ออดิโอ้ วิชวลไลเซชั่น
Audio Visualization

จุติพัฒน์ ฉัตรสมสง่า
JUTIPHAT CHATSOMSANGA
ชวกร ประเสริฐสุโข
CHAWAKORN PRASERTSUKO
ธนวัฒน์ สมโนชัย
THANAWAT SOMNOCHAI

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ออดิโอ้ วิชวลไลเซชั่น

Audio Visualization

จุติพัฒน์ ฉัตรสมสง่า
JUTIPHAT CHATSOMSANGA
ชวกร ประเสริฐสุโข
CHAWAKORN PRASERTSUKO
ธนวัฒน์ สมโนชัย
THANAWAT SOMNOCHAI

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Audio Visualization

JUTIPHAT CHATSOMSANGA

CHAWAKORN PRASERTSUKO

THANAWAT SOMNOCHAI

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ออดิโอ วิชวลไลเซชั่น	
รายชื่อนักศึกษา	นายจุติพัฒน์ ฉัตรสมสง่า	รหัสนักศึกษา 58010207
	นายชวกร ประเสริฐสุโข	รหัสนักศึกษา 58010263
	นายธนวัฒน์ สมโนชัย	รหัสนักศึกษา 58010517
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2561	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ. ดลชัย สุขเจริญผล	

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(ผศ. ดลชัย สุขเจริญผล)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ออดิโอ วิชวลไลเซชั่น	
รายชื่อนักศึกษา	นายจุติพัฒน์ ฉัตรสมสง่า	รหัสนักศึกษา 58010207
	นายชวกร ประเสริฐสุโข	รหัสนักศึกษา 58010263
	นายธนวัฒน์ สมโนชัย	รหัสนักศึกษา 58010517
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2561	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ. ดลชัย สุขเจริญผล	

บทคัดย่อ

ชิ้นงานโปรเจคนี้มีวัตถุประสงค์เพื่อเพิ่มอรรถรสในการรับชม รับฟังสื่อ เพลง และวีดีโอต่างๆ ด้วยการนำเส้นไฟแอลอีดี (LED Strip) มาแสดงแสงไฟให้เข้ากับเสียงเพลงที่เปิดตามเวลาจริง (Real-Time) โดยการแสดงแสงไฟนั้นจะแสดงตามช่วงความถี่ของเสียงที่เปิดขึ้นมาและแสดงออกมาในรูปแบบต่างๆ เนื่องจากตามงานแสดงดนตรีเล็กๆนิยมใช้หลอดไฟปิงปองเพื่อเพิ่มบรรยากาศ หรือแม้กระทั่งงานดนตรีใหญ่ๆ ก็มักใช้สปอร์ตไลท์ (Spotlight) โดยควบคุมให้เข้ากับจังหวะเพลง ดังนั้นจึงได้ออกแบบให้ชิ้นงานนี้เป็นอีกหนึ่งทางเลือกหรือเป็นส่วนหนึ่งของงานแสดงต่างๆได้ ไม่ว่าจะใช้กับห้องส่วนตัว งานดนตรีขนาดเล็ก หรือแม้กระทั่งงานดนตรีขนาดใหญ่ ก็สามารถเป็นส่วนหนึ่งในงานเหล่านั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	AUDIO VISUALIZATION	
Student	Mr. Jutiphat Chatsomsanga	Student ID. 58010207
	Mr. Chawakorn Prasertsuko	Student ID. 58010263
	Mr. Thanawat Somnochai	Student ID. 58010517
Degree	Bachelor of Engineering	
Program	Information Engineering	
Year	2018	
Thesis Advisor	Asst.Prof. Dolcha Sukcharoelphol	

Abstract

This project aims to increase entertaining experience to media, music, and videos or even the live show by making LED Strips play real-time light pattern according to music being played at that time. The light patterns displayed were controlled by frequency and amplitude of a sound wave, colors controlled by frequency and brightness controlled by amplitude (loudness). Today we need manual control by a human to control light to respond to show or music like the lighting in concert, therefore, we design this project to be an option or an additional part to those show. This project can be used in personal entertainment or even a concert.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้ดำเนินการจนสำเร็จลุล่วงได้ด้วยดี เพราะความร่วมมือของคณะผู้จัดทำ และความกรุณาจาก ผศ.ดลชัย สุขเจริญผล อาจารย์ที่ปรึกษาโครงงานผู้ซึ่งให้การสนับสนุน แก้ไขข้อบกพร่อง ตลอดจนอบรมสั่งสอนชี้นำแนวทาง และให้คำปรึกษาอันเป็นประโยชน์อย่างยิ่งตลอดการดำเนินงาน คณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณคณาจารย์ และเจ้าหน้าที่ประจำสาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้คำแนะนำอันเป็นประโยชน์ ตลอดจนเอื้อเฟื้ออุปกรณ์ และสถานที่สำหรับการดำเนินโครงงาน

ขอขอบคุณเพื่อน ๆ และพี่ ๆ ทุกคนที่คอยให้คำแนะนำ ให้ความช่วยเหลือในการทำโครงงาน และคอยให้กำลังใจที่ดีตลอดการดำเนินโครงงาน

สุดท้ายนี้ขอขอบพระคุณบิดา มารดา และบุคคลในครอบครัวทุกท่านที่ได้ดูแลเอาใจใส่ให้ความรัก พร้อมทั้งมอบโอกาสในการศึกษาอย่างเต็มที่ อีกทั้งยังให้กำลังใจ และเป็นแรงผลักดันที่ดีเสมอมา

คณะผู้จัดทำรู้สึกซาบซึ้ง และขอกราบขอบพระคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

จุติพัฒน์ ฉัตรสมสง่า
ชวกร ประเสริฐสุโข
ธนวัฒน์ สมโนชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญรูป ..... ฉ

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของปัญหา 1

1.2 จุดประสงค์ 1

1.3 ขอบเขตของโครงงาน 1

1.4 ผลที่คาดว่าจะได้รับ 2

1.5 อุปกรณ์ที่ต้องใช้ 2

1.6 ขั้นตอนการดำเนินงาน 3

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ 4

2.1 องค์ประกอบของเสียง 4

2.1.1 ระดับเสียง 4

2.1.2 ความเข้มเสียง 5

2.1.3 สีสันของเสียง 6

2.1.4 คุณภาพของเสียง 7

2.1.5 ความยาวเสียง 7

2.1.7 คลื่นเสียง 7

2.2 ชุดกรองความถี่แบบเมล 7

2.2.1 MFCC 8

2.3 โปรโตคอลในการสื่อสารข้อมูล 10

2.3.1 โปรโตคอล UDP. 10

2.4 ภาษาไพธอน 11

2.4.1 จุดเด่นของภาษาไพธอน 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบการทดลอง....13

3.1 การใช้งานโปรแกรม Anaconda....14

3.1.1 ส่วนเสริมของโปรแกรม Anaconda....14

3.1.2 โปรแกรม VS code....14

3.2 การออกแบบส่วนของซอฟท์แวร์....15

3.2.1 PyAudio....15

3.2.2 Numpy....15

3.2.3 PyQtGraph....23

3.3 การออกแบบส่วนของฮาร์ตแวร์....23

บทที่ 4 การใช้งานและการทดลอง....24

4.1 การทดลองตัวแสดงผลวิชวลไลเซชันแบบซอฟท์แวร์ที่ได้รับความถี่ต่าง....24

4.2 การทดลองรูปแบบการแสดงผลวิชวลไลเซชันแบบซอฟท์แวร์....25

4.3 การทดลองรูปแบบการแสดงผลวิชวลไลเซชันแบบฮาร์ตแวร์....28

4.4 การทดลองรับเสียงด้วยไมค์โครโฟน....31

บทที่ 5 บทวิจารณ์และสรุปผล....32

5.1 บทสรุป....32

5.1.1 การแสดงผลวิชวลไรเซชันแบบซอฟท์แวร์....32

5.1.2 การแสดงผลวิชวลไรเซชันแบบฮาร์ต์แวร์....32

5.2 ปัญหาและอุปสรรคในการทำงาน....32

5.3 สิ่งที่นำไปพัฒนาต่อในอนาคต....33

บทที่ 6 ภาคผนวก....35

บรรณานุกรม....15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ภาพแสดงการสั่นสะเทือนของเสียง....5
2.2 ความสัมพันธ์ระหว่าง ความถี่ scale และ mel-scale....8
2.3 Mel scale filterbank แต่ละจุดสูงสุดของความถี่ตรงกลางใน critical band....9
2.4 ภาพแสดงโปรโตคอล UDP....11
3.1 การทำงานของระบบ....12
3.2 ส่วนเสริมของโปรแกรม Anaconda....13
3.3 บล็อกไดอะแกรมส่วนของซอฟท์แวร์....14
3.4 การออกแบบวงจรวิชวลไลเซชัน....14
3.5 รูปตัวอย่างการใช้คำสั่ง numpy.copy....17
3.6 รูปตัวอย่างการใช้คำสั่ง numpy.mean....18
3.7 รูปแสดงตัวอย่างการใช้คำสั่ง numpy.around....19
3.8 รูปตัวอย่างแสดงการใช้งานคำสั่ง numpy.linspace....20
3.9 รูปตัวอย่างแสดงการใช้งานคำสั่ง numpy.linspace ออกมาเป็นรูปกราฟเพื่อให้เห็นภาพได้ดียิ่งขึ้น....20
3.10 รูปแสดงตัวอย่างการใช้คำสั่ง numpy.tile....21
3.11 รูปตัวแสดงอย่างการใช้คำสั่ง numpy.amax....22
3.12 รูปแสดงตัวอย่างการใช้ numpy.ceil....23
3.13 บล็อกไดอะแกรมส่วนของซอฟท์แวร์....23
3.14 การออกแบบวงจรวิชวลไลเซชัน....23
4.1 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์....24
4.2 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์เมื่อใช้ความถี่ 2000 Hz....25
4.3 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์เมื่อใช้ความถี่ 2000 Hz และ 400 Hz....25
4.4 รูปแบบการแสดงวิชวลไรเซชันแบบ Energy....26
4.5 รูปแบบการแสดงวิชวลไรเซชันแบบ Scroll....27
4.6 รูปแบบการแสดงวิชวลไรเซชันแบบ Spectrum....27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.7 ส่วนแสดงวิชวลไลเซชันแบบฮาร์ดแวร์....28
4.8 จอ LED แสดงวิชวลไรเซชันแบบ Energy....29
4.9 จอ LED แสดงวิชวลไรเซชันแบบ Scroll....30
4.10 จอ LED แสดงวิชวลไรเซชันแบบ Spectrum....30
6.1 แสดงผลงานวันโปรเจคเดย์วันแรก 1....34
6.2 แสดงผลงานวันโปรเจคเดย์วันแรก 2....34
6.3 แสดงผลงานวันโปรเจคเดย์วันแรก 3....35
6.4 แสดงผลงานวันโปรเจคเดย์วันที่สอง 1....35
6.5 แสดงผลงานวันโปรเจคเดย์วันที่สอง 2....36
6.6 แสดงผลงานวันโปรเจคเดย์วันที่สอง 3....36
6.7 ESP8266 Datasheet....37
บรรณานุกรม....38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันงาน live entertainments มีบทบาทและได้รับความนิยมมากขึ้นทั้งในการแสดง หรือ การโฆษณา เช่น งานคอนเสิร์ต BNK48 , งานเปิดตัวผลิตภัณฑ์ , วีดีทัศน์ประกอบ Motor show โปรเจคของเราจะทำให้การนำเสนอมีรูปแบบที่น่าสนใจมากยิ่งขึ้น โดยจะทำการประมวลผลเสียง และนำลักษณะเด่นของเสียง เช่น จังหวะเสียงเบสย่านความถี่ต่ำ เสียงช่วงคนร้อง และเสียงในย่านความถี่ที่ต้องการ ไปควบคุมลักษณะการเปลี่ยนแปลงของภาพกราฟฟิกหลายรูปแบบ อย่างต่อเนื่อง ตามลักษณะของเสียงนั้น ซึ่งจะทำให้สื่อนำเสนอหรือการแสดงนั้น มีความสัมพันธ์กันในด้านของภาพและเสียงสามารถทำให้ ผู้ชมสนใจในตัวการแสดงได้มากยิ่งขึ้น มีความสามารถในการโน้มน้าวผู้ชมให้คล้อยตามได้มากยิ่งขึ้น รู้สึกสนุกกับการแสดงได้มากยิ่งขึ้น โปรเจคนี้ ได้นำการนำเสนอ ในสองมิติ ซึ่งก็คือ ในเรื่อง เสียง และ ภาพ มาทำให้สอดคล้องกันมากยิ่งขึ้น ซึ่งจะสามารถ เพิ่มประสบการณ์อันน่าตื่นตาตื่นใจของผู้ชมต่อ การแสดงได้อย่างยอดเยี่ยม

1.2 จุดประสงค์

1.2.1 เพื่อนำไปใช้ประกอบการแสดง, คอนเสิร์ต เพื่อให้ผู้ชมได้รับประสบการณ์ที่ดี และมีอารมณ์ร่วม มากยิ่งขึ้น

1.2.2 เพื่อนำไปประกอบการแสดงทำให้ผู้ชมได้รับประสบการณ์ทางการบันเทิงที่แปลกใหม่สนุกสนานยิ่งขึ้น

1.2.3 เพื่อเพิ่มความหลากหลายในผลิตภัณฑ์ทางเสียงเพลงและการแสดง ทำให้ตอบสนองต่อความต้องการของผู้บริโภคได้อย่างกว้างขวาง และตอบสนองผู้บริโภคกลุ่มที่มีความต้องการพิเศษ เช่น งานคอนเสิร์ต สถานบันเทิงต่างๆ

1.3 ขอบเขตของโครงงาน

1.3.1 โปรแกรมสามารถแสดงผลของสัญญาณเสียงที่กำลังประมวลผลอยู่ออกมาทางหน้าจอได้

1.3.2 จัดทำในส่วนของ Graphical user interface เพื่อให้ผู้ใช้งานสามารถนำไปใช้งานได้สะดวกมากขึ้น

1.3.3 โปรแกรมสามารถแสดงผลกราฟิกที่ได้ประมวลผลมาได้อย่างหลายรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 ได้รับประสบการณ์ใหม่ๆในการรับชมหรือรับฟัง การแสดง, คอนเสิร์ต

1.4.2 เกิดรูปแบบของการแสดงผลจากเสียงที่หลากหลายยิ่งขึ้นซึ่งผู้ใช้สามารถเลือกใช้รูปแบบตามความต้องการของผู้ใช้

1.4.3 สามารถแสดงผลของความถี่เสียงได้ถูกต้อง

1.4.4 ในส่วนของโปรแกรมสามารถใช้งานได้ง่ายเพื่อรองรับกลุ่มผู้ใช้ได้หลากหลาย

1.5 อุปกรณ์ที่ต้องใช้

ซอฟต์แวร์

-Anaconda โปรแกรมจัดการ Python

-Arduino โปรแกรมควบคุมการทำงานของบอร์ด Arduino

ฮาร์ดแวร์

-NodeMCU esp8266

-WS2812B 5050 RGB LED STRIP

-5V power supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขั้นตอนการดำเนินงาน

ID	Task name	ส.ค. 2560	ก.ย. 2560	ต.ค. 2560	พ.ย. 2560	ธ.ค. 2560	ม.ค. 2561	ก.พ. 2561	มี.ค. 2561
1	ศึกษาหัวข้อและขอบเขต								
2	ศึกษา Java script								
3	ศึกษา Visualizing algorithm								
4	ออกแบบ UI								
5	นำผลงานแสดงผลออกมาทางหน้าจอ								
6	ปรับปรุงแก้ไข								
	จัดทำเอกสารการสอบวิชาโครงงาน (เทอม 1)								
7	ศึกษา Hardware								
8	สามารถแสดงผลออกทางจอ LED ได้								
9	ทดสอบการใช้งาน								
10	จัดทำต้นฉบับปริญญานิพนธ์								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้

2.1 องค์ประกอบของเสียง

เสียงเกิดจากการสั่นสะเทือนของวัตถุ เมื่อวัตถุสั่นสะเทือนจะทำให้เกิดการอัดตัวและการขยายตัวของคลื่นเสียงซึ่งถูกส่งไปยังหูคนโดยผ่านชั้นบรรยากาศ ดังนั้นเสียงที่ได้ยินก็เป็นผลจากการที่คลื่นเสียงถูกส่งจากวัตถุที่สั่นสะเทือนไปยังหูนั่นเอง เสียงที่ได้ยินจะเป็นเสียงสูง เสียงต่ำ เสียงเบา หรือมีคุณภาพเสียงในลักษณะต่างๆ อย่างไรนั้น จะขึ้นอยู่กับแหล่งกำเนิดเสียงและจำนวนรอบต่อวินาทีของการสั่นสะเทือนที่ทำให้เกิดเสียงนั้น เสียงแต่ละเสียงมีคุณสมบัติซึ่งสามารถแยกแยะ ได้ดังนี้

2.1.1 ระดับเสียง (Pitch)

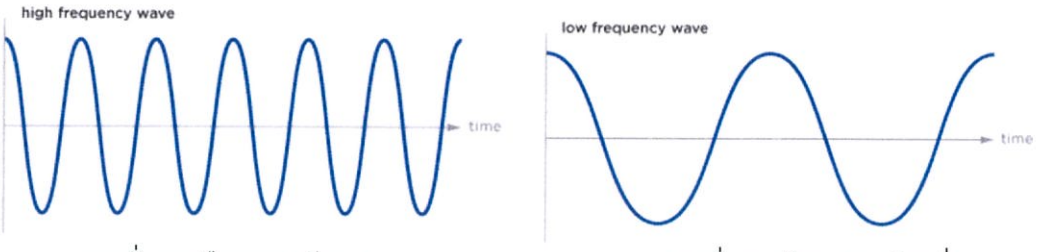
ระดับเสียง หมายถึง เสียงสูง เสียงต่ำ เราสามารถเปรียบเทียบเสียง 2 เสียงได้ว่า เสียงไหนมีระดับเสียงสูงกว่า และเสียงไหนมีระดับเสียงต่ำกว่าโดยการฟัง ถ้าระดับเสียงต่างกันมากก็จะฟังง่าย ถ้าต่างกันเพียงเล็กน้อยก็จะฟังยากกว่า

ระดับเสียง เกิดจากความถี่ของการสั่นสะเทือนของวัตถุ หรืออาจกล่าวได้ว่า ความถี่ของการสั่นสะเทือนเป็นตัวกำหนดระดับของเสียง วัตถุที่สั่นสะเทือนเร็วกว่าทำให้เกิดระดับเสียงสูงกว่าในขณะที่วัตถุที่สั่นสะเทือนช้ากว่าทำให้เกิดระดับเสียงสูงกว่า ถ้าความถี่มากขึ้นเป็นเท่าตัวระดับเสียงจะสูงขึ้น 1 ช่วงคู่แปด (Octave) เช่น โน้ตที่มีความถี่ 220 รอบต่อวินาที จะมีระดับเสียงสูงกว่าโน้ตที่มีความถี่ 110 รอบต่อวินาทีอยู่ 1 ช่วงคู่แปด และในทำนองเดียวกันกรณีที่ความถี่น้อยลง 1 เท่าตัวระดับเสียงจะต่ำลง 1 ช่วงคู่แปด ฉะนั้นโน้ตที่มีความถี่ 440 รอบต่อวินาที จะมีระดับเสียงต่ำกว่าโน้ตที่มีความถี่ 880 รอบต่อวินาทีเป็นระยะ 1 ช่วงคู่แปด

ระดับเสียงขึ้นอยู่กับความถี่ (frequency) ของการสั่นสะเทือน เช่น ถ้าการสั่นสะเทือนยิ่งเร็วเสียงจะยิ่งสูง แต่ถ้าการสั่นสะเทือนยิ่งช้าเสียงก็จะยิ่งต่ำ ความถี่ของการสั่นสะเทือนปกติจะวัดเป็นรอบ/วินาที (cycles/second) ในเปียโนมีความถี่ของการสั่นสะเทือนเสียงที่สูงที่สุดอยู่ที่ 4,186 รอบ/วินาที และความถี่ที่ต่ำสุดคือ 27 รอบ/วินาที

โดยทั่วไปแล้ววัตถุที่สั่นสะเทือนยิ่งมีขนาดเล็กก็จะทำให้เกิดการสั่นสะเทือนเร็วขึ้น สิ่งที่ตามมาก็คือเสียงสูงขึ้น เช่น ถ้าเราดีดสายไวโอลิน (Violin) เปล่าๆ จะพบว่าจะมีความถี่สูงหรือเสียงสูงกว่าเราดีดสายดับเบิลเบส (Double bass) เปล่าๆ ทั้งนี้เนื่องจากว่าสายของไวโอลินมีขนาดและความยาวน้อยกว่าดับเบิ้ลเบสนั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การสั่นสะเทือนของเสียงสูง การสั่นสะเทือนของเสียงต่ำ

รูปที่ 2.1 ภาพแสดงการสั่นสะเทือนของเสียง

2.1.2 ความเข้มเสียง (Volume or Inetensity)

ความเข้มเสียง หมายถึง เสียงเบา เสียงดัง ความเข้มเสียงเกิดจากแรงสั่นสะเทือนของวัตถุที่เป็นแหล่งกำเนิดเสียงถ้าวัตถุสั่นสะเทือนมากเสียงจะดังและในทางตงข้ามถ้าวัตถุสั่นสะเทือนน้อยเสียงจะเบา ความเข้มเสียงขึ้นอยู่กับแรงที่ส่งจากแหล่งกำเนิดเสียงไปยังหู เวลาที่เราตะโกนเส้นเสียงจะสั่นสะเทือนมากทำให้เกิดเสียงดังถ้าเราตีกลองโดยออกแรงมากขึ้นหนังกลองจะสั่นมากขึ้นทำให้เสียงกลองดังขึ้น ความเข้มเสียงวัดได้จากความกว้างคลื่นเสียง (Amplitude) ถ้าคลื่นเสียงยิ่งกว้างเสียงก็จะยิ่งดัง ในขณะที่เสียงเดินทางผ่านชั้นบรรยากาศนั้นความเข้มเสียงจะน้อยลงตามลำดับ กล่าวคือ เสียงจะเบาลง ดังนั้นระยะทางที่เสียงต้องเดินทางก็จะมีผลต่อความเข้มเสียงด้วย

ในระหว่างที่นักดนตรีบรรเลงดนตรีในวงให้มีความดังมากแล้วค่อยลงหรือมีการเปลี่ยนแปลงจำนวนชิ้นของเครื่องดนตรีที่เล่นให้น้อยลง ผลที่ตามมาก็คือมีการเปลี่ยนแปลงของความดัง-ค่อย (Dynamic) การเปลี่ยนแปลงดังกล่าวนี้เราสามารถทำให้เกิดได้โดยทันทีทันใดหรือให้เกิดทีละน้อยก็ได้ การทำให้เกิดเสียงที่เป็นลักษณะของความดัง-ค่อยของเสียงนี้มีผลทำให้เกิดความตื่นเต้น (excitement) โดยเฉพาะอย่างยิ่งเมื่อมีระดับเสียง (pitch) สูงขึ้น และถ้าหากเราค่อยๆลดความดังของเสียงลงทีละน้อยๆ เสียงต่ำลงก็สามารถทำให้เกิดความรู้สึกสงบ (sense of calm) ได้เช่นกัน

ขณะที่นักดนตรีบรรเลงอยู่ในวงดนตรีอยู่นั้นเขาสามารถจะปฏิบัติเสียงของการเล่นโน้ตตัวใดตัวหนึ่งให้เด่นขึ้นได้โดยวิธีการ “การเน้นเสียง” (dynamic accent) การเน้นเสียงนั้นเป็นการกระทำให้มีการเปลี่ยนแปลงความดัง – ค่อยขอเพลงมีผลทำให้บทเพลงนั้นๆ มีอารมณ์ของการแสดงออกเปลี่ยนแปลงตามไปด้วย ในการปฏิบัติการเน้นเสียงดังกล่าวนั้นปกติผู้ประพันธ์เพลงจะเขียนกำกับไว้ในโน้ตเพลง แต่ก็มีบ่อยครั้งที่ไม่มีการเขียนกำกับไว้แต่จะปล่อยให้เป็นการแสดงออกโดยอารมณ์ความรู้สึก (feelings) ของนักดนตรีเอง

ในการบันทึกโน้ตให้ผู้อื่นเล่นนั้นเป็นธรรมเนียมในดนตรีตะวันตกที่ผู้ประพันธ์เพลงจะต้องเขียนเครื่องหมายและคำสั่งต่างๆ กำกับไว้ในโน้ตเพลงโดยใช้คำศัพท์ภาษาอิตาเลียน (Itaian) โดยใช้คำย่อ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mortississimo	Fff	ดังที่สุด
Fortissimo	Ff	ดังมาก
Morte	F	ดัง
mezzo forte	Mf	ปานกลางค่อนข้างดัง
mezzo piano	Mp	ปานกลางค่อนข้างเบา
piano	P	เบา
Pmanissimo	Pp	เบามาก
Pianississimo	Ppp	เบาที่สุด

สำหรับช่วงใดของเพลงที่ต้องการให้มีความดัง ค่อยมากๆ ผู้ประพันธ์เพลงมักใช้ fff หรือ ffff และ ppp หรือ pppp กำกับลงไปตรงตำแหน่งนั้นๆ

นอกจากนี้แล้วยังมีเครื่องหมายที่ใช้สำหรับการเปลี่ยนแปลงของเสียงโดยให้เสียงที่ปฏิบัตินั้นค่อยๆ เกิดขึ้นทีละน้อยเช่น สำหรับช่วงใดของเพลงที่ต้องการให้มีความดัง - ค่อย มากๆ ผู้ประพันธ์เพลงมักใช้ fff หรือ ffff และ ppp หรือ pppp กำกับลงไปตรงตำแหน่งนั้นๆ

นอกจากนี้แล้วยังมีเครื่องหมายที่ใช้สำหรับการเปลี่ยนแปลงของเสียงโดยให้เสียงที่ปฏิบัตินั้นค่อยๆเกิดขึ้นทีละน้อย

- Cresscendo (เครเชนโด) แสดงการเพิ่มความดังของเสียงขึ้นทีละน้อย

- Decrescendo (เดเครเชนโด) แสดงการลดความดังของเสียงขึ้นทีละน้อย หรือ Diminendo (ดิมินูเอนโด)

2.1.3 สีสันของเสียง (Timbre or Tone Color)

สีสันเสียง หมายถึง เสียงที่มาจากแหล่งกำเนิดเสียงต่างๆกัน เช่น เสียงระนาด เสียงกีต้าร์ เสียงร้อง เสียงสัตว์ เสียงลม เป็นต้น คลื่นเสียงที่เกิดจากแหล่งที่มีสีสันเสียงต่างกันจะมีรูปร่างลักษณะต่างกัน เช่น คลื่นเสียงของไวโอลิน จะมีรูปร่างต่างจากคลื่นของทรัมเป็ต สื่อดนตรีที่ต่างกันทำให้เกิดสีสันเสียงต่างชนิดกัน สื่อดนตรี ก็คือ เครื่องดนตรีหรือเส้นเสียงดนตรีในกรณีการขับร้อง เครื่องดนตรีต่างชนิดกันจะทำให้เกิดสีสันเสียงต่างกัน เพลงเดียวที่บรรเลงโดยเครื่องดนตรีต่างชนิดกันจะทำให้เกิดรสชาติต่างกันเพราะสีสันเสียงต่างกัน

สิ่งที่สำคัญที่สุดที่เป็นตัวกำหนดเสียง ก็คือ พาร์เชียล(Partials) ซึ่งเป็นระดับเสียงจำนวนหนึ่ง ที่ประกอบกันเป็นเสียงที่เราได้ยิน เพราะฉะนั้น จำนวนระดับเสียงของพาร์เชียล ความเข้มเสียงของพาร์เชียล และการกระจายของพาร์เชียล จะเป็นตัวกำหนดสีสันเสียงทั้งสิ้น เสียงที่มีจำนวนพาร์เชียลน้อยจะมีเสียงใส เช่น เสียงของฟรุตที่เล่นโน้ตสูงๆ ส่วนเสียงที่มีจำนวนพาร์เชียลมาก จะมีเสียงที่หนักแน่นกว่า เช่น เสียงของเชลโล เสียงที่ไม่มีจำนวนพาร์เชียลเลย หรือเสียงบริสุทธิ์เกิดขึ้นได้จากเครื่องเสียงดนตรีไฟฟ้า และจะเป็นช่วงเสียงสูงมากๆ ที่ไม่ไพเราะสำหรับหูคนทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 คุณภาพของเสียง (Tone Quality)

คุณภาพของเสียง หมายถึง เสียงดีมาก ดีน้อย เช่น เสียงโน้ตตัวเดียวกันที่เกิดจากเปียโน 2 ตัว จะมีคุณภาพเสียงไม่เหมือนกันทีเดียว แม้จะมีระดับเสียง ความเข้มเสียง และสีสันเสียงเหมือนกันทุกประการ เสียงร้องหรือเสียงพูดของคนก็เช่นเดียวกัน บางคนมีคุณภาพเสียงที่ทุ้มนุ่มนวล ในขณะที่อีกคนหนึ่งมีคุณภาพเสียงที่ก้องกังวาน ในการปฏิบัติเครื่องดนตรี นักดนตรีต้องคำนึงอย่างมากถึงคุณภาพของเสียง เพื่อให้ได้น้ำเสียงที่ไพเราะน่าฟัง

2.1.5 ความยาวเสียง (Duration)

ความยาวเสียง เป็นพื้นฐานของดนตรีที่เกี่ยวข้องกับระยะเวลา เสียงแต่ละเสียงที่เกิดขึ้นต้องมีระยะเวลาซึ่งทำให้เกิดเสียงยาว เสียงสั้น เสียงไม่ว่าจะมีกี่ระดับเสียงที่แน่นอนหรือไม่ ก็ต้องมีความยาวเข้ามาเกี่ยวข้อง ความยาวเสียงเป็นที่มาของจังหวะ ซึ่งมีส่วนสำคัญอย่างยิ่งในดนตรีของทุกชาติ ทุกภาษา ความยาวเสียงในที่นี้ หมายรวมถึง ความเงียบของตัวหยุดด้วย เนื่องจากดนตรีเป็นผลของกระบวนการเสียง (Sound) สลับกับความเงียบ (Silence) ไม่ว่าจะเป็นเสียงหรือความเงียบ ย่อมต้องมีความยาวหรือระยะเวลาทั้งสิ้น

2.1.6 คลื่นเสียง (Sound Wave)

คลื่นเสียงเป็นคลื่นที่มีความถี่อยู่ในช่วง 20 ถึง 20,000 เฮิรตซ์ ซึ่งเป็นความถี่ที่มนุษย์สามารถได้ยินคลื่นเสียงเป็นคลื่นที่มีความถี่ต่ำ ทำให้ไม่สามารถเดินทางไปได้ไกล เพราะเกิดจากการจางหายของคลื่นได้ง่าย การกำเนิดของคลื่นเสียงนี้เกิดจากการสั่นสะเทือนของอากาศโดยรอบแล้วกระทบเยื่อแก้วหูของผู้ฟังให้สั่นสะเทือนตามระดับความถี่เสียงนี้จะแบ่งออกเป็น 3 ระดับคือความถี่ต่ำอยู่ในช่วงความถี่ประมาณ 20 ถึง 500 เฮิรตซ์ ความถี่ปานกลางประมาณ 500 ถึง 5,000 เฮิรตซ์ และความถี่สูงประมาณ 5,000 ถึง 20,000 เฮิรตซ์

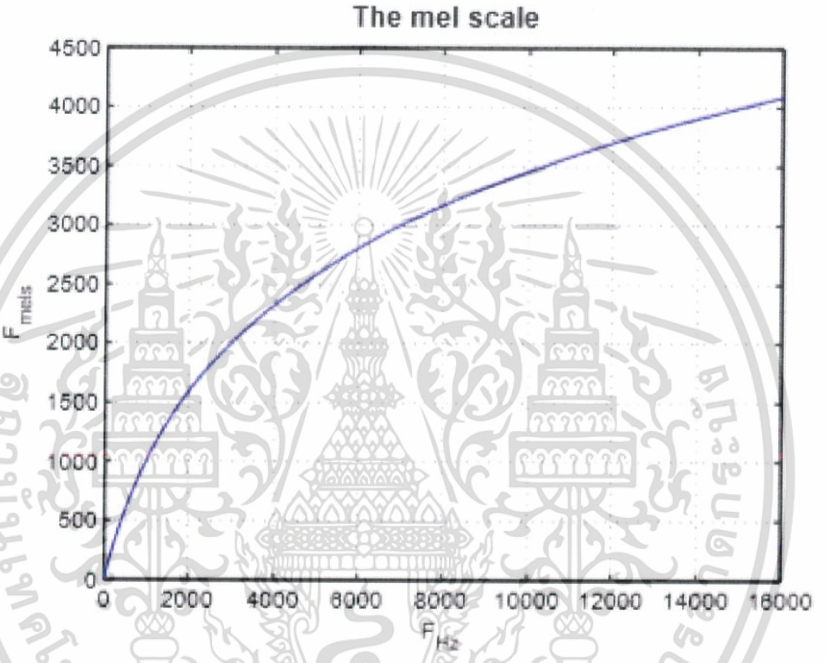
2.2 ความถี่แบบเมล

เนื่องจากหูของมนุษย์จะแยกรายละเอียดของสัญญาณเสียงความถี่ต่ำได้ดีกว่าความถี่สูง ดังนั้นจึงมีการออกแบบชุดตัวกรองที่ให้ความสำคัญกับสเปกตรัมย่านความถี่ต่ำ เรียกว่า mel-filter bank ที่จะไม่เน้นความถี่ช่วงกลางของตัวกรอง โดยความถี่กลางของตัวกรองแต่ละช่องนั้นอยู่บนสเกลความถี่แบบเมล (mel-scale) ซึ่งเป็นมาตราส่วนความถี่แบบไม่สม่ำเสมอ เพื่อแก้ปัญหานี้สเกลความถี่แบบเมลจึงถูกพัฒนาขึ้น ความถี่เมลนี้ได้เกิดจากการทดลองในความสามารถในการได้ยินของมนุษย์ในปี 1940 จากการทดลองแสดงให้เห็นว่า การได้ยินเสียงของมนุษย์นั้นได้ยินเสียงเป็นเส้นตรงความถี่ในช่วง 0-1000hz และมากกว่า 1000 hzนั้นจะไม่ใช่เส้นตรงแต่จะเป็น ตามสมการฟังก์ชัน log จึงคำนวนค่า mel-scale ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$F_{mel} = \frac{1000}{\log 2} \cdot \left[1 + \frac{F_{Hz}}{1000}\right] \quad (1)$$

เมื่อ F_{mel} คือผลลัพธ์ของความถี่ที่ใน mel-scale มีหน่วยเป็น mel และ F_{Hz} คือความถี่ปกติในหน่วย Hz ซึ่งจะได้ดังรูปที่ 2.2



รูปที่ 2.2 ความสัมพันธ์ระหว่าง ความถี่ scale และ mel-scale

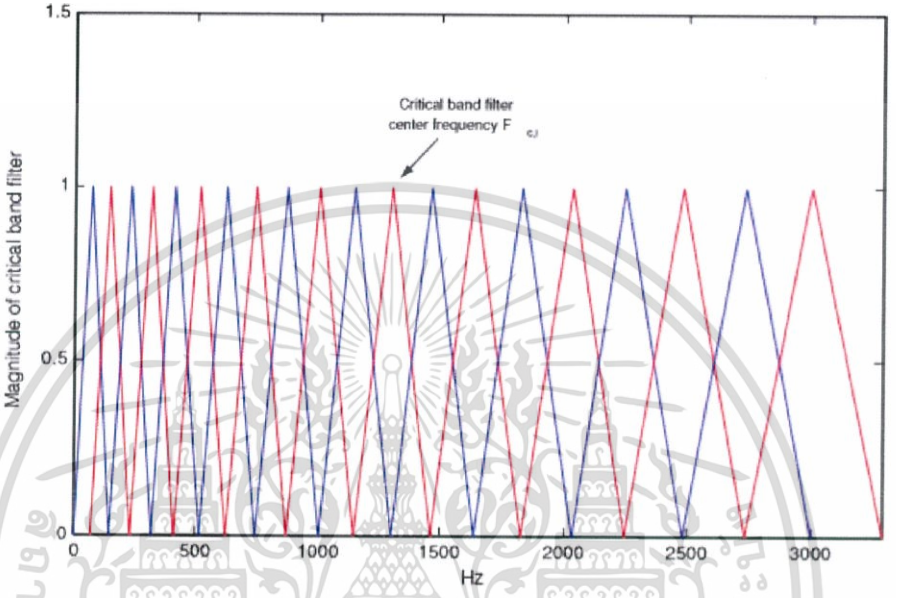
ด้วยการใช้ mel-scale สัมประสิทธิ์จาก LPC จะมีการกระจุกตัวอยู่ในความถี่ต่ำและบริเวณรอบ ๆ ที่มนุษย์สามารถได้ยิน ซึ่งอาจส่งผลให้มีการอธิบายสัญญาณได้ชัดเจนยิ่งขึ้น

2.2.1 MFCC

Mel Frequency Cepstral Coefficients (MFCC) เป็นความถี่ที่ได้มาจากการใช้ฟิลเตอร์แบงค์ จากรูปที่ 2.3 พบว่าพลังงานในคลื่นความถี่ที่สำคัญของความถี่ที่ทำให้มนุษย์สามารถรับรู้การได้ยิน อยู่ในช่วงความถี่ที่เป็นเส้นตรงที่ต่ำกว่า 1000 Hz และความถี่ตามสมการฟังชัน log ดังรูปที่ 2.2 ซึ่งนำ mel-scale มารวมกับคลื่นที่กระจายของ critical bands จะได้คลื่นที่เป็นเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Critical band คือ วงจรกรองแถบความถี่ผ่าน,ปรับความถี่รอบศูนย์กลาง ที่ความถี่ต่ำกว่า 1000 Hz critical band จะอยู่ในช่วง 100, 200, ... 1000 Hz ที่ความถี่มากกว่า 1 KHz จะอยู่ใน mel-scale วิธีการคำนวนพลังงาน MFCC ที่ใช้ในแต่ละ critical band สามารถทำได้ดังสมการ (2)



รูปที่ 2.3 Mel scale filterbank แต่ละจุดสูงสุดของความถี่ตรงกลางใน critical band

$$\boldsymbol{Y(i) = \sum_{k=0}^{N/2} \log|s(n)| \cdot H_i(k \cdot \frac{2\pi}{N'})} \quad (2)$$

เมื่อ $\boldsymbol{Y(i)}$ คือพลังงานทั้งหมดใน critical band, $\boldsymbol{N}$ คือความยาวเฟรม, $\boldsymbol{s(n)}$ คือสัญญาณ DFT สำหรับการคำนวณ MFCC, $\boldsymbol{H_i()}$ คือประสิทธิ์ภาพของ critical band ที่ i นั้นๆ และ $\boldsymbol{N'}$ คือจำนวนจุดที่ใช้ใน DFT

ต่อไปเราจะต้องคำนวณ IDTF ที่เกิดขึ้นจริงเพื่อให้ได้ค่าสัมประสิทธิ์ สำหรับเรื่องนี้เราต้องคิดแต่ละ crtirical band ซึ่งทำได้โดย

$$\widetilde{\boldsymbol{Y}}(\boldsymbol{k}) = \begin{cases} \boldsymbol{Y(i)} & \boldsymbol{, k = k_i} \\ \boldsymbol{0} & \boldsymbol{, otherk \in [0, N' - 1]} \end{cases} \quad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น cepstrum สุดท้ายสามารถหาได้จาก :

$$C_s(n) = \frac{1}{N'} \cdot \sum_{k=0}^{N'-1} \tilde{Y}(\boldsymbol{k}) e^{jk\left(\frac{2\pi}{N'}\right)n} \quad (4)$$

ถ้าใช้ cepstrum จริง ลำดับ $\tilde{Y}(\boldsymbol{k})$ คือ สมมาตร (คงที่) เกี่ยวกับศูนย์กลางความถี่ของ critical band เป็น $\boldsymbol{N'/2}$ สมการที่ 4 สามารถลดรูปได้ดังนี้

$$C_s(n) = \frac{1}{N'} \cdot \sum_{i=1,2,....N_{cb}} \tilde{Y}(\boldsymbol{k_i}) \cdot \cos(\boldsymbol{k_i} \cdot \frac{2\pi}{N'} n) \quad \text{... .....(5)}$$

เมื่อ N_{cb} คือจำนวนของ critical bands

2.3 โปรโตคอลในการสื่อสารข้อมูล

โปรโตคอล (Protocal) คือ ข้อตกลงในการสื่อสารระหว่างอุปกรณ์ในเครือข่าย ดังนั้น คอมพิวเตอร์หรืออุปกรณ์เครือข่ายจะติดต่อสื่อสารกันได้ต้องใช้โปรโตคอลตัวเดียวกัน โปรโตคอลเป็นได้ทั้งฮาร์ดแวร์และซอฟต์แวร์ซึ่งจะสอดคล้องกับโมเดล OSI โปรโตคอลอาจเป็นเพียงส่วนเดียวหรือประกอบขึ้นมาเป็นชุดก็ได้ โปรโตคอลสำคัญที่พบได้บ่อยในระบบเครือข่ายคอมพิวเตอร์ มีดังต่อไปนี้ โปรโตคอล TCP, โปรโตคอล UDP และ โปรโตคอล IP แต่ในที่นี้จะยกตัวอย่างแค่โปรโตคอล UDP

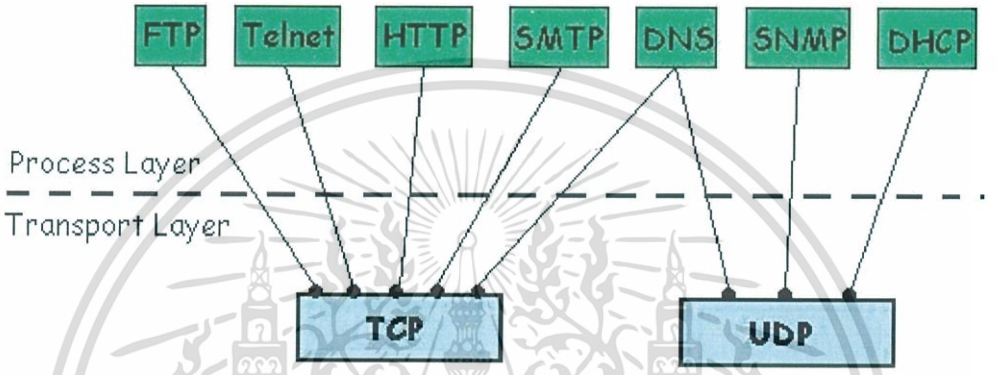
2.3.1 โปรโตคอล UDP

UDP เป็นการส่งข้อมูลที่ไม่มีการยืนยันการรับส่งข้อมูล คือผู้ส่งไม่สามารถรู้ได้ว่าข้อมูลได้ถึงผู้รับแล้วหรือไม่ เราจะต้องเขียนการตรวจสอบข้อมูลเอาเอง เช่น ถ้าผมส่งข้อมูลไป ถ้าฝั่งที่รับข้อมูลได้รับข้อมูลแล้ว ก็ให้ตอบกลับมาให้ผม ผมก็ทราบแล้วว่าข้อมูลไปถึง แต่ถ้าไม่มีการตอบกลับภายในระยะเวลาที่ผมกำหนด ก็ให้แจ้งว่าผู้รับไม่ได้รับข้อความ

UDP อยู่ใน Transport layer(ทรานสโพท เลเยอร์) ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล แต่ไม่มีกลไกความคุมการรับ ส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้ unreliable (อันรีไลเบิ้ล) และ connectionless (คอนเนคชั่นเลทด์) โดยปล่อยให้เป็นหน้าที่ของแอพพลิเคชั่นเลเยอร์ แต่ UDP มีข้อได้เปรียบในการส่งข้อมูลได้ทั้งแบบ unicast (ยูนิคาสต์), multicast (มัลติคาสต์) และ broadcast (บรอดคาสต์) อีกทั้งยังทำการติดต่อสื่อสารได้เร็วกว่า TCP (ที ซี พี) เนื่องจาก TCP ต้องเสีย overhead (โอเวอร์เฮด) ให้กับขั้นตอนการสื่อสารที่ทำให้ TCP มีความน่าเชื่อถือในการรับส่งข้อมูลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDP ซึ่งมีจุดเด่นที่ความเร็ว ขนาดเล็ก และไม่มีการทำงานเกี่ยวการส่งข้อมูลซ้ำหรือคำนวณอัตราการส่งข้อมูล ซึ่งจะเหมาะกับการส่งข้อมูลแบบ realtime (เรียลไทม์) ซึ่งข้อมูลที่สูญหายบางส่วนหรือข้อมูลที่เกิด delay (ดิเลย์) จะถูกละความสนใจไปมันจะส่งข้อมูลได้เร็วกว่า แบบ TCP และจะไม่มีการสร้าง Connection (คอนเนคชั่น) เกิดขึ้น ทำให้ข้อมูลที่วิ่งในเครือข่ายมีน้อยลง ด้วยเป็นการสื่อสารแบบ Connectionless คือข้อมูลจะถูกแบ่งเป็นชิ้นๆ ตามที่อยู่ปลายทาง แล้วผ่านตัวกลางไปยังปลายทาง อาจจะใช้เส้นทางคนละเส้นทางกันก็ได้ รวมทั้งข้อมูลแต่ละชิ้นอาจจะถึงก่อนหลังแตกต่างกันไปได้ด้วย ทำให้การเริ่มต้นส่งทำได้รวดเร็ว ไม่ต้องเสีย



รูปที่ 2.4 ภาพแสดงโปรโตคอล UDP

2.4 ภาษาไพธอน

ไพธอน (Python) เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา ไพธอน นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 ไพธอน นั้นเป็นภาษาแบบ interpreter ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่ ไพธอนนั้นมีคุณสมบัติเป็นภาษาเขียนโปรแกรมแบบไดนามิกส์และมีระบบการจัดการหน่วยความจำอัตโนมัติและสนับสนุนการเขียนโปรแกรมหลายรูปแบบ ที่ประกอบไปด้วย การเขียนโปรแกรมเชิงวัตถุ imperative การเขียนโปรแกรมแบบฟังก์ชัน และการเขียนโปรแกรมแบบขั้นตอน มันมีไลบรารี่ที่ครอบคลุมการทำงานอย่างหลากหลายตัวแปรในภาษา Python นั้นมีให้ใช้ในหลายระบบปฏิบัติการ ทำให้โค้ดของภาษา ไพธอน สามารถรันในระบบต่าง ๆ ได้อย่างกว้างขวาง CPython นั้นเป็นการพัฒนาในตอนต้นของ ไพธอน ซึ่งเป็นโปรแกรมแบบ open source และมีชุมชนสำหรับเป็นต้นแบบในการพัฒนา เนื่องจากมันได้มีการนำไปพัฒนากระจายไปอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลากหลาย CPython นั้นจึงถูกจัดการโดยองค์กรไม่แสวงหาผลกำไรอย่าง Python Software Foundation

2.4.1 จุดเด่นของภาษาไพธอน

ภาษาไพธอน ได้ออกแบบมาเพื่อให้ทำงานได้กับเว็ปแอปพลิเคชัน (Web Application) ที่มีลักษณะคล้ายกับภาษาเพิร์ล ภาษาพีเอชพี (PHP) ภาษาจาวา (Java) และภาเอเอสพี (ASP) เนื่องจากภาษาไพธอน เป็นภาษาที่ใหม่จึงมีคุณสมบัติที่ดีดังต่อไปนี้

1. สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือ สามารถทำงานได้ทุกซีพียู (CPU) หลายระบบปฏิบัติการ ไม่ว่าผู้เขียนโปรแกรมจะเขียนโปรแกรมจากแพลตฟอร์มใดก็ตามสามารถนำโปรแกรมที่เขียนนั้นไปทำงานต่างแพลตฟอร์มกันได้

2. ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมต้นฉบับ โดยปกติแล้วโปรแกรมภาษาทั่ว ๆ ไปจะต้องซื้อโปรแกรมต้นฉบับเพื่อนำมาติดตั้งในราคาที่แพงมากแต่โปรแกรมภาษาไพธอนสามารถดาวน์โหลด ได้โดยตรง แล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภทโอเพนซอรส์

3. ภาษาไพธอนได้นำเอาข้อดีของโปรแกรมในอดีตเข้ามาไว้ด้วยกัน เช่น ภาษา C++ ภาษา Java และภาษาเพิร์ล เป็นต้น

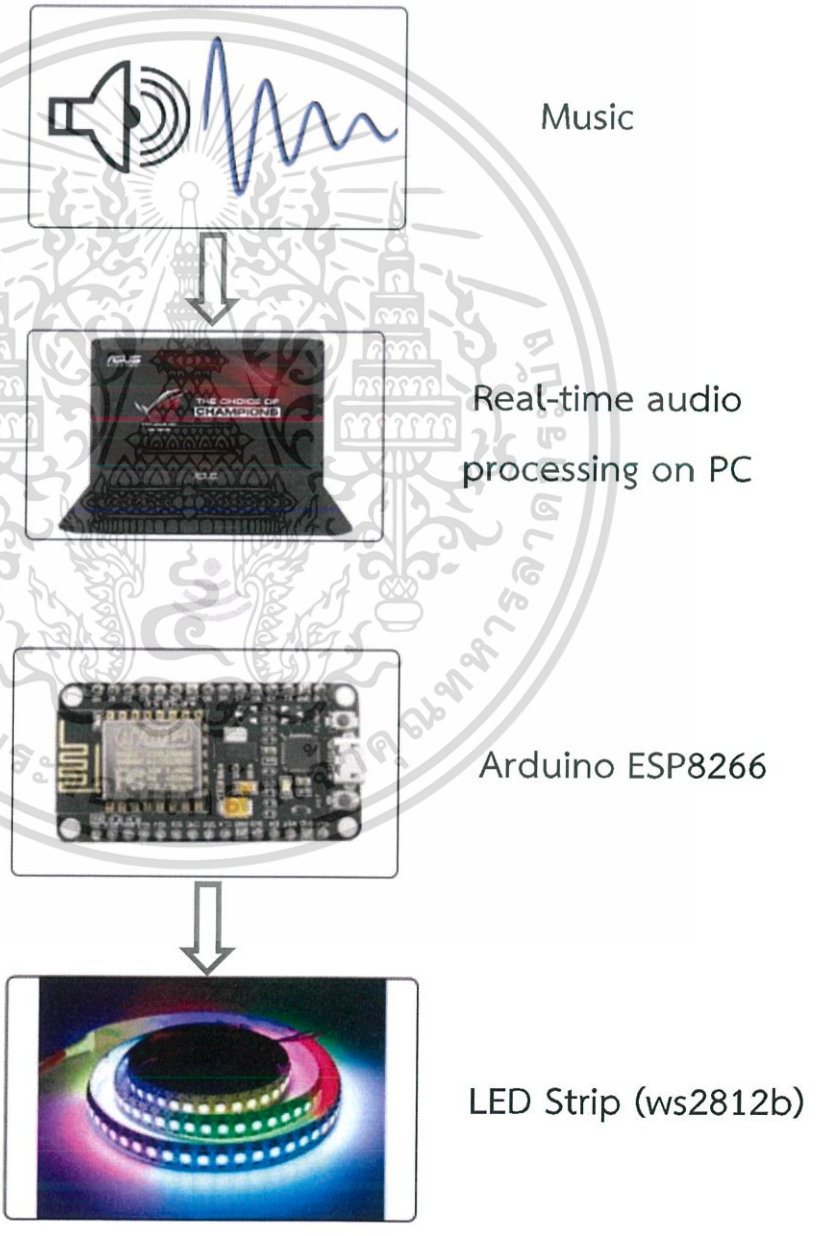
4. มีความปลอดภัยสูง เนื่องจากภาษาไพธอนทำงานอยู่ด้านแม่ข่าย (Server) เป็นหลักเมื่อมีการร้องขอจากเครื่องลูกข่าย (Client) จะประมวลผลที่เครื่องแม่ข่าย ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่องแม่ข่ายได้โดยตรงจึงมีความปลอดภัยสูงกว่า

5. ใช้ในการพัฒนาเว็บเซอร์วิส (Web Service) ซึ่งในปัจจุบันการพัฒนาซอฟท์แวร์ได้เน้นที่มีการแลกเปลี่ยนข้อมูลซึ่งกันและกันทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน ทำให้เกิดความสะดวกสบาย ไม่ต้องใช้ซอฟต์แวร์อื่น ๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไปเรียนรูปได้เร็วกว่าโปรแกรมภาษาอื่น ๆ เพราะมีโครงสร้างไม่ซับซ้อน ซึ่งโครงสร้างภาษาคล้ายคลึงกับภาษาซี ถ้าโปรแกรมเมอร์ที่เคยใช้ภาษาซี มาก่อนจะทำให้เรียนรู้ได้เร็วยิ่งขึ้น นอกจากนี้จากการเขียนโปรแกรมด้วยภาษาไพธอนจะมีความกระชับและสั้นกว่าภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบการทดลอง

โครงงานนี้ใช้โปรแกรม Anaconda .ในการทดลองและออกแบบโปรแกรม ออดิโอ วิชวล ไลเซชัน รวมทั้งใช้โปรแกรม Arduino .ในการควบคุมอุปกรณ์ ซึ่งโครงงานเล่มนี้ ได้ทำการทดลอง แบ่งออกเป็น 2 ส่วน คือส่วนที่เป็นซอฟท์แวร์เป็นส่วนประมวลผลและแสดงค่าออกทางหน้าจอ ส่วนที่เป็นฮาร์ดแวร์คือส่วนที่แสดงผลออกมาจริง ๆ โดยแผนภาพการทำงานของระบบดังรูปที่ 3.1



รูปที่ 3.1 การทำงานของระบบ

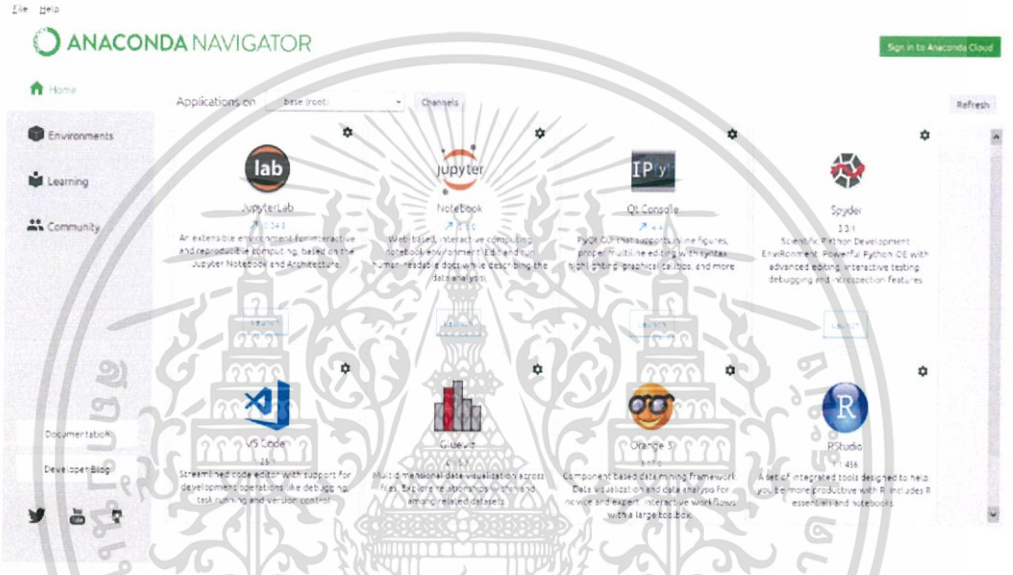
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การใช้งานโปรแกรม Anaconda เบื้องต้น

โปรแกรม Anaconda เป็นโปรแกรมที่ใช้จัดการภาษา python ซึ่งภาษา python มีความสามารถหลากหลายทั้ง รับเสียง การแสดงผลทางหน้าจอ เป็นต้น มีรูปแบบการใช้งานที่สะดวกเข้าใจง่าย แต่มีความยืดหยุ่นสูง รองรับการออกแบบที่มีความสลับซับซ้อนได้

3.1.1 ส่วนเสริมของโปรแกรม Anaconda

Anaconda มีส่วนเสริมของโปรแกรมไว้สำหรับ จัดการภาษา python จำนวนมาก แต่ในที่นี้กล่าวถึงเฉพาะส่วนเสริมที่จำเป็นต่อการทดลองนี้เท่านั้น โดยสามารถเลือกส่วนเสริมจาก Home



รูปที่ 3.2 ส่วนเสริมของโปรแกรม Anaconda

3.1.2 โปรแกรม VS Code

VS code หรือ Visual studio Code เป็นโปรแกรมที่อยู่ในส่วนเสริมของโปรแกรม Anacoda เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาก ไม่ว่าจะเป็น 1.การเปิดใช้งานภาษาอื่น ๆ ทั้งภาษา C++, C#, Java, Python, PHP หรือ Go 2.Themes 3.Debugger 4.Commands เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบส่วนของซอฟท์แวร์

ส่วนของซอฟท์แวร์นั้น จะทำการประมวลผลของเสียง ด้วย code ในภาษา python ในโปรแกรม VS Code และนำไปแสดงผลใน GUI เทคนิคที่ใช้ประกอบด้วย

3.2.1 PyAudio

ส่วนของการรับเสียงของการทำงานของโปรแกรม ใช้ PyAudio ซึ่งเป็น 1 ใน library ของ python ทำหน้าที่ในการรับเสียงและเล่นเสียงและส่งไปยังส่วนต่อไปทั้งยังแปลงสัญญาณคลื่นที่ดิบมาเป็นสัญญาณความถี่แบบเมลและส่งไปยังส่วนต่อไป

3.2.2 Numpy

ส่วนของการคำนวณทางคณิตศาสตร์ใช้ NumPy ซึ่งเป็น 1 ใน library ของ python ทำหน้าที่ เก็บข้อมูลสัญญาณดิจิตอลละส่งต่อไปยัง 2 ส่วน ส่วนของการทำกราฟและส่วนของฮาร์ดแวร์ ต่อไปนี้คือตัวอย่างที่นำมาใช้

3.2.2.1 คำสั่ง fft.rfftfreq

Syntax : numpy.fft.rfftfreq(n,d=1.0)

ทำการคืนค่า discrete fourier transform sample frequency เพื่อนำไปใช้กับ ฟังก์ชัน rfft และ irfft

Parameters:

n : int

ขนาดของ วินโดว์

d : scalar, optional

ความห่างกันของ sample (เป็นส่วนกลับของ sampling rate) มีค่าเริ่มต้นอัตโนมัติเป็น 1

Returns:

f : ndarray

อาเรย์ที่มีขนาดเป็น n//2 + 1 ซึ่งประกอบด้วย sample frequencies.

Examples

```
>>> signal = np.array([-2, 8, 6, 4, 1, 0, 3, 5, -3, 4], dtype=float)
>>> fourier = np.fft.rfft(signal)
>>> n = signal.size
>>> sample_rate = 100
>>> freq = np.fft.fftfreq(n, d=1./sample_rate)
>>> freq
array([  0.,  10.,  20.,  30.,  40., -50., -40., -30., -20., -10.])
>>> freq = np.fft.rfftfreq(n, d=1./sample_rate)
>>> freq
array([  0.,  10.,  20.,  30.,  40.,  50.])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 รูปแสดงตัวอย่างการใช้คำสั่ง numpy.fft.rfftfreq

3.2.2.2 คำสั่ง roll

Syntax : numpy.roll(a, shift, axis = none)

ทำการหมุน(Roll) อาเรย์ในแกนที่กำหนด โดยข้อมูลที่ถูก หมุนเกินตำแหน่งสุดท้าย จะกลับมาปรากฏใหม่ในตำแหน่งแรก

Parameters:

a : array_like

อาเรย์ที่จะทำการรับมาประมวลผล

shift : int or tuple of ints

ตัวเลขที่แสดงว่าจะทำการหมุนตรงไหน ฯ

axis : int or tuple of ints, optional

แกนที่เราจะทำการหมุน

Returns:

res : ndarray

ผลลัพธ์เป็นอาเรย์ ที่มีรูปร่างเหมือนกับ อาเรย์อินพุต(a)

Examples

```
>>> x = np.arange(10)                                          >>>
>>> np.roll(x, 2)
array([8, 9, 0, 1, 2, 3, 4, 5, 6, 7])

>>> x2 = np.reshape(x, (2,5))                                  >>>
>>> x2
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
>>> np.roll(x2, 1)
array([[9, 0, 1, 2, 3],
       [4, 5, 6, 7, 8]])
>>> np.roll(x2, 1, axis=0)
array([[5, 6, 7, 8, 9],
       [0, 1, 2, 3, 4]])
>>> np.roll(x2, 1, axis=1)
array([[4, 0, 1, 2, 3],
       [9, 5, 6, 7, 8]])
```

รูปที่ 3.4 รูปตัวอย่างการใช้งานคำสั่ง numpy.roll

3.2.2.3 คำสั่ง copy

Syntax : numpy.copy(a)

ทำการทำสำเนา object ที่เรากำหนด

Parameters:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a : array_like

ข้อมูลอินพุต

Returns:

arr : ndarray

สำเนาของข้อมูลชุด a

Examples

Create an array x, with a reference y and a copy z:

```
>>> x = np.array([1, 2, 3])
>>> y = x
>>> z = np.copy(x)
```

Note that, when we modify x, y changes, but not z:

```
>>> x[0] = 10
>>> x[0] == y[0]
True
>>> x[0] == z[0]
False
```

รูปที่ 3.5 รูปตัวอย่างการใช้คำสั่ง numpy.copy

3.2.2.4 คำสั่ง mean

Syntax : numpy.mean(a, axis=None, dtype=None, out=None, keepdims=<no value>)

ทำการหาค่าเฉลี่ยของชุดข้อมูล

Parameters:

a : array_like

อาเรย์ที่เราต้องการจะหาค่าเฉลี่ย ถ้าไม่เป็นอาเรย์ จะถูกแปลงให้เป็นอาเรย์

axis : None or int or tuple of ints, optional

แกนของอาเรย์ที่เราต้องการจะคำนวนค่าเฉลี่ย ถ้าไม่ได้ใส่ค่านี้ จะทำการแปลงอาเรย์เป็นมิติเดียวก่อน จึงคำนวน

dtype : data-type, optional

ชนิดของข้อมูลที่จะนำมาคำนวนค่าเฉลี่ย ถ้าข้อมูลที่จะนำมาคำนวนเป็น integer จะใช้ค่าเริ่มต้นที่ float64 ถ้าเป็น floating point จะมีค่าเป็นชนิดเดียวกับ floating point นั้นๆ

out : ndarray, optional

อาเรย์ที่จะนำมารับค่าอินพุต ถ้าไม่กำหนด คือไม่สร้างอาเรย์มารับค่า แต่ถ้ากำหนดค่า ต้องสัมพันธ์กับผลลัพธ์ที่คาดว่าจะออกมา

keepdims : bool, optional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Returns:

m : ndarray, see dtype parameter above

คืนค่าอาเรย์ที่มีค่าเฉลี่ยบรรจุอยู่ในนั้น

Examples

```
>>> a = np.array([[1, 2], [3, 4]])
>>> np.mean(a)
2.5
>>> np.mean(a, axis=0)
array([ 2.,  3.])
>>> np.mean(a, axis=1)
array([ 1.5,  3.5])
```

In single precision, mean can be inaccurate:

```
>>> a = np.zeros((2, 512*512), dtype=np.float32)
>>> a[0, :] = 1.0
>>> a[1, :] = 0.1
>>> np.mean(a)
0.54999924
```

Computing the mean in float64 is more accurate:

```
>>> np.mean(a, dtype=np.float64)
0.55000000074505806
```

รูปที่ 3.6 รูปตัวอย่างการใช้คำสั่ง numpy.mean

3.2.2.5 คำสั่ง around

Syntax : numpy.around(a, decimals=0, out=None)

ทำการปัดค่าตัวเลขที่ต้องการไปเป็นทศนิยมที่กำหนด

Parameters:

a : array_like

ข้อมูลอินพุต

decimals : int, optional

บอกว่าจะปัดเป็นทศนิยมกี่ตำแหน่ง

out : ndarray, optional

อาเรย์ที่เราต้องการนำไปรับค่าผลลัพธ์ ต้องมีรูปร่างเหมือนกับเอ้าท์พุต

Returns:

rounded_array : ndarray

คืนค่าอาเรย์ที่มีชนิดข้อมูลเดียวกับอินพุต ซึ่งมีค่าที่ถูกปัดบรรจุอยู่ในนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Examples

```
>>> np.around([0.37, 1.64])
array([ 0.,  2.])
>>> np.around([0.37, 1.64], decimals=1)
array([ 0.4,  1.6])
>>> np.around([.5, 1.5, 2.5, 3.5, 4.5]) # rounds to nearest even value
array([ 0.,  2.,  2.,  4.,  4.])
>>> np.around([1,2,3,11], decimals=1) # ndarray of ints is returned
array([ 1,  2,  3, 11])
>>> np.around([1,2,3,11], decimals=-1)
array([ 0,  0,  0, 10])
```

รูปที่ 3.7 รูปแสดงตัวอย่างการใช้คำสั่ง numpy.around

3.2.2.6 คำสั่ง linspace

Syntax : numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)

ทำการแบ่งช่วงที่กำหนดในพารามิเตอร์ start stop ออกมาเป็นตัวเลข ซึ่งจำนวนตัวเลขที่ออกมา จะมีจำนวนเท่ากลับ พารามิเตอร์ num โดยแต่ละตัวเลขจะมีความห่างเท่าๆกัน

Parameters:

start : array_like

ค่าเริ่มต้นของช่วงที่เราต้องการแบ่ง

stop : array_like

ค่าสุดท้ายของช่วงที่เราต้องการแบ่ง

num : int, optional

จำนวนตัวเลขที่เราต้องการจะแบ่งช่วงนั้นๆออกมาอย่างเท่ากัน เช่น ช่วง 2 ถึง 3 num = 5 จะแบ่งเป็น 2 2.25 2.5 2.75 3 ซึ่งก็คือมี 5 ตัวเลข ตามจำนวน num = 5

endpoint : bool, optional

ถ้าค่านี้ถูกตั้งเป็นเท็จ พารามิเตอร์ stop จะไม่ถูกรวมอยู่ในอาเรย์ตัวเลขที่ถูกแบ่งออกมา โดยปกติแล้ว ค่านี้เป็นจริง

retstep : bool, optional

ปกติเป็นเท็จ ถ้าตั้งค่าให้เป็นจริง จะแสดงให้ดูว่า แต่ละตัวเลขที่เป็นผลลัพธ์นั้นห่างกันเท่าไหร่

dtype : dtype, optional

ชนิดข้อมูลของเอ้าท์พุต ถ้าไม่ถูกกำหนด จะให้เป็นชนิดเดียวกับพารามิเตอร์อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Returns:

samples : ndarray

คืนค่าช่วงที่กำหนดในพารามิเตอร์ start stop ออกมาเป็นตัวเลข ซึ่งจำนวนตัวเลขที่ออกมา จะมีจำนวนเท่ากลับ พารามิเตอร์ num โดยแต่ละตัวเลขจะมีความห่างเท่าๆกัน

step : float, optional

คืนค่าช่วงที่ถูกแบ่งเท่าๆกันออกมาในกรณีที่พารามิเตอร์ retstep ถูกตั้งค่าให้เป็นจริง

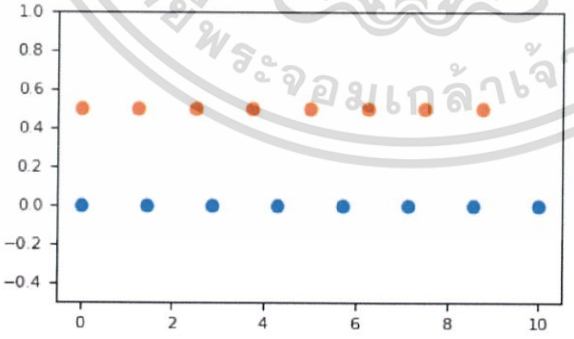
Examples

```
>>> np.linspace(2.0, 3.0, num=5)
array([ 2.  ,  2.25,  2.5 ,  2.75,  3.  ])
>>> np.linspace(2.0, 3.0, num=5, endpoint=False)
array([ 2. ,  2.2,  2.4,  2.6,  2.8])
>>> np.linspace(2.0, 3.0, num=5, retstep=True)
(array([ 2.  ,  2.25,  2.5 ,  2.75,  3.  ]), 0.25)
```

รูปที่ 3.8 รูปตัวอย่างแสดงการใช้งานคำสั่ง numpy.linspace

Graphical illustration:

```
>>> import matplotlib.pyplot as plt
>>> N = 8
>>> y = np.zeros(N)
>>> x1 = np.linspace(0, 10, N, endpoint=True)
>>> x2 = np.linspace(0, 10, N, endpoint=False)
>>> plt.plot(x1, y, 'o')
[<matplotlib.lines.Line2D object at 0x...>]
>>> plt.plot(x2, y + 0.5, 'o')
[<matplotlib.lines.Line2D object at 0x...>]
>>> plt.ylim([-0.5, 1])
(-0.5, 1)
>>> plt.show()
```



รูปที่ 3.9 รูปตัวอย่างแสดงการใช้งานคำสั่ง numpy.linspace ออกมาเป็นรูปกราฟเพื่อให้เห็นภาพได้ดียิ่งขึ้น

3.2.2.8 คำสั่ง tile

Syntax : numpy.tile(A, reps)

ทำการสร้างอาเรย์โดยการซ้ำค่า A เป็นจำนวน reps ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parameters:

A : array_like

อินพุตอาเรย์ที่เราต้องการ

reps : array_like

จำนวนครั้งที่ต้องการทำซ้ำ หรือ ใช้ค่าเป็นอาเรย์ก็ได้

Returns:

c : ndarray

คืนค่าอาเรย์ที่ถูกนำมาต่อกันแล้ว

Examples

```
>>> a = np.array([0, 1, 2])
>>> np.tile(a, 2)
array([0, 1, 2, 0, 1, 2])
>>> np.tile(a, (2, 2))
array([[0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2]])
>>> np.tile(a, (2, 1, 2))
array([[[0, 1, 2, 0, 1, 2]],
       [[0, 1, 2, 0, 1, 2]]])

>>> b = np.array([[1, 2], [3, 4]])
>>> np.tile(b, 2)
array([[1, 2, 1, 2],
       [3, 4, 3, 4]])
>>> np.tile(b, (2, 1))
array([[1, 2],
       [3, 4],
       [1, 2],
       [3, 4]])

>>> c = np.array([1,2,3,4])
>>> np.tile(c,(4,1))
array([[1, 2, 3, 4],
       [1, 2, 3, 4],
       [1, 2, 3, 4],
       [1, 2, 3, 4]])
```

รูปที่ 3.10 รูปแสดงตัวอย่างการใช้คำสั่ง numpy.tile

3.2.2.9 คำสั่ง numpy.amax

Syntax : numpy.amax(a, axis=None, out=None)

ฟังก์ชันนี้คือการหาค่า max ของตัวเลขใน แกน(axis) ที่กำหนด ถ้าไม่ได้กำหนดแกน อาเรย์จะถูกทำให้เป็นอาเรย์หนึ่งมิติก่อน แล้วจึงนำอาเรย์ที่ถูกทำเป็นหนึ่งมิตินั้นมาหาค่า max

Parameters:

a : array_like

ข้อมูลอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

axis : None or int or tuple of ints, optional

แกนที่เราต้องการจะนำข้อมูลมาคำนวนหาค่ามากที่สุด ถ้าไม่กำหนดพารามิเตอร์นี้ อาเรย์จะถูกทำให้เป็นหนึ่งมิติก่อน แล้วค่อยคำนวน แต่ถ้าข้อมูลเป็น tuple ของ int ค่า max จะถูกเลือกมาจาก ทุกๆแกน ไม่ได้ทำเป็น 1 มิติก่อน

out : ndarray, optional

ถ้าจะกำหนดอาเรย์เอาท์พุต ต้องกำหนดให้มีรูปร่างเดียวกัน และ ความยาวเท่ากัน กับผลลัพธ์ที่คาดว่าจะได้ออกมา ต้องไปดูรายละเอียดในไฟล์ doc.ufuncs

Returns:

amax : ndarray or scalar

คืนค่าสูงสุดของอินพุต a

Examples

```
>>> a = np.arange(4).reshape((2,2))
>>> a
array([[0, 1],
       [2, 3]])
>>> np.amax(a)           # Maximum of the flattened array
3
>>> np.amax(a, axis=0)   # Maxima along the first axis
array([2, 3])
>>> np.amax(a, axis=1)   # Maxima along the second axis
array([1, 3])
```

```
>>> b = np.arange(5, dtype=float)
>>> b[2] = np.NaN
>>> np.amax(b)
nan
>>> np.nanmax(b)
4.0
```

You can use an initial value to compute the maximum of an empty slice, or to initialize it to a different value:

```
>>> np.max([[-50], [10]], axis=-1, initial=0)
array([ 0, 10])
```

Notice that the initial value is used as one of the elements for which the maximum is determined, unlike for the default argument Python's max function, which is only used for empty iterables.

```
>>> np.max([5], initial=6)
6
>>> max([5], default=6)
5
```

รูปที่ 3.11 รูปตัวแสดงอย่างการใช้คำสั่ง numpy.amax

3.2.2.13 คำสั่ง ceil

Syntax : numpy.ceil(x)

Parameters :

X: array_like

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอินพุต เป็นอาเรย์

Returns :

Y:ndarray or scalar

ค่าจำนวนเต็มต่ำที่สุดที่ใกล้ตัวเลขในอาเรย์อินพุตที่สุด ทีละตัว

Examples

```
>>> a = np.array([-1.7, -1.5, -0.2, 0.2, 1.5, 1.7, 2.0])
>>> np.ceil(a)
array([-1., -1., -0.,  1.,  2.,  2.,  2.])
```

รูปที่ 3.12 รูปแสดงตัวอย่างการใช้ numpy.ceil

3.2.3 PyQtGraph

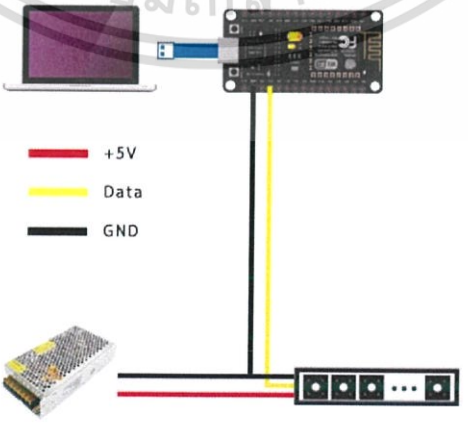
ส่วนของการสร้างกราฟ ใช้ PyQtGraph ซึ่งเป็น 1 ใน library ของ python ทำหน้าที่สร้างและแสดงผลกราฟ ในส่วนของ GUI เพื่อแสดงผลให้ user

รูปที่ 3.13 บล็อคไดอะแกรมส่วนของซอฟท์แวร์

3.3 การออกแบบส่วนของฮาร์ดแวร์

ส่วนของฮาร์ดแวร์ นั้นจะทำการแสดงผลของส่วนซอฟต์แวร์ที่ได้ประมวลผลมาแล้ว ซึ่งจะใช้ LED ในการแสดงผลออกมาแบบ real time และ responsive ต่อเสียง

3.3.1 การออกแบบวงจรวิชวลไลเซชัน



รูปที่ 3.14 การออกแบบวงจรวิชวลไลเซชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การใช้งานและการทดลอง

4.1 การทดลองตัวแสดงผลวิชวลไลเซชันแบบซอฟท์แวร์ที่ได้รับความถี่ต่างกัน

จุดประสงค์

เพื่อศึกษาแนวโน้มการเปลี่ยนแปลงของระดับความถี่ที่ได้รับเมื่อความมถี่มีการเพิ่มมากขึ้นเป็นไปตามทฤษฎีหรือไม่

สมมุติฐาน

กราฟที่แสดงจะเพิ่มขึ้นตามความถี่ที่ได้รับ

อุปกรณ์

คอมพิวเตอร์ 1 เครื่อง

วิธีการทดลอง

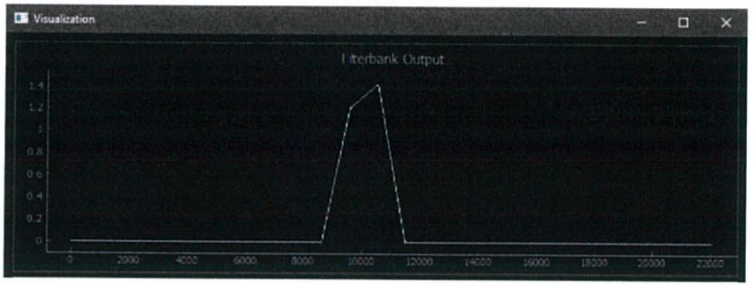
1.ปรับการตั้งค่าอินพุทธ์เสียงให้เป็น Stereo Mix
2.ทำการโปรแกรมการทำงานในส่วนของซอฟท์แวร์
3.ทำการเปิดเสียง frequency test
4.ดูผลการแสดง จากกราฟบน GUI

รูปที่ 4.1 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์

ผลการทดลอง

กราฟที่ขึ้นสัมพันธ์กับความถี่ที่ได้รับโดยความถี่ที่ได้นั้นมากจากสมการฟิลเตอร์แบงค์ที่นำมาคำนวณแสดงดังรูปที่ 4.2 และ รูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์เมื่อใช้ความถี่ 2000 Hz

รูปที่ 4.3 ภาพของโปรแกรมวิชวลไรเซชันแสดงความถี่ฟิลเตอร์แบงค์เมื่อใช้ความถี่ 2000 และ 4000 Hz

สรุปผลการทดลอง

จากรูปที่ 4.2 และ 4.4 ระดับของความถี่ที่ได้ในกราฟมีความสัมพันธ์กับความถี่ที่นำมาทำการคำนวณตามทฤษฎี

4.2 การทดลองรูปแบบการแสดงผลวิชวลไลเซชันแบบซอฟท์แวร์

จุดประสงค์

เพื่อให้มีความหลากหลายในการแสดงผลวิชวลไลเซชันและเพื่อให้มีความเข้าใจง่ายขึ้นในการดูกราฟความถี่

สมมุติฐาน

รูปแบบใหม่ๆของการแสดงผลซึ่งนำกราฟที่ผ่านการฟิลเตอร์แบงค์มาเพิ่มลูกเล่นให้หลากหลายมากขึ้น ซึ่งกราฟที่นำมาแสดงนั้นแบ่งเป็น 3 ช่วงจากกราฟที่ผ่านฟิลเตอร์แบงค์ แบ่งเป็นกราฟช่วงความถี่ต่ำ ใช้กราฟเป็นสีแดง กราฟช่วงความถี่กลาง ใช้กราฟเป็นสีเขียว กราฟช่วงความถี่สูง ใช้กราฟเป็นสีฟ้า และสามารถเลือกขนาดของช่วงที่จะแบ่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

คอมพิวเตอร์ 1 เครื่อง

วิธีการทดลอง

1.ปรับการตั้งค่าอินพุทธ์เสียงให้เป็น Stereo Mix

2.ทำการเปิดโปรแกรมในส่วนของซอฟท์แวร์

3.เลือกขนาดของช่วงความถี่ที่เราต้องการแบ่งช่วง

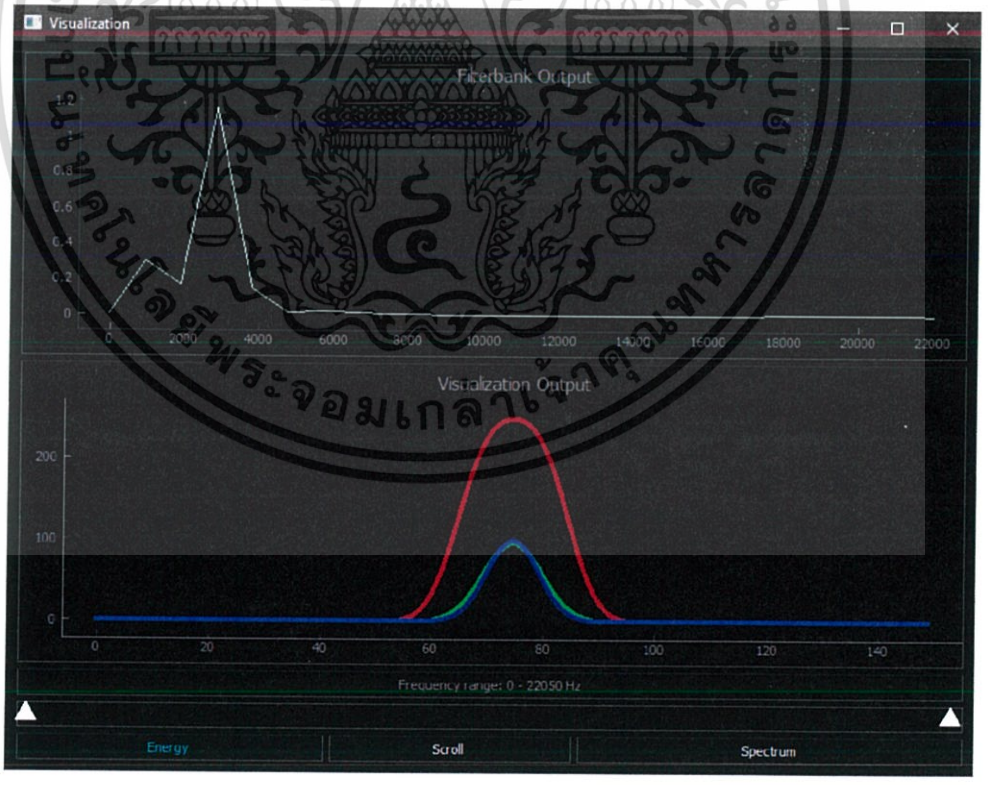
4.ทำการเปิดไฟล์เสียง

5.เลือกรูปแบบการแสดงผลของกราฟ

5.ดูผลการแสดง จากกราฟบน GUI

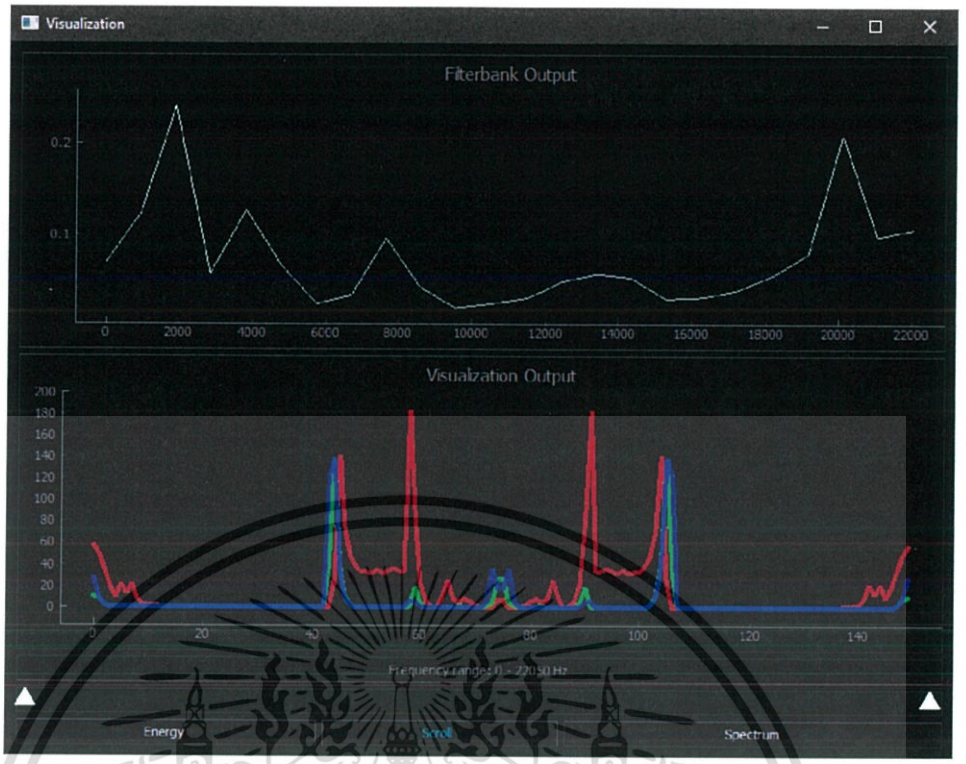
ผลการทดลอง

เมื่อทำการทดลองแล้วจะได้กราฟความถี่สามรูปแบบดังรูปที่ 4.4 , 4.5 และ 4.6 จะเห็นได้ว่ากราฟ Visualization Output ได้แบ่งความถี่ของกราฟ Filterbank Output แบ่งเป็น 3 ช่วงเท่าๆกัน

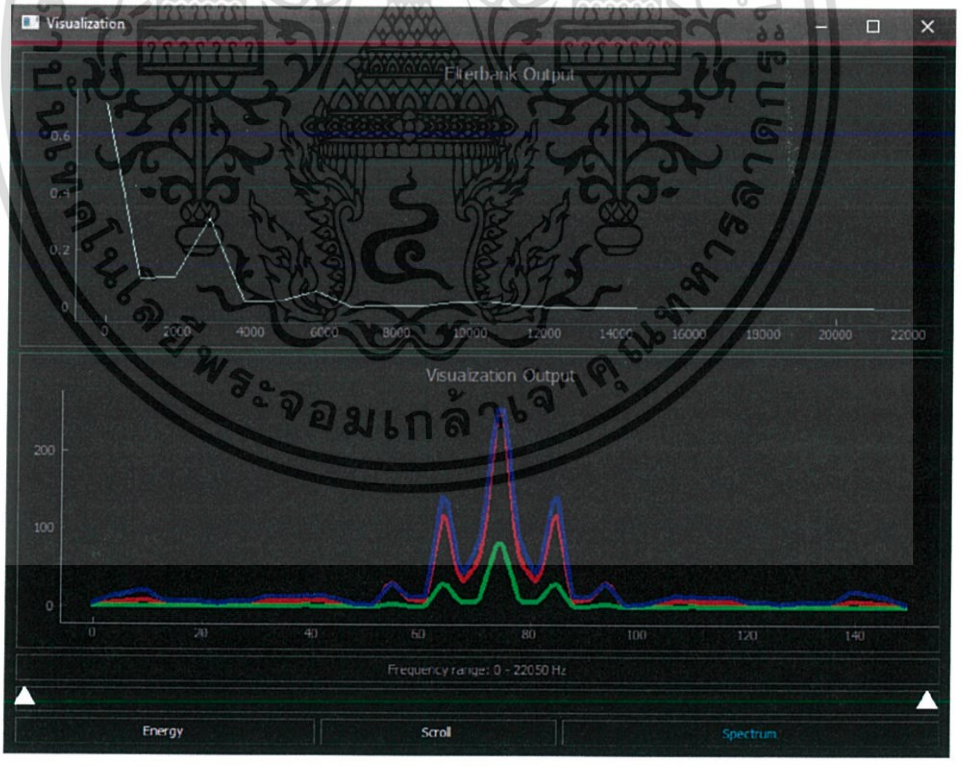


รูปที่ 4.4 รูปแบบการแสดงวิชวลไรเซชันแบบ Energy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 รูปแบบการแสดงวิชวลไรเซชันแบบ Scroll



รูปที่ 4.6 รูปแบบการแสดงวิชวลไรเซชันแบบ Spectrum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

เมื่อทำการทดลองแล้วพบว่า วิชวลไรเซชัน มีรูปแบบที่สวยขึ้นและหลากหลายขึ้นถึงแม้จะมีข้อผิดพลาดอยู่บ้างแต่ยอมรับได้เพราะฉะนั้น วิชวลไรเซชันนี้สามารถใช้งานได้จริง

4.3 การทดลองรูปแบบการแสดงผลวิชวลไลเซชันแบบฮาร์ตแวร์

จุดประสงค์

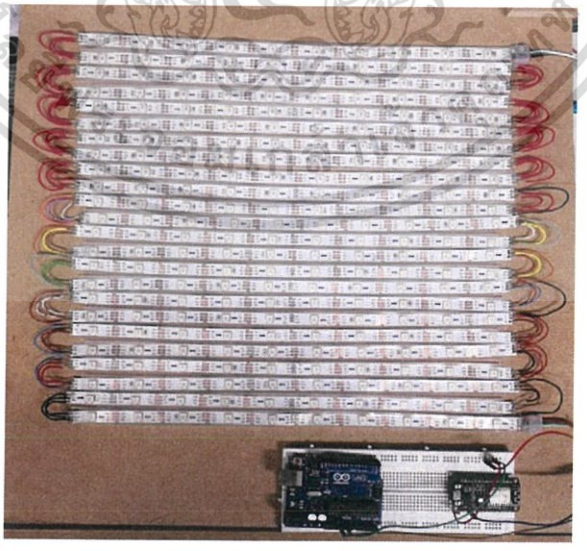
เพื่อแสดงรูปแบบที่ปรากฏในวิชวลไลเซชันแบบซอฟท์แวร์ให้มาเป็นแบบฮาร์ตแวร์และสามารถนำไปใช้ในชีวิตจริงได้

สมมุติฐาน

ตัว LED ต้องมีสีที่สัมพันธ์กับตัววิชวลไรเซชันแบบซอฟท์แวร์และแสดงผลออกมาได้แบบเรียลไทม์ และชัดเจนไม่ผิดเพี้ยนไป

อุปกรณ์

บอร์ดไมโครคอนโทรลเลอร์อีเอสพี 8266	1 ตัว
คอมพิวเตอร์	1 เครื่อง
WS2812B 5050 RGB LED STRIP	250 ดวง
แหล่งจ่ายไฟ 5 V	1 เครื่อง



รูปที่ 4.7 ส่วนแสดงวิชวลไลเซชันแบบฮาร์ตแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทดลอง

1.อัพโหลดโปรแกรมการทำงานให้อีเอสพี 8266 ทำหน้าที่ในการรับข้อมูลจากคอมพิวเตอร์เพื่อควบคุม LED และให้อีเอสพี 8266 เป็นตัวควบคุมการทำงานของ LED ทั้งหมด

2.ทำการติดตั้ง LED กับแหล่งจ่ายไฟ 5 v เพื่อจ่ายไฟให้ LED

3.ทำการติดตั้ง อีเอสพี 8266 เข้ากับคอมพิวเตอร์โดยผ่าน Wifi

4.ปรับการตั้งค่าอินพุทธ์เสียงให้เป็น Stereo Mix

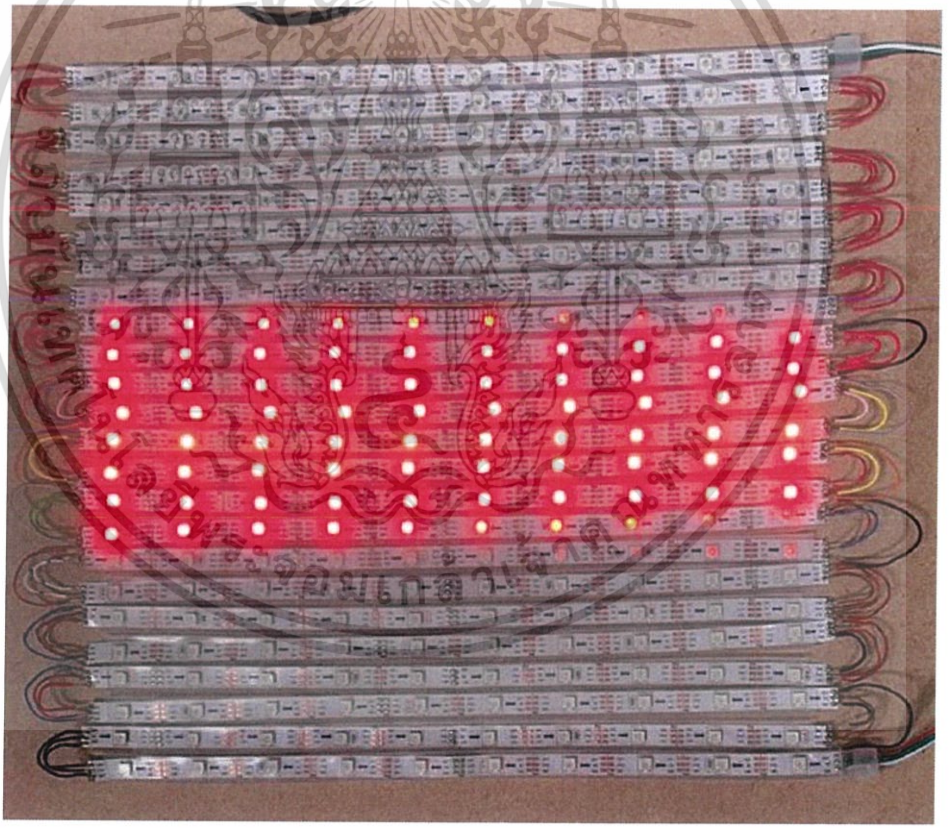
5.เปิดโปรแกรมแสดงผลวิชวลไรเซชันในคอมพิวเตอร์ขึ้นมาเพื่อประมวณผลเสียง

6.เปิดไฟล์เสียง

7.ดูผลการแสดงบน LED และบันทึกผล

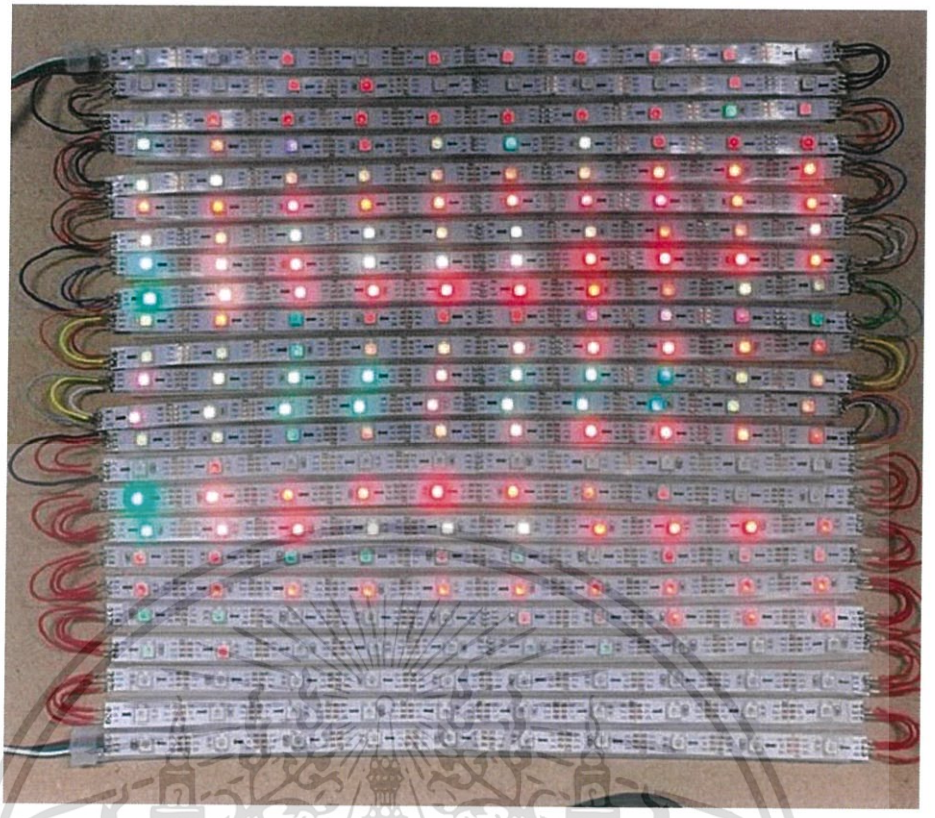
ผลการทดลอง

LED จะเปลี่ยนแสดงผลต่างๆตามวิชวลไรเซชัน แสดงได้ดังรูปด้านล่าง

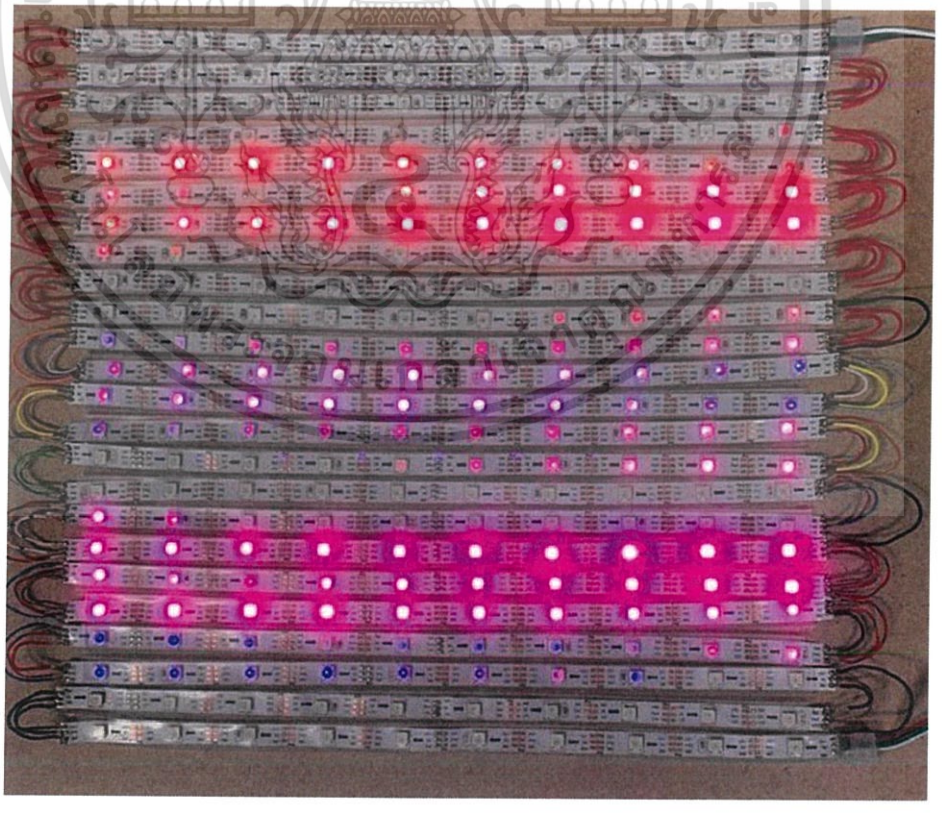


รูปที่ 4.8 จอ LED แสดงวิชวลไรเซชันแบบ Energy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 จอ LED แสดงวิชวลไรเซชันแบบ Scroll



รูปที่ 4.10 จอ LED แสดงวิชวลไรเซชันแบบ Spectrum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

เมื่อทำการทดลองแล้วจะเห็นได้ว่าสีของLED จะแสดงผลรวมของสีในกราฟ Visualization Output และความสว่างของแต่ละดวงขึ้นอยู่กับความดัง ของความถี่ในช่วงนั้นๆด้วย

4.4 การทดลองรับเสียงด้วยไมค์โครโฟน

จุดประสงค์

เพื่อรับเสียงจากไมค์โครโฟนและประมวณผลแสดงเป็นวิชวลไรเซชันแบบเรียลไทม์

สมมุติฐาน

สามารถแสดงผลได้เหมือนกันกับการใช้ไฟล์เสียง

อุปกรณ์

คอมพิวเตอร์	1 เครื่อง
ไมโครโฟน	1 ตัว

วิธีการทดลอง

1.ปรับการตั้งค่าอินพุทธ์เสียงให้เป็น Microphone
2.ทำการเปิดโปรแกรมในส่วนของซอฟท์แวร์
3.เลือกขนาดของช่วงความถี่ที่เราต้องการแบ่งช่วง
4.ทำการพูดใส่ไมโครโฟน
5.เลือกรูปแบบการแสดงผลของกราฟ
5.ดูผลการแสดง จากกราฟบน GUI

ผลการทดลอง

กราฟที่ขึ้นจะแสดงไปตามความถี่ของเสียงพูดดังรูปที่ 4.

สรุปผลการทดลอง

เมื่อทำการทดลองแล้วจะพบว่าวิชวลไรเซชันสามารถแสดงผลของเสียงเราได้ไม่ต่างกับแสดงผลจากไฟล์เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุปผล

5.1 บทสรุป

จากการศึกษาทดลองสามารถแบ่งออกเป็น 2 กรณี ดังนี้

5.1.1 การแสดงผลวิชวลไรเซชันแบบซอฟท์แวร์

คอมพิวเตอร์จะต้องมีประสิทธิภาพในการแสดงผลที่ดี ดังนั้นระยะเวลาการทำงานของโปรแกรมจึงเป็นปัจจัยหลักสำหรับการออกแบบ และพัฒนา อุปกรณ์ที่ประมวณผลข้อมูลๆต่างๆบนคอมพิวเตอร์ ซึ่งในส่วนของซอฟท์แวร์นั้นต้องส่งค่าต่อไปให้ยังส่วนของฮาร์ดแวร์

5.1.2 การแสดงผลวิชวลไรเซชันแบบฮาร์ดแวร์

เนื่องจากเป็นอุปกรณ์ที่ต้องอาศัย อินเตอร์เน็ตจึงต้องอาศัยอินเตอร์เน็ตพอสมควรเพราะส่งผลให้เกิดผลกระทบอย่างสำคัญต่อการแสดงผลบนจอ LED นอกจากนั้นแล้วบริเวณที่มีสัญญาณวายฟายอื่น สัญญาณวายฟายเหล่านั้นจะรบกวนและทำให้ระดับความเข้มของสัญญาณที่ส่งไปเพื่อการแสดงผลได้รับความผิดเพี้ยน นอกจากนั้นแล้วการบัดกรีจอ LED และสายไฟหรือแม้กระทั่งการจ่ายไฟนั้นยังมีผลกระทบต่อการแสดงผลอีกด้วย

จากปัญหาที่กล่าวมาข้างต้นสามารถแก้ไข และลดผลกระทบโดยการเชื่อมต่ออินเตอร์เน็ตที่มีความเสถียร ใช้แหล่งผลิตไฟสำหรับ LED โดยเฉพาะและตรวจเช็คว่ามีอะไรบกพร่องกับ LED จะทำให้การแสดงผลนั้นมีความน่าเชื่อถือ ไม่ผิดเพี้ยน เหมาะแก่การนำไปใช้ในชีวิตประจำวัน

อย่างไรก็ตามปัญหาที่เกิดไม่สามารถแก้ไขได้อย่างสมบูรณ์ทำได้เพียงลดผลกระทบเท่านั้น

5.2 ปัญหาและอุปสรรคในการทำงาน

- สัญญาณอินเตอร์เน็ตถ้าไม่เสถียรพอ การแสดงผลก็จะมีการผิดเพี้ยน
- คอมพิวเตอร์ไม่มีประสิทธิภาพมากพอสำหรับพัฒนาโปรแกรมส่งผลให้เกิดความล่าช้าในการพัฒนา
- การบัดกรี LED ถ้าบัดกรีไม่ดีสาย LED จะไหม้และเกิดการผิดเพี้ยน
- การศึกษาทฤษฎีเกี่ยวกับเสียงที่เกี่ยวข้องใช้เวลานาน
- ข้อจำกัดของโปรแกรมเกี่ยวกับสารแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

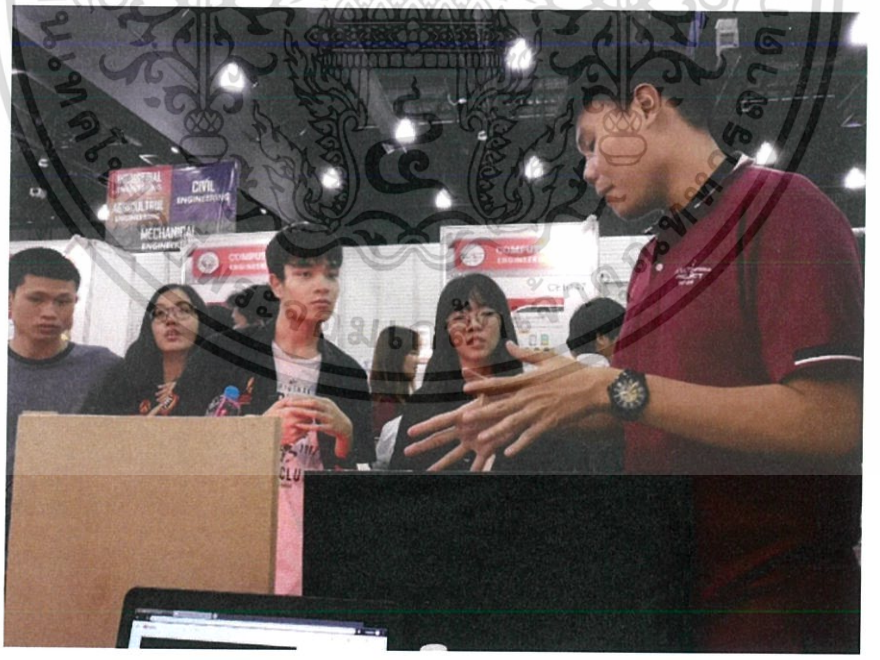
5.3 สิ่งที่สามารถนำไปพัฒนาต่อในอนาคต

- เพิ่มจำนวน LED เพื่อเพิ่มพื้นที่ในการแสดงผล
- เพิ่มรูปแบบในการแสดงผลให้มากขึ้น
- เพิ่มความแข็งแรงทนทานให้แก่ LED
- พัฒนาฮาร์ดแวร์ให้นำไปใช้ในชีวิตประจำวันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ภาคผนวก

รูปที่ 6.1 แสดงผลงานวันโปรเจคเดย์วันแรก 1



รูปที่ 6.2 แสดงผลงานวันโปรเจคเดย์วันแรก 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3 แสดงผลงานวันโปรเจคเดย์วันแรก 3

รูปที่ 6.4 แสดงผลงานวันโปรเจคเดย์วันที่สอง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

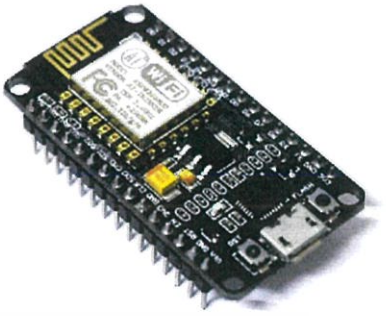
รูปที่ 6.5 แสดงผลงานวันโปรเจคเดย์วันที่สอง 2

รูปที่ 6.6 แสดงผลงานวันโปรเจคเดย์วันที่สอง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

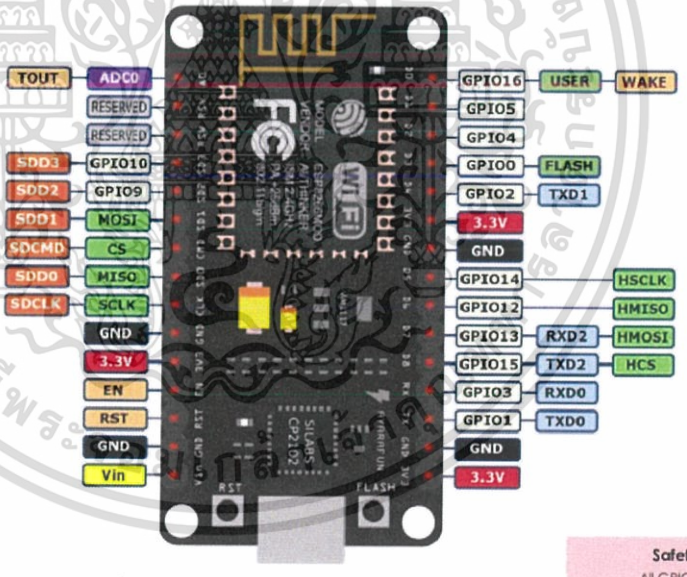


Features

- Version : DevKit v1.0
- Breadboard Friendly
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.

PINOUT DIAGRAM

NodeMCU ESP8266 v1.0



Safety Precaution
All GPIO runs at 3.3V !!

Source
https://iotbytes.wordpress.com/nodemcu-pinout/

รูปที่ 6.7 ESP8266 Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Haytham Fayek (2016). Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between : Retrieved from https://haythamfayek.com/

[2] Scott Lawson (2017). Real-time LED strip :Retrieved from https://github.com/scottlawsonbc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้