

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องแปลงสัญญาณmidi

MIDI Conversion Box



โดย
นาย วิทยา สุขธรรมรักษ์
นาย สุชาติ แน่นพิมาย

รฟ.
๗๕๔๑
๒๕๔๙

เลขหมู่.....
เลขทะเบียน..... 72840
วัน,เดือน,ปี..... 23 ส.ย. 2550

b..... 11๗๖๓1๖๑
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องแปลงสัญญาณmidi
MIDI Conversion Box



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2549

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องแปลงสัญญาณmidi
(MIDI Conversion Box)

ผู้จัดทำ

- 1.นาย วิทยา สุขธรรมรักษ์
- 2.นาย สุชาติ แน่นพิมาย

.....อาจารย์ที่ปรึกษา
(รศ.จิระวัฒน์ ปานกลาง)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องแปลงสัญญาณมิตี

นาย วิทยา สุขธรรมรักษ์ 46010707

นาย สุชาติ แน่นพิมาย 46010845

รศ.จิระวัฒน์ ปานกลาง อาจารย์ที่ปรึกษา

ปีการศึกษา 2549

บทคัดย่อ

เครื่องแปลงสัญญาณมิตี คือ อุปกรณ์ที่ใช้รับสัญญาณอนาลอกจากกีตาร์ไฟฟ้า นำมาประมวลผล เพื่อแปลงออกมาเป็นชุดคำสั่งดิจิทัล ในรูปแบบของรหัสมิตี ซึ่งมีชุดคำสั่งมิตีเป็นการบอกค่าตัวแปรต่างๆของโน้ตดนตรีสากล เช่นค่าตำแหน่งและความดังตัวโน้ต

การทำงาน เริ่มจาก การรับสัญญาณอนาลอกจากกีตาร์ไฟฟ้าที่มีช่วงความถี่ประมาณ 329.6-1318.4 Hz แล้วเข้าสู่วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล แล้วเข้าสู่การประมวลผลเพื่อนำสัญญาณดิจิทัลมาวิเคราะห์ หาระดับเสียงของโน้ตดนตรีสากล แล้วมาแปลงในรูปแบบรหัสมิตี ซึ่งสามารถนำไปใช้สั่งงานกับเครื่องดนตรีที่สามารถรับคำสั่งมิตีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

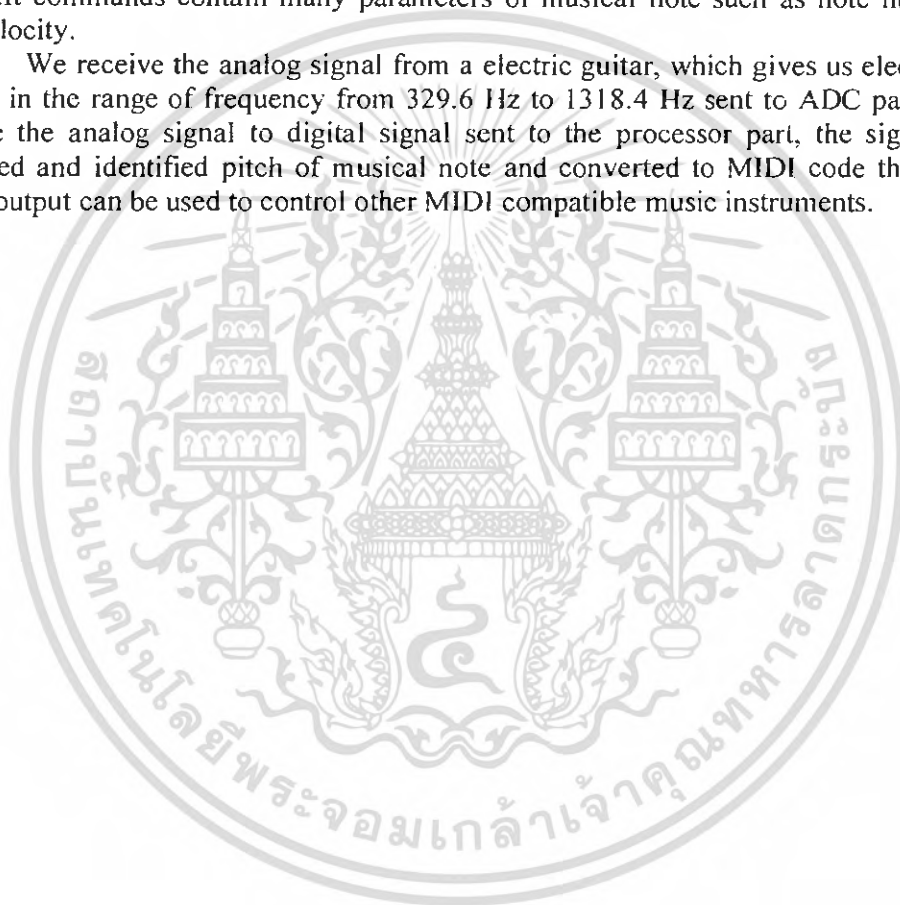
MIDI Conversion Box

Mr.Wittaya Sukthamruk 46010707
Mr.Suchart Nanpimai 46010845
Assoc.Prof.Jirawat Parnklang (Adviser)
Education Year 2006

Abstract

Midi conversion box is a device that receives analog signal from electric guitar. Process is signal processing and give us the digital command output in Midi Code .It commands contain many parameters of musical note such as note number and velocity.

We receive the analog signal from a electric guitar, which gives us electrical signal, in the range of frequency from 329.6 Hz to 1318.4 Hz sent to ADC part, we sample the analog signal to digital signal sent to the processor part, the signal is analyzed and identified pitch of musical note and converted to MIDI code that the MIDI output can be used to control other MIDI compatible music instruments.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการเรื่องแปลงสัญญาณมิดีนี้ จะไม่สามารถเดินหน้าถ้าขาด รศ.จิระวัฒน์ ปานกลาง ที่ให้คำปรึกษาและแนะนำมาโดยตลอด พร้อมทั้งการช่วยเหลือจากรุ่นพี่ เพื่อนๆ และพ่อแม่ที่เป็นบุคคลที่ให้การสนับสนุน ช่วยเหลือทุนทรัพย์ ให้กำลังใจ และเป็นห่วงตลอดมา

พวกเราขอขอบพระคุณ บุคคลที่ได้เอ่ยนามมาทั้งหมด รวมทั้งบุคคลที่ไม่ได้เอ่ยนาม มา ณ ที่นี้ด้วย



(นาย วิทยา สุขธรรมรักษ์)

(นาย สุชาติ แน่นพิมาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูปภาพ	V
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับมิดี้	2
2.1 ระบบมิดี้ (MIDI System)	2
2.2 รายละเอียดของระบบ (System Specifications)	2
2.3 การเชื่อมต่อกับอุปกรณ์ (Hardware Interface)	3
2.4 อุปกรณ์เชื่อมต่อและสายสัญญาณ (Connectors and Cables)	4
2.5 รูปแบบข้อความมิดี้ (MIDI Message Format)	5
2.6 มิดี้แชนเนล(MIDI Channel)	6
2.7 ข้อมูลเกี่ยวกับโน้ตในข้อความมิดี้	6
บทที่ 3 คุณสมบัติของเสียงดนตรี	14
3.1 ระดับเสียง(Pitch)	14
3.2 ความสัมพันธ์ระหว่างตัวโน้ตและความถี่	15
บทที่ 4 การแปลงฟาสต์ฟูรีเยอร์ (Fast Fourier Transform : FFT)	16
4.1 ฟาสต์ฟูรีเยอร์แบบฐานสอง (Radix 2 FFT)	16
4.2 ขั้นตอนวิธีการของการสับทางเวลา (Decimation in time : DIT)	21
บทที่ 5 การออกแบบและสร้าง	25
5.1 หลักการ	25
5.2 การออกแบบ	25
5.3 การออกแบบซอฟต์แวร์	26
5.4 เหตุที่ทำให้เกิดการอินเตอร์รัปต์	26
5.5 ฟังก์ชันการกลับบิท	29
5.6 ฟังก์ชัน Fast Fourier Transform	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7 วงจรขยายสัญญาณ(Amplifier)	33
บทที่ 6 การทดลอง	35
การทดลองที่ 1 เปรียบเทียบค่าความถี่และแอมพลิจูดระหว่างค่าที่วัดได้กับค่าที่คำนวณได้	35
การทดลองที่ 2 ค่าความถี่และแอมพลิจูดของสัญญาณที่ตัวรีเฟียที่เวลาต่างๆ	42
การทดลองที่ 3 ผลที่เกิดขึ้นเมื่อนำสัญญาณไปขยายจนเกิดการขลิบ	45
การทดลองที่ 4 ผลกระทบเมื่อเล่น 2 เสียงพร้อมกัน	51
การทดลองที่ 5 การหาค่าแอมพลิจูดสูงสุดในแต่ละ Grid	54
การทดลองที่ 6 การหาค่าเฉลี่ยแอมพลิจูดที่น้อยที่สุด ของทุก Grid	56
เพื่อใช้กำหนดค่า Threshold	
การทดลองที่ 7 การหาค่า Delay Time และเปรียบเทียบสัญญาณ Input กับ Output	59
บทที่ 7 สรุปผลการทดลอง	60
ภาคผนวก	
MIDI Code	a
กิตติกรรมประกาศ	q
บรรณานุกรม	r

สารบัญรูปลูกภาพ

เรื่อง	หน้า
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับมิดี	
รูปที่ 2.1 การส่งข้อความมิดีอย่างอนุกรมอะซิงโครนัส ในรูปแบบของ UART	3
รูปที่ 2.2 การเชื่อมต่อทางไฟฟ้าของมิดีที่พอร์ท IN, THRU, OUT	3
รูปที่ 2.3 ขอบของคลื่นพัลส์สี่เหลี่ยมถูกทำให้เกิด Rise time distortion	4
รูปที่ 2.4 รูปแบบทั่วไปของข้อความมิดี	5
บทที่ 4 การแปลงฟาสต์ฟูเรียร์	
รูปที่ 4.1 ผังการไหลแสดงวิธีการคำนวณ FFT	20
รูปที่ 4.2 หน่วยคำนวณผลคูณของขั้นตอนวิธีแบบสับทางเวลา	20
รูปที่ 4.3 ผังการไหลแสดงการคำนวณ DFT ขนาด 8 จุด รูปแสดง การแบ่งย่อยออกเป็นลำดับจนเหลือการคำนวณ DFT ขนาด 2 จุด	22
รูปที่ 4.4 การสลับตำแหน่งของลำดับ $x(n)$ ด้วยวิธีการกลับบิต	24
บทที่ 5 การออกแบบและสร้าง	
รูปที่ 5.1 Block Diagram การทำงานของระบบทั้งหมด	25
รูปที่ 5.2 โพลีชาร์ทของโปรแกรมหลัก	26
รูปที่ 5.3 โพลีชาร์ทของโปรแกรมการอินเตอร์รัปต์ไทมเมอร์	27
รูปที่ 5.4 โพลีชาร์ทของฟังก์ชัน Calculate	28
รูปที่ 5.5 โพลีชาร์ทการกลับบิต	30
รูปที่ 5.6 โพลีชาร์ทของ Fast Fourier Transform	32
รูปที่ 5.7 วงจรขยายที่ใช้ในการออกแบบ	33
รูปที่ 5.8 วงจรที่ใช้ในการออกแบบ	34
บทที่ 6 การทดลอง	
รูปที่ 6.1 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆ	39
รูปที่ 6.2 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ต E4 ในช่วงเวลาต่างๆ	42
รูปที่ 6.3 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆ เมื่อสัญญาณถูกขลิบ	47

- รูปที่ 6.4 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของคู่ตัวโน้ตต่างๆ
เมื่อเล่นพร้อมกัน 52
- รูปที่ 6.5 แสดงระยะห่างของเวลาในการบันทึกสัญญาณอนาล็อกกับสัญญาณมัลติ 59



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เรื่อง	หน้า
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับมิดี	
ตารางที่ 2.1 รูปแบบข้อความมิดี	6
ตารางที่ 2.2 การบอกระดับเสียงเป็นตัวเลขโน้ตมิดี	7
ตารางที่ 2.3 ตารางมิดีก่อนโทรลเลอร์ ในโหมด Control Change	8
ตารางที่ 2.4 ตารางมิดีก่อนโทรลเลอร์ ในโหมด Program Change	9
บทที่ 6 การทดลอง	
ตารางที่ 6.1 ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ต	36
ตารางที่ 6.2 ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ตเมื่อสัญญาณถูกขลิบ	45
ตารางที่ 6.3 ความถี่ของสัญญาณแต่ละตัวโน้ตเมื่อเล่น 2 เสียงพร้อมกัน	51
ตารางที่ 6.4 ค่าแอมพลิจูดเฉลี่ยสูงสุดที่สุดของ Grid ที่ตรงกับตัวโน้ต	54
ตารางที่ 6.5 ค่าแอมพลิจูดเฉลี่ยน้อยที่สุดของ Grid ที่ตรงกับตัวโน้ต	56
ตารางที่ 6.6 การหาค่า Threshold high	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในโลกของมิดีในทางดนตรี จะเป็น โลกที่แคบและอยู่กับผู้เล่นเครื่องดนตรีประเภทคีย์บอร์ด เป็นส่วนใหญ่ แต่ความจริงแล้วประโยชน์ของมิดีไม่น่าถูกจำกัดแค่อู่แคบนี้เท่านั้น เพราะมิดีนั้นสามารถนำไปใช้ควบคุมสิ่งต่างๆ เช่น งานด้านระบบเสียง ระบบแสง หรือ งานระบบควบคุมอื่นๆ เป็นต้น ดังนั้นโครงการนี้จึงเป็น โอกาสดีที่เราจะเปิดโลกของมิดีให้กว้างขึ้น

เนื่องจากมิดีนี้เป็นระบบที่ถูกออกแบบมาใช้กับระบบดิจิทัล เช่น คีย์บอร์ด จะเป็นไปได้หรือไม่ ที่นักดนตรีที่มีหัวใจเป็นมิดีแต่มีอุปกรณอื่นที่ไม่ใช่ประเภทเดียวกับคีย์บอร์ด เช่น กีตาร์ไฟฟ้า ซึ่งเป็นเครื่องดนตรีที่สร้างสัญญาณอนาล็อก เราจะทำอย่างไรกีตาร์ไฟฟ้าสามารถเข้าไปสู่โลกของมิดี

ดังนั้น จุดประสงค์ของโครงการนี้ คือเพื่อเชื่อมต่อกับนักดนตรีที่กำเนิดสัญญาณอนาล็อก เช่น กีตาร์ไฟฟ้า ไปสู่เครื่องดนตรีที่เป็นระบบดิจิทัล เช่นคีย์บอร์ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับมิดี

2.1 ระบบมิดี (MIDI System)

มิดี (MIDI : Musical Instrument Digital Interface) คือการเชื่อมต่อทางดิจิทัลของอุปกรณ์ทางดนตรี หรือเป็นการควบคุมระยะไกลด้วยดิจิทัลสำหรับระบบทางดนตรี ซึ่งอุปกรณ์ที่ควบคุมด้วยมิดีโดยทั่วไปจะใช้ไมโครคอนโทรลเลอร์ควบคุมผ่านทางช่องอินพุท และ เอาท์พุท มิดี เป็นที่นิยมและแพร่หลายมากในด้านการควบคุม นอกจากนี้ มิดียังสามารถประยุกต์ไปใช้งานประเภทอื่นได้ เช่นงานด้าน ระบบเสียง ระบบแสง แสงควบคุม ฯลฯ แม้ว่าคำสั่งมาตรฐานของมิดี จะเป็นคำสั่งทางด้านดนตรี แต่ก็สามารถนำไปปรับเปลี่ยนใช้กับงานประเภทอื่นได้

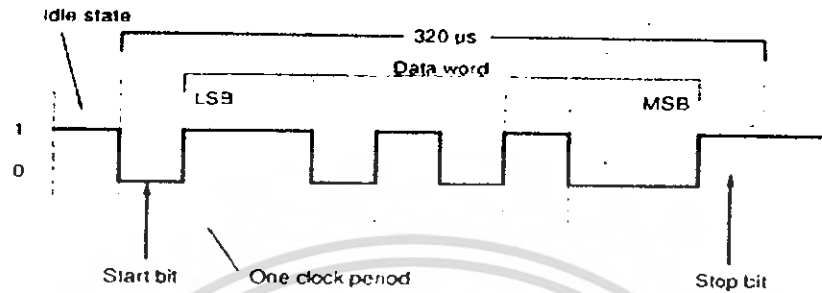
การที่มิดีใช้มาตรฐานการส่งแบบอนุกรมเนื่องจากเหตุผลทางด้านความประหยัด และความเป็นไปได้ในทางปฏิบัติ โดยมีจุดมุ่งหมายเพื่อให้อุปกรณ์ราคาถูกลง และสามารถรองรับผู้ใช้ได้หลากหลายเท่าที่จะทำได้ ในการส่งแบบขนานอาจมีประสิทธิภาพมากกว่าแต่เมื่อรวมถึงค่าผลรวมของการผลิตต่อชิ้น จะมีราคาที่สูงกว่าและขนาดใหญ่กว่าการใช้การเชื่อมต่อแบบอนุกรม เนื่องจากความง่ายและสะดวกต่อการติดตั้งอุปกรณ์มิดีนี้เอง จึงส่งผลให้มีการแพร่หลายอย่างรวดเร็วและใช้มาตรฐานที่นานาชาติยอมรับ

แตกต่างกับอุปกรณ์ในยุคก่อนซึ่งใช้สัญญาณอนาล็อก ในการส่งคำสั่งทุกชนิดก็จะอยู่ในรูปแบบเดียวกันบนสายเดียวกัน มิดีสามารถควบคุมอุปกรณ์มิดีที่เป็นหลายๆเสียงพร้อมกันได้ในเวลาจริงเสมือน โดยความเร็วในการส่งจะต้องมีการเหลื่อมล้ำของเวลาน้อยมาก จนหูของมนุษย์ไม่สามารถแยกแยะความแตกต่างนั้นได้ และยังสามารถกำหนดจุดหมายที่ต้องการจะให้แสดงผลโดยการเลือกช่องสัญญาณ

2.2 รายละเอียดของระบบ (System Specifications)

มาตรฐานของมิดีกำหนดว่าเป็นการส่งสัญญาณผ่านการเชื่อมต่อแบบอนุกรมทางเดียว (Unidirectional Serial Interface) โดยทำงานที่ความเร็ว 31.25 กิโลบิตต่อวินาที โดยผิดพลาด $\pm 1\%$ ด้วยความเร็วขนาดนี้ ถูกกำหนดเมื่อเทียบกับความเร็วของสัญญาณนาฬิกาของไมโครโปรเซสเซอร์ซึ่งโดยทั่วไปจะมีความเร็วของสัญญาณนาฬิกาอย่างน้อย 1 เมกะเฮิรตซ์ หรือ 2 เมกะเฮิรตซ์ จะมีความเร็วมากกว่าหลายเท่า โดยความเร็วของการส่งมิดีช้าเพียงพอที่จะไม่เกิดการล้นหรือเกินของข้อมูลในสายสัญญาณและอุปกรณ์ทางการเชื่อมต่อ แต่ก็ยังเร็วพอที่จะสามารถส่งข้อมูลทางดนตรี จากเครื่องหนึ่งไปยังเครื่องอื่นๆ โดยๆ ไม่สามารถสังเกตความแตกต่างของเวลาที่หน่วงได้ ข้อมูลจะถูกส่งไปได้ทางเดียวจากผู้ส่งไปยังผู้รับโดยไม่มีการย้อนกลับ เว้นแต่ว่าจะมีการเชื่อมต่อกลับจากภายนอก

ข้อความควบคุม (Control Messages) จะถูกส่งเป็นกลุ่มของไบต์ โดยเริ่มจากบิตเริ่มต้น 1 บิต และบิตปิดท้ายข้อมูลด้วยบิตสิ้นสุด 1 บิต เพื่อที่เป็นการเทียบเวลาให้ตรงกันของผู้รับสำหรับข้อมูล ที่ส่งในรูปแบบอะซิงโครนัส ดังรูป 2.1

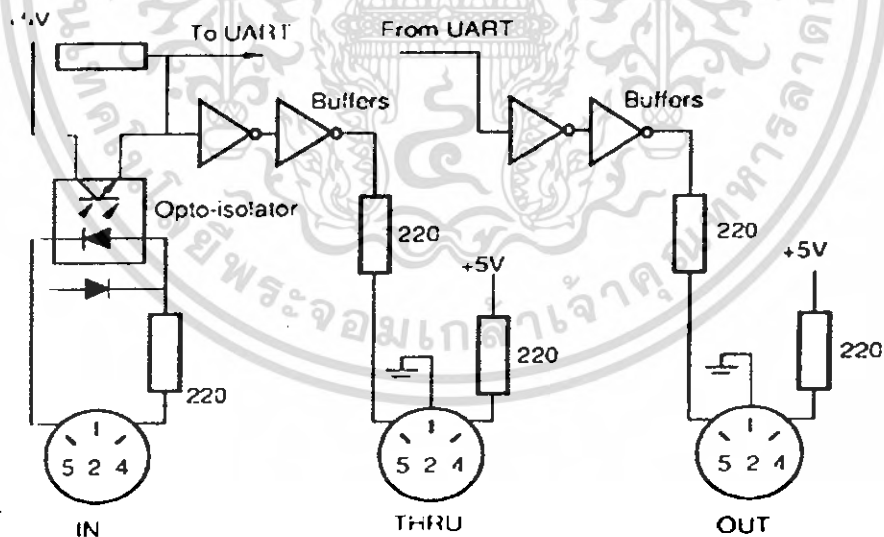


รูปที่ 2.1 การส่งข้อความมิตอย่างอนุกรมอะซิงโครนัส ในรูปแบบของ UART

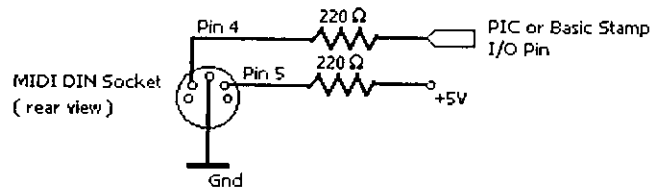
จากรูป 2.1 เราจะพบว่า มีการเพิ่มบิตเริ่มต้นและบิตสิ้นสุดรวมกับบิตข้อมูล 8 บิต เป็นข้อมูล 10 บิต ที่จะต้องส่งโดยรวม เวลาส่งจะได้ประมาณ 320 us มาตรฐานของข้อความมิตโดยทั่วไปจะประกอบด้วย 1 ไบต์ 2 ไบต์ หรือ 3 ไบต์

2.3 การเชื่อมต่อกับอุปกรณ์ (Hardware Interface)

เป็นการเชื่อมต่อกับอุปกรณ์ที่ชัดเจน โดยควรจะใช้งานได้ร่วมกับอุปกรณ์ทางมิตทุกชนิด และจะต้องมีระบบทางไฟฟ้าที่เข้ากันได้ในระบบ ดังรูป 2.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การเชื่อมต่อทางไฟฟ้าของมิดีที่พอร์ท IN,THRU,OUT

อุปกรณ์มิดีส่วนใหญ่จะมี 3 ช่อง คือ IN,THRU และ OUT ที่ช่อง OUT จะเป็นตัวส่งข้อมูลที่ตัวอุปกรณ์นั้นเป็นตัวสร้างขึ้นมาเอง ผ่าน UART ส่วนที่ช่อง THRU จะส่งต่อข้อมูลจากช่อง IN ไปยังอุปกรณ์อื่นๆ ได้ อุปกรณ์มิดีหลายๆจะไม่มีช่อง THRU แต่สามารถต่ออุปกรณ์ MIDI THRU Boxes เพื่อเพิ่มช่อง THRU ได้

ทางช่อง MIDI IN ต้องต่อผ่านทาง Opto-isolator เพื่อไปสู่ส่วนระบบไมโครคอนโทรลเลอร์ของอุปกรณ์ ทั้งนี้เพื่อให้แน่ใจว่าจะไม่มีการเชื่อมต่อทางไฟฟ้าระหว่างอุปกรณ์ และช่วยลดผลของการเกิดปัญหา ด้านเกิดอุปกรณ์ตัวหนึ่งมีข้อผิดพลาดทางไฟฟ้า ตัว Opto-isolator เป็นอุปกรณ์ที่ประกอบด้วย LED เพื่อใช้ควบคุมการเปิดปิดของสวิทช์ โดยขึ้นอยู่กับสัปดาห์ที่ตกคร่อมขั้วทั้งสอง เพื่อให้แสงกับ Photo-transistor ที่จะขึ้นกับสถานะของ LED ด้วยเหตุนี้การส่งข้อมูลทางแสงจึงดีกว่าการส่งข้อมูลทางไฟฟ้า ในรายละเอียดของมิดี ตัว Opto-isolator ใช้เวลาในช่วงขาขึ้นไม่เกิน 2 us ซึ่งเวลาในช่วงขาขึ้นใช้เวลานานอาจทำให้เกิดรูปผิดเพี้ยนที่ขอบของบิทข้อมูลได้ ดังรูป 2.3 ซึ่งจะเป็เหมือนกันสำหรับเวลาในช่วงขาลง



รูปที่ 2.3 ขอบของคลื่นพัลส์ที่เหลี่ยมถูกทำให้เกิด Rise time distortion

การบิดเบือนของรูปคลื่นในช่วงขาขึ้นนั้นส่งผลให้เกิดความไม่เสถียรทางเวลาของข้อมูล เมื่อกำลังสลับค่าที่ขอบจะไม่สามารถแยกแยะได้ว่าเป็น 0 หรือ เป็น 1 และถ้าเวลาในขอบขาขึ้นช้ามากเกินไป ค่าของข้อมูลจะผิดพลาดโดยผลลัพธ์ของอุปกรณ์จะยังไม่ขึ้นเต็มค่าก่อนที่ข้อมูลบิทถัดมาจะตามมา ถ้าอุปกรณ์มิดีหลายๆเครื่องต่อพ่วงกันแบบอนุกรม (เข้าทาง IN ออกช่อง THRU) ข้อมูลก็จะผ่าน Opto-isolator หลายตัวจะทำให้เกิดการบิดเบือนไปมาก หรือไม่ก็ทำให้เกิดการรับข้อมูลที่ผิดพลาดของตัวรับตัวสุดท้าย ดังนั้นจะขึ้นอยู่กับคุณภาพของ Opto-isolator และความสูญเสียของข้อมูลระหว่างทาง เมื่อคำนึงถึงความเป็นไปได้ของความหน่วงเวลาระหว่างช่อง IN และช่อง THRU จะต้องอยู่ในหน่วยของ us จะไม่มีผลต่อการได้ยินของมนุษย์

2.4 อุปกรณ์เชื่อมต่อและสายสัญญาณ (Connectors and Cables)

หัวต่อสำหรับมิดีเหมือนกับ DIN plugs 5 ขาที่ใช้สำหรับระบบไฮไฟบางระบบ และแม้ว่าสามารถใช้สายของไฮไฟได้ แต่สายที่มีคุณภาพดีกว่าก็ยังเป็นสิ่งจำเป็น ในมาตรฐานก็อนุญาตให้ใช้กับหัว XLR-type ได้ แม้ว่าจะใช้ในทางปฏิบัติได้ยาก แต่อุปกรณ์ส่วนใหญ่ ก็ใช้เพียงแต่ 3 พิน จาก 5 พิน ของ DIN plug

สายสัญญาณควรจะเป็นสาย Shielded twisted pair และชิลด์จะต่อกับ พินที่ 2 ของหัวทั้ง 2 ด้าน แม้ว่าช่อง IN จะไม่ได้ต่อลงกราวด์ที่พินที่ 2 เพื่อที่จะป้องกัน Ground Loop และสามารถใช้งานได้ทั้ง 2 ด้าน แนะนำว่าสายไม่เกินยาวเกิน 15 เมตร ถ้ายาวเกินจะเกิดข้อผิดพลาดขึ้นได้

2.5 รูปแบบข้อความมิดี (MIDI Message Format)

ข้อความมิดีจะมีรูปแบบไบท์ข้อความพื้นฐานอยู่ 2 รูปแบบ คือ ไบท์สถานะ (Status Byte) และ ไบท์ข้อมูล โดยไบท์สถานะนั้นจะขึ้นต้นบิตแรกเป็นไบนารี 1 ซึ่งจะตรงข้ามกับไบท์ข้อมูล ที่จะขึ้นต้นบิตแรกเป็นไบนารี 0 ดังนั้นจึงเหลือ 7 บิตที่สามารถใช้งานบอกค่าต่างๆได้ 128 ค่า

ข้อความมิดี แต่ละข้อความจะมีไบท์ข้อความอยู่ 1-3 ไบท์ จากรูป 2.4 ในไบท์แรกของข้อความมิดี จะเป็นไบท์สถานะ ซึ่งบรรจุข้อมูลเกี่ยวกับมิดีแชนแนล (MIDI Channel) ซึ่งจะระบุโดย 4 บิตแรก (บิต 3 - บิต 0) ซึ่งจะมีทั้งหมด 16 แชนแนล ส่วนในบิต 6 - บิต 4 จะทำหน้าที่บอกรูปแบบไบท์ของข้อความมิดีชุดนั้น



รูปที่ 2.4 รูปแบบทั่วไปของข้อความมิดี

- โดย บิต 's s s' ใช้บอกรูปแบบของข้อความ
- บิต 'n n n n' ใช้บอกเลขมิดีแชนแนล (MIDI Channel Nummer)
- บิต 'x x x x x x x' ใช้บอกระดับเสียงของตัว โน้ต
- บิต 'y y y y y y y' ใช้บอกค่าความดังของตัว โน้ต

ข้อมิดีมาตรฐานสามารถมีได้ถึง 3 ไบท์ ต่อ 1 ข้อความ แต่ก็เชื่อว่าทุกข้อความจะต้องยาว 3 ไบท์เสมอไป โดยรูปแบบของข้อความมิดีสามารถดูได้จากตาราง 2.1

ตารางที่ 2.1 รูปแบบข้อความมิดี

Message	Status	Data 1	Data 2
Note off	&8n	Note number	Velocity
Note on	&9n	Note number	Velocity
Polyphonic aftertouch	&An	Note number	Pressure
Control change	&Bn	Controller number	Data
Program change	&Cn	Program number	-
Channel aftertouch	&Dn	Pressure	-
Pith wheel	&En	LSbyte	MSbyte
System exclusive			
System exclusive start	&F0	Manufacturer ID	Data(Data)(Data)
End of SysEx	&F7	-	-
System common			
Quarter frame	&F1	Data	-
Song Pointer	&F2	LSbyte	MSbyte
Song select	&F3	Song number	-
Tune request	&F6	-	-
System realtime			
Timing Clock	&F8	-	-
Start	&FA	-	-
Continue	&FB	-	-
Stop	&FC	-	-
Active sensing	&FE	-	-
reset	&FF	-	-

2.6 มิดีเซนแนล(MIDI Channel)

มิดีเซนแนล คือ การประกาศรหัสในการรับส่งข้อมูลมิดี เพื่อให้เกิดการตอบสนอง และการละเว้นข้อการตอบสนอง ข้อมูลของอุปกรณ์การรับ จากอุปกรณ์การส่ง เพื่อให้เกิดการตอบสนองข้อมูลเฉพาะชั้นที่ระบุไว้ ใช้เมื่อต่อกับอุปกรณ์มิดีหลายตัว

2.7 ข้อมูลเกี่ยวกับโน้ตในข้อความมิดี

ในข้อความมิดี มีการบอกค่าต่างของตัวโน้ต โดยจะมีการระบุค่าไว้ที่ไบท์ข้อมูลที่ 1 และ 2 ยกเว้นค่าความยาวเสียงจะไม่มีการระบุโดยตรงแต่จะใช้เป็นการบอกเปิดปิดโน้ต (Note on/Note off) แทนโดยจะบอกค่าในบิต 6 – บิต 4 ของไบท์สถานะ โดยการบอกค่าต่างๆของโน้ตมีรายละเอียดดังนี้

1.เปิดโน้ต และ ปิดโน้ต (Note on and Note off) ใช้ในการบอกค่าความยาวของตัวโน้ต คือ โน้ตจะเริ่มเล่นเมื่อได้รับข้อความเปิดโน้ต และจะเล่นไปเรื่อยๆจนได้รับข้อความปิดโน้ตจึงทำการหยุดเล่น โดยข้อมูลการเปิดปิดโน้ต จะถูกระบุใน บิต 3 - บิต 0 ของไบท์สถานะในข้อความมิดี มีค่าดังในตารางที่ 2.1

2. ระดับเสียงของตัวโน้ต (Pitch) จะมีการระบุค่าไว้ทั้งหมด 128 ค่า ตั้งแต่ระดับเสียง C -2 ถึง G8 โดยทำการระบุค่าไว้ใน บิต 6 – บิต 0 ของไบท์ข้อมูลที่ 1 โดยมีค่าดังตารางที่ 2.2

3. ค่าความดังของตัวโน้ต (Velocity) จะสามารถระบุค่าได้ทั้งหมด 128 ค่า ความแตกต่าง โดยทำการระบุค่าไว้ใน บิต 6 – บิต 0 ของไบท์ข้อมูลที่ 2 โดยค่าความดังจะถูกส่งมาพร้อมกับชุดข้อมูล เปดโน้ต และจะคงค่าความดังนั้นๆ ไว้จนกว่าจะได้รับข้อมูลปิดโน้ต ส่วนค่าความดังที่มากับชุดข้อมูล ปิดโน้ตนั้นจะมีใช้ในวงแคบ เพราะมีอุปกรณ์บางชิ้นที่ตอบสนองข้อมูลนี้

4. ค่าแรงเบ็ดคีย์หลังเปิดโน้ต (Polyphonic aftertouch) เป็นข้อมูลที่มีใช้ในอุปกรณ์ที่มี ลักษณะเป็นคีย์บอร์ด โดยจะเป็นข้อมูลที่บอกถึงแรงกดคีย์ที่เปลี่ยนไปหลังจากมีการเปิดโน้ตไปแล้ว โดยจะมีการส่งข้อมูลใน บิต 3 – บิต 0 ของไบท์สถานะในข้อความมิติ โดยมีค่าตามตาราง 2.1

ตารางที่ 2.2 การบอกระดับเสียงเป็นตัวเลขโน้ตมิติ

ระดับเสียง (Pitch)	ตัวเลขโน้ตมิติ (MIDI note number)
C-2	0
C-1	12
C 0	24
C 1	36
C 2	48
C 3	60
C 4	72
C 5	84
C 6	96
C 7	108
C 8	120
G 8	127

ตารางที่ 2.3 ตารางมีดีคอนโทรลเลอร์ ในโหมด Control Change

MIDI Controller Numbers

The table below presents a summary of the MIDI Standard Controller codes in decimal and hexadecimal (h) form.

Decimal	Hex	Controller Name
0	00h	Bank Select (Controller # 32 more commonly used)
1	01h	Modulation Wheel
2	02h	Breath Controller
3	03h	Undefined
4	04h	Foot Controller
5	05h	Portamento Time
6	06h	Data Entry MSB
7	07h	Main Volume
8	08h	Balance
9	09h	Undefined
10	0Ah	Pan
11	0Bh	0Ch
12	0Ch	Effect Control 1
13	0Dh	Effect Control 2
14-15	0E-0Fh	Undefined
16-19	10-13h	General Purpose Controllers (Nos. 1-4)
20-31	14-1Fh	Undefined
32-63	20-3Fh	LSB for Controllers 0-31 (rarely implemented)
64	40h	Damper Pedal (Sustain) [Data Byte of 0-63=Off, 64-127=On]
65	41h	Portamento
66	42h	Sostenuto
67	43h	Soft Pedal
68	44h	Legato Footswitch
69	45h	Hold 2
70	46h	Sound Controller 1 (default: Sound Variation)
71	47h	Sound Controller 2 (default: Timbre/Harmonic Content)
72	48h	Sound Controller 3 (default: Release Time)
73	49h	Sound Controller 4 (default: Attack Time)
74	4Ah	Sound Controller 5 (default: Brightness)
75-79	4B-4Fh	Sound Controller 6-10 (no defaults)
80-83	50-53h	General Purpose Controllers (Nos. 5-8)
84	54h	Portamento Control
85-90	55-5Ah	Undefined
91	5Bh	Effects 1 Depth (previously External Effects Depth)
92	5Ch	Effects 2 Depth (previously Tremolo Depth)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

93	5Dh	Effects 3 Depth (previously Chorus Depth)
94	5Eh	Effects 4 Depth (previously Detune Depth)
95	5Fh	Effects 5 Depth (previously Phaser Depth)
96	60h	Data Increment
97	61h	Data Decrement
98	62h	Non-Registered Parameter Number LSB
99	63h	Non-Registered Parameter Number LSB
100	64h	Registered Parameter Number LSB
101	65h	Registered Parameter Number MSB
102-120	66-78h	Undefined

Channel Mode Messages

121	79h	Reset All Controllers
122	7Ah	Local Control
123	7Bh	All Notes Off
124	7Ch	Omn Off
125	7Dh	Omn On
126	7Eh	Mono On (Poly Off)
127	7Fh	Poly On (Mono Off)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ตารางมิดีคอนโทรลเลอร์ ในโหมด Program Change
Piano (0-7)

Dec	Hex	Program Change
0	0x00	acoustic-grand-piano
1	0x01	bright-acoustic-piano
2	0x02	electric-grand-piano
3	0x03	honky-tonk-piano
4	0x04	electric-piano-1
5	0x05	electric-piano-2
6	0x06	harpsichord
7	0x07	clavi

Chromatic Percussion (8-15)

Dec	Hex	Program Change
8	0x08	celesta
9	0x09	glockenspiel
10	0x0A	music-box
11	0x0B	vibraphone
12	0x0C	marimba
13	0x0D	xylophone
14	0x0E	tubular-bells
15	0x0F	dulcimer

Organ (16-23)

Dec	Hex	Program Change
16	0x10	drawbar-organ
17	0x11	percussive-organ
18	0x12	rock-organ
19	0x13	church-organ
20	0x14	reed-organ
21	0x15	accordion
22	0x16	harmonica
23	0x17	tango-accordion

Guitar (24-31)

Dec	Hex	Program Change
24	0x18	acoustic-guitar-nylon
25	0x19	acoustic-guitar-steel
26	0x1A	electric-guitar-jazz
27	0x1B	electric-guitar-clean
28	0x1C	electric-guitar-muted
29	0x1D	overdriven-guitar
30	0x1E	distortion-guitar
31	0x1F	guitar-harmonics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bass (32-39)

Dec	Hex	Program Change
32	0x20	acoustic-bass
33	0x21	electric-bass-finger
34	0x22	electric-bass-pick
35	0x23	fretless-bass
36	0x24	slap-bass-1
37	0x25	slap-bass-2
38	0x26	synth-bass-1
39	0x27	synth-bass-2

Solo strings (40-47)

Dec	Hex	Program Change
40	0x28	violin
41	0x29	viola
42	0x2A	cello
43	0x2B	contrabass
44	0x2C	tremolo-strings
45	0x2D	pizzicato-strings
46	0x2E	orchestral-strings
47	0x2F	timpani

Ensemble (48-55)

Dec	Hex	Program Change
48	0x30	string-ensemble-1
49	0x31	string-ensemble-2
50	0x32	synthstrings-1
51	0x33	synthstrings-2
52	0x34	choir-aahs
53	0x35	voice-oohs
54	0x36	synth-voice
55	0x37	orchestra-hit

Brass (56-63)

Dec	Hex	Program Change
56	0x38	trumpet
57	0x39	trombone
58	0x3A	tuba
59	0x3B	muted-trumpet
60	0x3C	french-horn
61	0x3D	brass-section
62	0x3E	synthbrass-1
63	0x3F	synthbrass-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reed (64-71)

Dec	Hex	Program Change
64	0x40	soprano-sax
65	0x41	alto-sax
66	0x42	tenor-sax
67	0x43	baritone-sax
68	0x44	oboe
69	0x45	english-horn
70	0x46	bassoon
71	0x47	clarinet

Pipe (72-79)

Dec	Hex	Program Change
72	0x48	piccolo
73	0x49	flute
74	0x4A	recorder
75	0x4B	pan-flute
76	0x4C	blown-bottle
77	0x4D	skakuhachi
78	0x4E	whistle
79	0x4F	ocarina

Synth Lead (80-87)

Dec	Hex	Program Change
80	0x50	lead-1-square
81	0x51	lead-2-sawtooth
82	0x52	lead-3-calliope
83	0x53	lead-4-chiff
84	0x54	lead-5-charang
85	0x55	lead-6-voice
86	0x56	lead-7-fifths
87	0x57	lead-8-bass

Synth Pad (88-95)

Dec	Hex	Program Change
88	0x58	pad-1-new-age
89	0x59	pad-2-warm
90	0x5A	pad-3-polysynth
91	0x5B	pad-4-choir
92	0x5C	pad-5-bowed
93	0x5D	pad-6-metallic
94	0x5E	pad-7-halo
95	0x5F	pad-8-sweep

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Synth Effects (96-103)

Dec	Hex	Program Change
96	0x60	fx-1-rain
97	0x61	fx-2-soundtrack
98	0x62	fx-3-crystal
99	0x63	fx-4-atmosphere
100	0x64	fx-5-brightness
101	0x65	fx-6-goblins
102	0x66	fx-7-echoes
103	0x67	fx-8-sci-fi

Ethnic (104-111)

Dec	Hex	Program Change
104	0x68	sitar
105	0x69	banjo
106	0x6A	shamisen
107	0x6B	koto
108	0x6C	kalimba
109	0x6D	bagpipe
110	0x6E	fiddle
111	0x6F	shantai

Percussive (112-119)

Dec	Hex	Program Change
112	0x70	tinkle-bell
113	0x71	agogo
114	0x72	steel-drums
115	0x73	woodblock
116	0x74	taiko-drum
117	0x75	melodic-tom
118	0x76	synth-drum
119	0x77	reverse-cymbal

Sound Effects (120-127)

Dec	Hex	Program Change
120	0x78	guitar-fret-noise
121	0x79	breath-noise
122	0x7A	seashore
123	0x7B	bird-tweet
124	0x7C	telephone-ring
125	0x7D	helicopter
126	0x7E	applause
127	0x7F	gunshot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

คุณสมบัติของเสียงดนตรี

คุณสมบัติเฉพาะของตัวโน้ตในแต่ละตัวนั้น มีอยู่ 2 คุณสมบัติด้วยกัน คือ ระดับเสียง (Pitch) และระยะเวลา (Duration) ดังนั้นการวิเคราะห์อย่างเหมาะสมจะทำให้สามารถระบุระดับเสียงและระยะเวลาของตัวโน้ตในแต่ละตัวได้

3.1 ระดับเสียง(Pitch)

ระดับเสียง ก็คือ ลักษณะของตัวโน้ตที่สามารถจัดเรียงเป็นระดับของเสียงดนตรีได้ ซึ่งจำกัดที่องค์กร America National Standards ให้ไว้ก็คือ “ระดับเสียง ก็คือ คุณสมบัติของการได้ยินในส่วนของเสียงซึ่งสามารถจัดแยกออกเป็นระดับจากต่ำไปสูง”

แนวคิดในการทำความเข้าใจกับเสียง ก็คือ อันดับแรกจะแนะนำให้รู้จักกับเสียงดนตรี ซึ่งความสามารถในการได้ยินของคนเรานั้นจะสามารถแยกระดับเสียงที่ต่างกันได้ค่อนข้างที่จะสูง (ประมาณ 1,500 ระดับเสียง) การได้ยินของคนธรรมดา นั้นจะสามารถรับรู้ความแตกต่างระหว่างเสียงที่มีความถี่ 440 Hz และ 441 Hz ซึ่งเสียงที่มีความถี่ 440 Hz นักดนตรีจะรู้จักกันดี คือ เสียง A4 หรือ ลา และตัวโน้ตถัดไปของเปียโน คือ A#4 มีความถี่ 466.2 Hz ทำให้สามารถสรุปได้อย่างคร่าวๆว่า การได้ยินของคนสามารถจะรับรู้ความแตกต่างของตัวโน้ต 2 ตัว ที่มีระดับเสียงต่างกัน 25 ระดับเสียงได้ ซึ่งคนส่วนใหญ่จะมีความสามารถนี้

ในทางวิทยาศาสตร์ แนวความคิดเกี่ยวกับระดับเสียง คือ ความสัมพันธ์เกี่ยวกับคุณสมบัติของเสียงทางกายภาพที่เรียกว่า “ความถี่” ซึ่งตามทฤษฎีนั้น คนเราสามารถได้ยินเสียงในช่วงความถี่ 20 Hz ถึง 20,000 Hz แต่ในความจริงแล้วใช้ได้เฉพาะกับเด็กและวัยรุ่นเท่านั้น เพราะเมื่อคนเรามีอายุมากขึ้น ความสามารถในการได้ยินก็จะลดน้อยลงไปด้วย ดังนั้นในช่วงอายุ 20 ปี ความถี่สูงสุดที่คนเราสามารถรับฟังได้จะตกลงมา เหลือเพียง 16,000 Hz และในผู้สูงอายุก็จะมีความสามารถในการได้ยินอยู่ที่ระหว่าง 20 Hz ถึง 8,000 Hz เท่านั้น

ระดับเสียงนั้น จะมีความสัมพันธ์กับความดังของเสียง เช่นระดับเสียงสูง (มากกว่า 2,000 Hz) จะสูงขึ้นเมื่อความดังของมันเพิ่มขึ้น และในทำนองเดียวกัน ระดับเสียงต่ำ (น้อยกว่า 2,000 Hz) จะต่ำลงอีกเมื่อลดความดังของมันลง ช่างปรากฏการณ์นี้ รู้จักกันในชื่อของ “Steven’s rule” หากมีความถี่เสียงน้อยกว่า 1,000 Hz หรือเสียงที่มีความถี่ สูงกว่า 2,000 Hz ก็จะเห็นปรากฏการณ์นี้ได้อย่างชัดเจน แต่หากใช้ความถี่อยู่ในช่วงระหว่าง 1,000 Hz ถึง 2,000 Hz จะไม่สามารถสังเกตเห็นความแตกต่างใดๆ

ความน่าสนใจอื่นๆ เกี่ยวกับปรากฏการณ์ของเสียง เช่น ปรากฏการณ์ Doppler Effect ที่ว่า หากระยะทางระหว่างแหล่งกำเนิดเสียงกับผู้ฟังนั้นเปลี่ยน ความสามารถในการรับรู้ระดับเสียงของผู้ฟัง ก็จะเปลี่ยนไปด้วย เช่น ระยะทางระหว่างแหล่งกำเนิดเสียงกับผู้ฟังนั้นเพิ่มขึ้น ความสามารถในการรับรู้ระดับเสียงของผู้ฟังก็จะลดลง

3.2 ความสัมพันธ์ระหว่างตัวโน้ตและความถี่

ได้มีการจัดแบ่งกลุ่มของตัวโน้ต 12 ตัว เป็นกลุ่มขึ้นมา เรียกว่า “หนึ่งออกเทฟ” (Octave) ซึ่งประกอบไปด้วยโน้ต A,A#,B,C,C#,D,D#,E,F,F#,G และ G# โน้ตเหล่านี้จะวนซ้ำกันใหม่ และจะระบุตัวเลขเฉพาะของแต่ละทบเสียง ซึ่งในช่วงของเสียงดนตรีทั้งหมด 10 ออกเทฟด้วยกัน จาก C0 (ประมาณ 16 Hz) ถึง C10 (ประมาณ 16kHz)

เราสามารถแยกแยะระดับเสียงดนตรีได้เนื่องจากแต่ละระดับเสียงจะมีความถี่ต่างกันออกไป ดังนั้นจึงได้มีการคำนวณค่าความถี่ของแต่ละระดับเสียงไว้ ตามมาตรฐานของดนตรีตะวันตก ถือว่า โน้ตตัวลา (A) ในออกเทฟ 4 ซึ่งมีความถี่ 440 Hz เป็นโน้ตพื้นฐาน และโน้ตตัวอื่นๆ ก็จะมีความสัมพันธ์กับความถี่ 440 Hz เช่นกัน เช่น โน้ตตัวลาในออกเทฟที่ 5 จะมีความถี่เป็น 2 เท่าของโน้ตตัวลาในออกเทฟที่ 4 ส่วนโน้ตตัวอื่นๆ ในออกเทฟ 4 ก็จะมีความสัมพันธ์กับโน้ตตัวลาในออกเทฟ 4 ในเชิงเอ็กซ์โพเนนเชียล (Exponential) ดังสมการที่ 3.1

$$f_i = 440 \times 2^{i/12} \quad (3.1)$$

f_i คือ ความถี่ของตัวโน้ต

i คือ หมายเลขของตัวโน้ต

เช่น โน้ต A4 ค่า i จะเท่ากับ 0 โดยค่า i จะเพิ่มขึ้นหนึ่งค่า เมื่อโน้ตสูงขึ้นครึ่งเสียง และ i จะมีค่าลดลงหนึ่งค่า เมื่อโน้ตต่ำลงครึ่งเสียง

บทที่ 4

การแปลงฟาสต์ฟูรีเยร์ (Fast Fourier Transform : FFT)

อุปกรณ์ทางคอมพิวเตอร์ในปัจจุบันมีราคาถูกลงและมีประสิทธิภาพสูง อีกทั้งความแม่นยำและความแน่นอนจากการคำนวณโดยใช้คอมพิวเตอร์จะมากกว่า จึงทำให้การวิจัยและพัฒนาของการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing : DSP) เพื่อนำไปใช้กับงานในสาขาวิชาต่างๆ โดยเฉพาะทางด้านอิเล็กทรอนิกส์ ตัวอย่างเช่น การประมวลผลสัญญาณเสียงแบบดิจิทัล การประมวลผลสัญญาณภาพ การประมวลผลสัญญาณเรดาร์ และการประมวลผลสัญญาณชีวภาพ เป็นต้น

การประมวลผลสัญญาณดังกล่าวมีการคำนวณการประสาน (Convolution) เป็นสำคัญ โดยทั่วไปแล้ว การแปลงฟูรีเยร์แบบเต็มหน่วย (DFT) นั้นสามารถนำมาใช้ในการคำนวณการประสานได้ แต่การคำนวณ DFT เองนั้น เมื่อข้อมูลหรือลำดับสัญญาณมีมากก็กินเวลาของคอมพิวเตอร์มาก ถ้าลำดับสัญญาณเข้ายาว N จุดนั้น คอมพิวเตอร์ต้องทำการคูณจำนวนเชิงซ้อน ถึง $N \times N$ ครั้ง และบวกจำนวนเชิงซ้อนอีก $N(N-1)$ ครั้ง

แต่ถ้าเราใช้ การแปลงฟาสต์ฟูรีเยร์ (FFT) ซึ่งเป็นผลงานของ J.W.Cooley กับ J.W.Tukey โดยการใช้ FFT จะทำให้การคำนวณ DFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้งเท่านั้น ซึ่งจำนวนครั้งในการคูณลดลงไปถึง $N/\log_2 N$ เท่า สำหรับค่า $N = 1,024$ จุดแล้ว การคูณจำนวนเชิงซ้อนลดลงไปถึง 102 เท่า ซึ่งทำให้การคำนวณทำได้เร็วขึ้น ดังนั้น FFT จึงเหมาะกับการใช้กับระบบเวลาจริง (Real-time system) ได้ อย่างไรก็ตาม ถึงแม้ FFT จะมีชื่อเรียกว่า การแปลงฟาสต์ฟูรีเยร์ก็ตาม แต่ตัว FFT เองนั้น ไม่ใช่การแปลงฟูรีเยร์ แต่แท้จริงแล้วเป็นเพียงวิธีการหรือลำดับการ ในการคำนวณที่ช่วยให้การคำนวณ DFT ทำได้เร็วขึ้น

4.1 ฟาสต์ฟูรีเยร์แบบฐานสอง (Radix 2 FFT)

การแปลงฟูรีเยร์แบบเต็มหน่วยสำหรับลำดับ $x(m)$ ที่ยาว N จุด สามารถนิยามได้ดังนี้

$$X(k) = \sum_{m=0}^{N-1} x(m)w^{mk} \quad (4.1)$$

โดยที่ตรงนั้น $k, m = 0, 1, \dots, N-1$ และจำนวนเชิงซ้อน $w = e^{-j2\pi/N}$ ในทางวิชาอิเล็กทรอนิกส์ ลำดับ $x(m)$ มักจะเกี่ยวข้องกับสัญญาณในแกนเวลา (time domain) ส่วน $X(k)$ เกี่ยวข้องกับสัญญาณทางความถี่ (frequency domain) หรือเรียกว่า สเปกตรัมของสัญญาณ (frequency spectrum) การเขียนสมการ 5.1 นั้น ได้ละพจน์ $1/N$ ไว้ ทั้งนี้เนื่องจากจุดที่สนใจก็คือต้องการหาวิธีการคำนวณ DFT ให้ได้เร็วที่สุด ในการนำ DFT ไปใช้งานนั้นพจน์ $1/N$ ต้องคูณกับลำดับ $X(k)$ เสมอ ดังนั้นจึงไม่สามารถ

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปรับปรุงความเร็วในการคำนวณจากพจน์ได้ สมการที่ 4.1 สามารถเขียนให้อยู่ในรูปของสมการเมตริกซ์ได้ดังนี้

$$\{X\} = \{A\} \{x\} \quad (4.2)$$

โดยที่ $\{X\}$ และ $\{x\}$ เป็นเวกเตอร์แนวตั้ง (column vector) ที่ประกอบขึ้นด้วยลำดับ $X(k)$ และ $x(m)$ ตามลำดับ และ $\{A\}$ เป็นเมตริกซ์จัตุรัส อันดับขนาด N ที่มีสมาชิกเป็นจำนวนเชิงซ้อน w^{mk} สำหรับกรณี $N=4$ ระบบสมการ 4.2 สามารถเขียนแยกออกได้เป็น

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (4.3)$$

การคำนวณจากสมการ 4.3 นี้โดยตรงนั้น สังเกตเห็นได้ว่า ต้องใช้การคูณจำนวนเชิงซ้อน $N \times N$ หรือ 16 ครั้ง และต้องบวกจำนวนเชิงซ้อน อีก $N(N-1)$ หรือ 12 ครั้ง

หลักการสำคัญประการหนึ่งของ FFT ที่จะลดจำนวนครั้งในการคูณจำนวนเชิงซ้อนก็โดยอาศัยคุณสมบัติภาวะของการเป็นคาบ (periodicity) ของจำนวนเชิงซ้อน w คือ $w^{mk} = [mk \bmod (N)]$ ซึ่ง $[mk \bmod (N)]$ หมายถึงส่วนที่เหลือหลังจากการหารเทอม mk ด้วย N ดังนั้นสมการ 4.3 อาจเขียนได้เป็น

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w^1 & w^2 & w^3 \\ 1 & w^2 & w^0 & w^2 \\ 1 & w^3 & w^2 & w^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \quad (4.4)$$

จากคุณสมบัติภาวะเป็นคาบนี้เอง ทำให้เราสามารถแยกตัวประกอบของเมตริกซ์ $\{A\}$ ออกเป็นเมตริกซ์หลายๆเมตริกซ์คูณกันและสมาชิกภายในเมตริกซ์จะมีค่าเป็นศูนย์มากที่สุด วิธีการแยกตัวประกอบนี้อาจสามารถกระทำได้ดังนี้

จากสมการที่ 4.3 เพื่อให้ได้เป็น FFT แบบการสับทางเวลา เราต้องจัดรูปเมตริกซ์ $\{A\}$ ใหม่ โดยการสลับสมาชิกในแถวตั้งใหม่ สมมติว่าได้ผลเป็นเมตริกซ์ $\{B\}$ หรือ

$$\{X(k)\} = \{B\} \{x(m)\}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^2 & w^1 & w^3 \\ w^0 & w^4 & w^2 & w^6 \\ w^0 & w^6 & w^3 & w^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}$$

แยกตัวประกอบเมตริกซ์ $\{B\}$ ได้

72840

$$\{B\} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w^1 & w^2 & w^3 \\ 1 & w^2 & w^0 & w^2 \\ 1 & w^3 & w^2 & w^1 \end{bmatrix} = \begin{bmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^0 \\ w^0 & w^2 \end{bmatrix} \begin{bmatrix} w^0 & 0 \\ 0 & w^1 \\ w^2 & 0 \\ 0 & w^3 \end{bmatrix} \begin{bmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^0 \\ w^0 & w^2 \end{bmatrix}$$

$$\{B\} = \begin{bmatrix} T'_{N/2} & K(T'_{N/2}) \\ T'_{N/2} & -K(T'_{N/2}) \end{bmatrix} = \begin{bmatrix} 1 & k \\ 1 & -k \end{bmatrix} \begin{bmatrix} T'_{N/2} & \phi \\ \phi & T'_{N/2} \end{bmatrix}$$

$$\{B\} = \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^1 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^3 \end{bmatrix} \begin{bmatrix} w^0 & w^0 & 0 & 0 \\ w^0 & w^2 & 0 & 0 \\ 0 & 0 & w^0 & w^0 \\ 0 & 0 & w^0 & w^2 \end{bmatrix}$$

วิธีการนี้การแยกตัวประกอบจะไม่กระทำโดยตรงจาก $\{A\}$ แต่จะมีการสับตำแหน่ง หรือจัดกลุ่มของเมตริกซ์ ด้วยวิธีการของ การกลับบิต (Bit Reversed) และเมตริกซ์ที่ได้จากการสลับแถว เรียบร้อยแล้วนี้ จึงนำมาแยกตัวประกอบอีกครั้งหนึ่ง ในที่นี้จะขออธิบายวิธีการแยกตัวประกอบด้วยการเขียนครรชนี k และ m ของสมการ 4.1 ด้วยเลขไบนารี (binary number) ดูตัวอย่างสำหรับกรณี $N=4$ ครรชนี k และ m จะมีค่าได้เพียง 0,1,2,3 เท่านั้น เพราะฉะนั้น เราสามารถแทนได้ด้วยเลขไบนารีสองหลัก คือ

$$k = (k_1, k_0) \quad , \quad m = (m_1, m_0) \quad (4.5)$$

โดยที่ k_1, k_0, m_1 และ m_0 เป็นเลขที่มีค่าได้แค่ 0 หรือ 1 เท่านั้น สมการ 4.5 นี้ สามารถเขียนได้ใหม่ เป็น

$$k = 2k_1 + k_0 \quad , \quad m = 2m_1 + m_0 \quad (4.6)$$

ค่าที่ได้นี้เมื่อนำไปแทนลงในสมการ 4.1 ทำให้ได้

$$X(k_1, k_0) = \sum_{m_0=1}^1 \sum_{m_1=1}^1 x(m_1, m_0) w^{(2m_1+m_0)(2k_1+k_0)} \quad (4.7)$$

เนื่องจาก

$$w^{(2m_1+m_0)(2k_1+k_0)} = w^{(2k_1+k_0)2m_1} w^{(2k_1+k_0)m_0} \\ = w^{4m_1k_1} w^{2m_1k_0} w^{(2k_1+k_0)m_0}$$

โดยที่ $w^{4m_1k_1} = 1$ เพราะฉะนั้นเขียนสมการ 4.7 ได้ใหม่เป็น

$$X(k_1, k_0) = \sum_{m_0=0}^1 \left(\sum_{m_1=0}^1 x(m_1, m_0) w^{2m_1k_0} \right) w^{(2k_1+k_0)m_0} \\ = \sum_{m_0=0}^1 (x_1(k_0, m_0)) w^{(2k_1+k_0)m_0} \quad (4.8)$$

โดยการสมมติให้ตัวแปร $x_1(k_0, m_0)$ เป็นการคำนวณระหว่างกลาง เพราะฉะนั้น สมการ 4.8 อาจเขียนเป็นสมการเมตริกซ์ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} X(0,0) \\ X(0,1) \\ X(1,0) \\ X(1,1) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w^1 \\ 0 & 0 & 1 & w^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^0 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{bmatrix} \begin{bmatrix} x(0,0) \\ x(0,1) \\ x(1,0) \\ x(1,1) \end{bmatrix} \quad (4.9)$$

โดยที่

$$\begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^0 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{bmatrix} \begin{bmatrix} x(0,0) \\ x(0,1) \\ x(1,0) \\ x(1,1) \end{bmatrix}$$

$$\begin{bmatrix} x_2(0,0) \\ x_2(0,1) \\ x_2(1,0) \\ x_2(1,1) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w^1 \\ 0 & 0 & 1 & w^3 \end{bmatrix} \begin{bmatrix} x(0,0) \\ x(0,1) \\ x(1,0) \\ x(1,1) \end{bmatrix}$$

ผลที่ได้จากสมการ 4.8 และ 4.9 หรือสมการแยกตัวประกอบ $\{A\}$ นี้ จะเห็นว่าสมาชิกตามแนวนอน (row) ของเมทริกซ์ที่ได้จากการแยกตัวประกอบนั้น มีเพียง 2 ตัวเท่านั้นที่มีค่าไม่เป็นศูนย์ โดยที่สมาชิกตัวหนึ่งมีค่าเป็นหนึ่งเสมอ ส่วนอีกตัวนั้นจะเป็นจำนวนเชิงซ้อน นั่นก็คือในการคูณเมทริกซ์ย่อย แต่ละเมทริกซ์ต้องการการคูณจำนวนเชิงซ้อนเพียง N ครั้ง ถ้าหาก N อยู่ในรูปของสองยกกำลังใดๆ ($N = 2^k$) และลองเขียนสมการ 4.7 สำหรับกรณีนี้ดูเห็นว่าสามารถเขียนแยกเป็นสมการย่อยได้ $\log_2 N$ สมการ เพราะฉะนั้น ด้วยวิธีการนี้ผลรวมของการคูณจำนวนเชิงซ้อนจึงมีค่าเท่ากับ $N \log_2 N$ ครั้งเท่านั้น จากตัวอย่างสำหรับ $N=4$ สมการ 4.9 ต้องใช้การคูณจำนวนเชิงซ้อน 8 ครั้ง และบวกจำนวนเชิงซ้อนอีก 8 ครั้ง ซึ่งจำนวนนี้น้อยกว่าการคูณค่าโดยตรง ความเป็นจริงแล้วจำนวนครั้งในการคูณจำนวนเชิงซ้อนอาจลดทอนไปได้อีก มาดูการคำนวณสัญญาณระหว่างกลาง $x_1(0,0)$ และ ซึ่ง

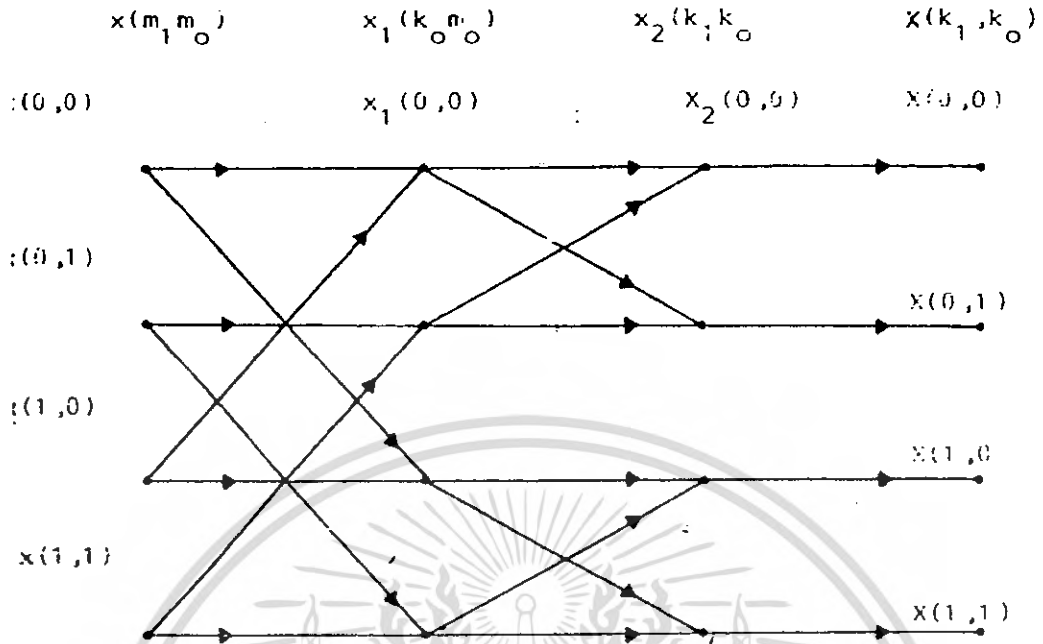
$$x_1(0,0) = x(0,0) + w^0 x(1,0) \quad (4.10a)$$

และ

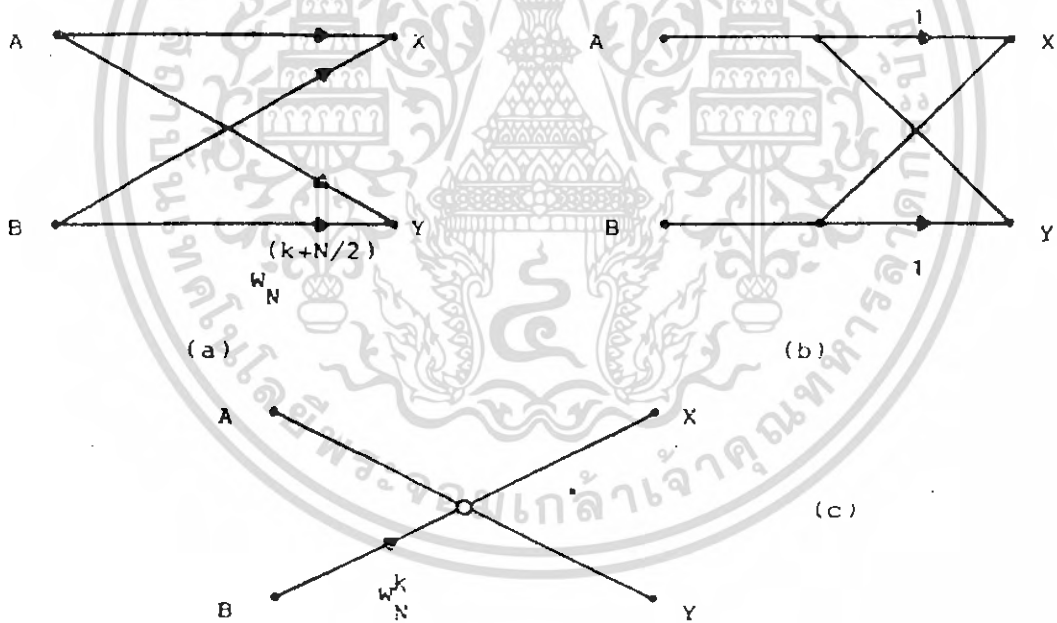
$$\begin{aligned} x_1(0,0) &= x(0,0) + w^2 x(1,0) \\ &= x(0,0) - w^0 x(1,0) \end{aligned} \quad (4.10b)$$

ผลจากการคูณสมบัติของจำนวนเชิงซ้อน $w^2 = -w^0$ ทำให้การคำนวณ $x_1(0,0)$ และ $x_1(0,1)$ ต้องการการคูณจำนวนเชิงซ้อนเพียงครั้งเดียวเท่านั้น อย่างไรก็ตามสำหรับกรณีต่างๆ ไปอาจกล่าวได้ว่าจำนวนครั้งในการคำนวณ DFT ขนาด N จุดโดยใช้ FFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้ง

ในรูปที่ 4.1 นั้นเป็น “ผังไหลของสัญญาณ” (Signal Flow Graph) ที่เขียนแสดงวิธีการคำนวณของ FFT ตามสมการ 4.8 โดยที่หัวลูกศรจะชี้ทิศทางคำนวณ ส่วนตัวอักษรกำกับจะเป็นตัวคูณค่าของสัญญาณที่ต้นทางของลูกศรนั้น



รูปที่ 4.1 ผังการไหลแสดงวิธีการคำนวณ FFT ตามสมการ 4.8



รูปที่ 4.2 หน่วยคำนวณสี่เหลี่ยมของขั้นตอนวิธีแบบสับทางเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ขั้นตอนวิธีของการลัดทางเวลา (Decimation in time : DIT)

วิธีการที่เสนอโดย Colley และ Tukey ซึ่งได้กล่าวมาแล้วนั้น ตามความเป็นจริงแล้ว เป็นการแบ่งลำดับสัญญาณ $x(m)$ ที่มีขนาดยาว N จุด ออกเป็น 2 ส่วน ถ้าเรานิยามให้ลำดับยาว $N/2$ จุด $x_e(m)$ และ $x_o(m)$ เป็นลำดับคู่ และลำดับคี่ของสัญญาณ $x(m)$ ตามลำดับ เพราะฉะนั้น

$$x_e(m) = x(2m) \quad (4.11a)$$

$$x_o(m) = x(2m+1) \quad (4.11b)$$

การคำนวณการแปลง DFT ของลำดับสัญญาณ $x(m)$ ที่ยาว N จุด จะเขียนได้เป็น

$$\begin{aligned} X(k) &= \sum_{\substack{m=0 \\ m_is_even}}^{N-1} x_e(m) w_N^{km} + \sum_{\substack{m=0 \\ m_is_odd}}^{N-1} x_o(m) w_N^{km} \\ &= \sum_{m=0}^{N/2-1} x(2m) w_N^{2mk} + \sum_{m=0}^{N/2-1} x(2m+1) w_N^{2(m+1)k} \end{aligned} \quad (4.12)$$

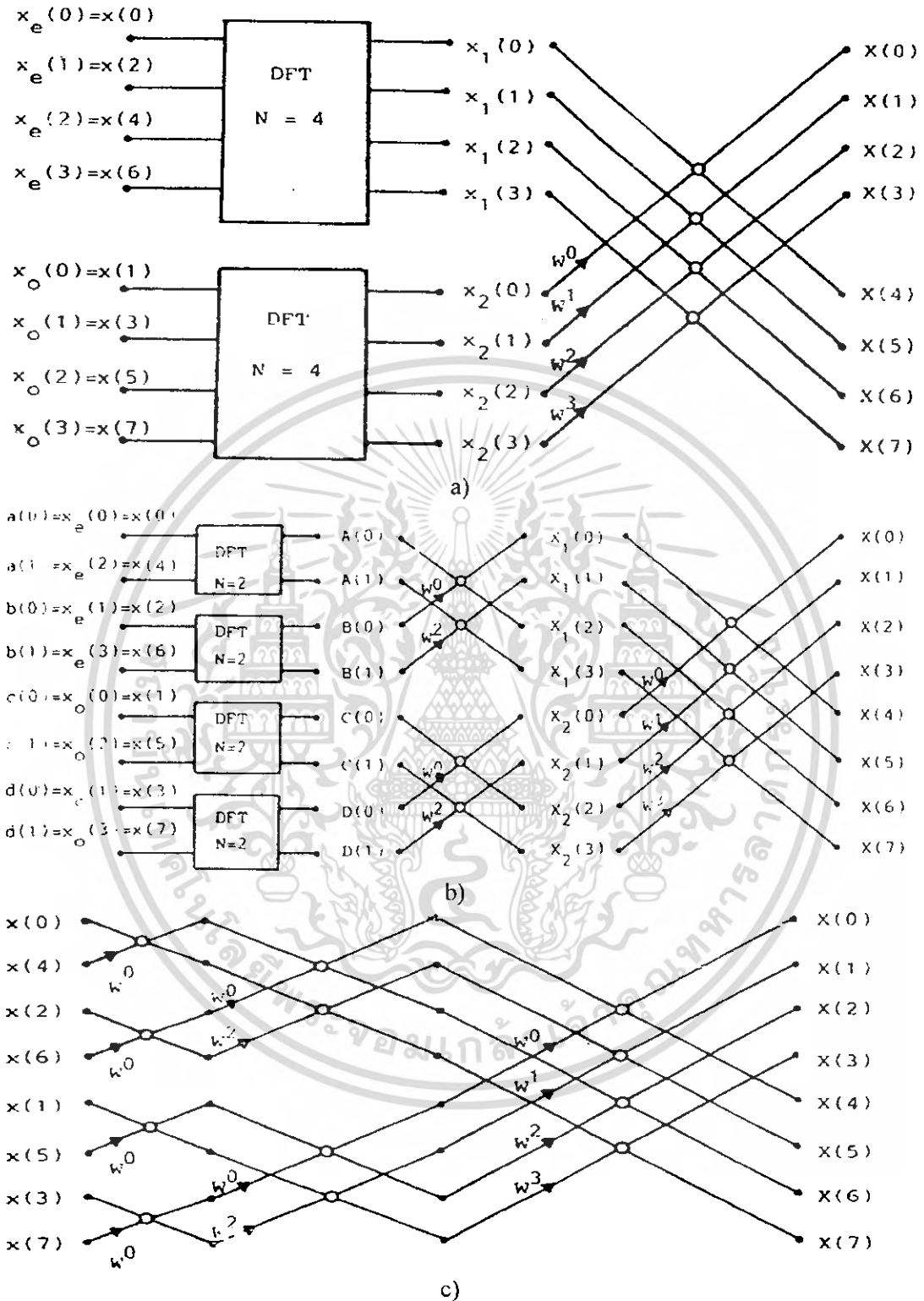
ซึ่งเทอม w_N^2 สามารถเขียนใหม่ได้เป็น

$$w_N^2 = e^{j\left(\frac{2\pi}{N}\right)} = e^{j\left(\frac{2\pi}{N/2}\right)} = w_{N/2}$$

สมการ 4.8 จึงจัดใหม่เป็น

$$\begin{aligned} X(k) &= \sum_{m=0}^{N/2-1} x_e(m) w_{N/2}^{mk} + w_N^k \sum_{m=0}^{N/2-1} x_o(m) w_{N/2}^{mk} \\ X(k) &= X_1(k) + w_N^k X_2(k) \end{aligned} \quad (4.13)$$

โดยที่ $X_1(k)$ และ $X_2(k)$ แทนการแปลง DFT ขนาด $N/2$ จุดของลำดับ $x_e(m)$ และ $x_o(m)$ ตามลำดับ สมการที่ 4.13 แสดงให้เห็นการคำนวณ DFT ขนาด N จุด นั้นสามารถแบ่งการคำนวณย่อยออกเป็น $N/2$ จุด 2 อันได้ และการคูณจำนวนเชิงซ้อนจะลดลงเหลือ $2(N/2)^2 = N^2/2$ ครั้ง ซึ่งจะเห็นว่าลดเวลาการคำนวณลงไปถึง 50 เปอร์เซ็นต์ ในทำนองเดียวกัน เราสามารถแบ่งลำดับ $x_e(m)$ และ $x_o(m)$ ออกเป็นลำดับคู่และลำดับคี่ลงไปเรื่อยๆ จนในที่สุดการคำนวณการแปลง DFT ขนาด N จุด ก็จะทำให้ได้ โดยการคำนวณการแปลง DFT ขนาด 2 จุด จำนวน $N/2$ ภาคด้วยกัน และการแบ่งย่อยเพื่อแบ่งลำดับ $x(m)$ ออกเป็นทีละครึ่งจนเหลือการคำนวณ DFT ขนาด 2 จุด ซึ่งจะทำให้การแบ่งออกได้ $\log_2 N$ ครั้ง (ดูรูป 4.3 ประกอบ)



รูปที่ 4.3 a)-c) ผังการไหลแสดงการคำนวณ DFT ขนาด 8 จุด รูปแสดงการแบ่งย่อยออกเป็นลำดับขั้น
เหลือการคำนวณ DFT ขนาด 2 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาต่อมาก็คือจะเอาการแปลง DFT ขนาด 2 จุด จำนวน $N/2$ ภาคนี้ มาต่อหรือมารวมตัวกัน เพื่อให้ได้การคำนวณ DFT ขนาด N จุดได้อย่างไร ตัวอย่างเช่น การนำเอา DFT ขนาด $N/2$ จุดมาต่อกันโดยตรงตามสมการ 4.13 ขอบมีข้อผิดพลาด เพราะว่า $X_1(k)$ และ $X_2(k)$ เป็น DFT ขนาด $N/2$ ที่นิยามสำหรับช่วง $0 \leq k \leq N/2$ เท่านั้น เพราะฉะนั้นเราต้องนิยามค่าของสมการ 4.13 สำหรับค่า $k > N/2$ คือ

$$X(k) = X_1(k) + w_N^k X_2(k) \quad 0 \leq k \leq N/2-1 \quad (4.14a)$$

และ
$$X(k) = X_1(k-N/2) + w_N^k X_2(k-N/2) \quad N/2 \leq k \leq N-1 \quad (4.14b)$$

พจน์ w_N^k ในสมการ 4.14 นี้มีชื่อเรียกว่า “ตัวประกอบบิด” (Twiddle Factors or Phase or Rotation Factors) ทั้งนี้พจน์นี้เป็นส่วนสำคัญที่ทำให้ DFT ขนาด $N/2$ จุด มารวมตัวขึ้นเป็น DFT ขนาด N จุด

ในสมการ 4.14 นี้ ถ้าหากเรานำเอาความสัมพันธ์ $w_N^{k+N/2} = -w_N^k$ แทนลงไปแล้ว สมการนี้สามารถเขียนใหม่เป็น

$$X(k) = X_1(k) + w_N^k X_2(k) \quad 0 \leq k \leq N/2-1 \quad (4.15a)$$

และ
$$X(k) = X_1(k-N/2) - w_N^{k-N/2} X_2(k-N/2) \quad N/2 \leq k \leq N-1 \quad (4.15b)$$

สมการที่ได้นี้เป็นหลักการสำคัญสำหรับ FFT แบบนี้ และได้กล่าวถึงบ้างแล้วในสมการ 4.10 สมการ 4.15 บอกเราว่าในการคำนวณ $X(k)$ ในสมการที่ 4.15a และ $X(k)$ ในสมการที่ 4.15b ที่สมนัยกันและห่างออกไป $N/2$ จุดนั้น สามารถทำได้โดยการใช้กรกฎจำนวนเชิงซ้อนเพียงครั้งเดียวเท่านั้น จากผลนี้เราสามารถนำไปสร้างหน่วยคำนวณที่มีชื่อเรียกว่า “ผีเสื้อ”(Butterfly) โดยหน่วยคำนวณนี้ประกอบด้วย ข้อมูลเข้าสองข้อมูล คือ A และ B และให้ข้อมูลออกเป็น X กับ Y โดยที่

$$X = A + w_N^k B \quad (4.16a)$$

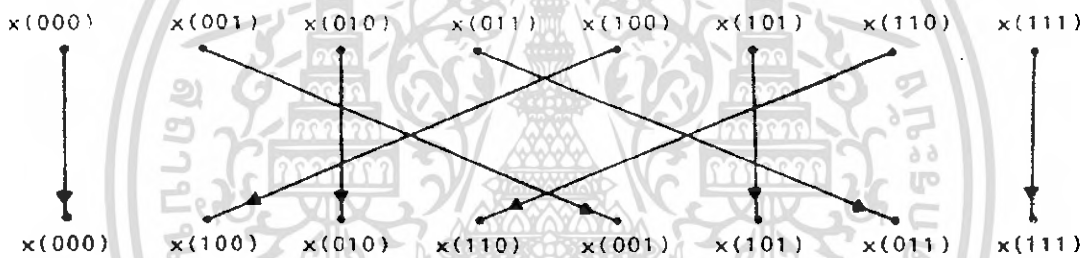
$$Y = A - w_N^k B \quad (4.16b)$$

หน่วยคำนวณผีเสื้อนี้ สามารถเขียนแทนได้ด้วย ผังการไหลของสัญญาณตามรูป 4.3

การคำนวณ DFT แบบ 8 จุด โดยอาศัยวิธีการที่ได้อธิบายมาแล้วนี้ได้แสดงไว้ด้วยผังการไหลของสัญญาณ ดังรูป 4.3 อย่างเป็นลำดับ โดยในเบื้องแรก ลำดับของข้อมูลเข้า $x(m)$ จะถูกแบ่งออกเป็นลำดับคู่ $x_e(m)$ และลำดับคี่ $x_o(m)$ อย่างละ 4 ข้อมูล ดังรูป 4.3(a) ลำดับ $x_e(m)$ และ $x_o(m)$ ก็ถูกแบ่งย่อยไปอีก ดังรูป 4.3(b) ซึ่งในที่สุดการคำนวณ DFT ขนาด 8 จุด จะถูกสับย่อยออกเป็นการคำนวณ DFT ขนาด 2 จุด 4 ภาค ด้วยกัน ซึ่งในรูป 4.3(c) ก็ได้แสดงผังไหลของสัญญาณสำหรับการคำนวณ DFT ขนาด 8 จุดอย่างสมบูรณ์ สังเกตเห็นว่าการคำนวณแบ่งเป็น 3 ส่วน และแต่ละส่วนประกอบด้วยหน่วยคำนวณผีเสื้ออยู่ $N/2$ หน่วย

ตามรูป 4.3(c) นี้มีข้อน่าสังเกตคือ ลำดับสัญญาณเข้า $x(m)$ ไม่ได้จัดเรียงอย่างเป็นต่อเนื่อง แต่ได้ถูกสลับที่กัน (Scrambled Order) การสลับที่กันนี้จะนำไปตามวิธีการที่เรียกว่า “การกลับบิต”

(Bit Reversed) นั่นคือถ้าเราแทนดรรชนี m ของลำดับ $x(m)$ ด้วยเลขไบนารี โดยที่จำนวนบิตของเลขไบนารีต้องเพียงพอที่จะแทนค่า N ได้ ในกรณี $N=8$ ก็ต้องแทนด้วยเลขไบนารีจำนวน 3 บิต จากนั้น ลำดับ $x(m)$ ใหม่ จะได้จากการกลับหลังเลขไบนารีของดรรชนี m เสีย ดังแสดงในรูป 4.4 นั่นคือ $x(001)$ จะถูกแทนที่ด้วย $x(100)$ หรือ $x(110)$ ถูกแทนที่ด้วย $x(011)$ เป็นต้น จากที่กล่าวแล้วปกติดรรชนี m มักจะเกี่ยวข้องกับเวลา และวิธีการของ FFT แบบนี้เป็นารลดเวลาการคำนวณโดยการสลับลำดับ $x(m)$ ออกเป็นกลุ่มย่อยๆ ที่แต่ละกลุ่มประกอบด้วยลำดับ $x(m)$ เพียง 2 ค่า การจัดกลุ่มนี้คล้ายกับการสุ่ม (Sampling) ลำดับเดิมด้วยอัตราที่ต่ำกว่า และถ้าหากเราถือว่าแต่ละกลุ่มใหม่ที่จัดขึ้นมาเป็นลำดับข้อมูล 1 ลำดับแล้ว ก็เท่ากับว่าเราได้ตัดทอนลำดับสัญญาณลงไป ดังนั้นวิธีการคำนวณแบบนี้ จึงมีชื่อเรียกว่า “การสลับทางเวลา”(Decimation in time : DIT) เราอาจเขียนแผนภาพแทนลำดับวิธีการของ DIT ได้ดังรูป 4.4 โดยที่ DFT ขนาด N จุดเดิมถูกแบ่งออกเป็น DFT ขนาด $N/2$ ภาค และนำมารวมกันโดยใช้แฟลคเตอร์บิต และลำดับการนี้จะทำงานกระทั่งผลสุดท้ายเป็นการแปลง DFT ขนาด 2 จุด ฟังสังเกตว่า การนำเอา DFT ขนาด $N/2$ จุดมาประกอบกันขึ้นนี้มีวิธีการหลายแบบขึ้นอยู่กับการจัดเรียงแฟลคเตอร์บิต การจัดเรียงแฟลคเตอร์บิตที่ต่างกันออกไปก็จะได้ FFT ชนิดที่ต่างกันออกไปด้วย



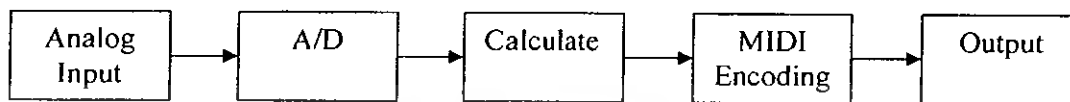
รูปที่ 4.4 การสลับตำแหน่งของลำดับ $x(m)$ ด้วยวิธีการกลับบิต

ที่ได้กล่าวไว้ว่า การคำนวณ DFT ขนาด N จุด โดยใช้ FFT นั้นจำนวนครั้งในการคูณจำนวนเชิงซ้อนมีแค่ $N \log_2 N$ นั้น ความเป็นจริงค่านี้นั้นเป็นเพียงค่าประมาณและใช้กับ FFT โดยทั่วไป ทั้งนี้ลองย้อนกลับไปดูลำดับการคำนวณตามรูป 4.3(c) แล้วจะเห็นว่า ส่วนแรกซึ่งเป็น DFT ขนาด 2 จุดนั้น การคำนวณมีเฉพาะการบวกและการลบจำนวนเชิงซ้อนเท่านั้น ในส่วนที่สองก็เช่นกันการคูณจำนวนเชิงซ้อนมีเฉพาะในส่วนที่สามและใช้การคำนวณเชิงซ้อนเพียง 2 ครั้งเท่านั้น แต่ถ้าใช้สูตรการคำนวณ $N \log_2 N$ ก็จะได้ 24 ครั้ง (หรือ 12 ครั้งเมื่อใช้ความสัมพันธ์ของหน่วยคำนวณสี่เหลี่ยม) ผลที่ได้แตกต่างกันมาก อย่างไรก็ตามสำหรับกรณีที่ N มีค่าใหญ่มาก การที่จะเขียนโปรแกรมหรือสร้างวงจรซึ่งสามารถใช้กับกรณีทั่วไป ได้ด้วยนั้น ข้อสังเกตเหล่านี้จำเป็นต้องมีโปรแกรมหรือวงจรส่วนหนึ่ง เพื่อมาวิเคราะห์ด้วยทำให้เกิดความยุ่งยากโดยไม่จำเป็น ดังนั้นจึงอาจสรุปได้ว่า การคูณจำนวนเชิงซ้อนของลำดับการ FFT จะมีประมาณ $N \log_2 N$ ครั้ง

บทที่ 5

การออกแบบและสร้าง

5.1 หลักการ



รูปที่ 5.1 Block Diagram การทำงานของระบบทั้งหมด

จากรูปจะทำการรับค่าจากสัญญาณอนาล็อกโดยมีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัล ใช้กระบวนการแปลงฟูรีเยอร์อย่างรวดเร็วเพื่อทำการแยกองค์ประกอบทางความถี่ของสัญญาณ และจับค่าของความถี่ที่มีแอมพลิจูดสูงที่สุด และนำไปเปรียบเทียบกับโน้ตดนตรีตัวโน้ต เมื่อได้ค่าแล้ว จะส่งสัญญาณของโน้ตตัวนั้นส่งไปยังพอร์ทเอาต์พุต ในรูปแบบของรหัสมีดี (Midi Code)

5.2 การออกแบบ

เนื่องจากรูปที่ 5.1 เราจะพบว่าเราจะต้องใช้อุปกรณ์ที่มีโมดูล A/D อยู่ภายใน และเนื่องจากโครงการนี้เป็นการออกแบบในระบบเวลาจริงเสมือน (Real Time System) จึงต้องใช้อุปกรณ์ที่เอื้ออำนวยต่อการคำนวณ และประมวลผลสัญญาณเชิงดิจิทัล ซึ่งในที่นี้เราจะใช้วิธีการคำนวณที่เหมาะสมและเร็วที่สุด คือ Fast Fourier Transform และมีการส่งสัญญาณออกพอร์ทอนุกรม ดังนั้นจึงเลือกใช้ไมโครคอนโทรลเลอร์ชนิด dsPIC30F4011 ซึ่งมีคุณสมบัติที่เอื้ออำนวย ดังนี้

- เป็นไมโครคอนโทรลเลอร์ ขนาด 16 บิต
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- หน่วยความจำโปรแกรมแบบ Flash ขนาด 48 kByte
- หน่วยความจำ SRAM ที่มีความเร็วสูง ขนาด 2,048 Byte
- ความถี่สัญญาณนาฬิกาจากภายนอกตั้งแต่ไฟตรงจนถึง 40MHz
- ความถี่สัญญาณภายในสามารถใช้งานกับวงจรเฟสล็อกภายใน ตั้งแต่ 4 MHz –

10 MHz เลือกได้ 3 ระดับ คือ 4,8 หรือ 16 เท่า

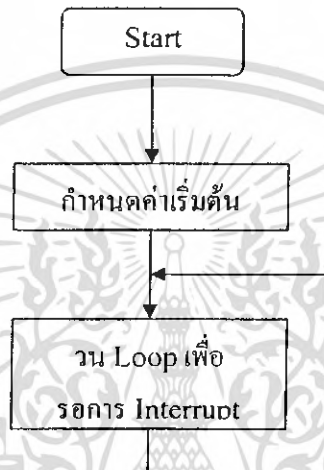
- ทำการคูณเลข 16 บิต ได้ภายในสัญญาณนาฬิกาเพียง 1 ไซเคิล
- ไทเมอร์/เคาน์เตอร์ 16 บิต 5 ตัว สามารถต่อใช้งานร่วมกันเป็น ไทเมอร์/เคาน์เตอร์ 32 บิต ได้
- มีโมดูลตรวจจับสัญญาณดิจิทัล
- มีวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีโมดูลสื่อสารข้อมูลอนุกรม UART
- ฯลฯ

5.3 การออกแบบซอฟต์แวร์

เมื่อเปิดเครื่องแล้ว เราจะรอรับสัญญาณจากกิตารไฟฟ้า เพื่อให้เกิดการอินเตอร์รัปต์ และโปรแกรมทั้งหมดก็จะทำงานภายในเงื่อนไขการเกิดอินเตอร์รัปต์นั้น ดังนั้น โฟลว์ชาร์ทโดยรวมจึงเป็นดังนี้



รูปที่ 5.2 ของโปรแกรมหลัก

ซึ่งเมื่อรับสัญญาณมาแล้วเราก็จะนำเข้าสู่ A/D แล้วทำการเก็บค่านี้ไว้ กำหนดเป็นค่า $x(0)$ แล้วทิ้งช่วงเวลาไว้ช่วงหนึ่ง ซึ่งมีค่าเท่ากับคาบการสุ่มสัญญาณ แล้วรับค่า input มาอีก ให้เป็น $x(1)$ ทำซ้ำเดิมไปเรื่อยๆ จนกระทั่งครบ 128 ค่า ($x(0)$ ถึง $x(127)$) ต่อจากนั้น ก็ทำการสลับบิท แล้วเข้าสู่ Fast Fourier Transform แล้วนำเอาลำดับที่มีค่าแอมพลิจูดมากที่สุด ไปหาเปรียบเทียบกับ เป็นโน้ตตัวไหน ต่อจากนั้นถอดรหัสเป็นรหัสมิดี้ (Midi Code) แล้วนำเข้าสู่ฟอร์ทอนุกรมเพื่อนำไปใช้กับอุปกรณ์ที่สนับสนุนกับรหัสมิดี้ต่อไป ซึ่งจะเป็นไปตามรูป 5.3

5.4 เหตุที่ทำให้เกิดการอินเตอร์รัปต์

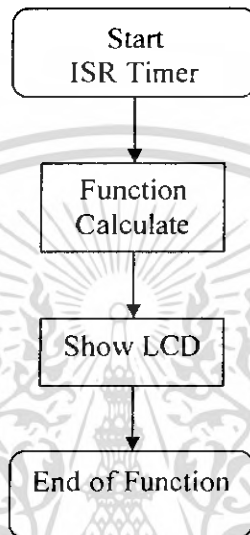
มี 2 สาเหตุ คือ

1. เกิดจากไทเมอร์นับครบเวลาที่เรากำหนด ซึ่งเรากำหนดเวลาที่ทำให้เกิดอินเตอร์รัปต์ไว้ ทุกๆ 375 μs หากในโปรแกรมอินเตอร์รัปต์ไทเมอร์นี้ จะประกอบไปด้วย

- การอ่านค่า A/D
- การแซมปลิงสัญญาณ
- การคำนวณและหาค่าแห่งของสัญญาณที่มีแอมพลิจูดสูงที่สุด

- การถอดรหัส
- การส่งค่าออกไปสู่ มิติพอร์ท
- การแสดงผลทาง LCD

เราจะเห็นว่าอินเทอร์รัปต์ไทมเมอร์นี้จะควบคุมการประมวลผลทั้งหมด จึงดูเหมือนว่ามันทำงานเปรียบเสมือนโปรแกรมปกติ ซึ่งมีโคอะแกรม ตามรูปข้างล่าง



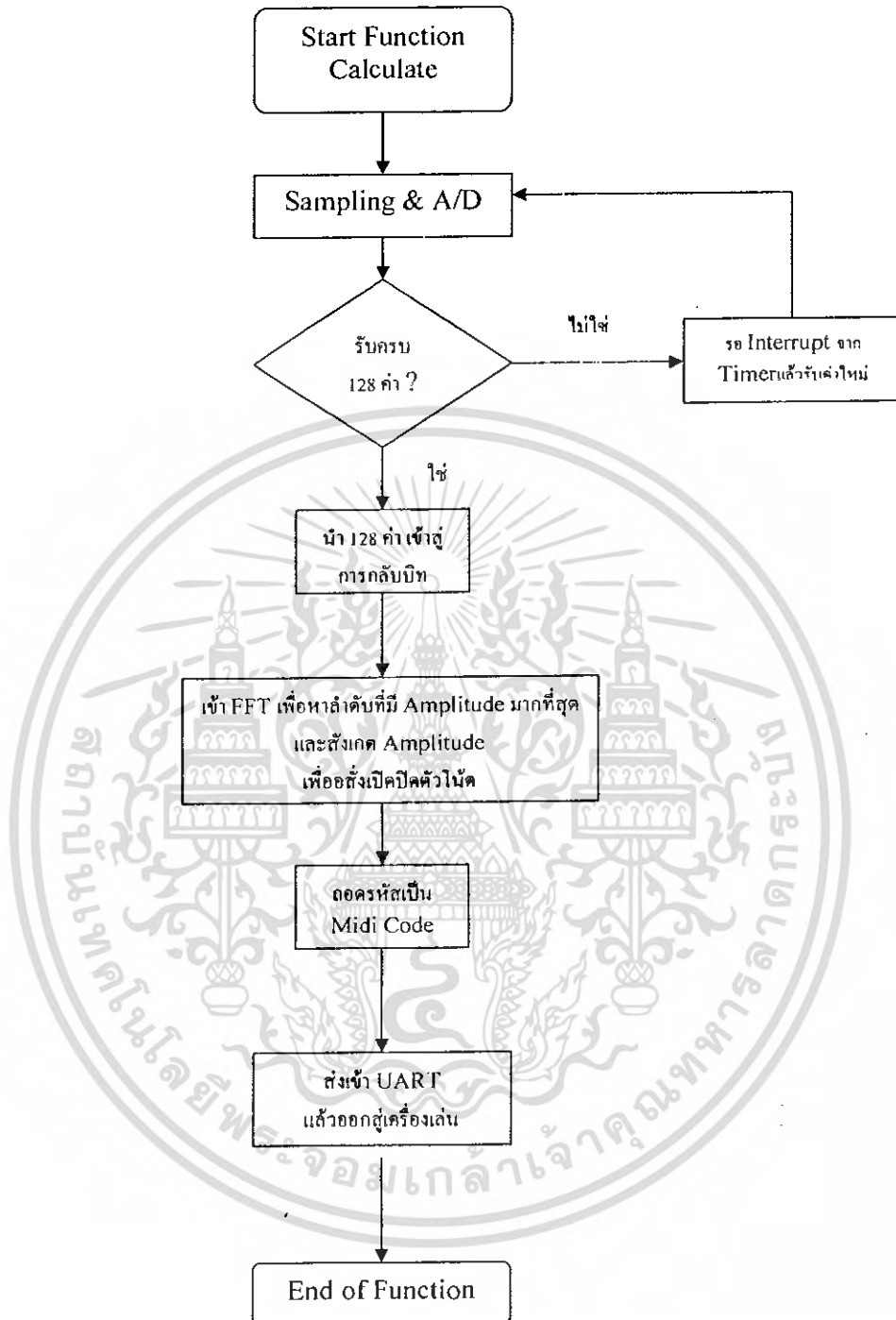
รูปที่ 5.3 โฟลว์ชาร์ทของโปรแกรมการอินเทอร์รัปต์ไทมเมอร์

2.เกิดจากสัญญาณอินเทอร์รัปต์จากภายนอก ซึ่งเราเอาไว้เพื่อเลือกเมนูที่จอ LCD เพื่อส่งค่าเลือกชนิดของมิติคอนโทรลเลอร์ ซึ่งถูกสร้างเป็นลักษณะปุ่ม 3 ปุ่ม ใช้รับสัญญาณอินเทอร์รัปต์จากภายนอก ซึ่งคอนโทรลเลอร์นี้มี 3 แหล่ง ซึ่งมีหน้าที่ ดังนี้

- อินเทอร์รัปต์หมายเลข 0 เอาไว้เลื่อนเมนูไปทางซ้าย
- อินเทอร์รัปต์หมายเลข 1 เอาไว้เลื่อนเมนูไปทางขวา
- อินเทอร์รัปต์หมายเลข 2 เอาไว้เลือกเมนู และออกจากเมนู

โดยการกดปุ่มเลือกเมนูซ้ายหรือขวานั้น คือ การส่งค่ามิติคอนโทรลเลอร์ที่แสดงอยู่ที่ LCD ไป 1 ครั้ง

โดยการกำหนดความสำคัญของการเกิดอินเทอร์รัปต์นั้น เราจะให้ การเกิดอินเทอร์รัปต์จากสัญญาณภายนอกมีความสำคัญมากกว่า การเกิดอินเทอร์รัปต์ไทมเมอร์ เพื่อเราสามารถส่งสัญญาณควบคุมมิติได้ตลอดเวลา แล้วจึงเล่นตัวโน้ตซึ่งอยู่ในโปรแกรมอินเทอร์รัปต์ไทมเมอร์ได้



รูปที่ 5.4 โฟลว์ชาร์ทของฟังก์ชัน Calculate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ฟังก์ชันการกลับบิท

จากการสังเกต ดังนี้ ถ้าเรารับมา 128 ค่า โดย

x(0) คือ x(0000000)	แปลงเป็น	x(0) คือ x(0000000)
x(1) คือ x(0000001)	แปลงเป็น	x(64) คือ x(1000000)
x(2) คือ x(0000010)	แปลงเป็น	x(32) คือ x(0100000)
x(3) คือ x(0000011)	แปลงเป็น	x(96) คือ x(1100000)
x(4) คือ x(0000100)	แปลงเป็น	x(16) คือ x(0010000)

x(15) คือ x(0001111)	แปลงเป็น	x(120) คือ x(1111000)
x(16) คือ x(0010000)	แปลงเป็น	x(4) คือ x(0000100)

พบว่า เราสามารถวน Loop 7 Loop ซ้อนกันได้ แล้วจึงกลับเข้าไปแปลงเป็นเลขฐานสิบ

ดังนี้

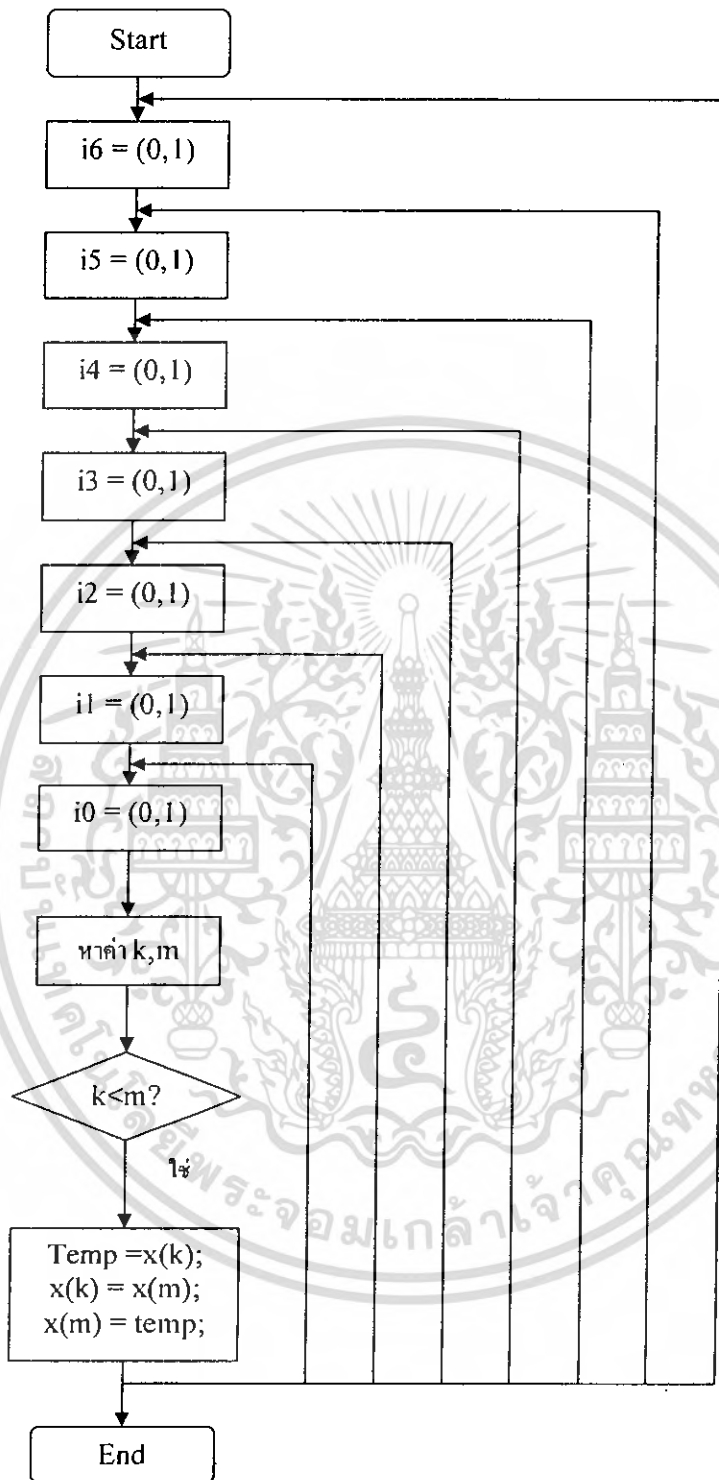
$$k = 64*i6 + 32*i5 + 16*i4 + 8*i3 + 4*i2 + 2*i1 + i0;$$

$$m = 64*i0 + 32*i1 + 16*i2 + 8*i3 + 4*i4 + 2*i5 + i6;$$

โดยที่ k คือ ค่าตำแหน่งที่รับเข้ามาแบบเรียงลำดับ จากวงจร A/D

m คือ ค่าตำแหน่งที่ต้องการสลับที่กับ k

และจากการสังเกต จะพบว่า เราต้องการสลับแค่กรณี $k < m$ เท่านั้น
ดังนั้น จึงเขียนโปรแกรมได้ดังนี้



รูปที่ 5.5 โฟลว์ชาร์ทการกลับบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 ฟังก์ชัน Fast Fourier Transform

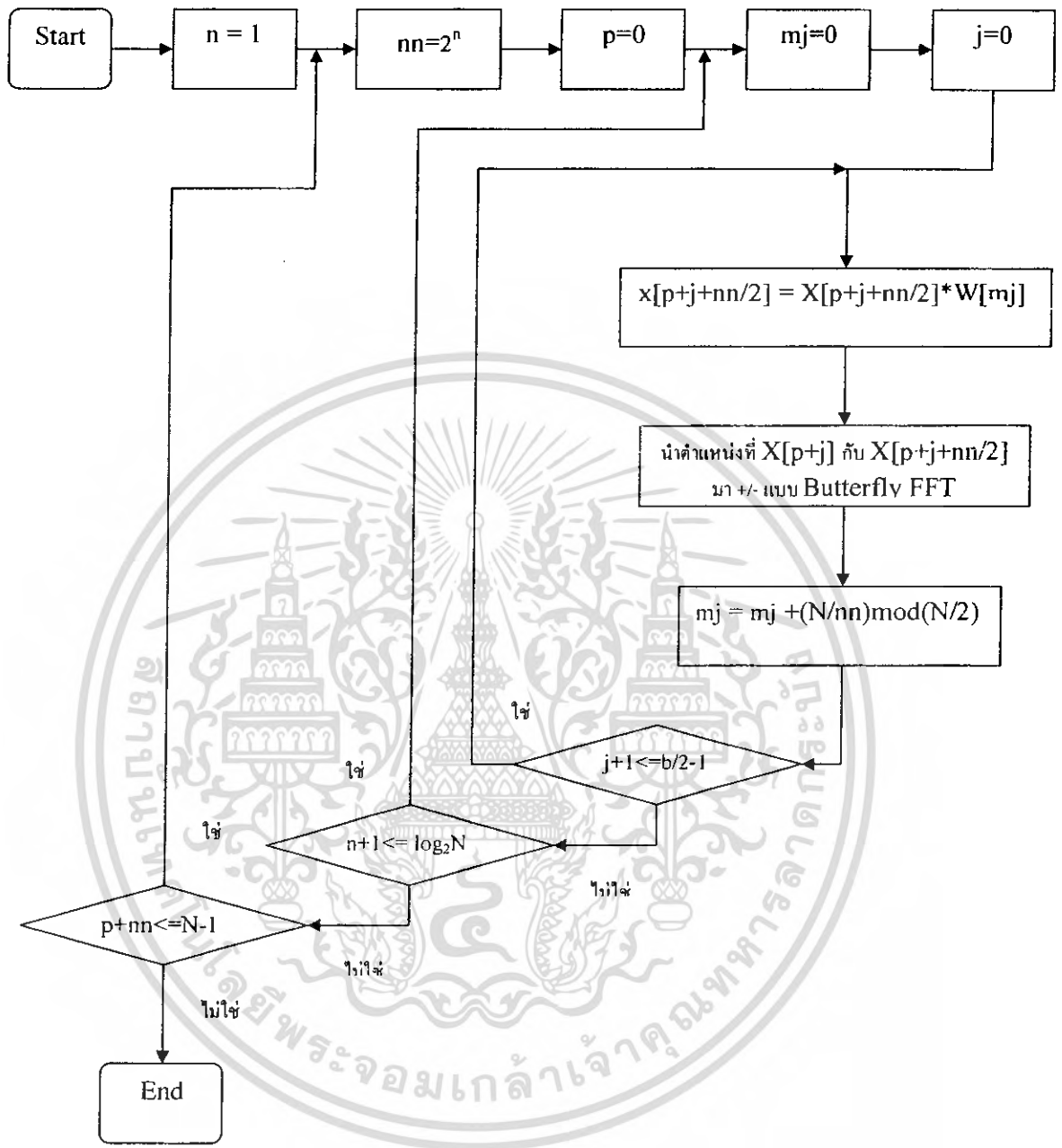
เราจะวนลูป 3 ลูป ที่ซ้อนกัน คือลูป n, p, j โดย n เป็นจำนวนครั้งในการคูณแต่ละชุด โดย จะมีค่าตั้งแต่ 0 เพิ่มทีละ 1 จนถึง $\log_2 N$ เมื่อ N เป็นจำนวนข้อมูล และ p จะแทนจุดเริ่มต้นของแต่ละชุด โดย มีค่าตั้งแต่ 0 ไปจนถึงไม่เกิน $N-1$ และเพิ่มขึ้นทีละ 2^n ส่วน j จะเป็นจำนวนครั้งของการคำนวณใน p หนึ่งครั้ง และค่า m_j เป็นตัวระบุค่าของ Twiddle Factor ที่ใช้ในแต่ละครั้ง ซึ่งจะมีค่าเพิ่มขึ้นทีละ $(N/2^n) \% (N/2)$ โดยที่เครื่องหมาย $\%$ คือการหารเอาเศษ

ในการคำนวณ 1 ครั้งของ Fast Fourier Transform จะทำเป็นคู่กัน สมมติว่า A คำนวณกับ B เราจะได้

$$\begin{aligned} A &= A + BW \\ B &= A - BW \end{aligned}$$

ดังนั้น เราจึงเขียนโฟลว์ชาร์ต ได้ดังนี้





รูปที่ 5.6 โพลีชาร์ทของ Fast Fourier Transform

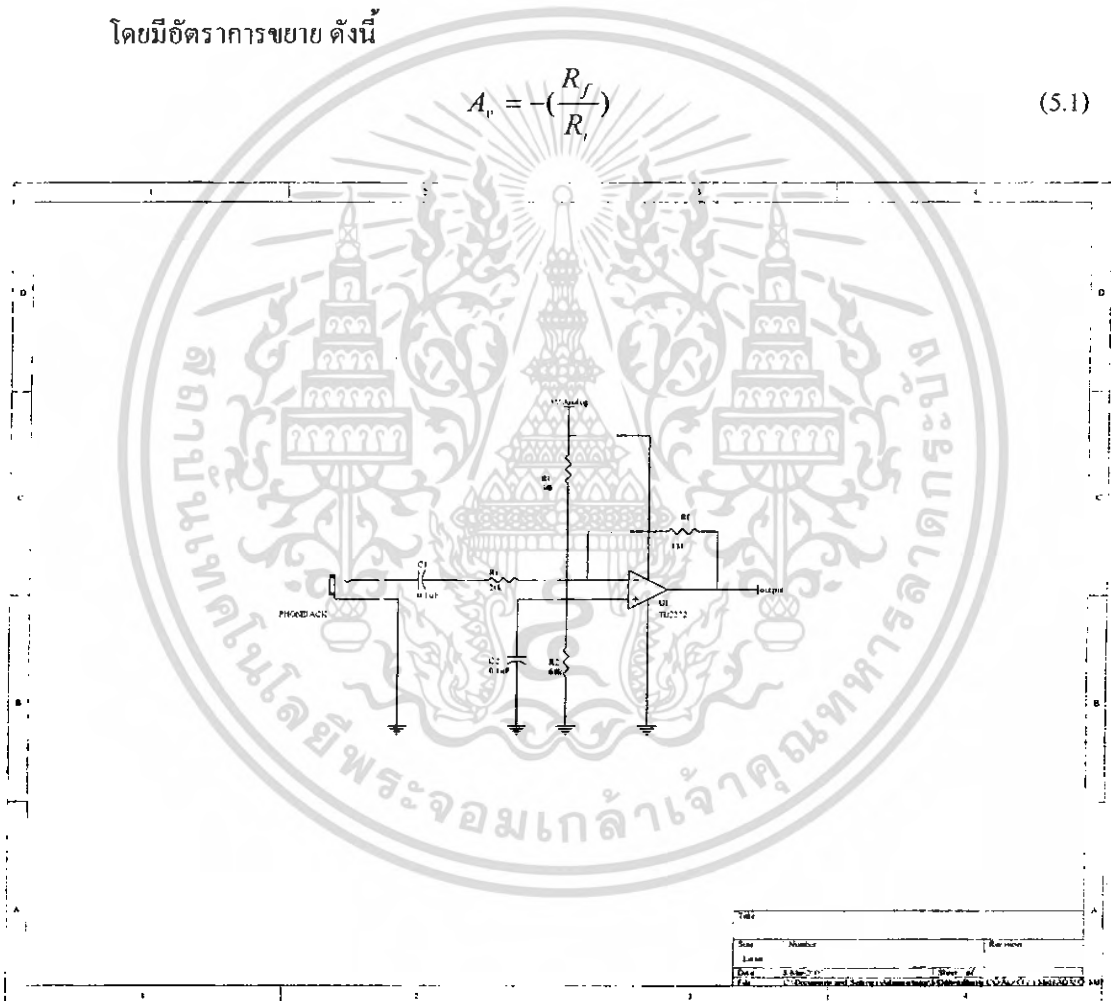
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7 วงจรขยายสัญญาณ(Amplifier)

เราจะใช้วงจรขยายแบบกลับเฟส(Inverting Amplifier) ซึ่งเป็นไปตามรูปที่ 5.7 โดยมี R_1 และ R_2 ทำหน้าที่ขั้วระดับของสัญญาณ ให้มีจุดปฏิบัติการอยู่ที่ 2.5 โวลต์ ส่วน C_1 ทำหน้าที่กรองสัญญาณรบกวนสัญญาณแรงดันไฟตรงจากกิตาร์ไฟฟ้าไม่ให้ผ่านไปสู่วงจรนี้ได้ และยังกรองสัญญาณรบกวนความถี่ต่ำด้วย ส่วน C_2 ทำหน้าที่กรองสัญญาณรบกวนซึ่งเกิดจากแหล่งจ่ายไฟเลี้ยง และสำหรับอัตราขยายนั้น เราจะสามารถรู้ได้ จากการทดลองดีคัตวโน้ตเข้าไปก่อน แล้วดูผลลัพธ์จากการขยายว่า มันเพี้ยน (ชนขอบหรือเปล่า) โดยวงจรขยายที่คืนัน จะต้องขยายโดยไม่ให้รูปสัญญาณที่ได้จากการขยายผิดเพี้ยนไปจากเดิม

โดยมีอัตราขยาย ดังนี้

$$A_v = -\left(\frac{R_f}{R_i}\right) \quad (5.1)$$



รูปที่ 5.7 วงจรขยายที่ใช้ในการออกแบบ

บทที่ 6

การทดลอง

การทดลองที่ 1 เปรียบเทียบค่าความถี่และแอมพลิจูดระหว่างค่าที่วัดได้กับค่าที่คำนวณได้

ในการทดลองเราจะนำเสียงกีตาร์ไฟฟ้าเก็บไว้ในคอมพิวเตอร์ โดยเก็บตั้งแต่ตัวโน้ต G3-D6 ซึ่งก็คือกีตาร์ไฟฟ้าสาย 3 เฟรทที่ 0 ถึง สาย 1 เฟรทที่ 22 โดยตั้งค่าความถี่การแซมปลิงขณะอัดเสียงไว้ที่ 96,000 Hz ที่ความดังหรือแอมพลิจูดเดียวกันแล้วนำมาคำนวณโดยใช้โปรแกรม MATLAB คำนวณหาค่าแอมพลิจูดและความถี่ โดยเขียนโปรแกรมให้เก็บค่า Sampling 128 ค่า ที่ความถี่การแซมปลิง 2666.67 Hz ซึ่งแต่ละช่วงจะใช้เวลา 48 ms โดยใช้ Code Program

```
t = 1
for n=1:t
    m = 4608*(n-1)
    a = 1+m
    b = 4608+m
    y = wavread('G3',[a,b]);
    for i=1:128
        x(i) = y(36*i);
    end
    YY = fft(x,128);
    Pyy = abs(YY)/128;
    f = (0:(2666.66/128):1333.33);
    figure(1);
    subplot(t,2,2*n);
    plot(f,Pyy(1:65),f,Pyy(1:65),'xr')
    title('Frequency content of signal ')
    xlabel('Frequency (Hz)')
    subplot(t,2,(2*n)-1);
    plot(x)
    title('signal of G3 Guitar'); end
```

ตารางที่ 6.1 ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ต

โน้ต	ความถี่ของ ตัวโน้ต (Hz)	ความถี่ที่วัด ได้ (Hz)	ความถี่ขอบ ล่าง (Hz)	แอมพลิจูด ของขอบ ล่าง	ความถี่ขอบ บน (Hz)	แอมพลิจูด ของขอบบน
G3	199.9577	195.96	187.50	0.0358	208.33	0.0246
G#3	207.6523	208.98	208.33	0.0398	229.17	0.0055
A3	220.0000	217.45	208.33	0.0224	229.17	0.0358
A#3	233.0819	234.38	229.17	0.0379	250.00	0.0107
B3	246.9417	248.70	229.17	0.0033	250.00	0.0324
C4	261.6256	263.67	250.00	0.0237	270.83	0.0424
C#4	277.1826	278.65	270.83	0.0312	291.67	0.0167
D4	293.6648	294.92	270.83	0.0071	291.67	0.0452
D#4	311.1270	311.84	291.67	0.0015	312.5	0.0436
E4	329.6275	330.07	312.5	0.0018	333.33	0.0163
F4	349.2282	352.21	333.33	0.0010	354.17	0.0162
F#4	369.9944	371.09	354.17	0.0036	375.00	0.0191
G4	391.9954	391.28	375.00	0.0052	395.83	0.0209
G#4	415.3046	416.67	416.67	0.0239	437.50	0.0007
A4	440.0000	442.71	437.50	0.0230	458.33	0.0075
A#4	466.1637	466.80	458.33	0.0215	479.17	0.0164
B4	493.8833	494.14	479.17	0.0092	500.00	0.0231
C5	523.2511	524.09	520.83	0.0248	541.67	0.0045
C#5	554.3652	553.39	541.67	0.0150	562.50	0.0206

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1(ต่อ) ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ต

โน้ต	ความถี่ของ ตัวโน้ต (Hz)	ความถี่ที่วัด ได้ (Hz)	ความถี่ขอบ ล่าง (Hz)	แอมพลิจูด ของขอบ ล่าง	ความถี่ขอบ บน (Hz)	แอมพลิจูด ของขอบบน
D5	587.3295	585.29	583.33	0.0271	604.17	0.0059
D#5	622.2539	621.745	604.17	0.0032	625.00	0.0283
E5	659.2551	661.46	645.83	0.0083	666.66	0.0280
F5	698.4564	697.27	687.50	0.0177	708.33	0.0196
F#5	739.9888	734.37	729.16	0.0251	750.00	0.0085
G5	783.9908	784.51	770.83	0.0094	791.66	0.0258
G#5	830.6093	835.29	833.33	0.0269	854.16	0.0044
A5	880.0000	875.65	875.00	0.0028	895.83	0.0030
A#5	932.3275	932.29	916.66	0.0218	937.50	0.0157
B5	987.7666	976.56	979.16	0.0308	1000.00	0.0019
C6	1046.5022	1069.66	1062.50	0.0214	1083.33	0.0142
C#6	1108.7305	1107.42	1104.16	0.0028	1125.00	0.0376
D6	1174.6590	1165.89	1166.66	0.0095	1187.50	0.0291

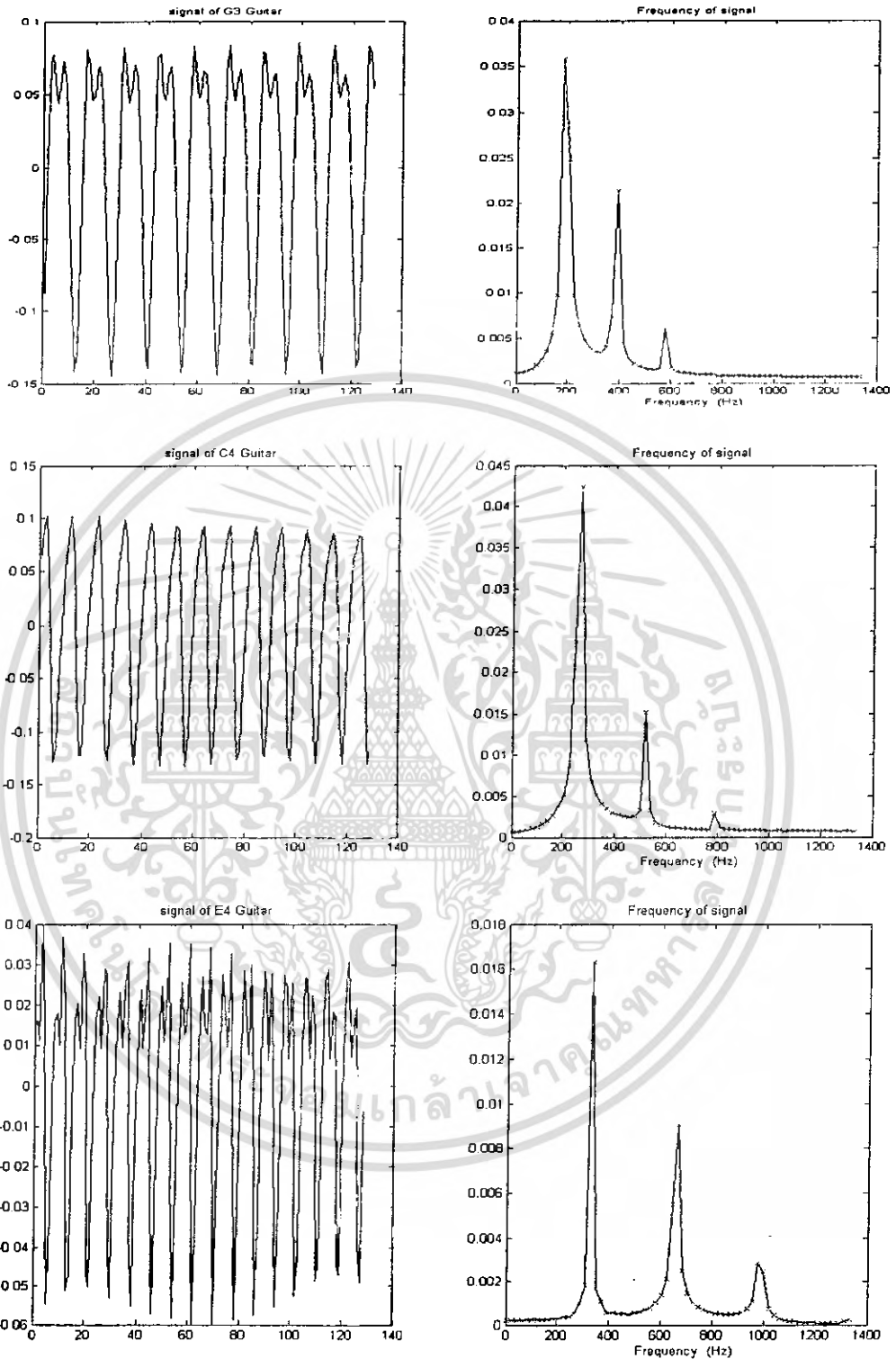
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการวัดค่าความถี่นั้นเราจะใช้จำนวน Sample มากกว่า 128 ค่าเพื่อให้ได้ความละเอียดในการ Sampling มากขึ้น ทำให้ค่าที่วัดได้ใกล้เคียงค่าความถี่จริงของเสียงกีตาร์ไฟฟ้า ซึ่งผู้ทดลองเลือกใช้ จำนวน Sample 4096 ค่า ดังนั้น Code Program จะเปลี่ยน ไปเล็กน้อยดังนี้

```
t = 1
for n=1:t
    m = 147456*(n-1)
    a = 1+m
    b = 147456+m
    y = wavread('G3',[a,b]);
    for i=1:4096
        x(i) = y(36*i);
    end

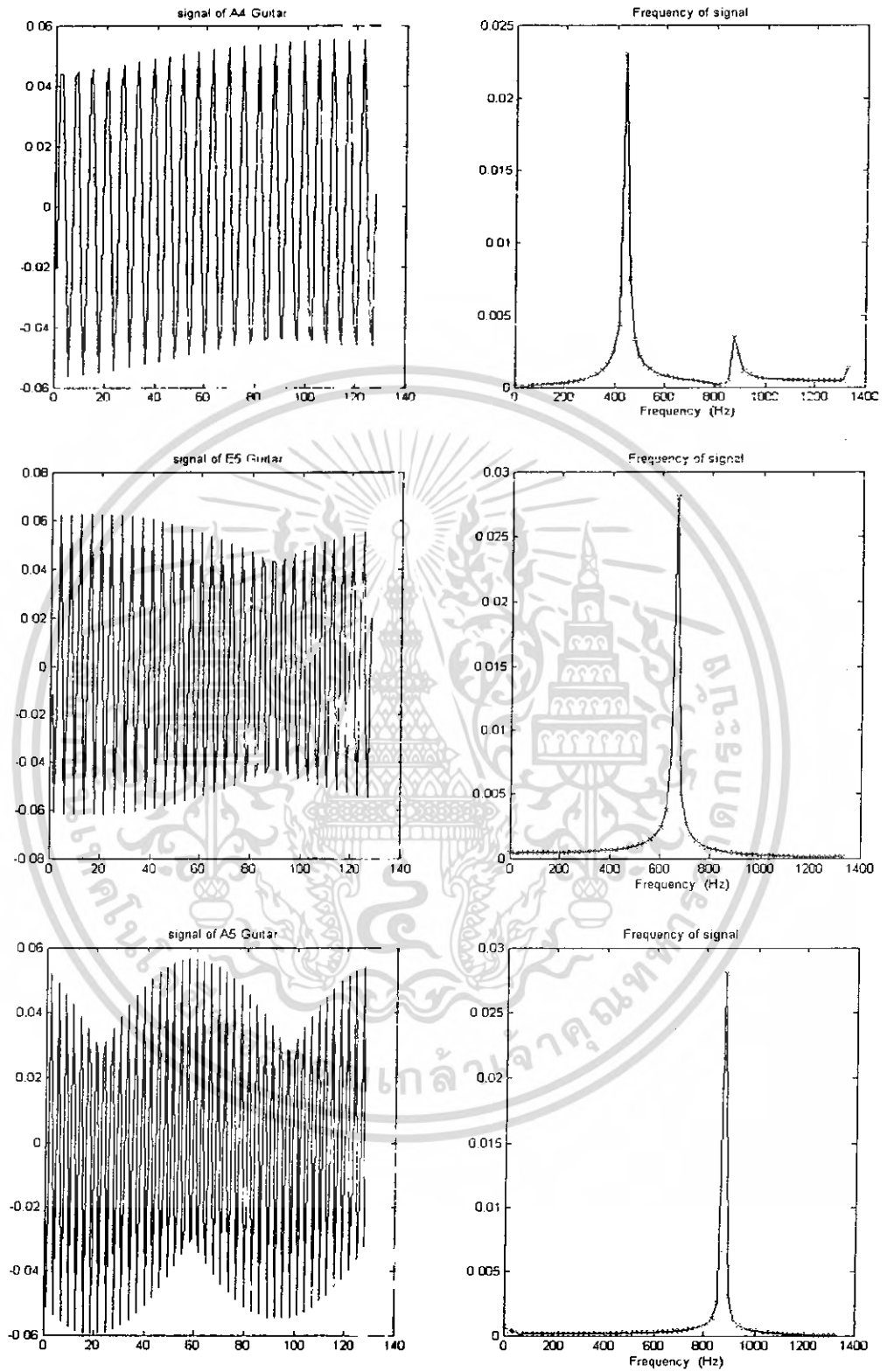
    YY =fft(x,4096);
    Pyy = abs(YY) /4096;
    f = (0:(2666.666/4096):1333.333);
    figure(1);
    subplot(t,2,2*n);
    plot(f,Pyy(1:2049),f,Pyy(1:2049),'xr')
    title('Frequency content of signal ')
    xlabel('Frequency (Hz)')
    subplot(t,2,(2*n)-1);
    plot(x)
    title('signal of G3 Guitar')
end
```





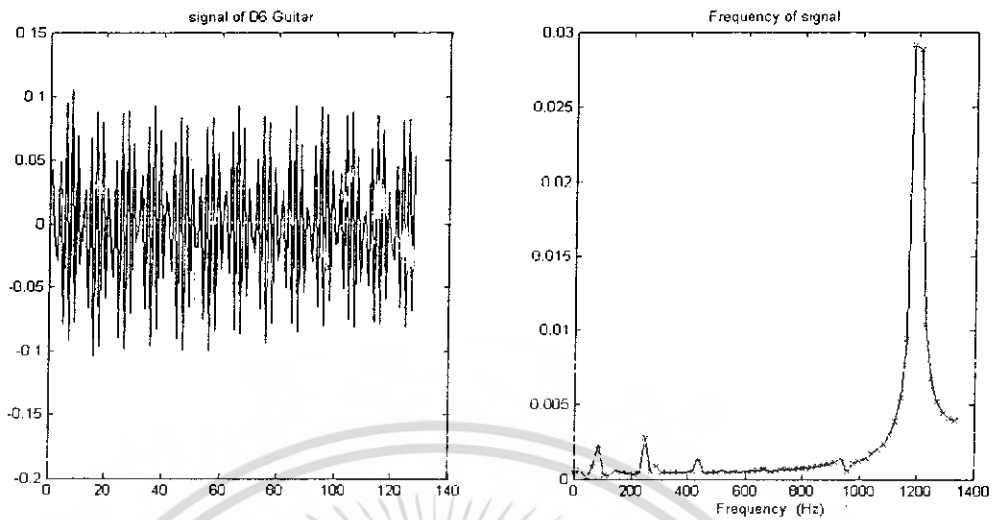
รูปที่ 6.1 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



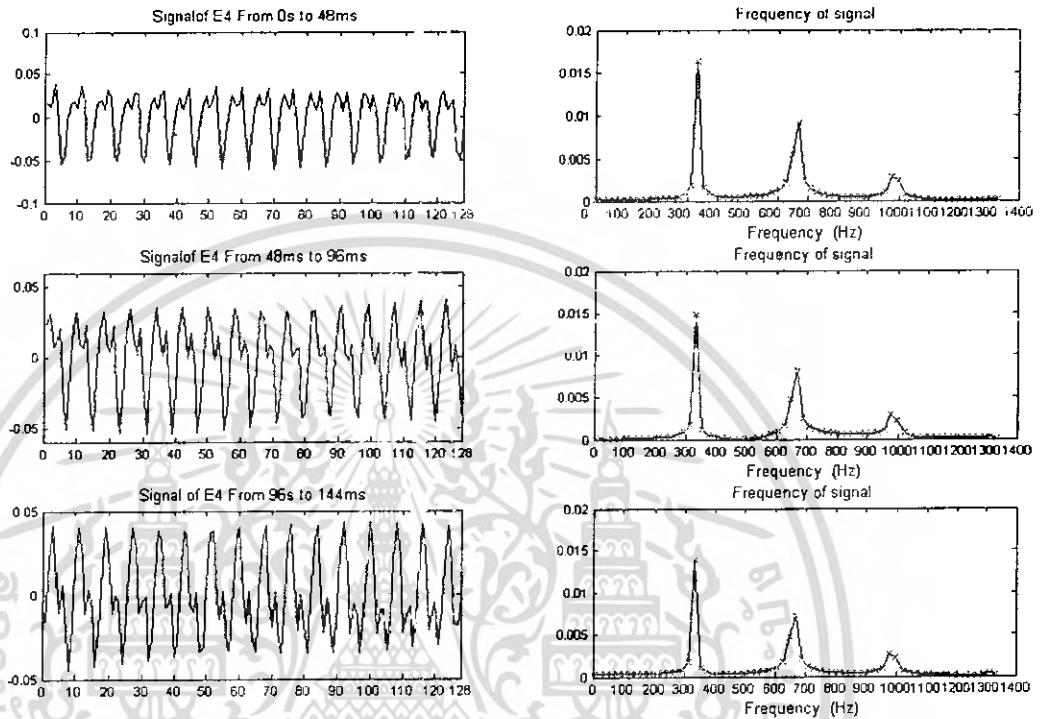
รูปที่ 6.1(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆ



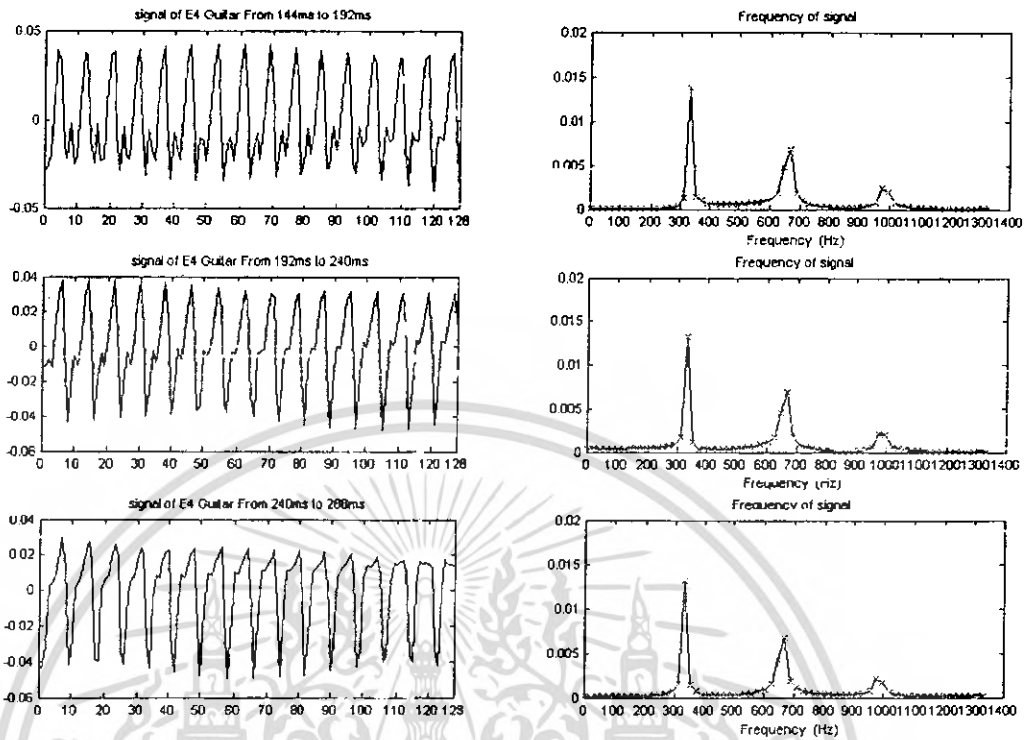
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 ค่าความถี่และแอมพลิจูดของสัญญาณกีตาร์ไฟฟ้าที่เวลาต่างๆ

ในการทดลองนี้เราพิจารณารูปของสัญญาณกีตาร์ไฟฟ้าที่เวลาต่างๆ โดยจะนำโน้ต E4 มาแบ่งเป็น 9 ช่วง ช่วงละ 128 sample ซึ่งแต่ละช่วงจะใช้เวลา 48 ms

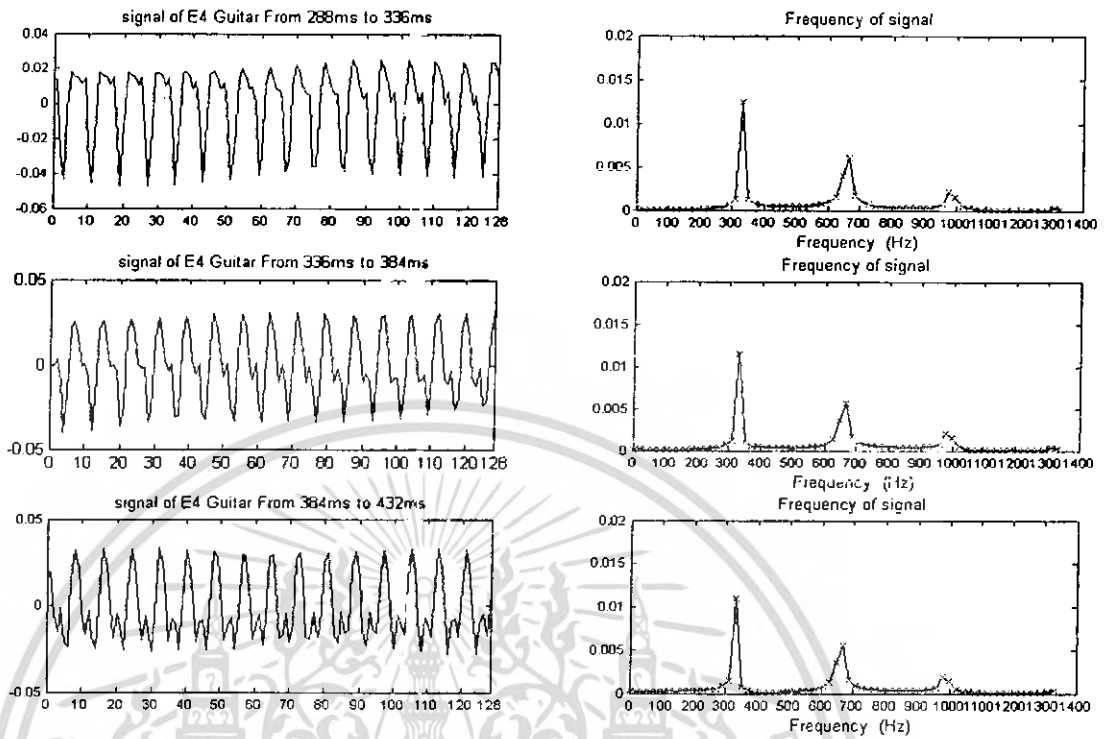


รูปที่ 6.2 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ต E4 ในช่วงเวลาต่างๆ



รูปที่ 6.2(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ต E4 ในช่วงเวลา
ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ต E4 ในช่วงเวลา
ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3 ผลที่เกิดขึ้นเมื่อนำสัญญาณไปขยายจนเกิดการขลิบ
ในการทดลองนี้เราจะนำไปขยายก่อนแล้วจึงนำไปคำนวณใน MATLAB

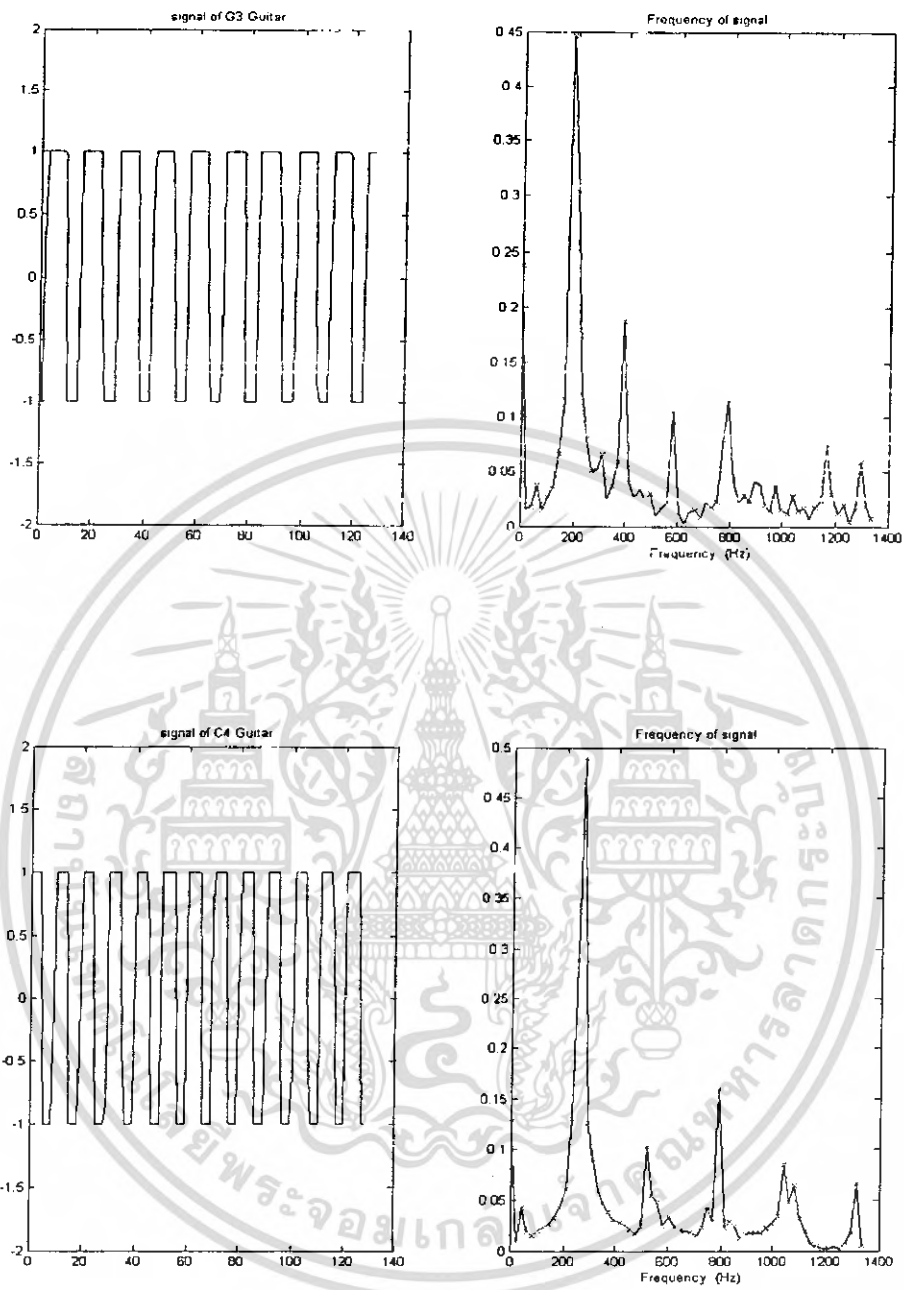
ตารางที่ 6.2 ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ตเมื่อสัญญาณถูกขลิบ

โน้ต	ความถี่ของ ตัวโน้ต (Hz)	ความถี่ที่วัด ได้ (Hz)	ความถี่ขบ ล่าง (Hz)	แอมพลิจูด ของขบ ล่าง	ความถี่ขบ บน (Hz)	แอมพลิจูด ของขบ บน
G3	199.9577	195.96	187.50	0.4470	208.33	0.3040
G#3	207.6523	208.98	208.33	0.5923	229.17	0.0177
A3	220.0000	217.45	208.33	0.3005	229.17	0.4639
A#3	233.0819	234.38	229.17	0.5582	250.00	0.1617
B3	246.9417	248.70	229.17	0.0493	250.00	0.6134
C4	261.6256	263.67	250.00	0.2888	270.83	0.4896
C#4	277.1826	278.65	270.83	0.4307	291.67	0.2373
D4	293.6648	294.92	270.83	0.0865	291.67	0.5653
D#4	311.1270	311.84	291.67	0.0223	312.5	0.5983
E4	329.6275	330.07	312.5	0.0935	333.33	0.4937
F4	349.2282	352.21	333.33	0.0108	354.17	0.4835
F#4	369.9944	371.09	354.17	0.0865	375.00	0.5599
G4	391.9954	391.28	375.00	0.1555	395.83	0.5709
G#4	415.3046	416.67	416.67	0.6049	437.50	0.0080
A4	440.0000	442.71	437.50	0.5546	458.33	0.2613
A#4	466.1637	466.80	458.33	0.4456	479.17	0.3459
B4	493.8833	494.14	479.17	0.2134	500.00	0.5507

ตารางที่ 6.2(ต่อ) ความถี่และแอมพลิจูดของสัญญาณแต่ละตัวโน้ตเมื่อสัญญาณถูกขลิบ

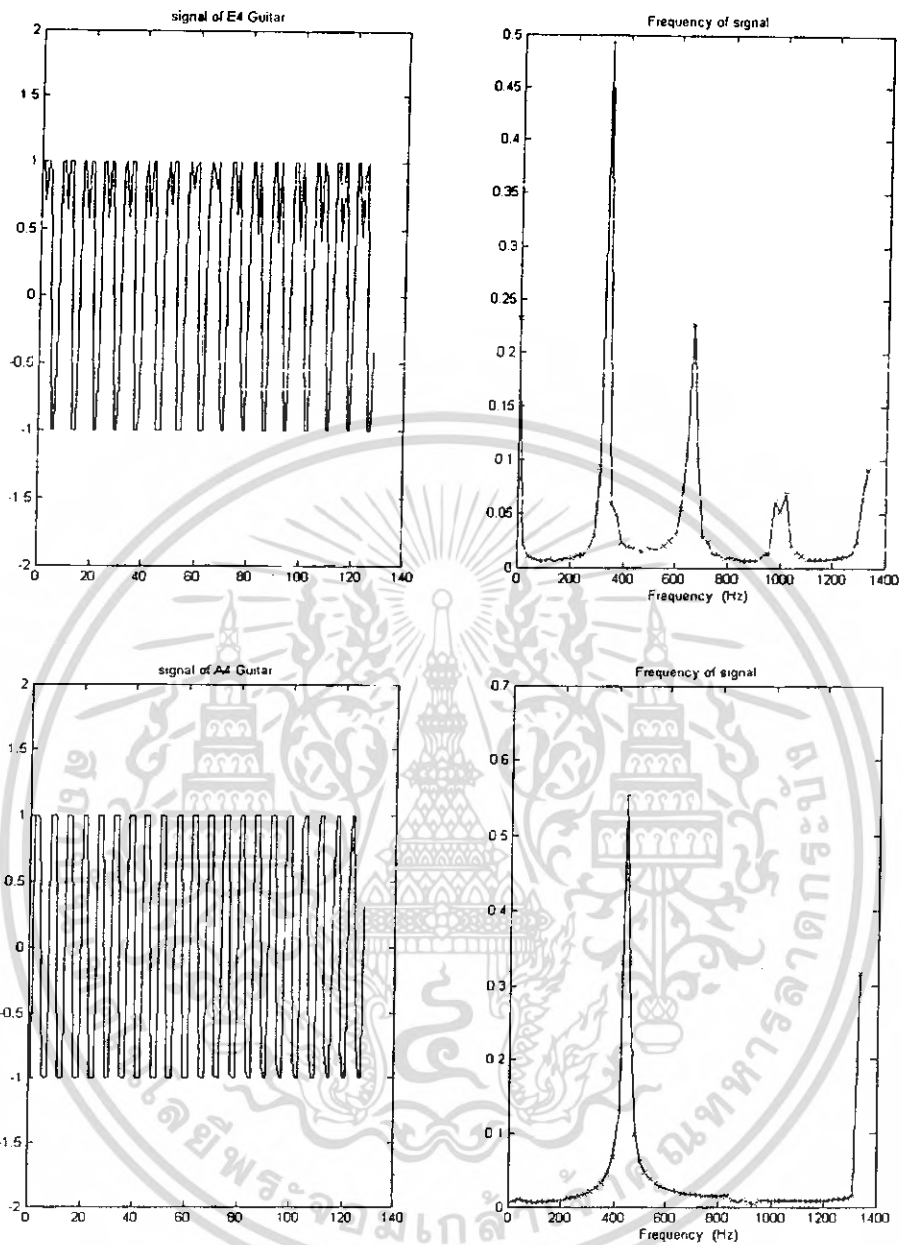
โน้ต	ความถี่ของ ตัวโน้ต (Hz)	ความถี่ที่วัด ได้ (Hz)	ความถี่ขอบ ล่าง (Hz)	แอมพลิจูด ของขอบ ล่าง	ความถี่ขอบ บน (Hz)	แอมพลิจูด ของขอบ บน
C5	523.2511	524.09	520.83	0.6015	541.67	0.1084
C#5	554.3652	553.39	541.67	0.3321	562.50	0.4612
D5	587.3295	585.29	583.33	0.5969	604.17	0.1269
D#5	622.2539	621.745	604.17	0.0677	625.00	0.6199
E5	659.2551	661.46	645.83	0.1286	666.66	0.6131
F5	698.4564	697.27	687.50	0.3869	708.33	0.4135
F#5	739.9888	734.37	729.16	0.5623	750.00	0.1933
G5	783.9908	784.51	770.83	0.2064	791.66	0.5551
G#5	830.6093	835.29	833.33	0.6042	854.16	0.1021
A5	880.0000	875.65	875.00	0.6158	895.83	0.0751
A#5	932.3275	932.29	916.66	0.4595	937.50	0.3352
B5	987.7666	976.56	979.16	0.6232	1000.00	0.0361
C6	1046.5022	1069.66	1062.50	0.4769	1083.33	0.3125
C#6	1108.7305	1107.42	1104.16	0.3000	1125.00	0.4916
D6	1174.6590	1165.89	1166.66	0.1230	1187.50	0.4111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



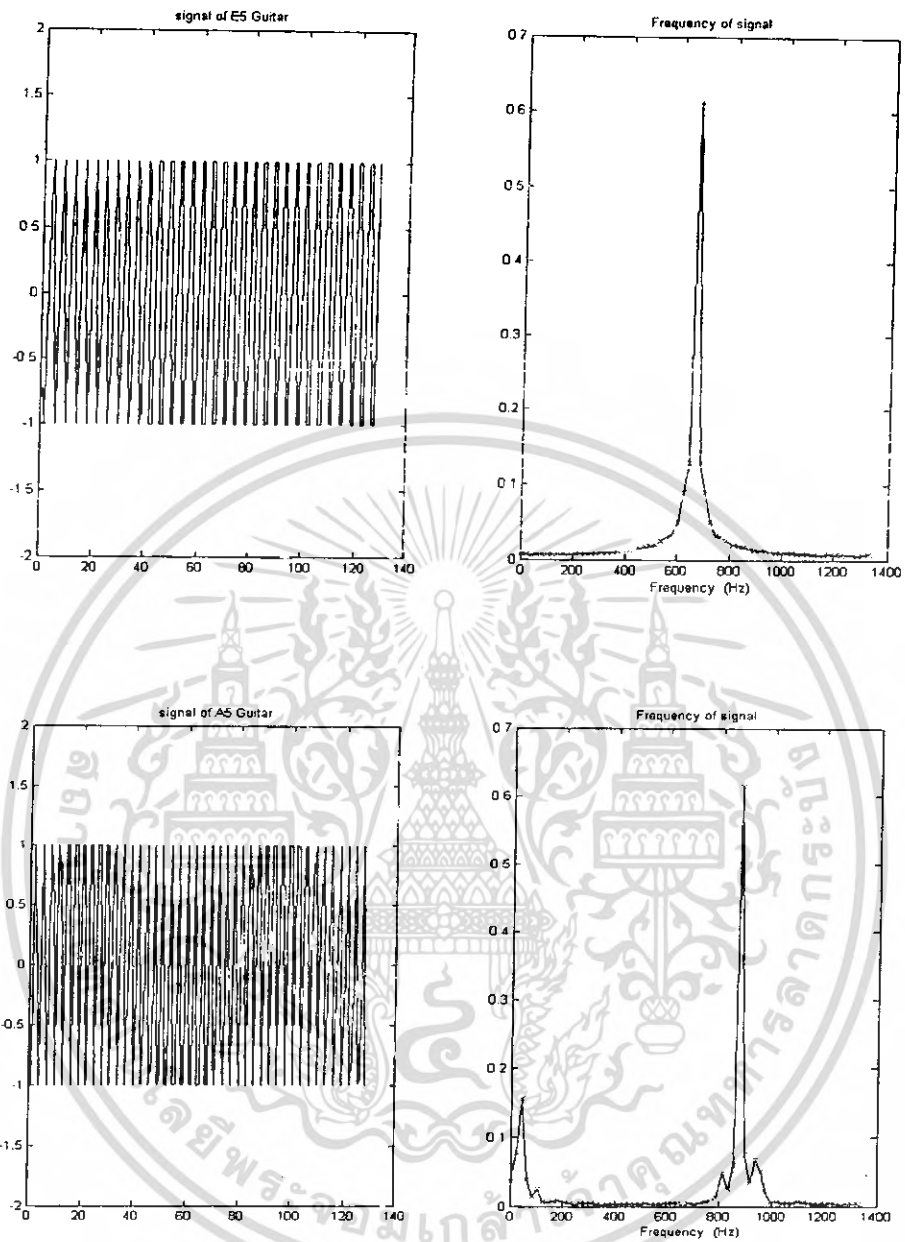
รูปที่ 6.3 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆเมื่อสัญญาณถูกขลิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



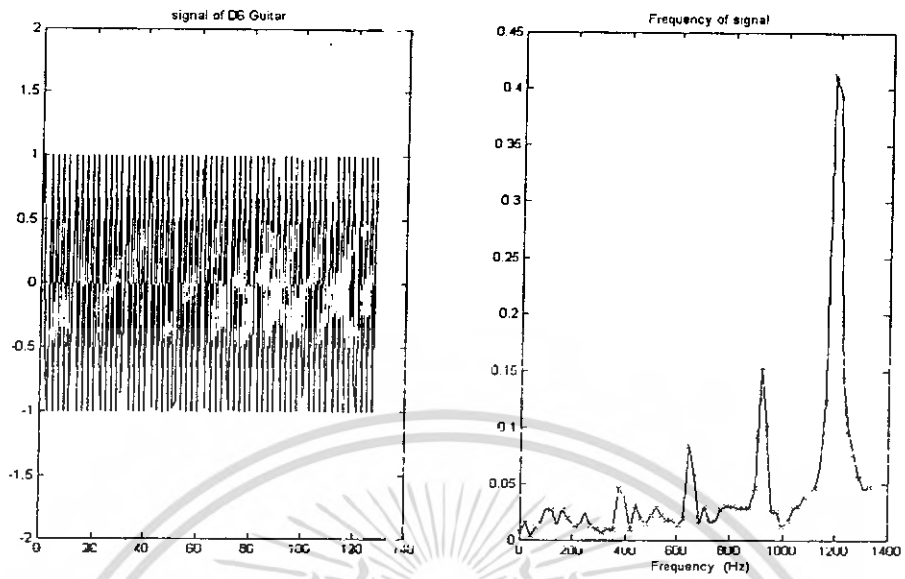
รูปที่ 6.3(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของคอร์ดโน้ตต่างๆเมื่อสัญญาณถูกขลิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆเมื่อสัญญาณถูก
ขลิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของตัวโน้ตต่างๆเมื่อสัญญาณถูก

ขลิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

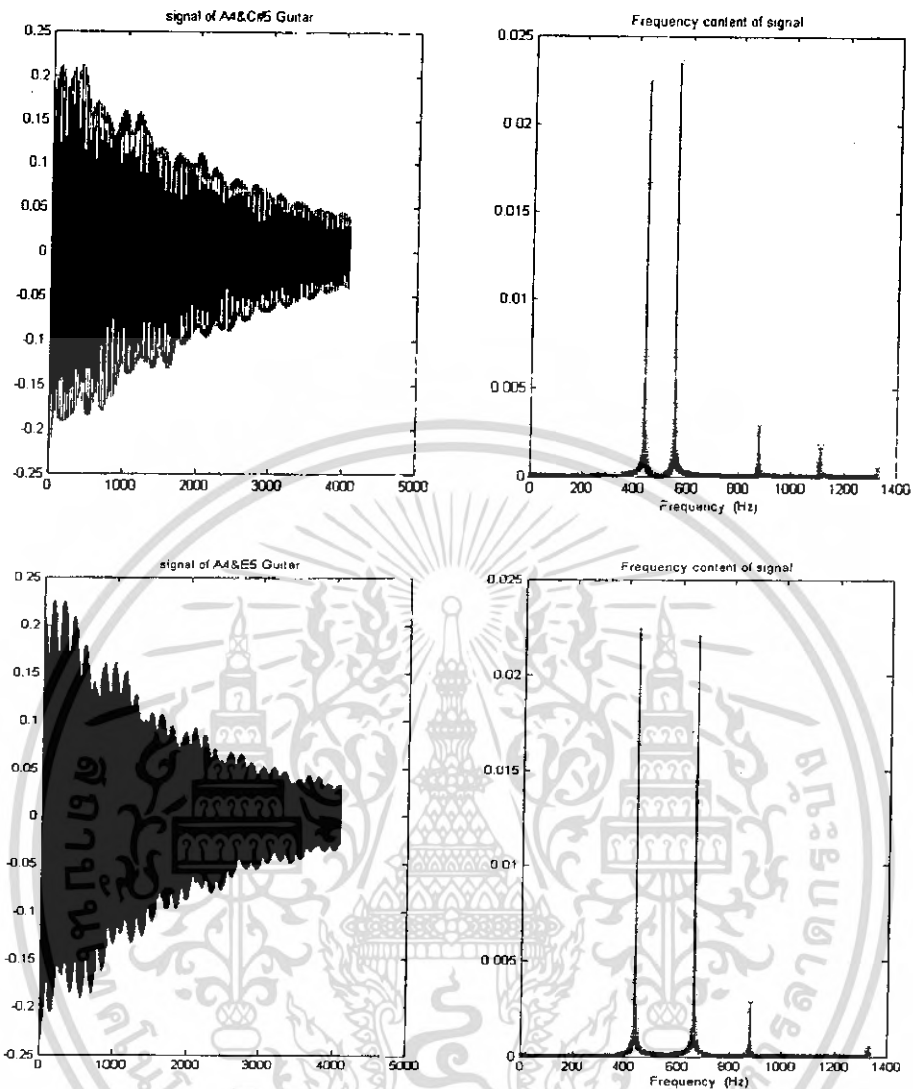
การทดลองที่ 4 ผลกระทบเมื่อเล่น 2 เสียงพร้อมกัน

ในการทดลองนี้เราจะบันทึกเสียงกีตาร์ไฟฟ้าพร้อมกัน 2 เสียงก่อนแล้วจึงนำไปคำนวณใน MATLAB

ตารางที่ 6.3 ความถี่ของสัญญาณแต่ละตัว โน้ตเมื่อเล่น 2 เสียงพร้อมกัน

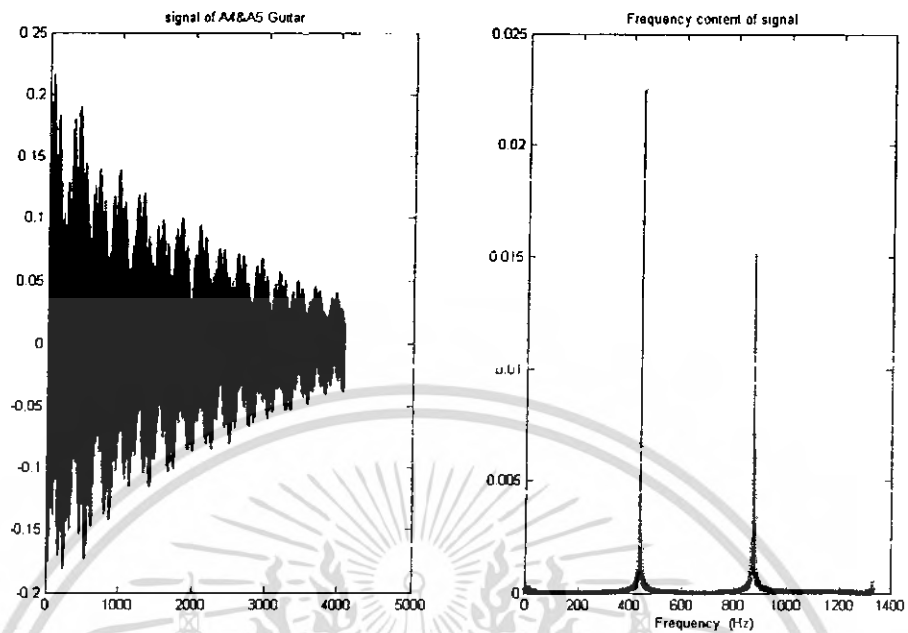
โน้ต	ชั้นคู่	ความถี่ของตัวโน้ต (Hz)	ความถี่ที่วัดได้ (Hz)
A4, B4	2	440.00, 493.88	494.14
A4, C#5	3	440.00, 554.37	553.39
A4, D5	4	440.00, 587.33	442.71
A4, E5	5	440.00, 659.26	442.71
A4, F#5	6	440.00, 739.99	442.71
A4, G#5	7	440.00, 830.61	442.71
A4, E5	8	440.00, 880.00	442.71





รูปที่ 6.4 กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของคู่ตัวโน้ตต่างๆเมื่อเล่นพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4(ต่อ) กราฟระหว่างรูปสัญญาณกับแอมพลิจูดของแต่ละความถี่ของคู่ตัวโน้ตต่างๆเมื่อเล่นพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5 การหาค่าแอมพลิจูดสูงสุดในแต่ละ Grid

การทดลองนี้จะทำการเก็บข้อมูลของแอมพลิจูดสูงสุด 5 ครั้งจากการคำนวณในไมโครคอนโทรลเลอร์ ในแต่ละ Grid เพื่อนำมาใช้ในการปรับค่าแอมพลิจูดของสัญญาณในแต่ละ Grid ให้อยู่ในสเกลเดียวกัน โดยการเก็บค่าแอมพลิจูดสูงสุด (วัดด้วยความแรงสูงสุดในการเล่นกีตาร์ไฟฟ้าโดยทั่วไป) จากนั้นก็ทำการเก็บค่า 5 ครั้ง แล้วจึงนำมาหาค่าเฉลี่ย จะได้ผลลัพธ์ดังตารางที่ 6.4

ตารางที่ 6.4 ค่าแอมพลิจูดเฉลี่ยสูงสุดของ Grid ที่ตรงกับคิ้วโน้ต

Note(Grid)	Amp1	Amp2	Amp3	Amp4	Amp5	Amp(เฉลี่ย)
E4(16)	9461	10121	7821	7651	11201	9261
F4(17)	1821	3251	3421	2691	3721	2981
F#4(18)	3621	5821	2981	3243	3651	3863
G4(19)	11714	10516	11275	12361	13473	11867
G#4(20)	9561	4321	6410	4851	5561	6140
A4(21)	10721	10763	8376	11706	11456	10604
A#4(23)	9907	9751	12607	10856	11751	10974
B4(24)	7521	6521	5728	8506	7856	7226
C5(25)	11321	7528	7458	6532	6478	7863
C#5(27)	18520	14251	8957	10587	16278	13718
D5(28)	17271	12352	10735	15262	19072	14938
D#5(30)	17265	13421	17521	17280	16547	16478
E5(32)	21057	18727	30053	29876	16520	23246
F5(34)	43712	41276	31257	46071	32976	39058
F#5(36)	12867	12872	12493	13025	14020	13055
G5(38)	12628	13452	12765	12553	11646	12608
G#5(40)	14542	10268	13506	16502	9961	12956
A5(43)	19207	22072	25026	24060	20556	22184
A#5(45)	13251	15472	12472	19074	13552	14764
B5(48)	11026	15075	13527	13175	10023	12691
C6(51)	14352	12627	10057	14578	12638	12850

ตารางที่ 6.4(ต่อ) ค่าแอมพลิจูดเฉลี่ยสูงสุดของ Grid ที่ตรงกับตัวโน้ต

C6#(54)	11206	13025	8525	13008	12523	11567
D6(57)	6752	5402	9161	2346	8312	6395
D#6(61)	3528	6502	3908	5862	7802	5520
E6(64)	15252	13520	15072	18062	13041	14989



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6 การหาค่าเฉลี่ยแอมป์ลิจูดที่น้อยที่สุด ของทุก Grid เพื่อใช้กำหนดค่า Threshold

การทดลองนี้จะทำการเก็บข้อมูลของแอมป์ลิจูดน้อยสุด 5 ครั้งจากการคำนวณในไมโครคอนโทรลเลอร์ (คิดด้วยความแรงต่ำสุดในการเล่นกีตาร์ไฟฟ้าโดยทั่วไป) แล้วนำค่าเฉลี่ยในแต่ละ Grid มาหารค่าแอมป์ลิจูดสูงสุดที่ได้จากการทดลองที่ 5 เพื่อปรับให้แอมป์ลิจูดเฉลี่ยน้อยที่สุด ในแต่ละ Grid อยู่บนสเกลเดียวกัน ก็จะได้ค่า Threshold ของแต่ละ Grid แล้วจึงนำ Threshold ที่ได้มาหาค่าเฉลี่ย ก็จะได้ Threshold high กลางที่ใช้เป็น Threshold ในการ On โน้ต ดังตารางที่ 6.5 ในแต่ละ Grid มาหา

ตารางที่ 6.5 ค่าแอมป์ลิจูดเฉลี่ยน้อยที่สุดของ Grid ที่ตรงกับตัวโน้ต

Note(Grid)	Amp1	Amp2	Amp3	Amp4	Amp5	Amp(เฉลี่ย)
E4(16)	1621	2451	2782	1091	2473	2083
F4(17)	2821	1019	1350	1642	1653	1697
F#4(18)	3528	2412	4202	4452	4091	3737
G4(19)	2827	1937	2472	1659	2157	2210
G#4(20)	2806	2752	2853	2972	2846	2846
A4(21)	2542	1972	3572	2694	1529	2466
A#4(23)	1134	1392	1387	2022	2225	1632
B4(24)	2324	2426	1748	1753	1592	1969
C5(25)	2047	2597	2892	2452	2022	2402
C#5(27)	1721	1052	1043	1459	1096	1274
D5(28)	1472	1471	1857	1279	1250	1466
D#5(30)	2003	1372	1625	2021	1527	1710
E5(32)	1647	1052	1539	1023	1657	1384
F5(34)	1555	2122	2577	1657	2542	2091
F#5(36)	1741	1825	2320	1195	1199	1652
G5(38)	1428	1728	1325	1018	1054	1311
G#5(40)	2295	1994	2274	1254	1234	1811
A5(43)	1515	1819	1144	1219	1182	1376
A#5(45)	3357	5564	2726	1782	3352	3356
B5(48)	2872	1751	3728	2524	2612	2697
C6(51)	1082	1602	1127	1543	1828	1436

ตารางที่ 6.5(ต่อ) ค่าแอมพลิจูดเฉลี่ยน้อยที่สุดของ Grid ที่ตรงกับตัวโน้ต

C6#(54)	802	1069	1024	1082	1502	1096
D6(57)	1480	777	635	982	1272	1029
D#6(61)	407	735	250	532	990	583
E6(64)	833	1031	892	685	658	820



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.6 การหาค่า Threshold high

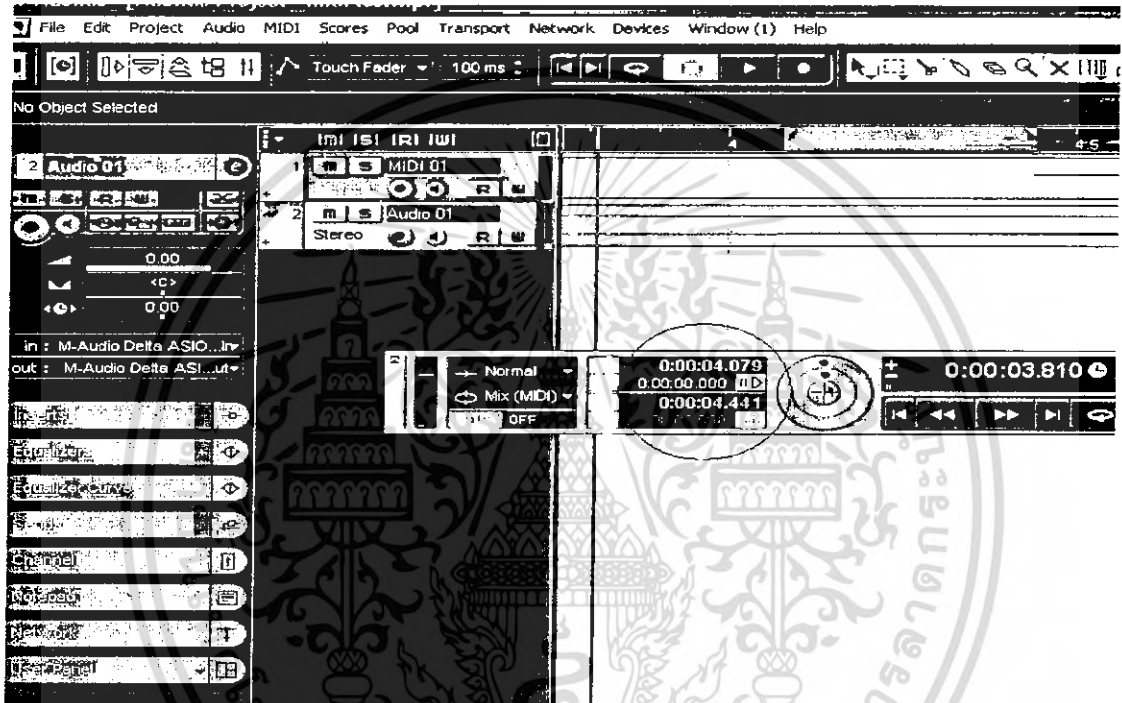
Note(Grid)	Amp(min)	Amp(Peak)	Min/Peak
E4(16)	2083	9261	0.2249
F4(17)	1697	2981	0.5693
F#4(18)	3737	3863	0.9658
G4(19)	2210	11867	0.1862
G#4(20)	2846	6140	0.4635
A4(21)	2466	10604	0.2326
A#4(23)	1632	10974	0.1487
B4(24)	1969	7226	0.2725
C5(25)	2402	7863	0.3055
C#5(27)	1274	13718	0.0929
D5(28)	1466	14938	0.0981
D#5(30)	1710	16478	0.1038
E5(32)	1384	23246	0.0595
F5(34)	2091	39058	0.0535
F#5(36)	1652	13055	0.1265
G5(38)	1311	12608	0.1039
G#5(40)	1811	12956	0.1398
A5(43)	1376	22184	0.0620
A#5(45)	3356	14764	0.2273
B5(48)	2697	12691	0.2079
C6(51)	1436	12850	0.1118
C#6(54)	1096	11567	0.0940
D6(57)	1029	6395	0.1609
D#6(61)	583	5520	0.1056
E6(64)	820	14989	0.0547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7 การหาค่า Delay Time และเปรียบเทียบสัญญาณ Input กับ Output

ทำการป้อนสัญญาณ Input ด้วยการเล่นกีตาร์ไฟฟ้าตั้งแต่เข้าสู่วงจร แล้วป้อน Output ในรูปแบบมิดีไปบันทึกเป็นรูปแบบมิดีและบันทึกสัญญาณอนาล็อกในรูปแบบเวฟไฟล์ไปพร้อมๆกันในซอฟต์แวร์บันทึกเสียง โดยในโปรแกรมที่ใช้จะมีสเกลเวลาบอกเวลา ทำให้สามารถหาค่า Delay Time ระหว่างสัญญาณอนาล็อก กับค่ามิดีที่บันทึกได้ ดังรูปที่ 6.5

โดยจะมีค่า Delay Time ประมาณ $4.441 - 4.079 = 362 \text{ ms}$



รูปที่ 6.5 แสดงระยะเวลาห่างของเวลาในการบันทึกสัญญาณอนาล็อกกับสัญญาณมิดี

บทที่ 7

สรุปผลการทดลอง

จากการทดลองที่ 1 เราได้หาแอมพลิจูดของแต่ละความถี่ของแต่ละสัญญาณของแต่ละตัวโน้ต ซึ่งเราได้ลดความถี่การแซมปลิง จาก 96,000 Hz ให้เหลือ 2666.67 Hz โดยใช้จำนวนการสุ่ม 128 ค่า ดังนั้นในแต่ละค่าของการสุ่ม ความถี่จะห่างกัน $2666.67/128 = 20.83$ Hz ซึ่งช่องว่างของความถี่นี้เพียงพอต่อการแยกระดับเสียงในแต่ละตัวโน้ตของสาย 1 ได้ แต่จะมีปัญหาในสาย 2 และ สาย 3 เนื่องจากสาย 2 และ สาย 3 อยู่ในช่วงความถี่ที่ต่ำ และแต่ละตัวโน้ตในช่วงความถี่นี้ จะมีระยะห่างของความถี่ของโน้ตแต่ละตัว น้อยกว่า 20.83 Hz ดังนั้นในช่วงความถี่ต่ำ อาจจะมีการติดเพี้ยนของแอมพลิจูดของตัวโน้ต เนื่องจากความละเอียดของการสุ่มไม่เพียงพอต่อความห่างของความถี่ในช่วงนี้ ทำให้ความถี่ของขอบบนและขอบล่างในช่วงความถี่ต่ำ เกิดความผิดพลาด จะเห็นได้จากผลการทดลองในตารางที่ 6.1 ว่า ความถี่ขอบบนและขอบล่างของตัวโน้ต G3 ถึง D4 มีค่าผิดไปจากค่าที่ควรจะเป็น เช่น ความถี่ขอบบนของตัวโน้ต A#3 มีค่าเท่ากับ 250 Hz ซึ่งมีค่ามากกว่า ความถี่ของตัวโน้ต B3 เป็นต้น ซึ่งถ้าเราต้องการให้สามารถแยกความถี่ของสาย 2 และ สาย 3 เราอาจต้องใช้จำนวนแซมปลิงมากกว่า 128 ค่า

จากการทดลองที่ 2 เมื่อเรานำสัญญาณกีตาร์ไฟฟ้าที่เวลาต่างๆของโน้ต E4 มาแบ่งเป็น 9 ช่วง ช่วงละ 128 sample ซึ่งแต่ละช่วงจะใช้เวลา 48 ms พบว่าเมื่อเวลาผ่านไป ความถี่ Harmonic จะมีพลังงานลดลงเรื่อยๆ ในขณะที่ความถี่ Fundamental จะมีพลังงานค่อนข้างคงที่ ดังนั้น หากมีการเลือกช่วงเวลาในการแซมปลิงที่เหมาะสม จะสามารถแก้ปัญหาในเรื่องของสัญญาณที่มีความถี่ Harmonic ที่อาจมีพลังงานมากกว่าพลังงานของความถี่ Fundamental ได้

จากการทดลองที่ 3 เมื่อเรานำสัญญาณไปขยายจนเกิดการขลิบของสัญญาณ เมื่อเรานำไปคำนวณโดย MATLAB พบว่ามีความถี่ Harmonic ออกมามากขึ้น เนื่องจากสัญญาณที่นำมาขยายนั้น ถูกคูล แต่ความถี่ Fundamental จะมีพลังงานมากกว่าความถี่ Harmonic มากขึ้น ดังนั้นถ้าเรานำสัญญาณไปขยายให้เหมาะสมก่อน ก่อนนำไปคำนวณจะทำให้สามารถหาความถี่ได้ถูกต้องมากขึ้น

จากการทดลองที่ 4 เมื่อเรานำกีตาร์มาเล่น 2 เสียง พร้อมกัน แล้วนำสัญญาณที่มีตัวโน้ต 2 ตัว ไปหาความถี่ พบว่าความถี่ที่ได้มีค่าเข้าใกล้ความถี่ของตัวโน้ตใดตัวโน้ตหนึ่ง ซึ่งความถี่ที่ได้นั้น เป็นความถี่ของตัวโน้ตที่มีพลังงานสูงกว่า ดังนั้นเราไม่สามารถที่จะหาความถี่ของตัวโน้ตที่เล่นพร้อมกันมากกว่า 1 เสียงได้

จากการทดลองที่ 5 ทำการคำนวณหาค่าแอมพลิจูดสูงสุดของแต่ละตัวโน้ตโดยใช้ไมโครคอนโทรลเลอร์คำนวณและแสดงผลออกทางจอ LCD ทำให้สามารถแยกแยะโน้ตที่มีแอมพลิจูดสูงสุดได้โดยง่าย

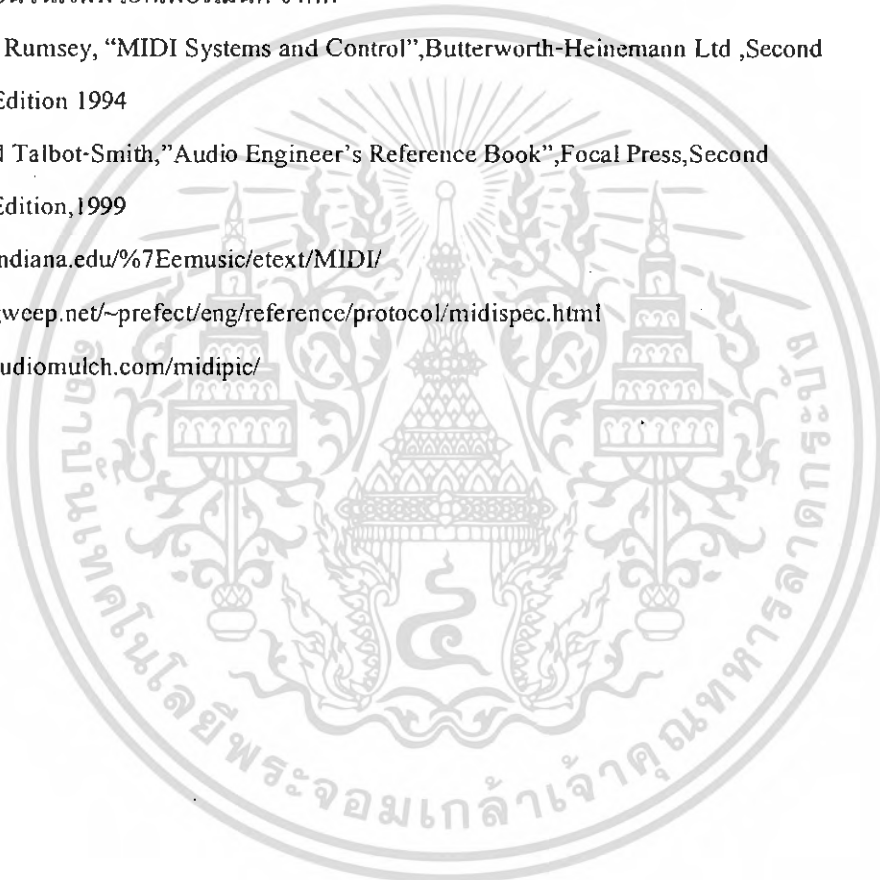
จากการทดลองที่ 6 ทำการคำนวณหาค่าแอมพลิจูดค่าสุดของแต่ละตัวโน้ตโดยใช้ไมโครคอนโทรลเลอร์คำนวณและแสดงผลออกทางจอ LCD แล้วนำค่าเฉลี่ยในแต่ละตัวโน้ตที่ได้มาหาค่าเฉลี่ยที่ได้จากการทดลองที่ 5 ก็จะได้ค่า Threshold ของแต่ละตัวโน้ตเพื่อที่จะนำไปใช้ในการสั่ง on โน้ตต่อไป

จากการทดลองที่ 7 เราจะพบว่าไมโครคอนโทรลเลอร์จะใช้เวลาในการประมวลผลทั้งหมดประมาณ 362ms ซึ่งเกิดจากการ Sampling ด้วยเวลา $0.375 \times 48 = 48$ ms ส่วนที่เหลือนั้น เป็นการใช้เวลาในการคำนวณในฟังก์ชันฟาสต์ฟูเรียร์ทรานฟอร์มและการส่งสัญญาณมีดี



บรรณานุกรม

- 1.ศาสตราจารย์ ดร.วิมลภ สุระกำพลธร, วิเชียร เลหา โกศล “ทฤษฎีเบื้องต้น การแปลงฟาสท์ฟูริเยร์”
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง พิมพ์ครั้งที่ 1 พ.ศ.2527
- 2.รองศาสตราจารย์ เอก ไชยสวัสดิ์ “สัญญาณและระบบ” พิมพ์ครั้งที่ 2 พ.ศ.2543
- 3.นคร ภัคดีชาติ, ฌัฐพล วงศ์สุนทรชัย, ชัยวัฒน์ ลัมพรจิตรวิไล “คู่มือการทดลอง dsPIC
Microcontroller เบื้องต้น ด้วยโปรแกรมภาษา C กับ MPLAB C30” บริษัท
อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด
- 4.Francis Rumsey, “MIDI Systems and Control”, Butterworth-Heinemann Ltd ,Second
Edition 1994
- 5.Michael Talbot-Smith, “Audio Engineer’s Reference Book”, Focal Press, Second
Edition, 1999
6. www.indiana.edu/%7Eemusic/text/MIDI/
7. www.gweep.net/~prefect/eng/reference/protocol/midispec.html
8. www.audiomulch.com/midipic/





ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Program ที่ใช้ในการออกแบบ

```
#include <p30f4011.h>
```

```
#include <lcd4.h>
```

```
#include <adc10.h>
```

```
#include <timer.h>
```

```
#include <uart.h>
```

```
#include <ports.h>
```

```
#include <math.h>
```

```
int Sw1,Sw11,Sw2,Sw21,Sw22,Sw23,Sw31,Sw41,Sw5x,Sw52,Skb1,Skb2,Skb3,Swp,Swp,index,
```

```
float xr[128],xj[128],AmpPeak;
```

```
int result;
```

```
int i0,i1,i2,i3,i4,i5,i6,k0,m0;
```

```
float temp;
```

```
unsigned int b,b2,i,k,km,wp,wpp,j,jm,counter;
```

```
float wr[64] = { 1.000,0.999,0.995,0.989,0.981,0.970,0.957,0.942,
                0.924,0.904,0.882,0.858,0.831,0.803,0.773,0.741,
                0.707,0.672,0.634,0.596,0.556,0.514,0.471,0.428,
                0.383,0.337,0.290,0.243,0.195,0.147,0.098,0.049,
                0.000,-0.049,-0.098,-0.147,-0.195,-0.243,-0.290,-0.337,
                -0.383,-0.428,-0.471,-0.514,-0.556,-0.596,-0.634,-0.672,
                -0.707,-0.741,-0.773,-0.803,-0.831,-0.858,-0.882,-0.904,
                -0.924,-0.942,-0.957,-0.970,-0.981,-0.989,-0.995,-0.999},
                wj[64] = { 0.000,-0.049,-0.098,-0.147,-0.195,-0.243,-0.290,-0.337,
                -0.383,-0.428,-0.471,-0.514,-0.556,-0.596,-0.634,-0.672,
                -0.707,-0.741,-0.773,-0.803,-0.831,-0.858,-0.882,-0.904,
                -0.924,-0.942,-0.957,-0.970,-0.981,-0.989,-0.995,-0.999,
                -1.000,-0.999,-0.995,-0.989,-0.981,-0.970,-0.957,-0.942,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

-0.924,-0.904,-0.882,-0.858,-0.831,-0.803,-0.773,-0.741,
-0.707,-0.672,-0.634,-0.596,-0.556,-0.514,-0.471,-0.428,
-0.383,-0.337,-0.290,-0.243,-0.195,-0.147,-0.098,-0.049}
,Ar,Aj,Br,Bj,wjxr,wjxj,wrxr,wrxj,xar,xaj,xbr,xbj:

float xar,xai,xbi,xbr;

unsigned int GridPeak,Amp,GP,w;

int ia[25]={16,17,18,19,20,21,23,24,25,27,28,30,32,
34,36,38,40,43,45,48,51,54,57,61,64},ib,OldGrid,NewGrid,iGridPeak,ja;

//int ib;

float Peak[25] = {2083,1697,3707,2210,2846,2466,1632,1969,2402,1274,1466,1710,1384,
2091,1652,1311,1811,1376,3356,2697,1436,1096,1029,583,820},TL;

void adc_init(void)
{
    unsigned int Channel, PinConfig, Scanselect, Adcon3_reg, Adcon2_reg, Adcon1_reg;

    ADCON1bits.ADON = 0;

    Channel = ADC_CH0_POS_SAMPLEA_AN0 &
              ADC_CH0_NEG_SAMPLEA_NVREF ;

    SetChanADC10(Channel);

    ConfigIntADC10(ADC_INT_DISABLE);

```

```

PinConfig      =    ENABLE_AN0_ANA ;

Scanselect    =    SKIP_SCAN_AN1 &
                    SKIP_SCAN_AN2 &
                    SKIP_SCAN_AN3 &
                    SKIP_SCAN_AN4 &
                    SKIP_SCAN_AN5 &
                    SKIP_SCAN_AN6 &
                    SKIP_SCAN_AN7;

Adcon3_reg    =    ADC_SAMPLE_TIME_10 &
                    ADC_CONV_CLK_INTERNAL_RC &
                    ADC_CONV_CLK_13Tcy;

Adcon2_reg    =    ADC_VREF_AVDD_AVSS &
                    ADC_SCAN_ON &
                    ADC_ALT_BUF_OFF &
                    ADC_ALT_INPUT_OFF &
                    ADC_CONVERT_CH0 &
                    ADC_SAMPLES_PER_INT_8;

Adcon1_reg    =    ADC_MODULE_ON &
                    ADC_IDLE_CONTINUE &
                    ADC_FORMAT_INTG &
                    ADC_CLK_MANUAL &
                    ADC_SAMPLE_SIMULTANEOUS &
                    ADC_AUTO_SAMPLING_ON;

OpenADC10(Adcon1_reg, Adcon2_reg, Adcon3_reg, PinConfig, Scanselect);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void uart1_init()
{
    unsigned int baudvalue;
    unsigned int UIMODEvalue;
    unsigned int UISTAvalue;

    CloseUART1();
    ConfigIntUART1(UART_RX_INT_EN &
                   UART_RX_INT_PRI &
                   UART_TX_INT_EN &
                   UART_TX_INT_PRI);

    baudvalue = 39;

    UIMODEvalue = UART_EN &
                  UART_IDLE_CON &
                  UART_RX_TX &
                  UART_DIS_WAKE &
                  UART_DIS_LOOPBACK &
                  UART_DIS_ABAUD &
                  UART_NO_PAR_8BIT &
                  UART_1STOPBIT;

    UISTAvalue = UART_INT_TX_BUF_EMPTY &
                  UART_TX_PIN_NORMAL &
                  UART_TX_ENABLE &
                  UART_INT_RX_3_4_FUL &
                  UART_ADR_DETECT_DIS &
                  UART_RX_OVERRUN_CLEAR;

```

```

    OpenUART1(U1MODEvalue, U1STAValue, baudvalue);
}

void timer1_init(void)
{
    unsigned int match_value;

    ConfigIntTimer1(T1_INT_PRIOR_6 & T1_INT_ON);

    WriteTimer1(0);
    match_value = 937;

    OpenTimer1(T1_ON &
               T1_GATE_OFF &
               T1_IDLE_STOP &
               T1_PS_1_1 &
               T1_SYNC_EXT_OFF &
               T1_SOURCE_INT, match_value);
}

void SendUART(unsigned int Status,unsigned int data1,unsigned int data2)
{
    WriteUART1(Status+Sw11);
    while(BusyUART1());
    WriteUART1(data1);
    while(BusyUART1());
    if(data2<0x80)
    {
        WriteUART1(data2);
        while(BusyUART1());
    }
}

```

```

    }
}

void inttolcdx(unsigned char addr,long value,long base)

```

```

{
    unsigned int bufdiv[7],i=0;
    unsigned char bufchar[8];
    for(i=0;i<7;i++)
    {
        bufdiv[i] = value%base,
        value = value/base;
        bufchar[6-i] = bufchars[bufdiv[i]];
    }
    for(i=0;i<3;i++)
    {
        lcd_text(0x20);
    }
    bufchar[7] = 0x00;
    lcd_puts(addr,bufchar);
}

```

```

void BitReverse(void)
{
    unsigned int ii,jj,kk,bb,bb2;
    float xt;
    for(ii=7;ii>=1;ii--)
    {
        bb = 1;
        for(jj=1;jj<=ii;jj++)
        {
            bb = bb*2;

```

```

    }
    bb2 = bb/2-1;
    for(jj=1;jj<=bb2;jj++)
    {
        kk = jj*2;
        xt = xr[jj];
        xr[jj] = xr[kk];
        xr[kk] = xt;
    }
}
}

void Fast_Fourier(void)
{
    jm = 127;
    for(i=1;i<=7;i++)
    {
        b = 1;
        for(j=1;j<=i;j++)
        {
            b = b*2;
        }
        b2 = b/2;
        km = b2-1;
        wpp = (128/b)%(128/2);
        for(j=0;j<=jm;j+=b)
        {
            wp = 0;
            for(k=0;k<=km;k++)
            {
                xar = (xr[j+k+b2]*wr[wp])-(xj[j+k+b2]*wj[wp]);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    xaj      = (xr[j+k+b2]*wj[wp])+(xj[j+k+b2]*wr[wp]);
    xr[j+k+b2] := xar;
    xj[j+k+b2] := xaj;
    xar = xr[j+k] + xr[j+k+b2];
    xaj = xj[j+k] + xj[j+k+b2];
    xbr = xr[j+k] - xr[j+k+b2];
    xbj = xj[j+k] - xj[j+k+b2];
    xr[j+k] := xar;
    xj[j+k] := xaj;
    xr[j+k+b2] = xbr;
    xj[j+k+b2] = xbj;
    wp = wp + wpp;
}
}
}

void SizeofXr(void)
{
    for(i=0;i<=127;i++)
    {
        xar = xr[i];
        xai = -xj[i];
        xbr = (xr[i]*xar)-(xj[i]*xai);
        xbi = (xr[i]*xai)+(xj[i]*xar);
        xr[i] = fabsf(xbr);
        xj[i] = fabsf(xbi);
    }
}

void SelectGrid(void)

```

```

{
    AmpPeak = 0;
    for(ja=0;ja<25;ja++)
    {
        if(xr[ia[ja]]>AmpPeak)
        {
            GP = ja;
            GridPeak = ia[ja];
            AmpPeak = xr[GridPeak];
        }
    }
    Amp = (long)AmpPeak;
}

void AmpDivPeak(void)
{
    AmpPeak = 0;
    for(ib=0;ib<25;ib++)
    {
        xr[ia[ib]] = xr[ia[ib]]/Peak[ib];
        if(xr[ia[ib]]>AmpPeak)
        {
            TL = AmpPeak;
            GridPeak = ia[ib];
            iGridPeak = ib;
            AmpPeak = xr[GridPeak];
        }
    }
}

void SelectNote(void)

```

```

{
    float TH[25] = {2083,1697,3707,2210,2846,2466,1632,1969,2402,1274,1466,1710,1384,
                    2091,1652,1311,1811,1376,3356,2697,1436,1096,1029,583,820};

    if(AmpPeak>=TH[GP])
    {
        if(OldGrid!=GP)
        {
            SendUART(0x80,(unsigned char)(76+GP),0x64);
        }
        SendUART(0x90,(unsigned char)(76+GP),0x64);
        w = 1;
        OldGrid = GP;
    }
    else
    {
        w = 2;
        if(AmpPeak<20)
        {
            SendUART(0x80,(unsigned char)(76+GP),0x64);
            GP = 25;
            w = 3;
        }
    }
}
}

```

```

void ShowWork(void)

```

```

{
    if(GP<25)
    {
        inttolcd(0xC0,76+GP,10);
        inttolcd(0x8D,result,10);
    }
}

```

```

        lcd_puts(0x87,"  ");
        inttolcd(0xC9,w,10);
        lcd_puts(0xCA,"  ");
        inttolcd(0xCD,GridPeak,10);
        inttolcdx(0x80,Amp,10);
    }
    else
    {
        lcd_puts(0x80,"  It's no note");
        lcd_puts(0xC0,"  ");
    }
}

void ShowNote(void)
{
    lcd_puts(0x80,"  note");
    switch(GP)
    {
        case 0:
            lcd_puts(0xC0,"  E4");
            break;
        case 1:
            lcd_puts(0xC0,"  F4");
            break;
        case 2:
            lcd_puts(0xC0,"  F#4");
            break;
        case 3:
            lcd_puts(0xC0,"  G4");
            break;
        case 4:

```

```
        lcd_puts(0xC0,"      G#4");
        break;
case 5:
        lcd_puts(0xC0,"      A4");
        break;
case 6:
        lcd_puts(0xC0,"      A#4");
        break;
case 7:
        lcd_puts(0xC0,"      B4");
        break;
case 8:
        lcd_puts(0xC0,"      C5");
        break;
case 9:
        lcd_puts(0xC0,"      C#5");
        break;
case 10:
        lcd_puts(0xC0,"      D5");
        break;
case 11:
        lcd_puts(0xC0,"      D#5");
        break;
case 12:
        lcd_puts(0xC0,"      E5");
        break;
case 13:
        lcd_puts(0xC0,"      F5");
        break;
case 14:
        lcd_puts(0xC0,"      F#5");
```

```

        break;
case 15:
    lcd_puts(0xC0,"    G5");
    break;
case 16:
    lcd_puts(0xC0,"    G#5");
    break;
case 17:
    lcd_puts(0xC0,"    A5");
    break;
case 18:
    lcd_puts(0xC0,"    A#5");
    break;
case 19:
    lcd_puts(0xC0,"    B5");
    break;
case 20:
    lcd_puts(0xC0,"    C6");
    break;
case 21:
    lcd_puts(0xC0,"    C#6");
    break;
case 22:
    lcd_puts(0xC0,"    D6");
    break;
case 23:
    lcd_puts(0xC0,"    D#6");
    break;
case 24:
    lcd_puts(0xC0,"    E6");
    break;

```

```

default:
    lcd_puts(0xC0,"  No note");
    break;
}
}

void ShowLCD(void)
{
    if(LATDbits.LATD1==0)
    {
        ShowNote();
    }
    else if(LATDbits.LATD1==1)
    {
        ShowWork();
    }
}

void _ISR_U1TXInterrupt(void)
{
    IFS0bits.U1TXIF = 0;
}

void _ISR_U1RXInterrupt(void)
{
    IFS0bits.U1RXIF = 0;
}

void _ISR_INT2Interrupt(void)
{
    delay_ms(200);
}

```

```

LATDbits.LATD1 = !LATDbits.LATD1;
IFS1bits.INT2IF = 0;
}

```

```

void _ISR_T1Interrupt(void)

```

```

{
    ADCON1bits.SAMP = 1;
    while(!ADCON1bits.SAMP);
    ConvertADC10();
    while(ADCON1bits.SAMP);
    while(BusyADC10());
    if((index%8)==0)
    {
        result = ReadADC10(0);
        xr[index/8] = ((float)result)/64;
    }
    if(index >= 1016)
    {
        TRISDbits.TRISD2 = !TRISDbits.TRISD2;
        index = 0;
        BitReverse();
        Fast_Fourier();
        SizeofXr();
        SelectGrid();
        SelectNote();
        ShowLCD();
    }
    else
    {
        index++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IFS0bits.T1IF = 0;
}

int main(void)
{
    adc_init();
    uart1_init();
    lcd_init();
    lcd_clear();

    TRISEbits.TRISE8 = 1;
    TRISDbits.TRISD0 = 1;
    TRISDbits.TRISD1 = 1;

    index = 0;
    ConfigINT2(RISING_EDGE_INT & EXT_INT_ENABLE & EXT_INT_PRI_7);
    timer1_init();
    while(1);
    return 0;
}
```