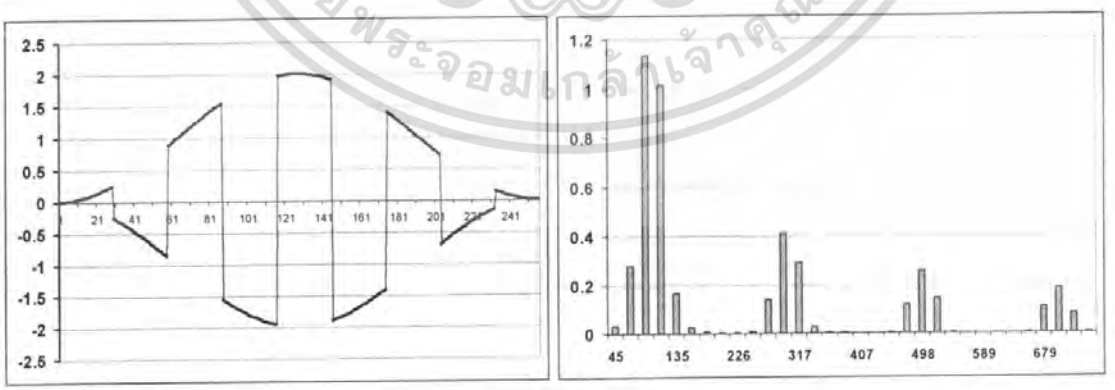
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.11 การแปลงฟูริเยร์ของคลื่นรูปสี่เหลี่ยม 100 Hz จำนวน 2 คาบ



รูปที่3.12 การแปลงฟูริเยร์ของคลื่นรูปสี่เหลี่ยม100 Hz ที่เก็บข้อมูลได้ไม่ครบคาบเป็นจำนวนเต็ม



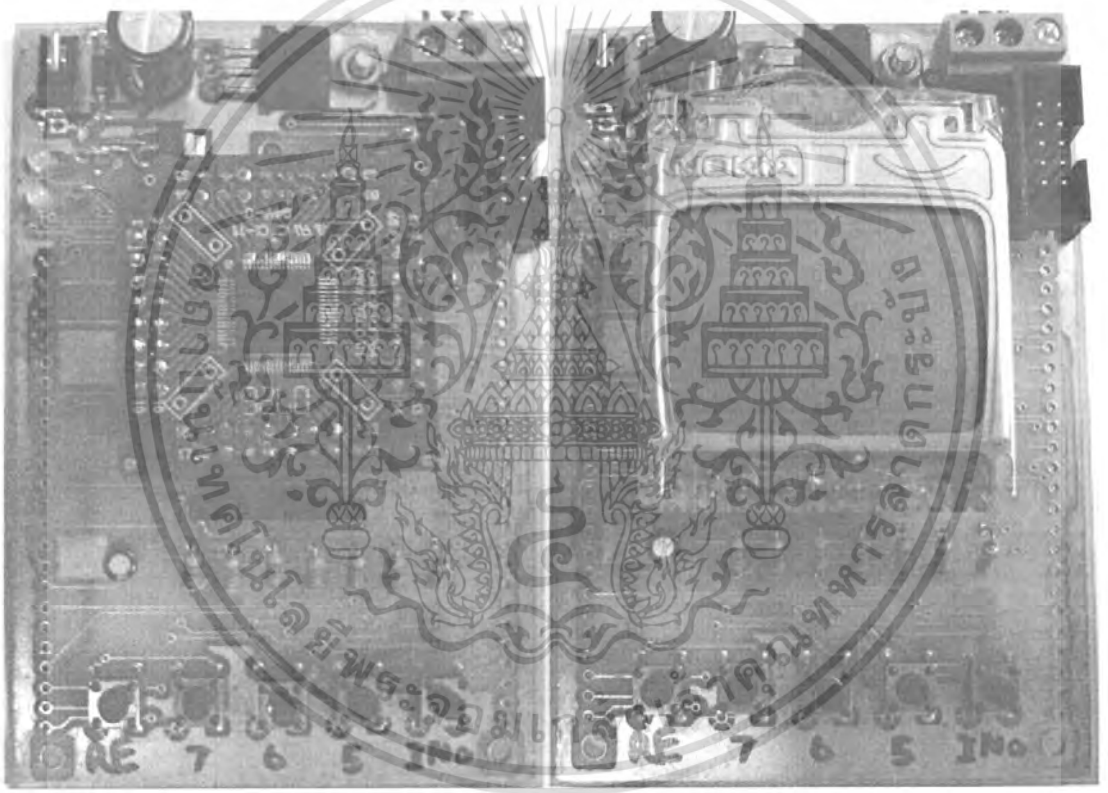
รูปที่3.13 การแปลงฟูริเยร์ของสัญญาณจากรูปที่3.8 เมื่อนำมาผ่านหน้าต่างกรองความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและอภิปรายผล

4.1 การทดลองแปลงฟูริเยร์แบบรวดเร็วโดยใช้dsPIC30F6015

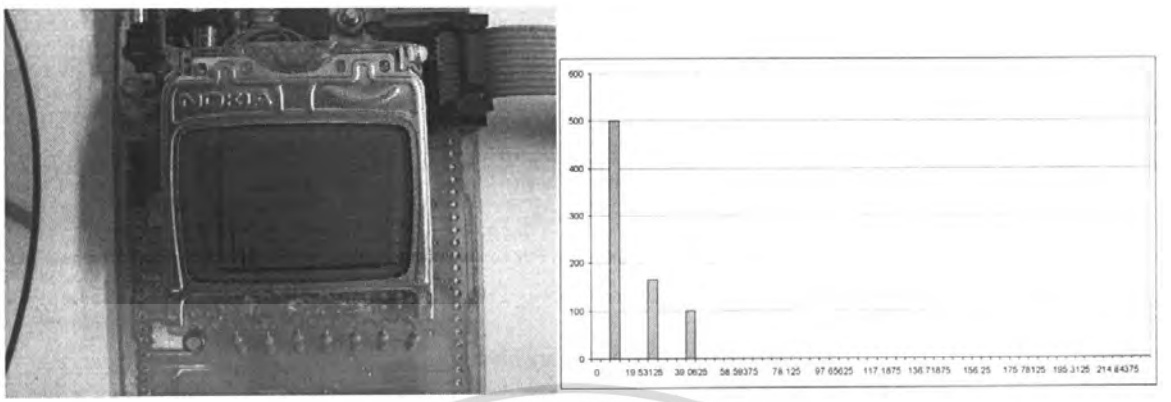
หลังจากที่ได้จัดทำแผงวงจร dsPIC30F6015 และเขียนโปรแกรมควบคุมเกี่ยวกับการจอแสดงผล LCD 3310 และการแปลงฟูริเยร์แบบรวดเร็ว



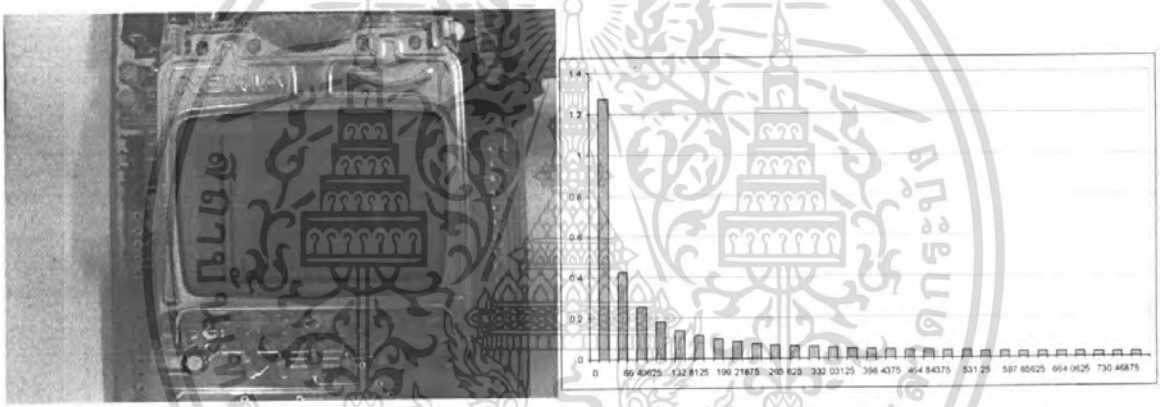
รูปที่4.1 แสดงลักษณะแผงวงจร dsPIC30F6015

เนื่องจากโครงงานพิเศษนี้เป็นโครงงานเกี่ยวกับการวิเคราะห์สัญญาณด้วยการแปลงฟูริเยร์แบบรวดเร็วจึงได้ทำการทดลองวิเคราะห์สัญญาณของคลื่นรูปไซน์และคลื่นรูปสี่เหลี่ยมให้อยู่ในรูปของแกนความถี่

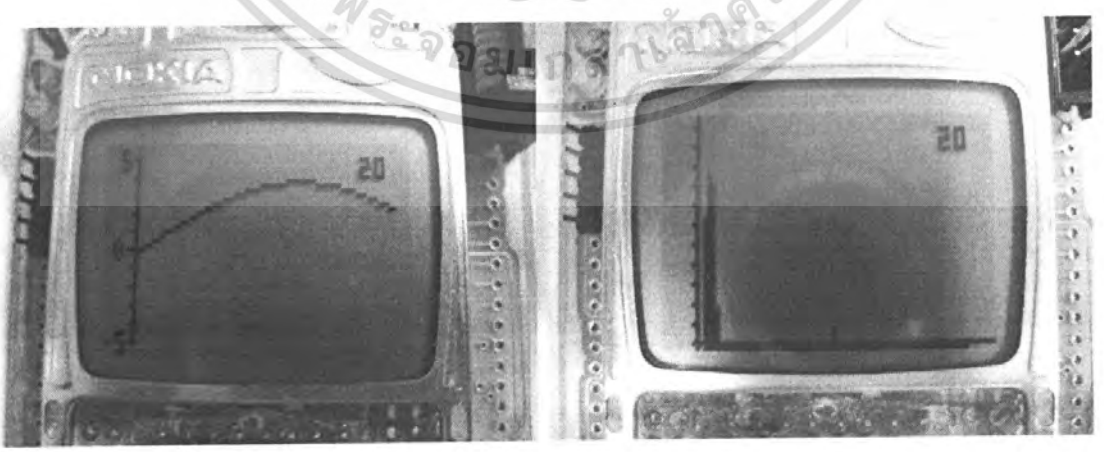
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.2 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 15Hz 30Hz และ 45 Hz รวมกัน

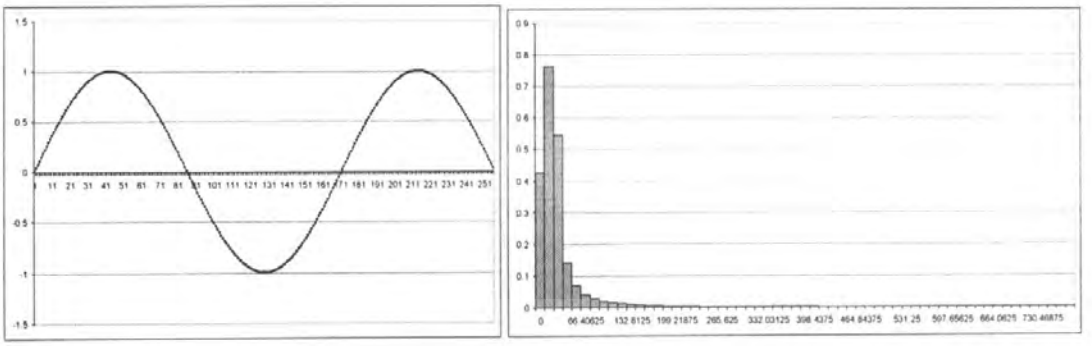


รูปที่4.3 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม

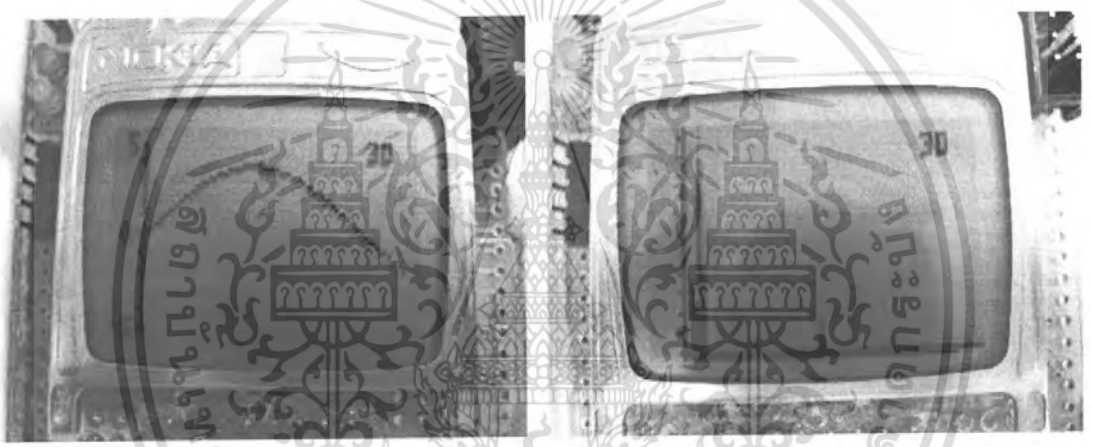


รูปที่4.4 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 20 Hz

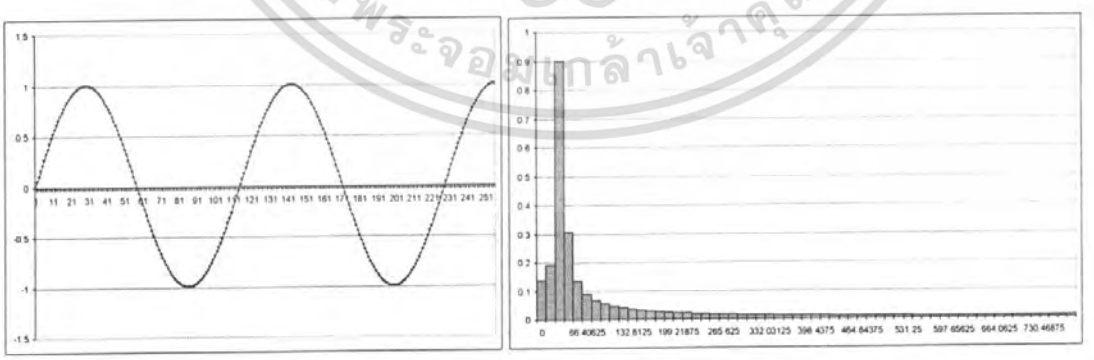
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.5 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 20 Hz โดยซอฟแวร์

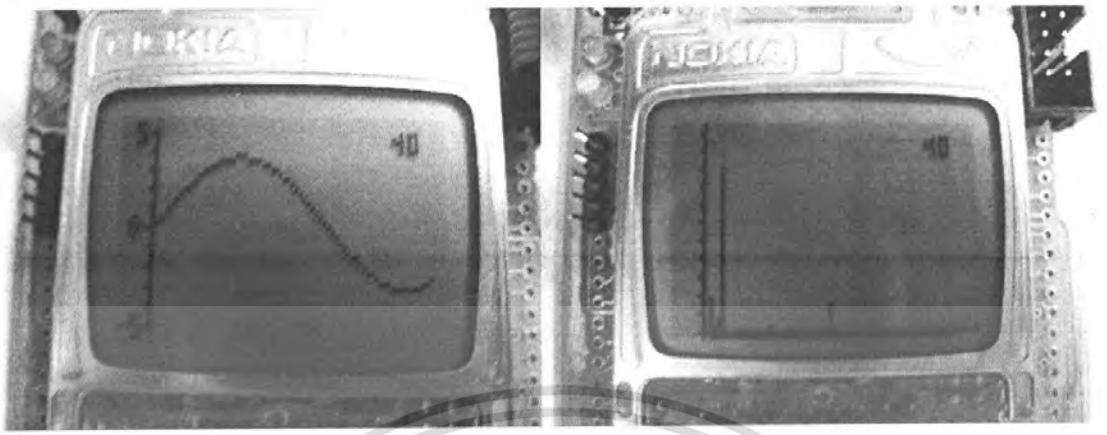


รูปที่4.6 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 30 Hz

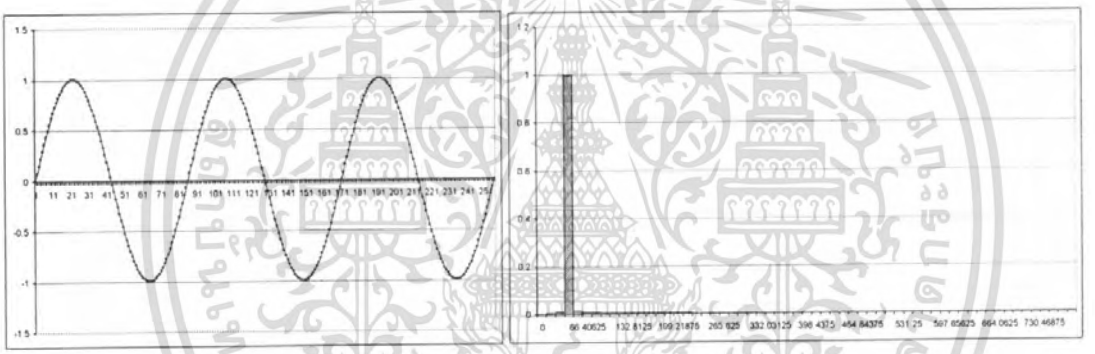


รูปที่4.7 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 30 Hz โดยซอฟแวร์

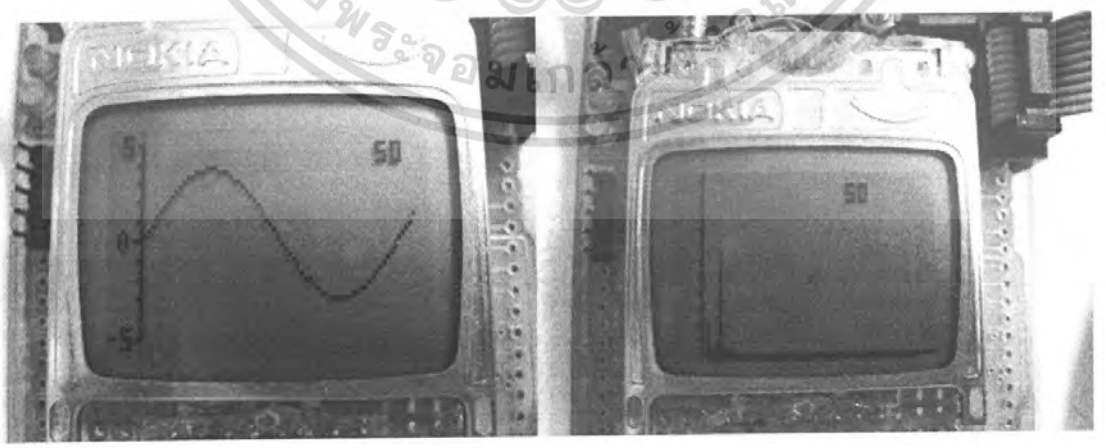
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.8 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 40 Hz

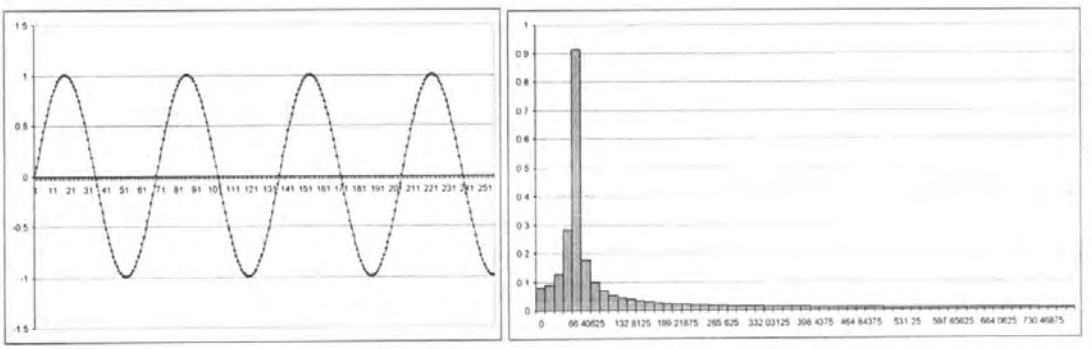


รูปที่4.9 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 40 Hz โดยซอฟแวร์

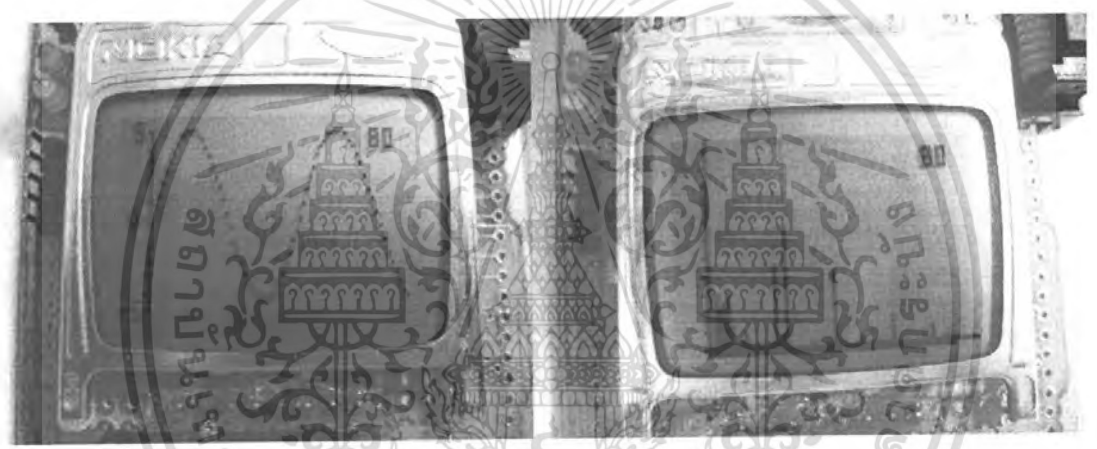


รูปที่4.10 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 50 Hz

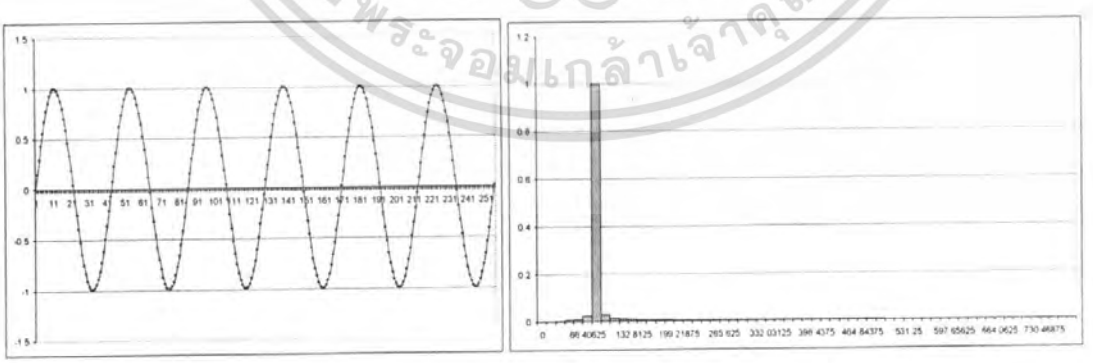
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.11 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 50 Hz โดยซอฟแวร์

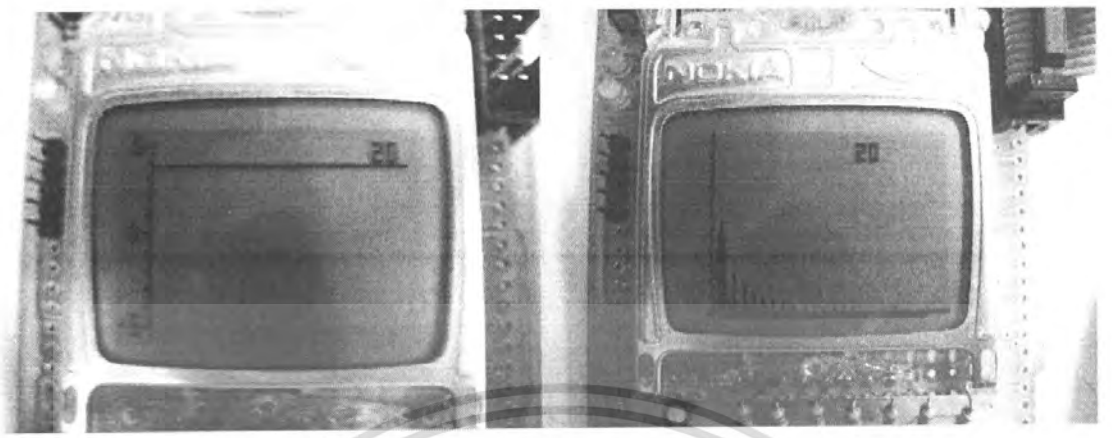


รูปที่4.12 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 80 Hz

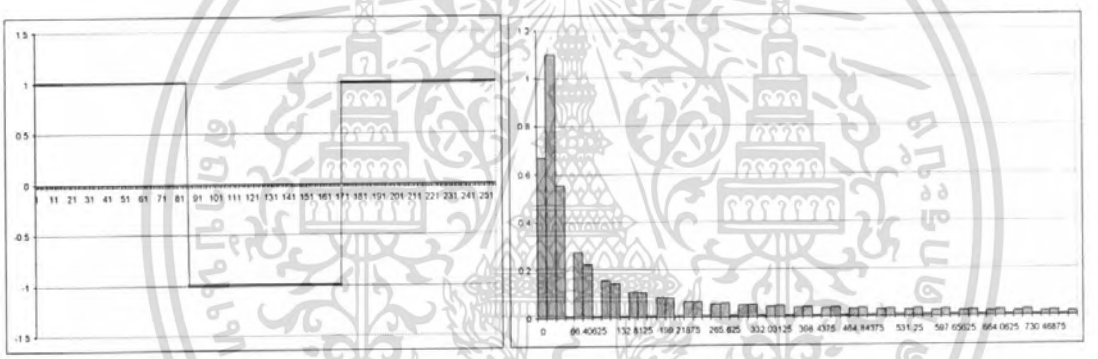


รูปที่4.13 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ 80 Hz โดยซอฟแวร์

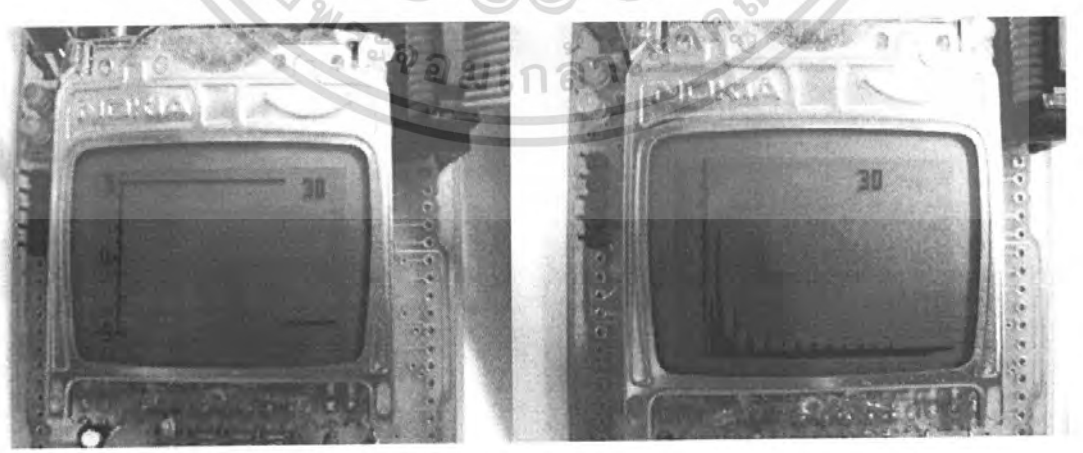
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.14 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 20 Hz

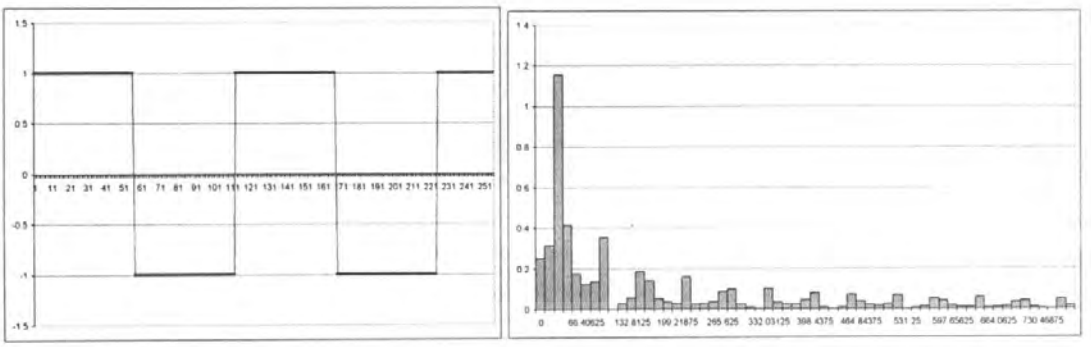


รูปที่4.15 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 20 Hz โดยซอฟแวร์

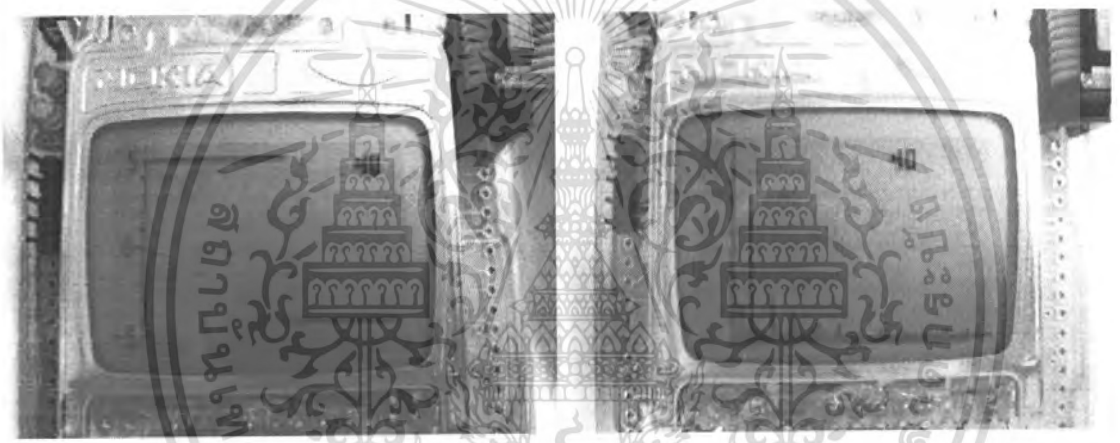


รูปที่4.16 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 30 Hz

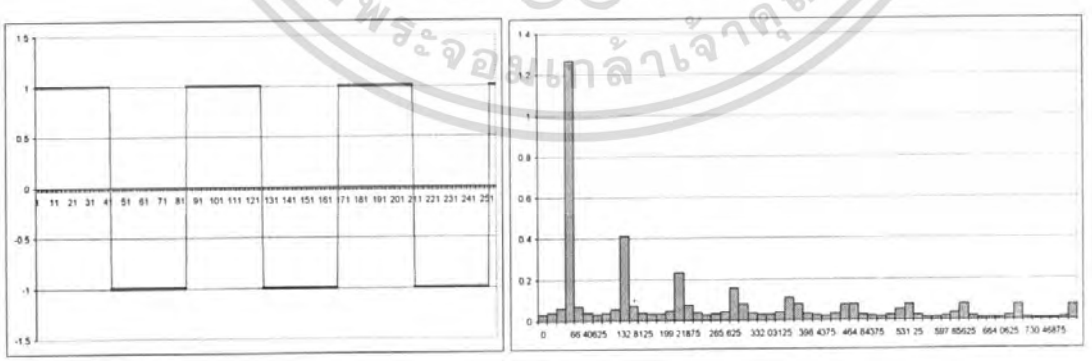
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.17 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 30 Hz โดยซอฟแวร์

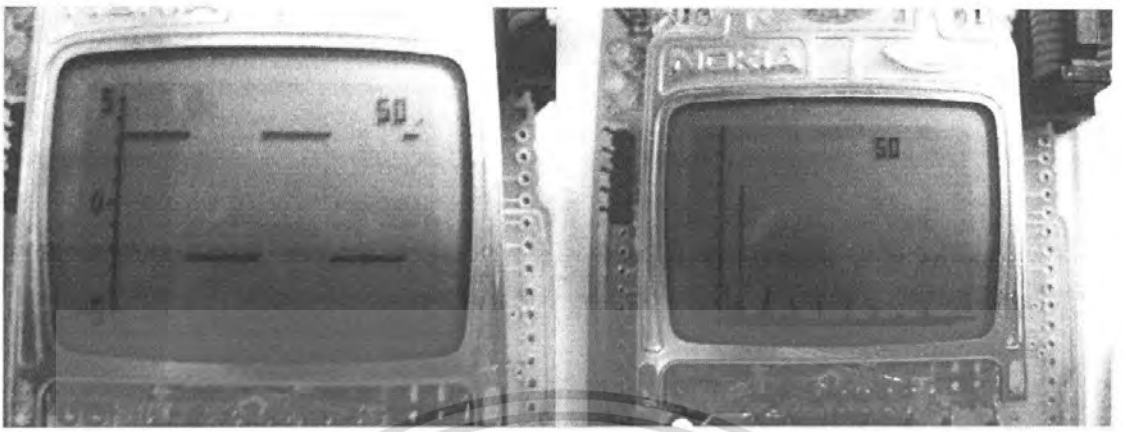


รูปที่4.18 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 40 Hz

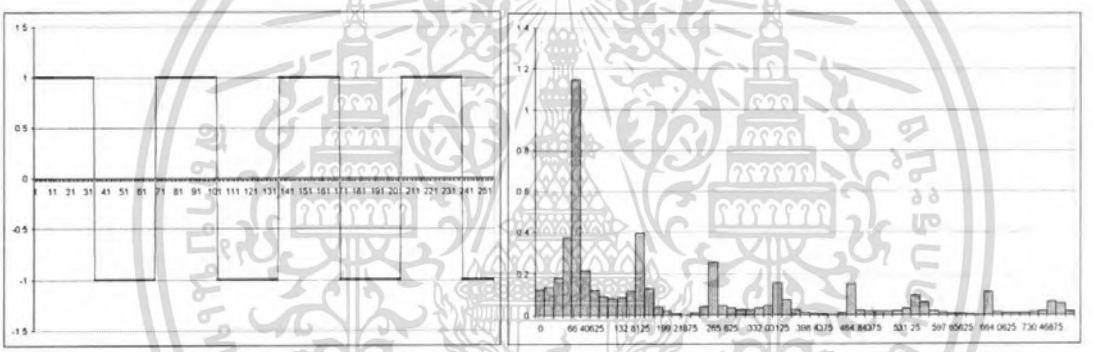


รูปที่4.19 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 40 Hz โดยซอฟแวร์

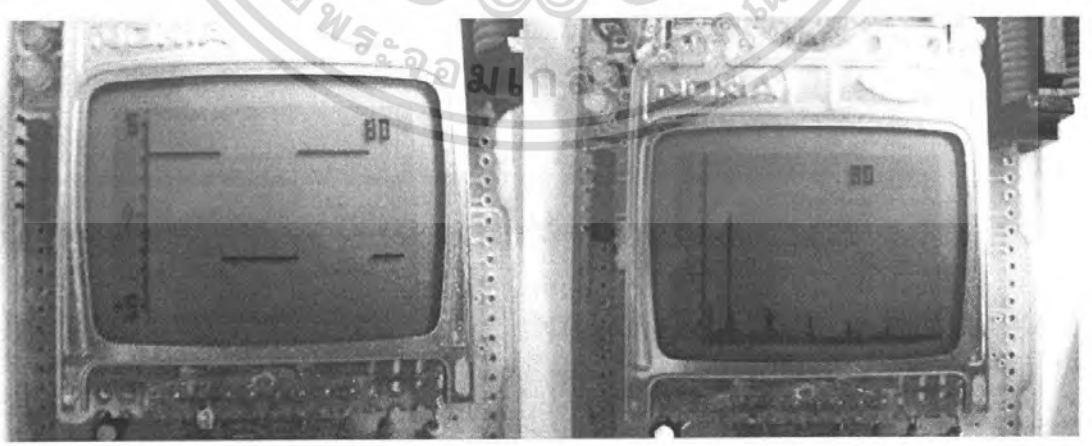
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.20 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 50 Hz

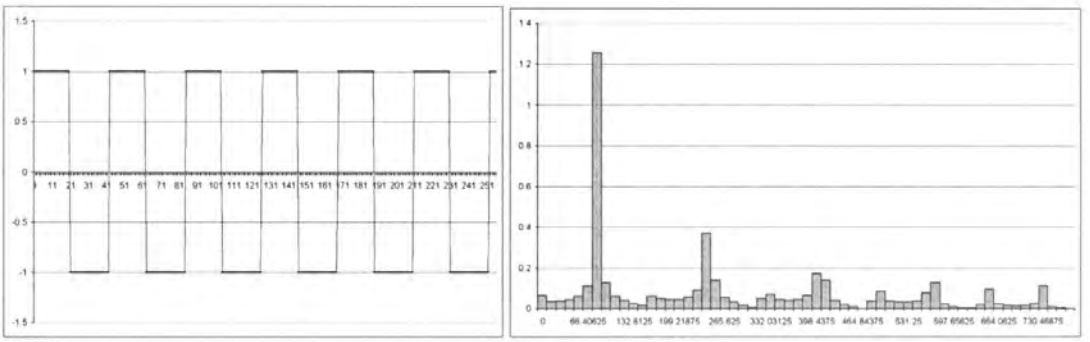


รูปที่4.21 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 50 Hz โดยซอฟแวร์

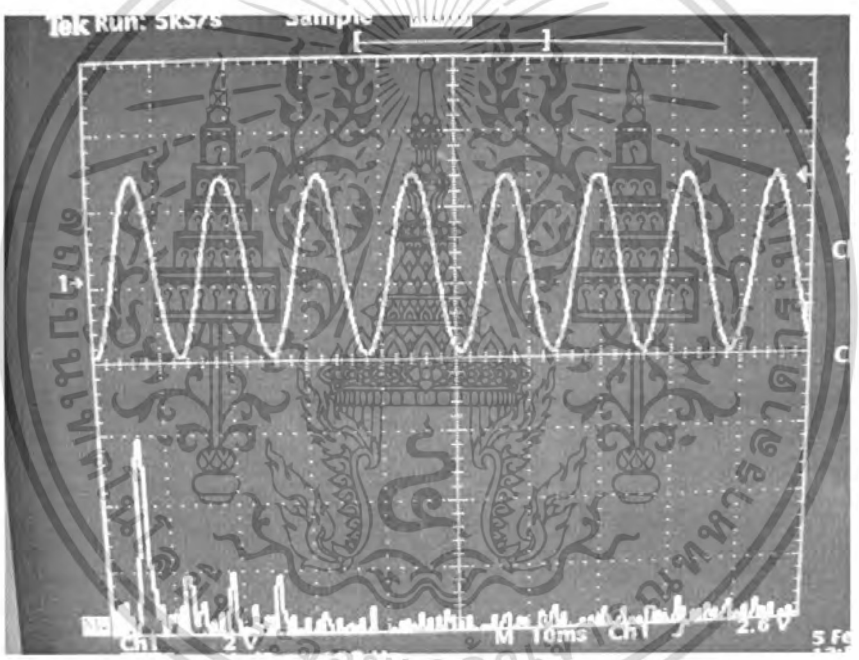


รูปที่4.22 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 80 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

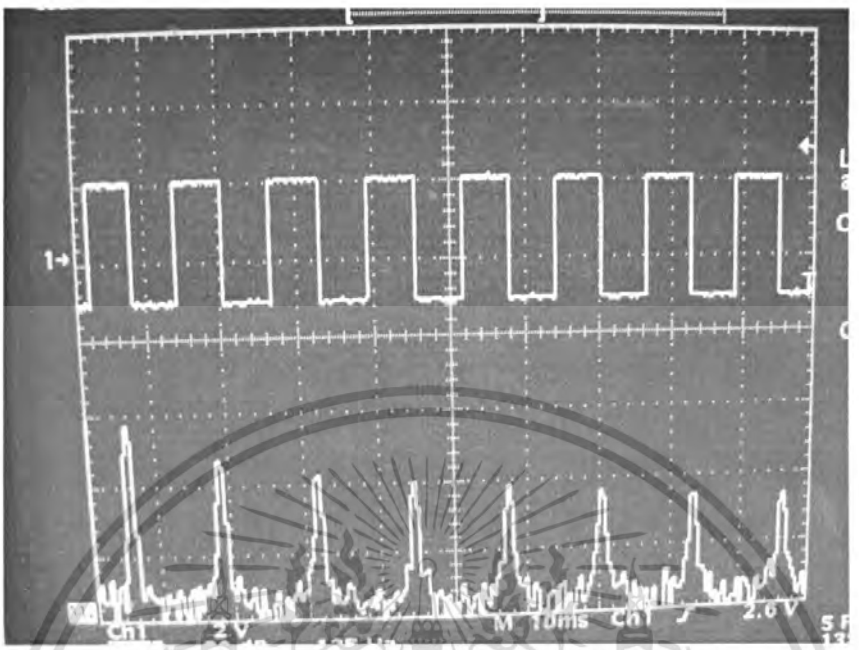


รูปที่4.23 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยม 80 Hz โดยซอฟแวร์



รูปที่4.24 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปไซน์ในเครื่องออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.25 การแปลงฟูริเยร์แบบรวดเร็วของสัญญาณรูปสี่เหลี่ยมในเครื่องออสซิลโลสโคป

จากการทดลองวิเคราะห์สัญญาณไฟฟ้าจากเครื่องมือที่เราสร้างขึ้นซึ่งสามารถแยกสเปคตรัมได้ซึ่งมีแนวโน้มใกล้เคียงกับเครื่องออสซิลโลสโคป ทำให้รู้สัญญาณความถี่ที่เข้ามา ซึ่งจะมีความถี่เป็น จำนวนเท่าของความถีมูลฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

โครงงานพิเศษนี้สามารถที่จะแสดงรูปของคลื่นต่างและทำการแปลงฟูริเยร์แบบรวดเร็ว โดยการรับสัญญาณไฟฟ้าเข้ามาแล้วแปลงสัญญาณจากอนาล็อกไปเป็นดิจิตอลเพื่อใช้ในกาคำนวณ จำนวนค่าที่เก็บเข้ามาทั้งหมดมี 256 จำนวน ซึ่งเพียงพอที่จะใช้ในการวิเคราะห์สัญญาณที่มีความถี่ต่ำ เครื่องที่ใช้ในการวิเคราะห์สัญญาณนี้ ใช้ไมโครคอนโทรลเลอร์ dsPIC30F6015 และแสดงผลทางจอ LCD NOKIA 3310 ซึ่งทำให้มีขนาดเล็กสามารถพกพาได้และยังมีราคาประหยัด และยังเป็นการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล dsPIC เพื่อใช้แทนออสซิลโลสโคป

เนื่องจากโครงงานพิเศษนี้เน้นไปทางการแปลงฟูริเยร์เพื่อแปลงสัญญาณที่อยู่ในรูปของแกนเวลาไปเป็นแกนความถี่ เพื่อจะวิเคราะห์ดูว่าสัญญาณที่เข้ามาประกอบด้วยความถี่อะไรบ้าง

ข้อเสนอแนะและแนวทางการพัฒนา

1. ความเร็วในการเก็บข้อมูลยังช้าอยู่ควรเพิ่มความเร็วในการเก็บข้อมูล
2. ถ้าเพิ่มจำนวนข้อมูลที่นำมาคำนวณจะทำให้เครื่องมีประสิทธิภาพมากขึ้น
3. โครงงานพิเศษนี้ใช้จอ LCD NOKIA 3310 ซึ่งมีราคาถูกแต่มีความละเอียดน้อย
4. ควรพัฒนาในส่วนของการเก็บข้อมูลซึ่งจะมีประโยชน์มาก
5. ควรเพิ่มเรื่องฟิลเตอร์เข้าไปเพื่อตัดสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

John Wiley & Sons **"DSP Applications Using C and TMS320C6X"**

Wichit Sirichote. **"Build Your Own Microcontroller Projects."** [Online]
http://www.kmitl.ac.th/~kswichit.

Somsuk Chumchoy. **"Basic Digital Signal Processing"**
Major Electronic Faculty of Engineering KMITL.

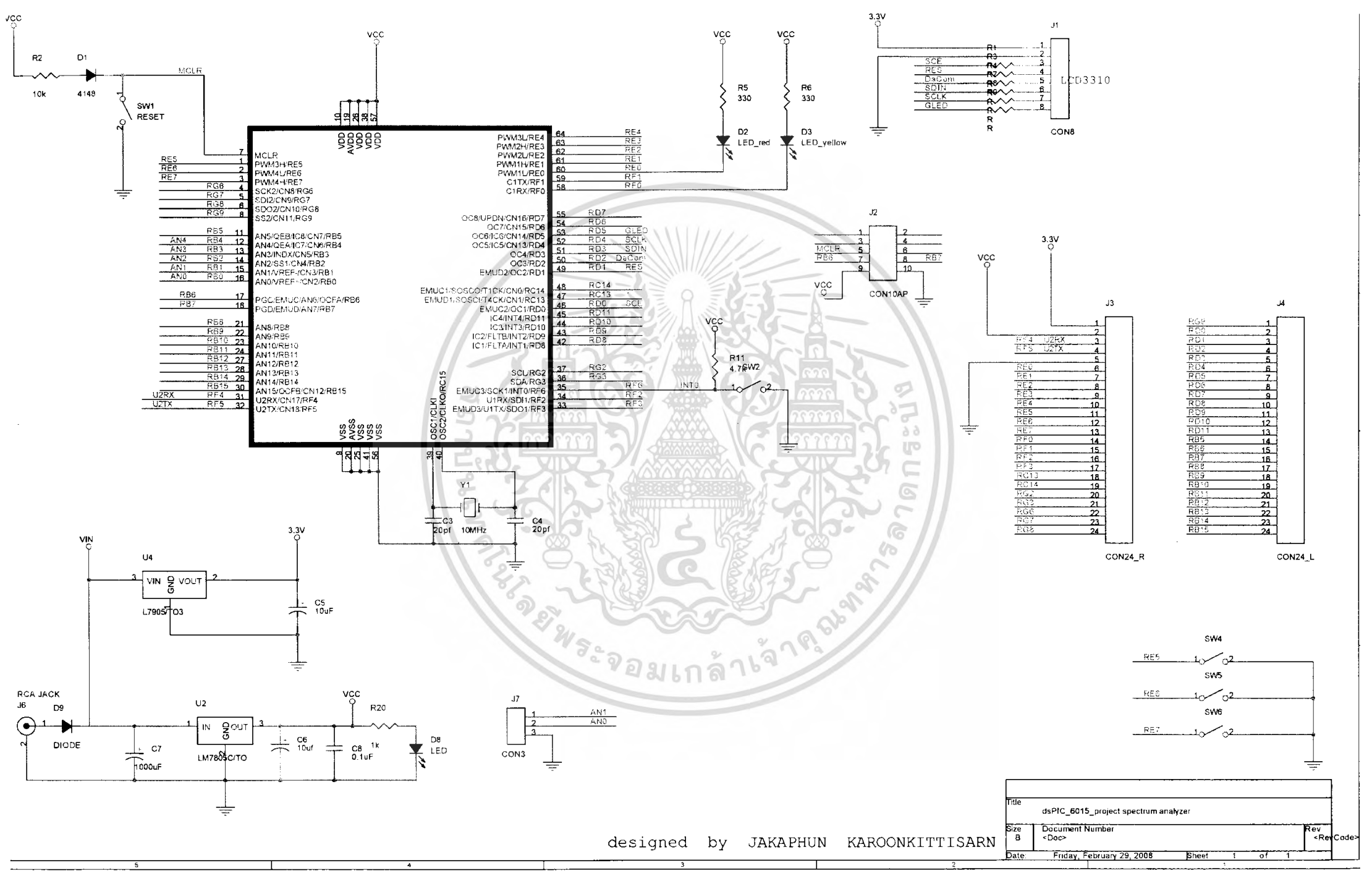
Nakhon Pukdeechate & Chaiwut Limpornjitwilai. **"dsPIC Microcontroller"**

Ratprachar Charncholsamut **"Design of a compact FFT/ IFFT processor"**
Major Electronic Faculty of Engineering KMITL.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
R2
10k
D1
4148
MCLR
SW1
RESET
VDD
AVDD
VSS
AVSS
MCLR
PWM3H/RE5
PWM4L/RE6
PWM4H/RE7
SCK2/CN8/RG6
SDI2/CN9/RG7
SDO2/CN10/RG8
SS2/CN11/RG9
AN5/QEB/IC8/CN7/RB5
AN4/QEA/IC7/CN6/RB4
AN3/INDX/CN5/RB3
AN2/SS1/CN4/RB2
AN1/VREF-/CN3/RB1
AN0/VREF+/CN2/RB0
PGC/EMUC/AN6/OCFA/RB6
PGD/EMUD/AN7/RB7
AN8/RB8
AN9/RB9
AN10/RB10
AN11/RB11
AN12/RB12
AN13/RB13
AN14/RB14
AN15/OCFB/CN12/RB15
U2RX/CN17/RF4
U2TX/CN18/RF5
PWM3L/RE4
PWM2H/RE3
PWM2L/RE2
PWM1H/RE1
PWM1L/RE0
C1TX/RF1
C1RX/RF0
OC8/UPDN/CN16/RD7
OC7/CN15/RD6
OC6/IC6/CN14/RD5
OC5/IC5/CN13/RD4
OC4/RD3
OC3/RD2
EMUD2/OC2/RD1
EMUC1/SOSCO/T1CK/CN0/RC14
EMUD1/SOSCI/T4CK/CN1/RC13
EMUC2/OC1/RD0
IC4/INT4/RD11
IC3/INT3/RD10
IC2/FLTB/INT2/RD9
IC1/FLTA/INT1/RD8
SCL/RG2
SDA/RG3
EMUC3/SCK1/INT0/RF6
U1RX/SDI1/RF2
EMUD3/U1TX/SDO1/RF3
OSC1/CLKI
OSC2/CLKO/RC15
Y1
10MHz
C3
20pf
C4
20pf
R5
330
R6
330
D2
LED_red
D3
LED_yellow
3.3V
J1
LCD3310
CON8
SCE
RES
D/Com
SDIN
SCLK
GLED
J2
CON10AP
MCLR
RB6
RB7
R11
4.7K
SW2
INT0
J3
J4
CON24_R
CON24_L
U4
VIN
GND
VOUT
L7905/TO3
C5
10uF
RCA JACK
J6
D9
DIODE
U2
IN
GND
OUT
LM7805C/TO
C7
1000uF
C6
10uf
C8
0.1uF
R20
1k
D8
LED
J7
CON3
AN1
AN0
SW4
SW5
SW6
RE5
RE6
RE7
designed by JAKAPHUN KAROONKITTISARN
Title
dsPIC_6015_project spectrum analyzer
Size
B
Document Number
<Doc>
Rev
<RevCode>
Date:
Friday, February 29, 2008
Sheet
1
of
1

Bill Of Materials

Item	Quantity	Reference	Part
1	2	C4,C3	20pf
2	2	C6,C5	10uF
3	1	C7	1000uF
4	1	C8	0.1uF
5	1	D1	4148
6	1	D2	LED_red
7	1	D3	LED_yellow
8	1	D8	LED
9	1	D9	DIODE
10	1	J1	CON8
11	1	J2	CON10AP
12	1	J3	CON24_R
13	1	J4	CON24_L
14	1	J6	RCA JACK
15	1	J7	CON3
16	6	R1,R3,R4,R7,R8,R9	3.3k 1/8w
17	1	R2	10k 1/8w
18	2	R5,R6	330 1/8w
19	1	R11	4.7k 1/8w
20	1	R20	1k 1/8w
21	1	SW1	RESET
22	4	SW2,SW4,SW5,SW6	SW KEY-SPST
23	1	U1	DSPic30F6015
24	1	U2	LM7805C/TO
25	1	U4	LM117
26	1	Y1	10MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<built_in.h>
char mode1=0;
char buffer_p[10];
unsigned Samples[512];
unsigned freq;
char txt[5];
unsigned Written[128];


char picture[ ] =
{
0x00,0x00,0x00,0x00,0x2E,
0x2A,0x3A,0x00,0x88,0xF8,
0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x88,0xFF,
0x00,0x00,0x00,0xC0,0x20,
0xC0,0x00,0x80,0x88,0xFF,
0x00,0x00,0x00,0x01,0x02,
0x01,0x00,0x00,0x88,0xFF,
0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x88,0xFF,
0x00,0x08,0x08,0x00,0x2E,
0x2A,0x3A,0x00,0x08,0x0F,

};

char font[ ] =
{
0xf8, 0x88, 0xf8,    // 0
 0x90, 0xf8, 0x80,    // 1
 0xe8, 0xa8, 0xb8,    // 2
 0x88, 0xa8, 0xf8,    // 3
 0x30, 0x20, 0xf8,    // 4
 0xb8, 0xa8, 0xe8,    // 5
 0xf8, 0xa8, 0xe8,    // 6
 0x08, 0xe8, 0x18,    // 7
 0xf8, 0xa8, 0xf8,    // 8
 0xb8, 0xa8, 0xf8,    // 9
 0x00, 0x00, 0x00,
};

char graph1[]={
0x00,0x00,0x00,0x00,0x00,0x08,
0x00,0x00,0x00,0x00,0x00,0x04,
0x00,0x00,0x00,0x00,0x00,0x02,
0x00,0x00,0x00,0x00,0x00,0x01,
0x00,0x00,0x00,0x00,0x80,0x00,
0x00,0x00,0x00,0x00,0x40,0x00,
0x00,0x00,0x00,0x00,0x20,0x00,
0x00,0x00,0x00,0x00,0x10,0x00,
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0x00,0x00,0x00,0x00,0x08,0x00,
0x00,0x00,0x00,0x00,0x04,0x00,
0x00,0x00,0x00,0x00,0x02,0x00,
0x00,0x00,0x00,0x00,0x01,0x00,
0x00,0x00,0x00,0x80,0x00,0x00,
0x00,0x00,0x00,0x40,0x00,0x00,
0x00,0x00,0x00,0x20,0x00,0x00,
0x00,0x00,0x00,0x10,0x00,0x00,
0x00,0x00,0x00,0x08,0x00,0x00,
0x00,0x00,0x00,0x04,0x00,0x00,
0x00,0x00,0x00,0x02,0x00,0x00,
0x00,0x00,0x00,0x01,0x00,0x00,
0x00,0x00,0x80,0x00,0x00,0x00,
0x00,0x00,0x40,0x00,0x00,0x00,
0x00,0x00,0x20,0x00,0x00,0x00,
0x00,0x00,0x10,0x00,0x00,0x00,
0x00,0x00,0x08,0x00,0x00,0x00,
0x00,0x00,0x04,0x00,0x00,0x00,
0x00,0x00,0x02,0x00,0x00,0x00,
0x00,0x00,0x01,0x00,0x00,0x00,
0x00,0x80,0x00,0x00,0x00,0x00,
0x00,0x40,0x00,0x00,0x00,0x00,
0x00,0x20,0x00,0x00,0x00,0x00,
0x00,0x10,0x00,0x00,0x00,0x00,
0x00,0x08,0x00,0x00,0x00,0x00,
0x00,0x04,0x00,0x00,0x00,0x00,
0x00,0x02,0x00,0x00,0x00,0x00,
0x00,0x01,0x00,0x00,0x00,0x00,
0x80,0x00,0x00,0x00,0x00,0x00,
0x40,0x00,0x00,0x00,0x00,0x00,
0x20,0x00,0x00,0x00,0x00,0x00,
0x10,0x00,0x00,0x00,0x00,0x00,
0x08,0x00,0x00,0x00,0x00,0x00,
};
char graph2[]={
0x00,0x00,0x00,0x00,0x00,0xc0,
0x00,0x00,0x00,0x00,0x00,0xe0,
0x00,0x00,0x00,0x00,0x00,0xf0,
0x00,0x00,0x00,0x00,0x00,0xf8,
0x00,0x00,0x00,0x00,0x00,0xfc,
0x00,0x00,0x00,0x00,0x00,0xfe,
0x00,0x00,0x00,0x00,0x00,0xff,
0x00,0x00,0x00,0x00,0x80,0xff,
0x00,0x00,0x00,0x00,0xc0,0xff,
0x00,0x00,0x00,0x00,0xe0,0xff,
0x00,0x00,0x00,0x00,0xf0,0xff,
0x00,0x00,0x00,0x00,0xf8,0xff,
0x00,0x00,0x00,0x00,0xfc,0xff,
0x00,0x00,0x00,0x00,0xfe,0xff,
0x00,0x00,0x00,0x00,0xff,0xff,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0x00,0x00,0x00,0x80,0xff,0xff,
0x00,0x00,0x00,0xc0,0xff,0xff,
0x00,0x00,0x00,0xe0,0xff,0xff,
0x00,0x00,0x00,0xf0,0xff,0xff,
0x00,0x00,0x00,0xf8,0xff,0xff,
0x00,0x00,0x00,0xfc,0xff,0xff,
0x00,0x00,0x00,0xfe,0xff,0xff,
0x00,0x00,0x00,0xff,0xff,0xff,
0x00,0x00,0x80,0xff,0xff,0xff,
0x00,0x00,0xc0,0xff,0xff,0xff,
0x00,0x00,0xe0,0xff,0xff,0xff,
0x00,0x00,0xf0,0xff,0xff,0xff,
0x00,0x00,0xf8,0xff,0xff,0xff,
0x00,0x00,0xfc,0xff,0xff,0xff,
0x00,0x00,0xfe,0xff,0xff,0xff,
0x00,0x00,0xff,0xff,0xff,0xff,
0x00,0x80,0xff,0xff,0xff,0xff,
0x00,0xc0,0xff,0xff,0xff,0xff,
0x00,0xe0,0xff,0xff,0xff,0xff,
0x00,0xf0,0xff,0xff,0xff,0xff,
0x00,0xf8,0xff,0xff,0xff,0xff,
0x00,0xfc,0xff,0xff,0xff,0xff,
0x00,0xfe,0xff,0xff,0xff,0xff,
0x00,0xff,0xff,0xff,0xff,0xff,
0x80,0xff,0xff,0xff,0xff,0xff,
0xc0,0xff,0xff,0xff,0xff,0xff,
0xe0,0xff,0xff,0xff,0xff,0xff,
0xf0,0xff,0xff,0xff,0xff,0xff,
0xf8,0xff,0xff,0xff,0xff,0xff,
0xfc,0xff,0xff,0xff,0xff,0xff,
0xfe,0xff,0xff,0xff,0xff,0xff,
0xff,0xff,0xff,0xff,0xff,0xff,
0xff,0xff,0xff,0xff,0xff,0xff,
0xff,0xff,0xff,0xff,0xff,0xff,
};
int hunningw[256]={
0.04015162 ,0.040606389,0.04136403, 0.042424084,0.043785909,0.045448677,
0.047411382,0.049672832,0.052231656,0.055086302,0.058235038,0.061675956,
0.065406969,0.069425812,0.07373005 ,0.078317072,0.083184095,0.088328169,
0.093746173,0.099434821,0.105390663,0.111610087,0.118089322,0.124824437,
0.131811349,0.139045819,0.146523459,0.154239736,0.162189969,0.170369336,
0.178772878,0.187395496,0.196231962,0.205276917,0.214524875,0.223970228,
0.233607246,0.243430086,0.25343279 ,0.263609291,0.273953418,0.284458896,
0.295119356,0.305928331,0.316879266,0.327965519,0.339180367,0.350517009,
0.361968568,0.3735281 ,0.385188593,0.396942977,0.408784123,0.420704848,
0.432697923,0.444756075,0.456871991,0.469038322,0.48124769 ,0.49349269 ,
0.505765896,0.518059864,0.530367138,0.542680255,0.554991746,0.567294145,
0.579579991,0.591841832,0.604072232,0.616263773,0.628409062,0.640500733,
0.652531451,0.664493922,0.676380889,0.688185145,0.699899528,0.711516936,
0.723030322,0.734432703,0.745717165,0.756876864,0.767905031,0.778794977,

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ทำซ้ำ ดัดแปลง หรือนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0.7895401  ,0.800133881,0.810569896,0.820841815,0.830943409,0.840868552,
0.850611224,0.860165516,0.869525634,0.878685901,0.887640761,0.896384784,
0.904912667,0.913219237,0.921299457,0.929148426,0.936761384,0.944133714,
0.951260944,0.958138753,0.964762969,0.971129574,0.977234708,0.983074667,
0.988645909,0.993945057,0.998968896,1.003714379,1.008178628,1.012358936,
1.016252768,1.019857761,1.02317173 ,1.026192664,1.028918732,1.03134828 ,
1.033479835,1.035312104,1.036843976,1.038074522,1.039002995,1.039628832,
1.039951654,1.039971265,1.039687653,1.03910099 ,1.038211632,1.037020119,
1.035527172,1.033733697,1.031640783,1.029249698,1.026561892,1.023578996,
1.020302819,1.016735348,1.012878745,1.008735351,1.004307678,0.999598411,
0.994610407,0.98934669 ,0.983810453,0.978005054,0.971934013,0.965601012,
0.959009893,0.952164652,0.945069441,0.937728564,0.930146471,0.922327763,
0.91427718 ,0.905999605,0.897500059,0.888783695,0.879855801,0.870721792,
0.861387205,0.851857704,0.842139067,0.832237188,0.822158073,0.811907835,
0.80149269 ,0.790918955,0.780193042,0.769321456,0.758310792,0.747167727,
0.735899018,0.7245115  ,0.713012079,0.70140773 ,0.689705491,0.677912458,
0.666035783,0.654082671,0.642060369,0.62997617 ,0.617837402,0.605651427,
0.593425636,0.581167443,0.568884283,0.556583605,0.54427287 ,0.531959543,
0.519651092,0.507354983,0.495078672,0.482829605,0.470615211,0.458442898,
0.446320047,0.434254011,0.422252109,0.410321618,0.398469774,0.386703766,
0.375030729,0.363457743,0.351991826,0.340639933,0.329408947,0.318305681,
0.307336868,0.296509161,0.285829127,0.275303242,0.264937891,0.254739359,
0.244713833,0.234867392,0.225206008,0.21573554 ,0.206461733,0.19739021 ,
0.188526473,0.179875899,0.171443732,0.163235088,0.155254945,0.147508142,
0.139999377,0.132733205,0.125714033,0.118946117,0.112433563,0.106180319,
0.100190178,0.094466773,0.089013576,0.083833893,0.078930867,0.074307469,
0.069966506,0.065910608,0.062142237,0.058663677,0.055477038,0.052584253,
0.049987077,0.047687083,0.045685668,0.043984045,0.042583246,0.041484121,
0.040687336,0.040193375,0.040002537,0.040114937,
};
unsigned spectrum[75];
void OscillatorFailTrap() org 0x06 { // if oscillator fails, the code jumps here
 trisf = 0;
   asm{
   MOV [w15-34], w13
   LSR  w13, #8, w13
   MOV w13, LATF
   ;LSR  w15, #8, w13
   ;MOV w13, LATB

  }
 while(1);
}

void AddressTrap() org 0x08 {  // if the addressing mode is wrong, the code jumps
here
   trisd = 0;
   asm{
   MOV [w15-34], w13
   ;LSR  w13, #8, w13
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   MOV w13, LATd
  }
  while(1);
}

void StackErrorTrap() org 0x0A { // stack overflow, underflow...
trisf = 0;
   asm{
   MOV [w15-34], w13
   LSR  w13, #8, w13
   MOV w13, LATF
   ;LSR  w15, #8, w13
   ;MOV w13, LATB

  }  while(1);
}

void MathErrorTrap() org 0x0C { // div by zero etc...
 trisf = 0;
   asm{
   MOV [w15-34], w13
   LSR  w13, #8, w13
   MOV w13, LATF
   ;LSR  w15, #8, w13
   ;MOV w13, LATB

  }
  while(1);
}
void MainInit()
{
  Twiddle_Factors_Init();
}
unsigned ReadAdc() {

  ADCON1.F1 = 1;              // Start AD conversion
  while (ADCON1.F0 == 0)          // Wait for ADC to finish
    asm nop;
  return ADCBUF0;             // Get ADC value
}//~


//---------- Fills "Samples" with input samples in manner Re, Im, Re, Im... where
Im = 0
void SampleInput()
{
char j=0;
int adc0,adc1=0,adc2;
   do
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      adc2=Adc_Read(0);
      if(adc2<adc1-5)
      {
         j=1;
      }
      adc1=adc2;
    }while(j==0);
    j=0;
   do
   {
     adc2=Adc_Read(0);
     if(adc2>adc1 && adc2>490)
     {
           int i = 0;
           while (i <= 512)
           {
             adc0=  Adc_Read(0);
             Samples[i++] =  adc0-512;
             Samples[i++] = 0;
           }
           j=1;
     }
     adc1=adc2;
  }while(j==0);

}//~
lcd_write_data(char data1);
lcd_write_command(char command1);
lcd_initial();
lcd_clear_screen();
lcd_write_picture();
lcd_gotoxy(char x,char y);
lcd_put_char(char character);
lcd_print_string(char *string);
read_wave();
putpoint(char a,char x);
putpoint2(char a,char x);
putgraph();
spoint1();
print_s();
print_s2();
sprint();
print_lcd(int a);
plotspectrum();
float int2float(int input);
lcd_write_picture_s();
scankey();
frequency1();
hunning();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main() {   MainInit();

        TRISD = 0xffc0;
        TRISE = 0xfffe;
        TRISB = 0xffff;
        LATD=0xff00;//LCD3310_RES = 0;
        delay_us(20);
        LATD=0xff26;    //set LCD
        LATE |= 0x0001;
        delay_ms(300);
        lcd_initial();
        lcd_clear_screen();
          lcd_write_picture();

 while (1) {                // Infinite loop
   scankey();
        SampleInput();


        if(mode1==0)
        {
          frequency1();
          putgraph();
          lcd_gotoxy(70,0);
          print_lcd(freq);
        }

        else  if(mode1==1)
        {
          frequency1();
          FFT(8, TwiddleCoeff_256, Samples);
          BitReverseComplex(8, Samples);
          spoint1();
          plotspectrum();
          lcd_gotoxy(70,0);
          print_lcd(freq);
        }
        else  if(mode1==2)
        {
          frequency1();
          hunning();
          FFT(8, TwiddleCoeff_256, Samples);
          BitReverseComplex(8, Samples);
          spoint1();
          plotspectrum();
          lcd_gotoxy(70,0);
          print_lcd(freq);
         }

 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
hunning()
{
        int i,n;
        i=0;
        for(n=0;n<256;n++)
        {
            Samples[i]=Samples[i]*hunningw[n];
            i=i+2;
        }
}

float int2float(int input)
{

  if (input < 0)
    input = - input;
  return (input /512.0);
}
frequency1()
{
  char j;
  int i,r,f,trick,trick1;
  i=0;
  j=0;
  trick=Samples[0];
  trick1=Samples[0]-5;
  do
  {
     r=Samples[i];
     if(r<=trick1)
     {
     j=1;
     }
     i++;
     i++;
  }while(j==0);
  j=0;
  do
  {
     r=Samples[i];
     if(r>=trick)
      {
        j=1;
      }
     i++;
     i++;
  }while(j==0);
 freq=6730/i;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
scankey()
{
        int x,y;
        x=PORTE;
        x&=0x00e0;
        y=x;
        if(x!=0x00e0)
        {
          if(x==0x00c0)
          {
             mode1=0;
             lcd_clear_screen();
             lcd_write_picture();
          }
          if(x==0x00a0)
          {
             mode1=1;
             lcd_clear_screen();
             lcd_write_picture_s();
          }
          if(x==0x0060)
          {
             mode1=2;
             lcd_clear_screen();
             lcd_write_picture_s();
          }
          do
          {
             x=PORTE;
             x&=0x00e0;
          }
          while(x==y);
        }
}
spoint1()
{
        char a,j;
        int e,f,i,sf,pm;
        float re,im,re2,im2,am;
        a=0;
        j=0;
        while(a<70)
        {
            e=Samples[j++];

            f=Samples[j++];

            re=int2float(e);
            im=int2float(f);
            re2=re*re;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            im2=im*im;
            am=sqrt(re2+im2);
            pm=am*100.;
            sf=pm;
            spectrum[a]=sf;
            a++;
        }
}
plotspectrum()
{
            char a;
            a=0;
            while(a<=65)
            {
               putpoint2(spectrum[a],a+11);
               a++;
            }
}
 putgraph()
{
          char m,n,o;
        for(m=0;m<140;m=m+2)
            {
               o=m/2;
               n = (Samples[m]+512)/25;
                     putpoint(n,o+10);
            }
}
putpoint(char a,char x)
{
          char b;
          int c;
          c=a*6;
           for(b=0;b<=5;b++)
          {
                    lcd_gotoxy(x,b);
                    lcd_write_data(graph1[c]);
                    c++;
          }
}
putpoint2(char a,char x)
{
          char b;
          int c;
          c=a*6;
           for(b=0;b<=5;b++)
          {
                    lcd_gotoxy(x,b);
                    lcd_write_data(graph2[c]);
                    c++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}
lcd_write_data(char Data1)
{
 char ac ;

 LATD |=0x0004;
 for (ac=0;ac<8;ac++)
 {
    if(Data1&0x80)
    {
       LATD |=0x0008;
    }
    else
    {
       LATD &=0xfff7;
    }
         LATD |=0x0010;
         delay_us(20);
    LATD &=0xffef;
    delay_us(20);
    Data1<<=1;
 }
 }

lcd_write_command(char command1)
{
 char ab;

 LATD &=0xfffb;
 for (ab=0;ab<8;ab++)
 {
    if(command1&0x80)
    {
       LATD |=0x0008;
    }
    else
    {
       LATD &=0xfff7;
    }
       LATD |=0x0010;
          delay_us(20);
    LATD &=0xffef;
    delay_us(20);
    command1<<=1;
 }
}

lcd_initial(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  LATD &=0xfffd;
  delay_us(20);
  LATD |=0x0002;
  delay_us(20);

  lcd_write_command(0x21);
  lcd_write_command(128+38);
  lcd_write_command(4+3);
  lcd_write_command(16+3);

  lcd_write_command(32+0);
  lcd_write_command(12);
}


lcd_clear_screen(void)
{
 unsigned int  i=0;

 lcd_write_command(128+0);
 lcd_write_command(64+0);

 for(i=0;i<504;i++)
 lcd_write_data(0);
}

lcd_write_picture()
{
   char a,b,c,i;
        a=0;
        b=0;
        do
        {
                lcd_gotoxy(a,b);
                c=0;
                i=b*10;
                do
                {
                        lcd_write_data(picture[i]);
                        c++;
                        i++;
                }while(c<10);
                a=0;
                b++;
        }while(b<=5);
}
lcd_write_picture_s()
{
      char a,b;
      a=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while(a<=5)
    {
      lcd_gotoxy(9,a);
      lcd_write_data(0xff);
      a++;
    }
    a=0;
    while(a<=5)
    {
      lcd_gotoxy(8,a);
      lcd_write_data(0x44);
      a++;
    }
    b=10;
    while(b<=83)
    {
      lcd_gotoxy(b,5);
      lcd_write_data(0x40);
      b++;
    }
}

lcd_gotoxy(char x,char y)
{
 lcd_write_command(128+x);
 lcd_write_command(64+y);
}

lcd_put_char(char character)
{
 char font_size_count = 0;
 int  font_data_index;

 font_data_index = character-48;
 font_data_index = font_data_index*3;

 while(font_size_count<3)
 {
  lcd_write_data(font[font_data_index]);
  font_size_count++;
  font_data_index++;
 }
 lcd_write_data(0);
}
lcd_print_string(char *string)
{
      char i;
      i=0;
       do
       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcd_put_char(buffer_p[i]);
            i++;
          }
        while(buffer_p[i]!=0x00);

}
print_lcd(int a)
{
          sprinti(buffer_p,"%d",a);
          lcd_print_string(buffer_p);
 }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010A/6015
Data Sheet

High-Performance, 16-Bit Digital Signal Controllers

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010A/6015 Enhanced Flash 16-bit Digital Signal Controller (DSC)

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 Programmers 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
- 144 Kbytes on-chip Flash program space (Instruction words)
- 8 Kbytes of on-chip data RAM
- 4 Kbytes 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)
  - 7.37 MHz internal RC with PLL active (4x, 8x, 16x)
- 44 interrupt sources:
  - 5 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 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
- 2 CAN modules, 2.0B compliant (dsPIC306010A)
- 1 CAN module, 2.0B compliant (dsPIC306015)

Motor Control PWM Module Features:

- 8 PWM output channels:
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 4 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 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analog Features:

- 10-bit Analog-to-Digital Converter (ADC) with 4 S/H Inputs:
  - 1 Msps conversion rate
  - 16 input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

Special Microcontroller 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)
- 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	A/D 10-bit 1 Msps	Quad Enc	UART	SPI	I^2C™	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
dsPIC30F5016	80	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010A	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2
dsPIC30F6015	64	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	1

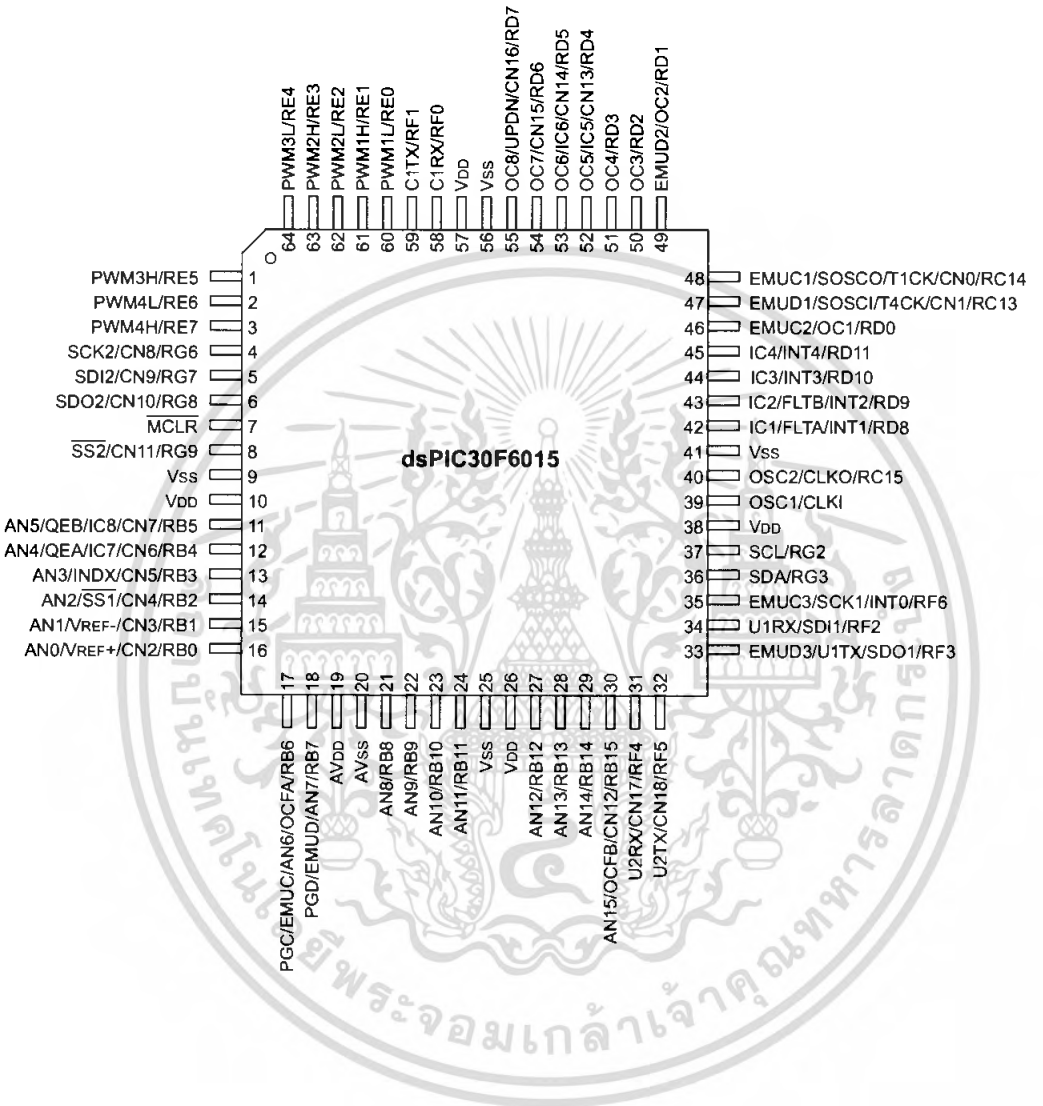
* This table provides a summary of the dsPIC30F peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagram

64-Pin TQFP

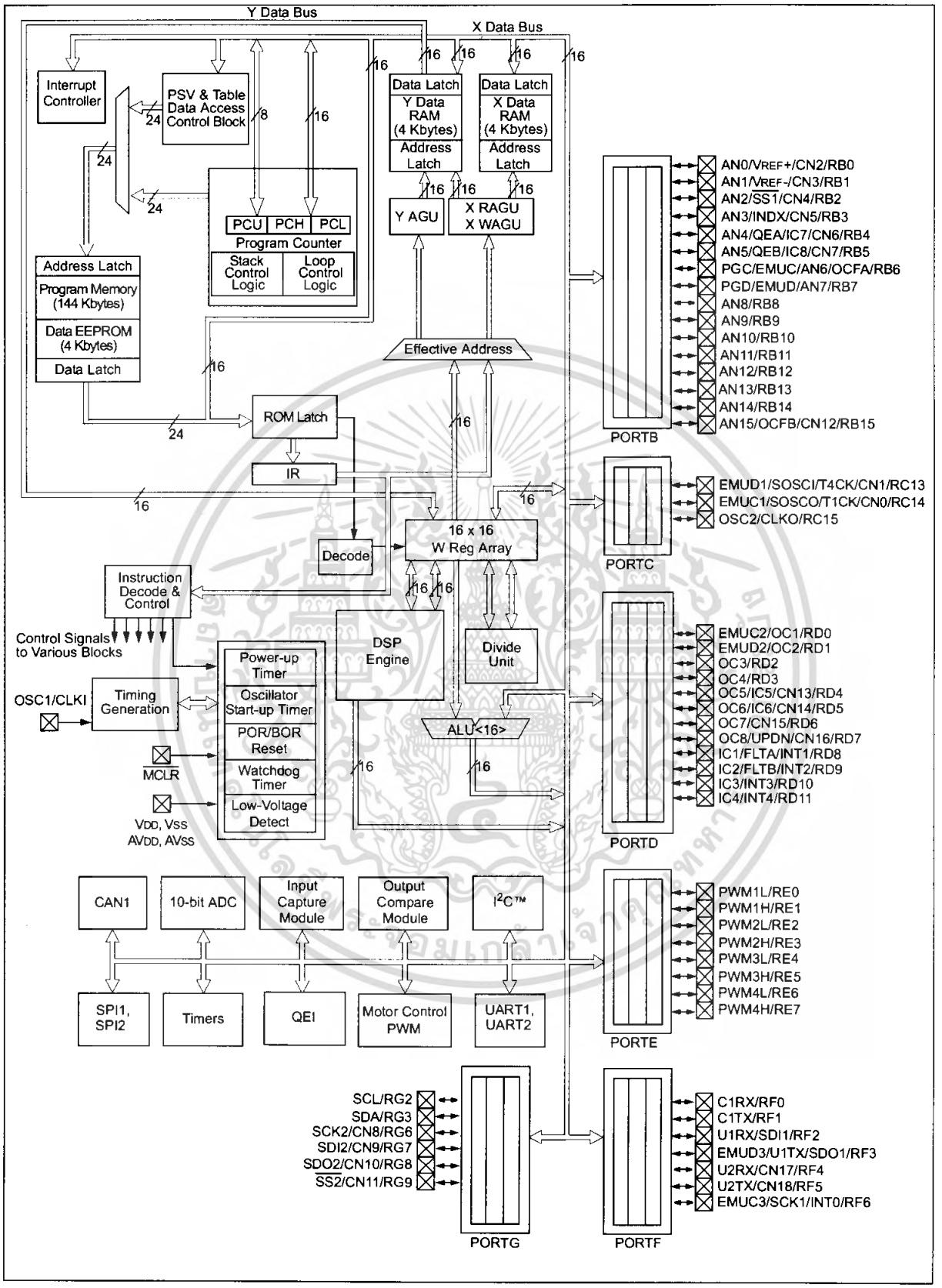


Note: Pinout subject to change.

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1-2: dsPIC30F6015 BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1-1 provides a brief description of the device I/O pinout and the functions that are multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN15	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-CN23	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
C1RX C1TX C2RX C2TX	I O I O	ST — ST —	CAN1 bus receive pin. CAN1 bus transmit pin. CAN2 bus receive pin. CAN2 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-IC8	I	ST	Capture inputs 1 through 8.
INDX QEA QEB UPDN	I I I O	ST ST ST CMOS	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. Position Up/Down Counter Direction State.
INT0 INT1 INT2 INT3 INT4	I I I I I	ST ST ST ST ST	External interrupt 0. External interrupt 1. External interrupt 2. External interrupt 3. External interrupt 4.

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

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
FLTA	I	ST	PWM Fault A input.
FLTB	I	ST	PWM Fault B input.
PWM1L	O	—	PWM 1 Low output.
PWM1H	O	—	PWM 1 High output.
PWM2L	O	—	PWM 2 Low output.
PWM2H	O	—	PWM 2 High output.
PWM3L	O	—	PWM 3 Low output.
PWM3H	O	—	PWM 3 High output.
PWM4L	O	—	PWM 4 Low output.
PWM4H	O	—	PWM 4 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	I	ST	Compare Fault A input (for Compare channels 1, 2, 3 and 4).
OCFB	I	ST	Compare Fault B input (for Compare channels 5, 6, 7 and 8).
OC1-OC8	O	—	Compare outputs 1 through 8.
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.
RA9-RA10	I/O	ST	PORTA is a bidirectional I/O port.
RA14-RA15	I/O	ST	
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC1	I/O	ST	PORTC is a bidirectional I/O port.
RC3	I/O	ST	
RC13-RC15	I/O	ST	
RD0-RD15	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE9	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF8	I/O	ST	PORTF is a bidirectional I/O port.
RG0-RG3	I/O	ST	PORTG is a bidirectional I/O port.
RG6-RG9	I/O	ST	
SCK1	I/O	ST	Synchronous serial clock input/output for SPI #1.
SDI1	I	ST	SPI #1 Data In.
SDO1	O	—	SPI #1 Data Out.
$\overline{SS1}$	I	ST	SPI #1 Slave Synchronization.
SCK2	I/O	ST	Synchronous serial clock input/output for SPI #2.
SDI2	I	ST	SPI #2 Data In.
SDO2	O	—	SPI #2 Data Out.
$\overline{SS2}$	I	ST	SPI #2 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.
T4CK	I	ST	Timer4 external clock 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

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
U1RX	I	ST	UART1 Receive.
U1TX	O	—	UART1 Transmit.
U1ARX	I	ST	UART1 Alternate Receive.
U1ATX	O	—	UART1 Alternate Transmit.
U2RX	I	ST	UART2 Receive.
U2TX	O	—	UART2 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

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

2.0 CPU ARCHITECTURE OVERVIEW

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 Programmers Reference Manual"* (DS70157).

This document summarizes the CPU and peripheral functions of the dsPIC30F6010A/6015. For a complete description of this functionality, please refer to the *"dsPIC30F Family Reference Manual"* (DS70046).

2.1 Core Overview

The core has a 24-bit instruction word. The Program Counter (PC) is 23 bits wide with the Least Significant bit (LSb) always clear (see **Section 3.1 "Program Address Space"**), and the Most Significant bit (MSb) is ignored during normal program execution, except for certain specialized instructions. Thus, the PC can address up to 4M instruction words of user program space. An instruction prefetch mechanism is used to help maintain throughput. Program loop constructs, free from loop count management overhead, are supported using the DO and REPEAT instructions, both of which are interruptible at any point.

The working register array consists of 16x16-bit registers, each of which can act as data, address or offset registers. One working register (W15) operates as a Software Stack Pointer for interrupts and calls.

The data space is 64 Kbytes (32K words) and is split into two blocks, referred to as X and Y data memory. Each block has its own independent Address Generation Unit (AGU). Most instructions operate solely through the X memory AGU, which provides the appearance of a single unified data space. The Multiply-Accumulate (MAC) class of dual source DSP instructions operate through both the X and Y AGUs, splitting the data address space into two parts (see **Section 3.2 "Data Address Space"**). The X and Y data space boundary is device-specific and cannot be altered by the user. Each data word consists of 2 bytes, and most instructions can address data either as words or bytes.

There are two methods of accessing data stored in program memory:

- The upper 32 Kbytes of data space memory can be mapped into the lower half (user space) of program space at any 16K program word boundary, defined by the 8-bit Program Space Visibility Page (PSVPAG) register. This lets any instruction access program space as if it were data space, with a limitation that the access requires an additional cycle. Moreover, only the lower 16 bits of each instruction word can be accessed using this method.
- Linear indirect access of 32K word pages within program space is also possible using any working register, via table read and write instructions. Table read and write instructions can be used to access all 24 bits of an instruction word.

Overhead-free circular buffers (Modulo Addressing) are supported in both X and Y address spaces. This is primarily intended to remove the loop overhead for DSP algorithms.

The X AGU also supports Bit-Reversed Addressing on destination Effective Addresses, to greatly simplify input or output data reordering for radix-2 FFT algorithms. Refer to **Section 4.0 "Address Generator Units"** for details on Modulo and Bit-Reversed Addressing.

The core supports Inherent (no operand), Relative, Literal, Memory Direct, Register Direct, Register Indirect, Register Offset and Literal Offset Addressing modes. Instructions are associated with predefined addressing modes, depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, 3-operand instructions are supported, allowing C = A + B operations to be executed in a single cycle.

A DSP engine has been included to significantly enhance the core arithmetic capability and throughput. It features a high-speed 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bidirectional barrel shifter. Data in the accumulator or any working register can be shifted up to 16 bits right or 16 bits left in a single cycle. The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance. The MAC class of instructions can concurrently fetch two data operands from memory, while multiplying two W registers. To enable this concurrent fetching of data operands, the data space has been split for these instructions and linear for all others. This has been achieved in a transparent and flexible manner, by dedicating certain working registers to each address space for the MAC class of instructions.

The core does not support a multi-stage instruction pipeline. However, a single stage instruction prefetch mechanism is used, which accesses and partially decodes instructions a cycle ahead of execution, in order to maximize available execution time. Most instructions execute in a single cycle, with certain exceptions.

The core features a vectored exception processing structure for traps and interrupts, with 62 independent vectors. The exceptions consist of up to 8 traps (of which 4 are reserved) and 54 interrupts. Each interrupt is prioritized based on a user-assigned priority between 1 and 7 (1 being the lowest priority and 7 being the highest) in conjunction with a predetermined 'natural order'. Traps have fixed priorities, ranging from 8 to 15.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Programmer's Model

The programmer's model is shown in Figure 2-1 and consists of 16x16-bit working registers (W0 through W15), 2x40-bit accumulators (AccA and AccB), STATUS register (SR), Data Table Page register (TBLPAG), Program Space Visibility Page register (PSVPAG), DO and REPEAT registers (DOSTART, DOEND, DCOUNT and RCOUNT), and Program Counter (PC). The working registers can act as data, address or offset registers. All registers are memory mapped. W0 acts as the W register for file register addressing.

Some of these registers have a shadow register associated with each of them, as shown in Figure 2-1. The shadow register is used as a temporary holding register and can transfer its contents to or from its host register upon the occurrence of an event. None of the shadow registers are accessible directly. The following rules apply for transfer of registers into and out of shadows.

- `PUSH.S` and `POP.S`
  W0, W1, W2, W3, SR (DC, N, OV, Z and C bits only) are transferred.
- `DO` instruction
  DOSTART, DOEND, DCOUNT shadows are pushed on loop start and popped on loop end.

When a byte operation is performed on a working register, only the Least Significant Byte of the target register is affected. However, a benefit of memory mapped working registers is that both the Least and Most Significant Bytes can be manipulated through byte-wide data memory space accesses.

2.2.1 SOFTWARE STACK POINTER/ FRAME POINTER

The dsPIC® DSC devices contain a software stack. W15 is the dedicated Software Stack Pointer (SP), and will be automatically modified by exception processing and subroutine calls and returns. However, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies the reading, writing and manipulation of the Stack Pointer (e.g., creating stack frames).

Note:	In order to protect against misaligned stack accesses, W15<0> is always clear.

W15 is initialized to 0x0800 during a Reset. The user may reprogram the SP during initialization to any location within data space.

W14 has been dedicated as a Stack Frame Pointer as defined by the `LNK` and `ULNK` instructions. However, W14 can be referenced by any instruction in the same manner as all other W registers.

2.2.2 STATUS REGISTER

The dsPIC DSC core has a 16-bit STATUS register (SR), the LSB of which is referred to as the SR Low Byte (SRL) and the MSB as the SR High Byte (SRH). See Figure 2-1 for SR layout.

SRL contains all the MCU ALU operation status flags (including the Z bit), as well as the CPU Interrupt Priority Level Status bits, IPL<2:0>, and the Repeat Active Status bit, RA. During exception processing, SRL is concatenated with the MSB of the PC to form a complete word value which is then stacked.

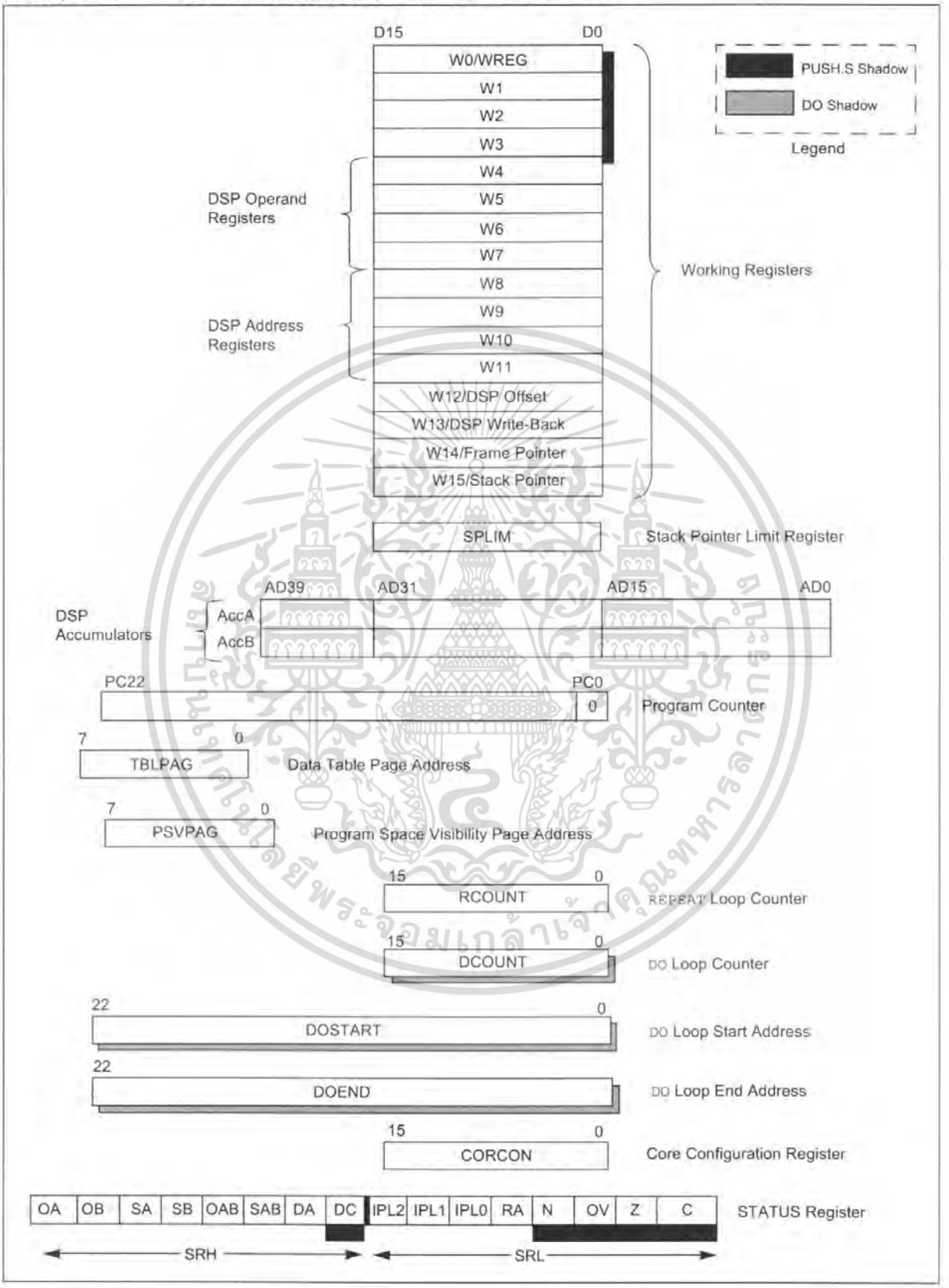
The upper byte of the SR register contains the DSP adder/subtractor Status bits, the DO Loop Active bit (DA) and the Digit Carry (DC) Status bit.

2.2.3 PROGRAM COUNTER

The Program Counter is 23 bits wide. Bit 0 is always clear. Therefore, the PC can address up to 4M instruction words.

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

FIGURE 2-1: dsPIC30F6010A/6015 PROGRAMMER'S MODEL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Divide Support

The dsPIC DSC devices feature a 16/16-bit signed fractional divide operation, as well as 32/16-bit and 16/16-bit signed and unsigned integer divide operations, in the form of single instruction iterative divides. The following instructions and data sizes are supported:

1. DIVF – 16/16 signed fractional divide
2. DIV.sd – 32/16 signed divide
3. DIV.ud – 32/16 unsigned divide
4. DIV.s – 16/16 signed divide
5. DIV.u – 16/16 unsigned divide

The divide instructions must be executed within a REPEAT loop. Any other form of execution (e.g., a series of discrete divide instructions) will not function correctly because the instruction flow depends on RCOUNT. The divide instruction does not automatically set up the RCOUNT value, and it must, therefore, be explicitly and correctly specified in the REPEAT instruction, as shown in Table 2-1 (REPEAT will execute the target instruction {operand value + 1} times). The REPEAT loop count must be set up for 18 iterations of the DIV/DIVF instruction. Thus, a complete divide operation requires 19 cycles.

Note:	The divide flow is interruptible. However, the user needs to save the context as appropriate.

TABLE 2-1: DIVIDE INSTRUCTIONS

Instruction	Function
DIVF	Signed fractional divide: Wm/Wn → W0; Rem → W1
DIV.sd	Signed divide: (Wm+1:Wm)/Wn → W0; Rem → W1
DIV.s	Signed divide: Wm/Wn → W0; Rem → W1
DIV.ud	Unsigned divide: (Wm+1:Wm)/Wn → W0; Rem → W1
DIV.u	Unsigned divide: Wm/Wn → W0; Rem → W1

2.4 DSP Engine

The DSP engine consists of a high-speed 17-bit x 17-bit multiplier, a barrel shifter, and a 40-bit adder/subtractor (with two target accumulators, round and saturation logic).

The dsPIC30F devices have a single instruction flow which can execute either DSP or MCU instructions. Many of the hardware resources are shared between the DSP and MCU instructions. For example, the instruction set has both DSP and MCU multiply instructions which use the same hardware multiplier.

The DSP engine also has the capability to perform inherent accumulator-to-accumulator operations, which require no additional data. These instructions are ADD, SUB and NEG.

The DSP engine has various options selected through various bits in the CPU Core Configuration register (CORCON), as listed below:

1. Fractional or Integer DSP Multiply (IF).
2. Signed or Unsigned DSP Multiply (US).
3. Conventional or Convergent Rounding (RND).
4. Automatic Saturation On/Off for AccA (SATA).
5. Automatic Saturation On/Off for AccB (SATB).
6. Automatic Saturation On/Off for Writes to Data Memory (SATDW).
7. Accumulator Saturation mode Selection (ACCSAT).

Note:	For CORCON layout, see Table 3-3.

A block diagram of the DSP engine is shown in Figure 2-2.

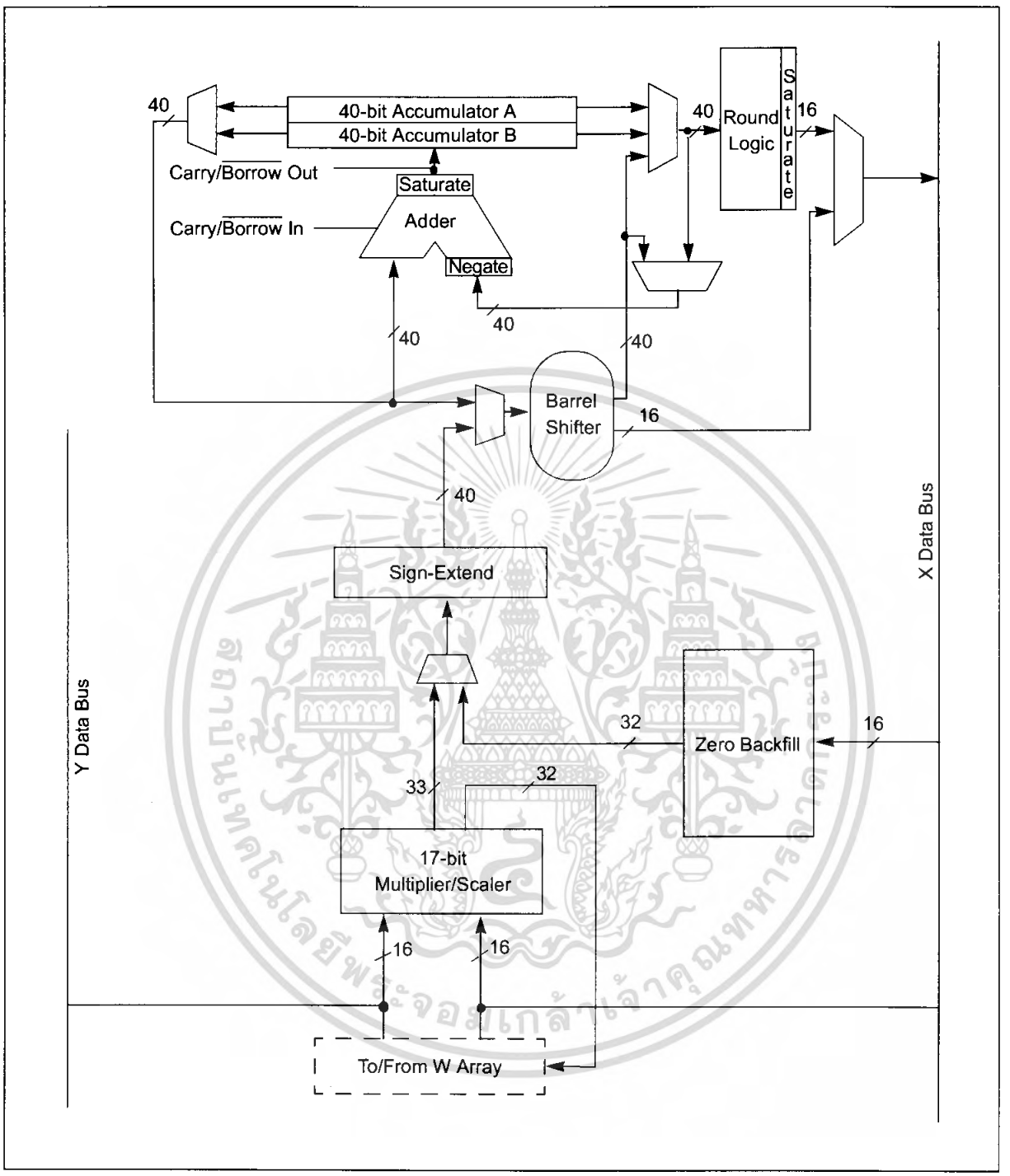
TABLE 2-2: DSP INSTRUCTION SUMMARY

Instruction	Algebraic Operation
CLR	A = 0
ED	A = $(x - y)^2$
EDAC	A = A + $(x - y)^2$
MAC	A = A + (x * y)
MOVSAC	No change in A
MPY	A = x * y
MPY.N	A = – x * y
MSC	A = A – x * y

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 2-2: DSP ENGINE BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 MULTIPLIER

The 17x17-bit multiplier is capable of signed or unsigned operations and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17x17-bit multiplier/scaler is a 33-bit value, which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSB is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF), including 0. For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,645 (0x7FFF FFFF).

When the multiplier is configured for fractional multiplication, the data is represented as a two's complement fraction, where the MSB is defined as a sign bit and the radix point is implied to lie just after the sign bit (QX format). The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1-2^{1-N})$. For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF), including 0 and has a precision of 3.01518×10^{-5}. In Fractional mode, a 16x16 multiply operation generates a 1.31 product, which has a precision of 4.65661×10^{-10}.

The same multiplier is used to support the MCU multiply instructions, which include integer 16-bit signed, unsigned and mixed sign multiplies.

The MUL instruction may be directed to use byte or word-sized operands. Byte operands will direct a 16-bit result, and word operands will direct a 32-bit result to the specified register(s) in the W array.

2.4.2 DATA ACCUMULATORS AND ADDER/SUBTRACTOR

The data accumulator consists of a 40-bit adder/subtractor with automatic sign extension logic. It can select one of two accumulators (A or B) as its pre-accumulation source and post-accumulation destination. For the ADD and LAC instructions, the data to be accumulated or loaded can be optionally scaled via the barrel shifter, prior to accumulation.

2.4.2.1 Adder/Subtractor, Overflow and Saturation

The adder/subtractor is a 40-bit adder with an optional zero input into one side and either true or complement data into the other input. In the case of addition, the carry/borrow input is active-high and the other input is true data (not complemented), whereas in the case of subtraction, the carry/borrow input is active-low and the other input is complemented. The adder/subtractor generates Overflow Status bits, SA/SB and OA/OB, which are latched and reflected in the STATUS register.

- Overflow from bit 39: this is a catastrophic overflow in which the sign of the accumulator is destroyed.
- Overflow into guard bits 32 through 39: this is a recoverable overflow. This bit is set whenever all the guard bits are not identical to each other.

The adder has an additional saturation block which controls accumulator data saturation, if selected. It uses the result of the adder, the Overflow Status bits described above, and the SATA/B (CORCON<7:6>) and ACCSAT (CORCON<4>) mode control bits to determine when and to what value to saturate.

Six STATUS register bits have been provided to support saturation and overflow. They are:

1. OA:
   AccA overflowed into guard bits
2. OB:
   AccB overflowed into guard bits
3. SA:
   AccA saturated (bit 31 overflow and saturation)
   or
   AccA overflowed into guard bits and saturated (bit 39 overflow and saturation)
4. SB:
   AccB saturated (bit 31 overflow and saturation)
   or
   AccB overflowed into guard bits and saturated (bit 39 overflow and saturation)
5. OAB:
   Logical OR of OA and OB
6. SAB:
   Logical OR of SA and SB

The OA and OB bits are modified each time data passes through the adder/subtractor. When set, they indicate that the most recent operation has overflowed into the accumulator guard bits (bits 32 through 39). The OA and OB bits can also optionally generate an arithmetic warning trap when set and the corresponding overflow trap flag enable bit (OVATE, OVBTE) in the INTCON1 register (refer to **Section 5.0 "Interrupts"**) is set. This allows the user to take immediate action, for example, to correct system gain.

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

The SA and SB bits are modified each time data passes through the adder/subtractor, but can only be cleared by the user. When set, they indicate that the accumulator has overflowed its maximum range (bit 31 for 32-bit saturation, or bit 39 for 40-bit saturation) and will be saturated if saturation is enabled. When saturation is not enabled, SA and SB default to bit 39 overflow and thus indicate that a catastrophic overflow has occurred. If the COVTE bit in the INTCON1 register is set, SA and SB bits will generate an arithmetic warning trap when saturation is disabled.

The Overflow and Saturation Status bits can optionally be viewed in the STATUS register (SR) as the logical OR of OA and OB (in bit OAB) and the logical OR of SA and SB (in bit SAB). This allows programmers to check one bit in the STATUS register to determine if either accumulator has overflowed, or one bit to determine if either accumulator has saturated. This would be useful for complex number arithmetic which typically uses both the accumulators.

The device supports three Saturation and Overflow modes.

1. Bit 39 Overflow and Saturation:
   When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFFFF) or maximally negative 9.31 value (0x8000000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. This is referred to as 'super saturation' and provides protection against erroneous data or unexpected algorithm problems (e.g., gain calculations).
2. Bit 31 Overflow and Saturation:
   When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFFF) or maximally negative 1.31 value (0x0080000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used so the OA, OB or OAB bits are never set.
3. Bit 39 Catastrophic Overflow
   The bit 39 Overflow Status bit from the adder is used to set the SA or SB bit, which remain set until cleared by the user. No saturation operation is performed and the accumulator is allowed to overflow (destroying its sign). If the COVTE bit in the INTCON1 register is set, a catastrophic overflow can initiate a trap exception.

2.4.2.2 Accumulator 'Write-Back'

The MAC class of instructions (with the exception of MPY, MPY.N, ED and EDAC) can optionally write a rounded version of the high word (bits 31 through 16) of the accumulator that is not targeted by the instruction into data space memory. The write is performed across the X bus into combined X and Y address space. The following addressing modes are supported:

1. W13, Register Direct:
   The rounded contents of the non-target accumulator are written into W13 as a 1.15 fraction.
2. [W13]+ = 2, Register Indirect with Post-Increment:
   The rounded contents of the non-target accumulator are written into the address pointed to by W13 as a 1.15 fraction. W13 is then incremented by 2 (for a word write).

2.4.2.3 Round Logic

The round logic is a combinational block, which performs a conventional (biased) or convergent (unbiased) round function during an accumulator write (store). The Round mode is determined by the state of the RND bit in the CORCON register. It generates a 16-bit, 1.15 data value which is passed to the data space write saturation logic. If rounding is not indicated by the instruction, a truncated 1.15 data value is stored and the least significant word is simply discarded.

Conventional rounding takes bit 15 of the accumulator, zero-extends it and adds it to the ACCxH word (bits 16 through 31 of the accumulator). If the ACCxL word (bits 0 through 15 of the accumulator) is between 0x8000 and 0xFFFF (0x8000 included), ACCxH is incremented. If ACCxL is between 0x0000 and 0x7FFF, ACCxH is left unchanged. A consequence of this algorithm is that over a succession of random rounding operations, the value will tend to be biased slightly positive.

Convergent (or unbiased) rounding operates in the same manner as conventional rounding, except when ACCxL equals 0x8000. If this is the case, the LSb (bit 16 of the accumulator) of ACCxH is examined. If it is '1', ACCxH is incremented. If it is '0', ACCxH is not modified. Assuming that bit 16 is effectively random in nature, this scheme will remove any rounding bias that may accumulate.

The SAC and SAC.R instructions store either a truncated (SAC) or rounded (SAC.R) version of the contents of the target accumulator to data memory, via the X bus (subject to data saturation, see **Section 2.4.2.4 "Data Space Write Saturation"**). Note that for the MAC class of instructions, the accumulator write-back operation will function in the same manner, addressing combined MCU (X and Y) data space though the X bus. For this class of instructions, the data is always subject to rounding.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.4 Data Space Write Saturation

In addition to adder/subtractor saturation, writes to data space may also be saturated, but without affecting the contents of the source accumulator. The data space write saturation logic block accepts a 16-bit, 1.15 fractional value from the round logic block as its input, together with overflow status from the original source (accumulator) and the 16-bit round adder. These are combined and used to select the appropriate 1.15 fractional value as output to write to data space memory.

If the SATDW bit in the CORCON register is set, data (after rounding or truncation) is tested for overflow and adjusted accordingly. For input data greater than 0x007FFF, data written to memory is forced to the maximum positive 1.15 value, 0x7FFF. For input data less than 0xFF8000, data written to memory is forced to the maximum negative 1.15 value, 0x8000. The MSb of the source (bit 39) is used to determine the sign of the operand being tested.

If the SATDW bit in the CORCON register is not set, the input data is always passed through unmodified under all conditions.

2.4.3 BARREL SHIFTER

The barrel shifter is capable of performing up to 16-bit arithmetic or logic right shifts, or up to 16-bit left shifts in a single cycle. The source can be either of the two DSP accumulators or the X bus (to support multi-bit shifts of register or memory data).

The shifter requires a signed binary value to determine both the magnitude (number of bits) and direction of the shift operation. A positive value will shift the operand right. A negative value will shift the operand left. A value of '0' will not modify the operand.

The barrel shifter is 40 bits wide, thereby obtaining a 40-bit result for DSP shift operations and a 16-bit result for MCU shift operations. Data from the X bus is presented to the barrel shifter between bit positions 16 to 31 for right shifts, and bit positions 0 to 15 for left shifts.

© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

PCD8544
48 × 84 pixels matrix LCD controller/driver

Product specification
File under Integrated Circuits, IC17

1999 Apr 12

Philips
Semiconductors

PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

48 × 84 pixels matrix LCD controller/driver PCD8544

1 FEATURES

- Single chip LCD controller/driver
- 48 row, 84 column outputs
- Display data RAM 48 × 84 bits
- On-chip:
  - Generation of LCD supply voltage (external supply also possible)
  - Generation of intermediate LCD bias voltages
  - Oscillator requires no external components (external clock also possible).
- External $\overline{RES}$ (reset) input pin
- Serial interface maximum 4.0 Mbits/s
- CMOS compatible inputs
- Mux rate: 48
- Logic supply voltage range V_{DD} to V_{SS}: 2.7 to 3.3 V
- Display supply voltage range V_{LCD} to V_{SS}
  - 6.0 to 8.5 V with LCD voltage internally generated (voltage generator enabled)
  - 6.0 to 9.0 V with LCD voltage externally supplied (voltage generator switched-off).
- Low power consumption, suitable for battery operated systems
- Temperature compensation of V_{LCD}
- Temperature range: –25 to +70 °C.

2 GENERAL DESCRIPTION

The PCD8544 is a low power CMOS LCD controller/driver, designed to drive a graphic display of 48 rows and 84 columns. All necessary functions for the display are provided in a single chip, including on-chip generation of LCD supply and bias voltages, resulting in a minimum of external components and low power consumption.

The PCD8544 interfaces to microcontrollers through a serial bus interface.

The PCD8544 is manufactured in n-well CMOS technology.

3 APPLICATIONS

- Telecommunications equipment.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCD8544U	–	chip with bumps in tray; 168 bonding pads + 4 dummy pads	–

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 BLOCK DIAGRAM

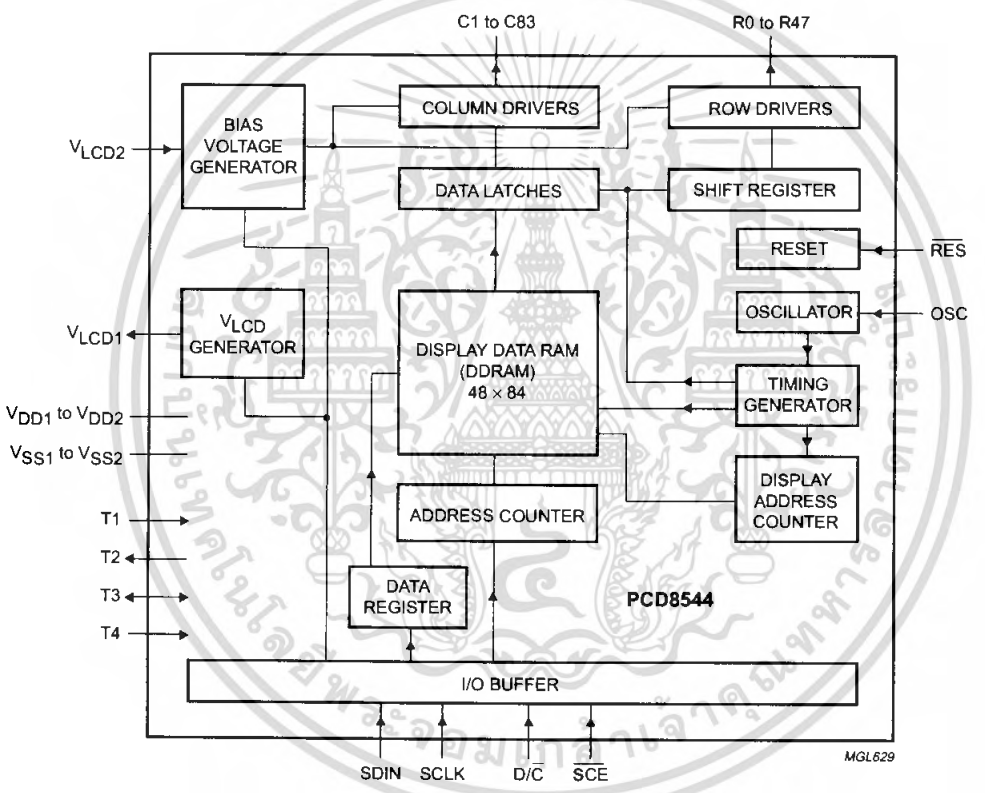


Fig.1 Block diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 PINNING

SYMBOL	DESCRIPTION
R0 to R47	LCD row driver outputs
C0 to C83	LCD column driver outputs
V_{SS1}, V_{SS2}	ground
V_{DD1}, V_{DD2}	supply voltage
V_{LCD1}, V_{LCD2}	LCD supply voltage
T1	test 1 input
T2	test 2 output
T3	test 3 input/output
T4	test 4 input
SDIN	serial data input
SCLK	serial clock input
D/$\overline{C}$	data/command
$\overline{SCE}$	chip enable
OSC	oscillator
$\overline{RES}$	external reset input
dummy1, 2, 3, 4	not connected

Note

1. For further details, see Fig.18 and Table 7.

6.1 Pin functions

6.1.1 R0 TO R47 ROW DRIVER OUTPUTS

These pads output the row signals.

6.1.2 C0 TO C83 COLUMN DRIVER OUTPUTS

These pads output the column signals.

6.1.3 V_{SS1}, V_{SS2}: NEGATIVE POWER SUPPLY RAILS

Supply rails V_{SS1} and V_{SS2} must be connected together.

6.1.4 V_{DD1}, V_{DD2}: POSITIVE POWER SUPPLY RAILS

Supply rails V_{DD1} and V_{DD2} must be connected together.

6.1.5 V_{LCD1}, V_{LCD2}: LCD POWER SUPPLY

Positive power supply for the liquid crystal display. Supply rails V_{LCD1} and V_{LCD2} must be connected together.

6.1.6 T1, T2, T3 AND T4: TEST PADS

T1, T3 and T4 must be connected to V_{SS}, T2 is to be left open. Not accessible to user.

6.1.7 SDIN: SERIAL DATA LINE

Input for the data line.

6.1.8 SCLK: SERIAL CLOCK LINE

Input for the clock signal: 0.0 to 4.0 Mbits/s.

6.1.9 D/$\overline{C}$: MODE SELECT

Input to select either command/address or data input.

6.1.10 $\overline{SCE}$: CHIP ENABLE

The enable pin allows data to be clocked in. The signal is active LOW.

6.1.11 OSC: OSCILLATOR

When the on-chip oscillator is used, this input must be connected to V_{DD}. An external clock signal, if used, is connected to this input. If the oscillator and external clock are both inhibited by connecting the OSC pin to V_{SS}, the display is not clocked and may be left in a DC state. To avoid this, the chip should always be put into Power-down mode before stopping the clock.

6.1.12 $\overline{RES}$: RESET

This signal will reset the device and must be applied to properly initialize the chip. The signal is active LOW.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 FUNCTIONAL DESCRIPTION

7.1 Oscillator

The on-chip oscillator provides the clock signal for the display system. No external components are required and the OSC input must be connected to V_{DD}. An external clock signal, if used, is connected to this input.

7.2 Address Counter (AC)

The address counter assigns addresses to the display data RAM for writing. The X-address X_6 to X_0 and the Y-address Y_2 to Y_0 are set separately. After a write operation, the address counter is automatically incremented by 1, according to the V flag.

7.3 Display Data RAM (DDRAM)

The DDRAM is a 48 × 84 bit static RAM which stores the display data. The RAM is divided into six banks of 84 bytes (6 × 8 × 84 bits). During RAM access, data is transferred to the RAM through the serial interface. There is a direct correspondence between the X-address and the column output number.

7.4 Timing generator

The timing generator produces the various signals required to drive the internal circuits. Internal chip operation is not affected by operations on the data buses.

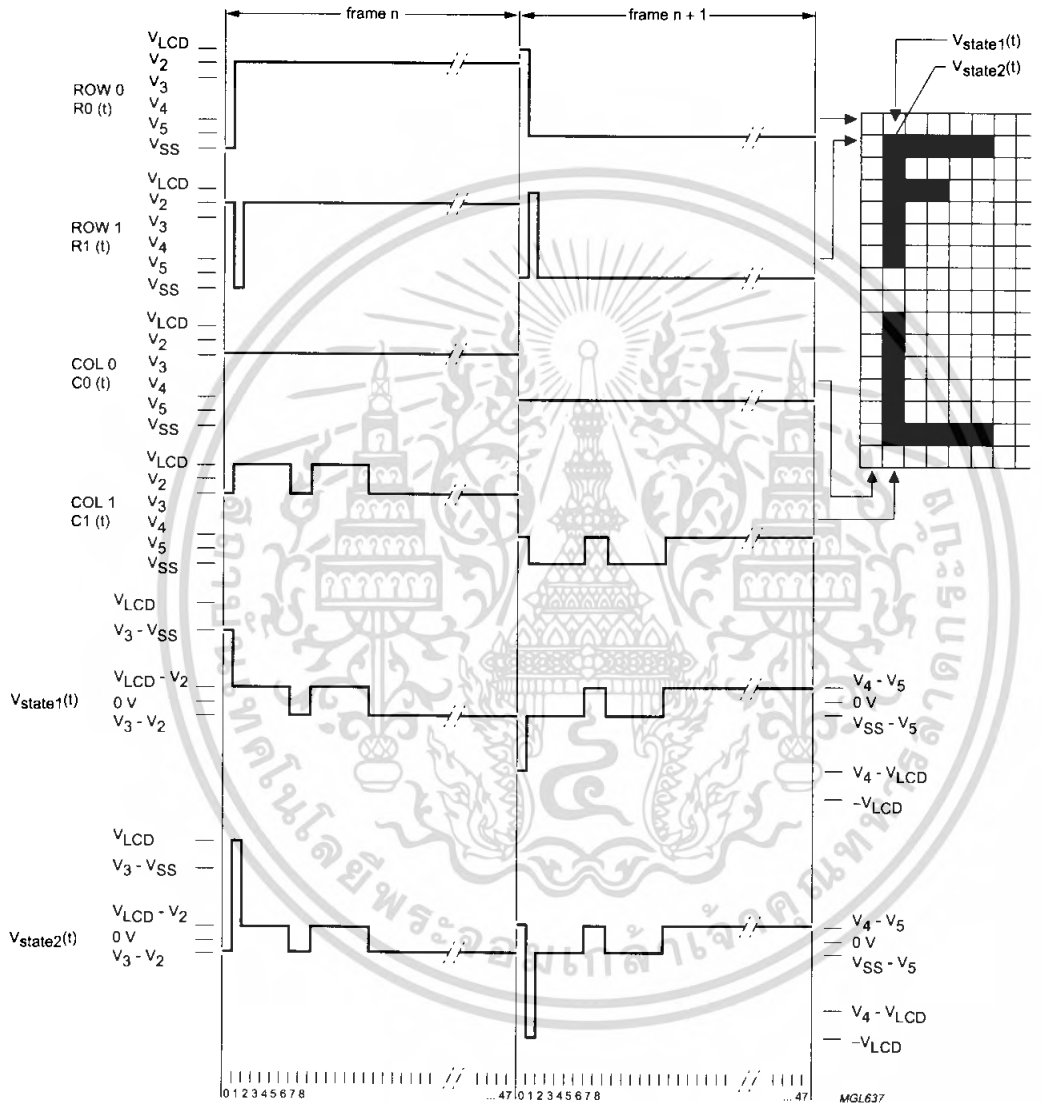
7.5 Display address counter

The display is generated by continuously shifting rows of RAM data to the dot matrix LCD through the column outputs. The display status (all dots on/off and normal/inverse video) is set by bits E and D in the 'display control' command.

7.6 LCD row and column drivers

The PCD8544 contains 48 row and 84 column drivers, which connect the appropriate LCD bias voltages in sequence to the display in accordance with the data to be displayed. Figure 2 shows typical waveforms. Unused outputs should be left unconnected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$V_{state1}(t) = C1(t) - R0(t)$.
$V_{state2}(t) = C1(t) - R1(t)$.

Fig.2 Typical LCD driver waveforms.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

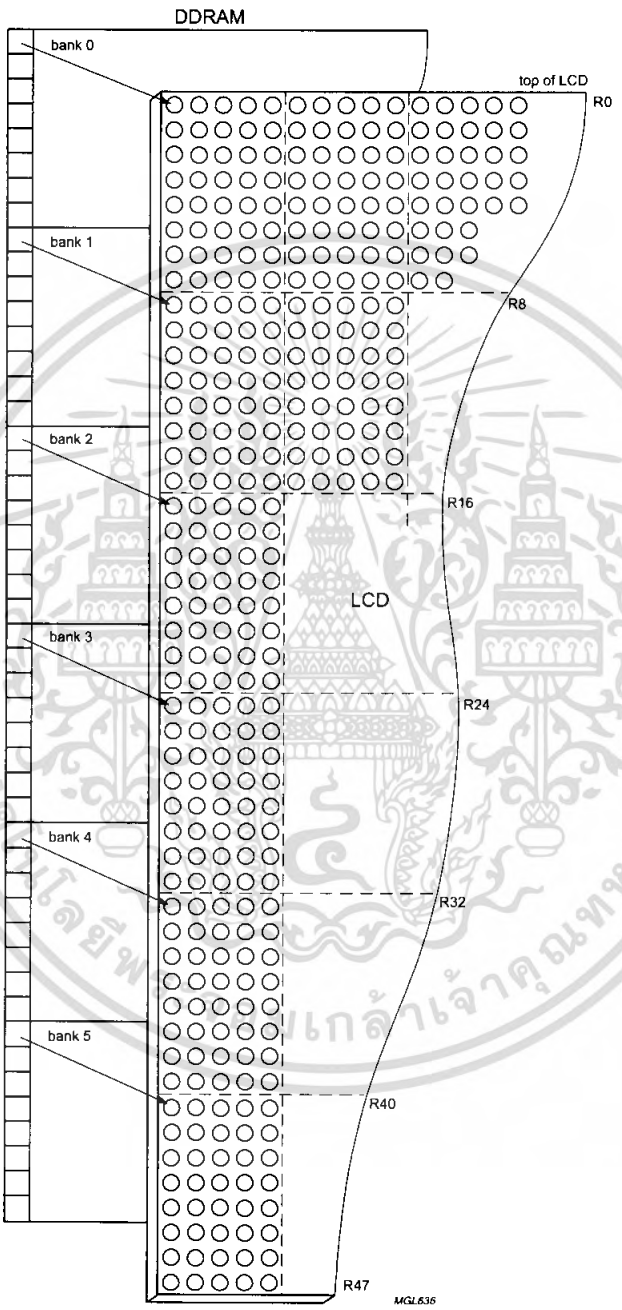


Fig.3 DDRAM to display mapping.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7 Addressing

Data is downloaded in bytes into the 48 by 84 bits RAM data display matrix of PCD8544, as indicated in Figs. 3, 4, 5 and 6. The columns are addressed by the address pointer. The address ranges are: X 0 to 83 (1010011), Y 0 to 5 (101). Addresses outside these ranges are not allowed. In the vertical addressing mode (V = 1), the Y address increments after each byte (see Fig.5). After the last Y address (Y = 5), Y wraps around to 0 and X increments to address the next column. In the horizontal addressing mode (V = 0), the X address increments after each byte (see Fig.6). After the last X address (X = 83), X wraps around to 0 and Y increments to address the next row. After the very last address (X = 83 and Y = 5), the address pointers wrap around to address (X = 0 and Y = 0).

7.7.1 DATA STRUCTURE

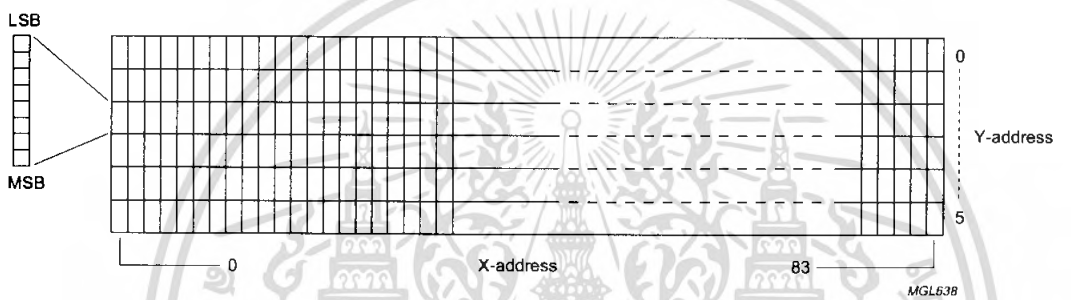


Fig.4 RAM format, addressing.

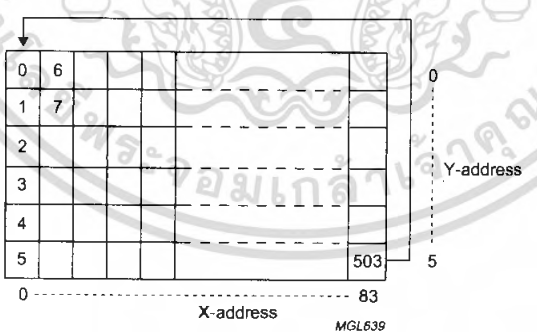


Fig.5 Sequence of writing data bytes into RAM with vertical addressing (V = 1).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

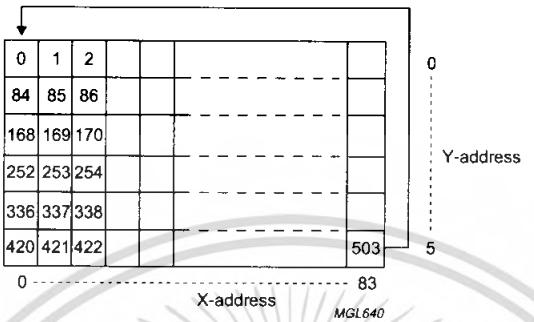
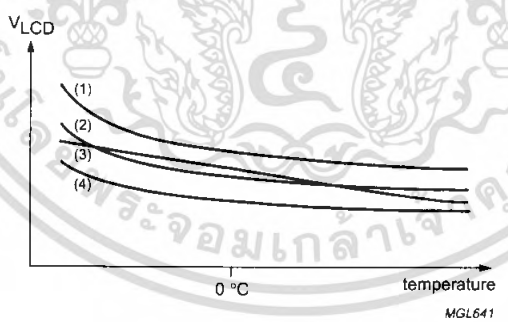


Fig.6 Sequence of writing data bytes into RAM with horizontal addressing (V = 0).

7.8 Temperature compensation

Due to the temperature dependency of the liquid crystals' viscosity, the LCD controlling voltage V_{LCD} must be increased at lower temperatures to maintain optimum contrast. Figure **7** shows V_{LCD} for high multiplex rates. In the PCD8544, the temperature coefficient of V_{LCD}, can be selected from four values (see Table 2) by setting bits TC_1 and TC_0.



(1) Upper limit.
(2) Typical curve.
(3) Temperature coefficient of IC.
(4) Lower limit.

Fig.7 V_{LCD} as function of liquid crystal temperature (typical values).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

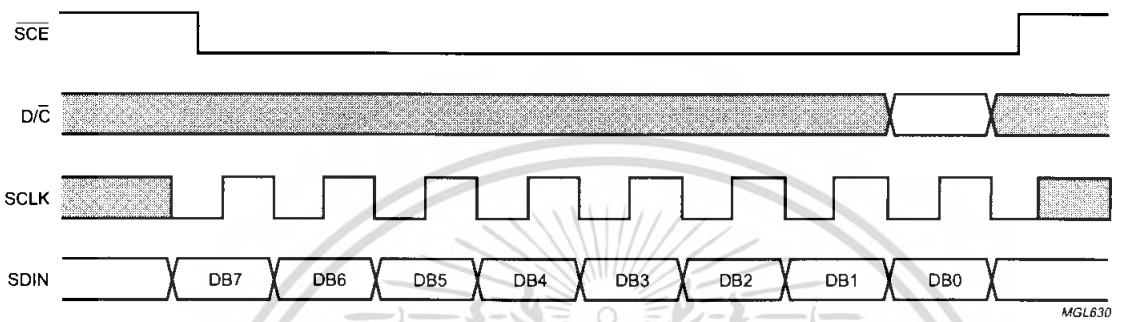


Fig.10 Serial bus protocol - transmission of one byte.

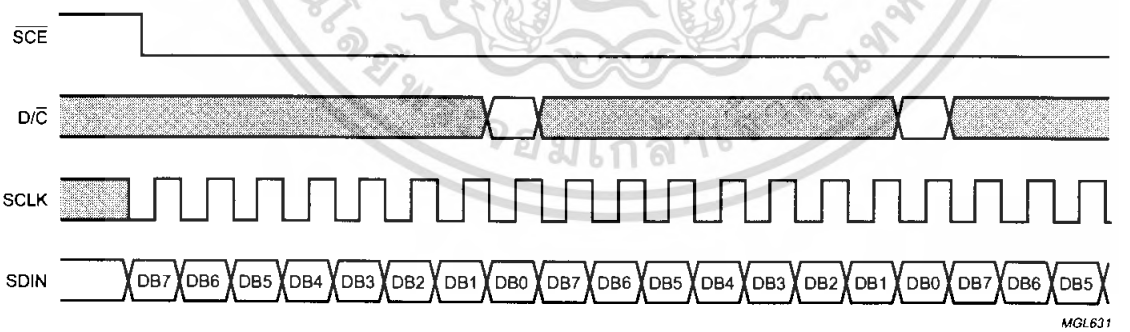


Fig.11 Serial bus protocol - transmission of several bytes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

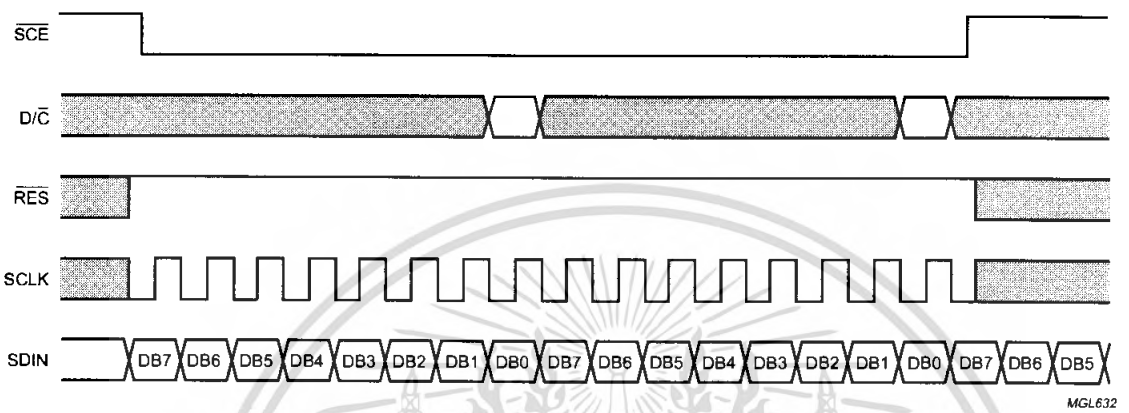


Fig.12 Serial bus reset function ($\overline{SCE}$).

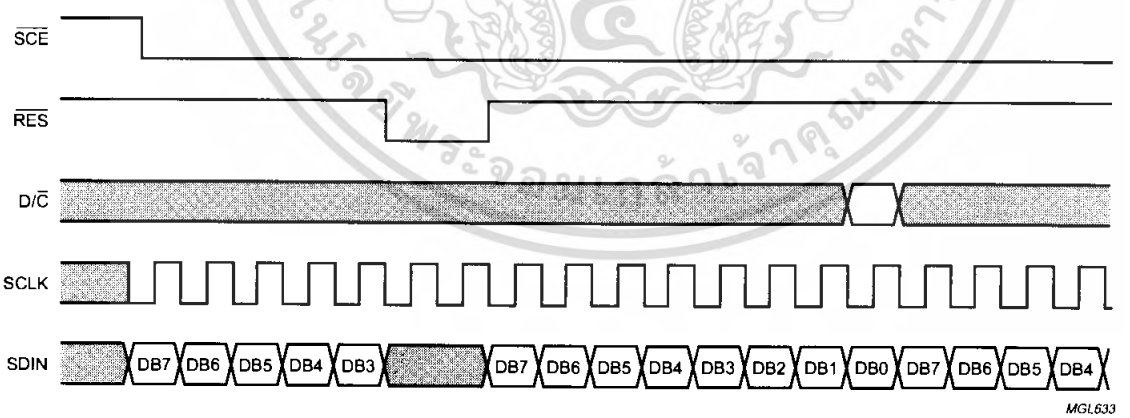


Fig.13 Serial bus reset function ($\overline{RES}$).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1 Initialization

Immediately following power-on, the contents of all internal registers and of the RAM are undefined. **A $\overline{\text{RES}}$ pulse must be applied**. Attention should be paid to the possibility that the **device may be damaged** if not properly reset.

All internal registers are reset by applying an external $\overline{\text{RES}}$ pulse (active LOW) at pad 31, within the specified time. However, the RAM contents are still undefined. The state after reset is described in Section 8.2.

The $\overline{\text{RES}}$ input must be $\leq 0.3V_{DD}$ when V_{DD} reaches V_{DDmin} (or higher) within a maximum time of 100 ms after V_{DD} goes HIGH (see Fig.16).

8.2 Reset function

After reset, the LCD driver has the following state:

- Power-down mode (bit PD = 1)
- Horizontal addressing (bit V = 0) normal instruction set (bit H = 0)
- Display blank (bit E = D = 0)
- Address counter X_6 to X_0 = 0; Y_2 to Y_0 = 0
- Temperature control mode (TC_1 TC_0 = 0)
- Bias system (BS_2 to BS_0 = 0)
- V_{LCD} is equal to 0, the HV generator is switched off (V_{OP6} to V_{OP0} = 0)
- After power-on, the RAM contents are undefined.

8.3 Function set

8.3.1 Bit PD

- All LCD outputs at V_{SS} (display off)
- Bias generator and V_{LCD} generator off, V_{LCD} can be disconnected
- Oscillator off (external clock possible)
- Serial bus, command, etc. function
- Before entering Power-down mode, the RAM needs to be filled with '0's to ensure the specified current consumption.

8.3.2 Bit V

When V = 0, the horizontal addressing is selected. The data is written into the DDRAM as shown in Fig.6. When V = 1, the vertical addressing is selected. The data is written into the DDRAM, as shown in Fig.5.

8.3.3 Bit H

When H = 0 the commands 'display control', 'set Y address' and 'set X address' can be performed; when H = 1, the others can be executed. The 'write data' and 'function set' commands can be executed in both cases.

8.4 Display control

8.4.1 Bits D and E

Bits D and E select the display mode (see Table 2).

8.5 Set Y address of RAM

Y_n defines the Y vector addressing of the display RAM.

Table 3 Y vector addressing

Y_2	Y_1	Y_0	BANK
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

8.6 Set X address of RAM

The X address points to the columns. The range of X is 0 to 83 (53H).

8.7 Temperature control

The temperature coefficient of V_{LCD} is selected by bits TC_1 and TC_0.

8.8 Bias value

The bias voltage levels are set in the ratio of R - R - nR - R - R, giving a 1/(n + 4) bias system. Different multiplex rates require different factors n (see Table 4). This is programmed by BS_2 to BS_0. For Mux 1 : 48, the optimum bias value n, resulting in 1/8 bias, is given by:

$$n = \sqrt{48} - 3 = 3.928 = 4 \quad (1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12.1 Serial interface

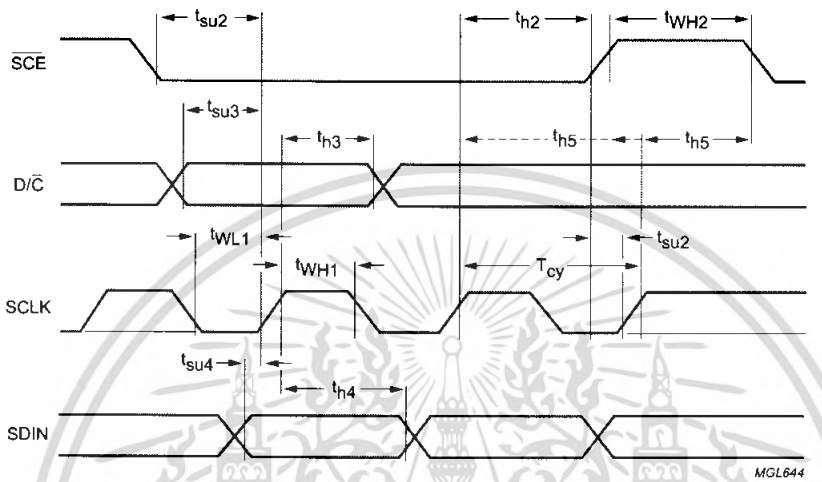


Fig.15 Serial interface timing.

12.2 Reset

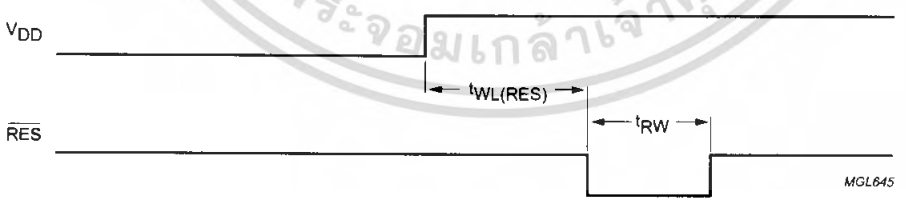


Fig.16 Reset timing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

October 2002

LM1117/LM1117I
800mA Low-Dropout Linear Regulator

General Description

The LM1117 is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as National Semiconductor's industry standard LM317.

The LM1117 is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V.

The LM1117 offers current limiting and thermal shutdown. Its circuit includes a zener trimmed bandgap reference to assure output voltage accuracy to within ±1%.

The LM1117 series is available in LLP, TO-263, SOT-223, TO-220, and TO-252 D-PAK packages. A minimum of 10µF tantalum capacitor is required at the output to improve the transient response and stability.

Features

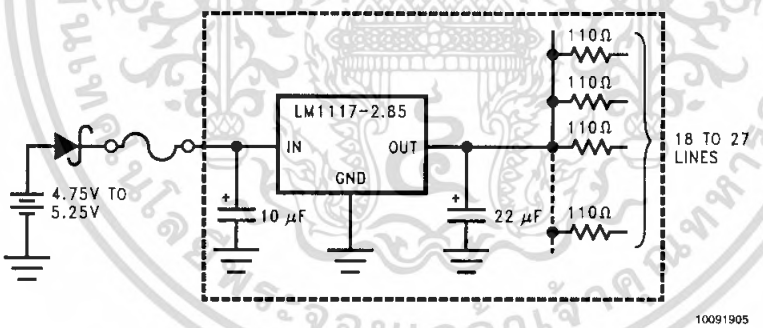
- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 and LLP Packages
- Current Limiting and Thermal Protection
- Output Current 800mA
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)
- Temperature Range
  - — LM1117 0°C to 125°C
  - — LM1117I −40°C to 125°C

Applications

- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

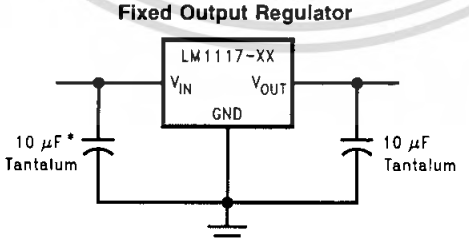
Typical Application

Active Terminator for SCSI-2 Bus



10091905

Fixed Output Regulator



* Required if the regulator is located far from the power supply filter.

10091928

© 2002 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้