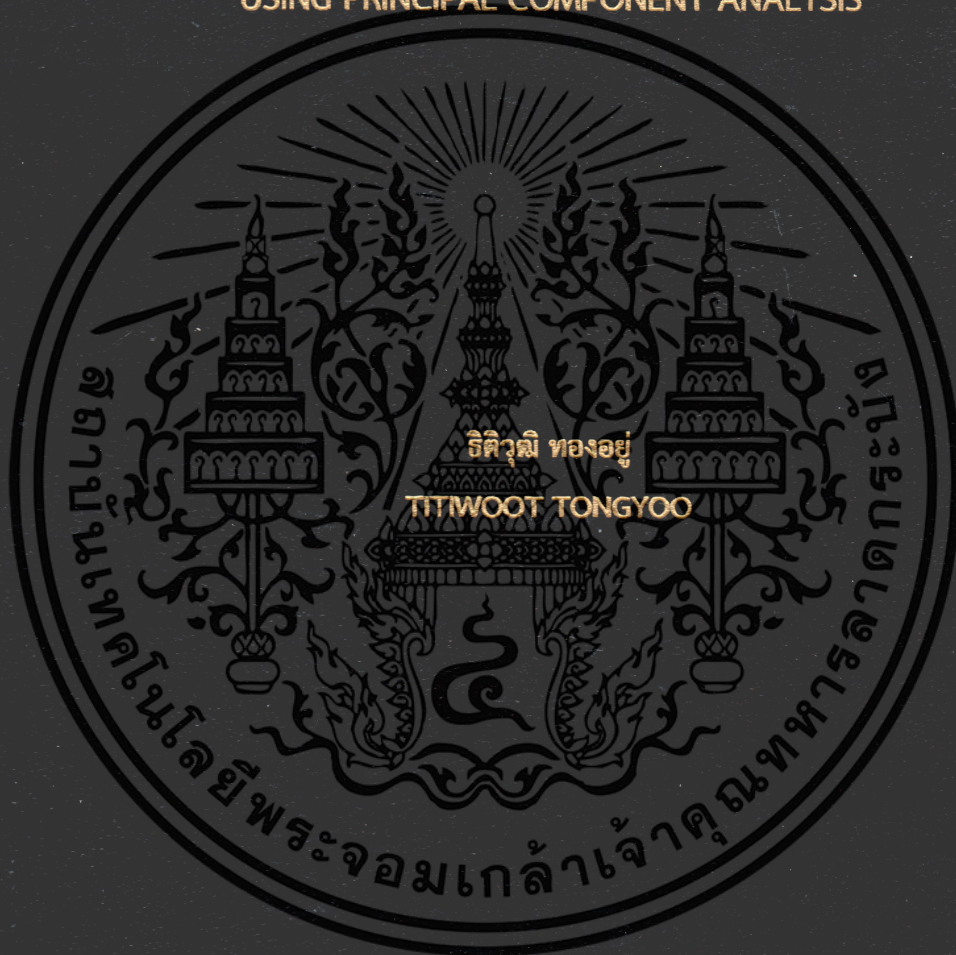


การพัฒนาแบบแยกแยะกลิ่นสำหรับโมเลกุลของกลิ่นที่มีความสามารถ
ในการระเหยปานกลางและต่ำโดยใช้การวิเคราะห์องค์ประกอบหลัก

DEVELOPMENT OF ODOR CLASSIFICATION SYSTEM
FOR MODERATE AND LOW VOLATILE ODOR MOLECULES
USING PRINCIPAL COMPONENT ANALYSIS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

KMITL-2019-EN-M-040-005

การพัฒนาระบบแยกแยะกลิ่นสำหรับโมเลกุลของกลิ่นที่มีความสามารถ
ในการระเหยปานกลางและต่ำโดยใช้การวิเคราะห์องค์ประกอบหลัก

DEVELOPMENT OF ODOR CLASSIFICATION SYSTEM
FOR MODERATE AND LOW VOLATILE ODOR MOLECULES
USING PRINCIPAL COMPONENT ANALYSIS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาระบบแยกแยะกลิ่นสำหรับโมเลกุลของกลิ่นที่มีความสามารถ
ในการระเหยปานกลางและต่ำโดยใช้การวิเคราะห์องค์ประกอบหลัก

DEVELOPMENT OF ODOR CLASSIFICATION SYSTEM
FOR MODERATE AND LOW VOLATILE ODOR MOLECULES
USING PRINCIPAL COMPONENT ANALYSIS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPMENT OF ODOR CLASSIFICATION SYSTEM
FOR MODERATE AND LOW VOLATILE ODOR MOLECULES
USING PRINCIPAL COMPONENT ANALYSIS



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

2019

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2019

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การพัฒนาระบบแยกแยะกลิ่นสำหรับโมเลกุลของกลิ่นที่มีความสามารถในการระเหยปานกลางและต่ำโดยใช้การวิเคราะห์องค์ประกอบหลัก
นักศึกษา	นายธิตวุฒิ ทองอยู่
รหัสประจำตัว	59601246
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
พ.ศ.	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.ยศศิริ อาริยะกุล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอเกี่ยวกับการพัฒนาระบบแยกแยะกลิ่นสำหรับแยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยปานกลางและต่ำโดยใช้วิธีวิเคราะห์องค์ประกอบหลัก ระบบดังกล่าวประกอบด้วยอาร์เรย์ของเซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์ วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง หน่วยประมวลผลและอัลกอริทึมสำหรับแยกแยะกลิ่น อย่างไรก็ตามการแยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยต่ำนั้นทำได้ยาก เนื่องจากมีความเข้มข้นของกลิ่นในอากาศน้อย ในวิทยานิพนธ์ฉบับนี้จึงจะใช้วิธีการคัดเลือกคุณสมบัติของข้อมูลที่ได้จากการทดลองวัด มาทำการวิเคราะห์องค์ประกอบหลักอย่างเหมาะสม ซึ่งคาดว่าจะสามารถทำให้แยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยต่ำได้ดีขึ้น โดยได้ทำการทดลองเพื่อเปรียบเทียบความถูกต้องของการแยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยปานกลางและต่ำโดยใช้ระบบแยกแยะกลิ่นที่พัฒนาขึ้น ผลการทดลองชี้ให้เห็นว่าระบบสามารถแยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยปานกลาง ได้แก่ อะซีโตน แอมโมเนีย เอทานอล และเฮกเซน ได้ อัตราความถูกต้องเฉลี่ย 100%, 100%, 100% และ 80% ตามลำดับ เมื่อทำการวิเคราะห์องค์ประกอบหลักโดยใช้ค่าแอมพลิจูดเชิงสัมพันธ์ และสามารถแยกแยะกลิ่นที่โมเลกุลมีความสามารถในการระเหยต่ำ ได้แก่ มะกรูด ยูคาลิปตัส สระระแห่น และกระดังงา ได้ อัตราความถูกต้องเฉลี่ย 100%, 75%, 44.44% และ 33.33% ตามลำดับ เมื่อทำการวิเคราะห์องค์ประกอบหลักโดยใช้ค่าแอมพลิจูดเชิงสัมบูรณ์

Thesis	Development of odor classification system for moderate and low volatile odor molecules using principal component analysis
Student	Mr. Titiwoot Tongyoo
Student ID.	59601246
Degree	Master of Engineering
Program	Electronics Engineering
Year	2019
Thesis Advisor	Dr. Yossiri Ariyakul

ABSTRACT

This thesis presents about development of odor classification system for moderate and low volatile odor molecules using principal component analysis. The system consists of metal oxide sensor array, related electronic circuits, processor, and algorithms for odor classification. However, identifying low volatile odor molecules is difficult. Due to the low concentration of odor in the air. In this thesis, principal component analysis based on selection of the attributes from the measured data was studied. The method is expected to improve the identification of the low volatile odor molecules. The experiments were conducted to compare the accuracy of the identification of the moderate and low volatile odor molecules by using proposed odor classification system. The experimental results revealed that the accuracy of classifying moderate volatile odor molecules which are acetone, ammonia, ethanol, and hexane are 100%, 100%, 100%, and 80%, respectively, when principal component analysis based on relative amplitude is utilized and the accuracy classifying low volatile odor molecules which are acetone, ammonia, ethanol, and hexane are 100%, 100%, 100%, and 80%, respectively, when principal component analysis based on absolute amplitude is utilized.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร.ยศศิริ อาริยะกุล และคณาจารย์ในภาควิชาวิศวกรรมอิเล็กทรอนิกส์ ที่ให้ความรู้ ให้คำแนะนำ ให้ความช่วยเหลือจนกระทั่งผลงานสำเร็จลุล่วง ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของวิทยานิพนธ์ รวมถึงผู้มีพระคุณทุกท่านที่ได้เอื้อยยามไว้ ณ ที่นี้

และขอกราบขอบพระคุณผู้ปกครองที่อำนวยความสะดวกในการอุปการะทุนค่าใช้จ่ายในการเล่าเรียนจนกระทั่งผลงานสำเร็จลุล่วง

นายธิตวุฒิ ทองอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย.....	1
1.4 ผลที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 กลิ่น.....	3
2.2 การรับรู้กลิ่นของมนุษย์.....	5
2.3 ระบบแยกแยะกลิ่น.....	6
2.4 เซ็นเซอร์กลิ่น.....	7
2.4.1 เซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์.....	8
2.4.2 เซ็นเซอร์กลิ่นชนิดควออร์ตคริสตัลไมโครบาลานซ์.....	9
2.4.3 เซ็นเซอร์กลิ่นชนิดโพลิเมอร์นำไฟฟ้า.....	13
2.4.4 เซ็นเซอร์กลิ่นที่ใช้หมวดของแมลง.....	14
2.5 การแยกแยะกลิ่น.....	15
2.5.1 โพลาร์โรสฟลोट.....	15
2.5.2 การวิเคราะห์คลัสเตอร์ตามลำดับชั้น.....	15
2.5.3 การแปลงแบบคาสูเนนโลเอฟ.....	17
2.5.4 การวิเคราะห์ห้องค์ประกอบหลัก.....	18
2.6 FPGA.....	21
บทที่ 3 วิธีการดำเนินการวิจัย.....	22
3.1 ระบบรับรู้กลิ่น.....	22
3.2 วงจรสำหรับใช้งานเซ็นเซอร์.....	23
3.3 หน่วยประมวลผลแบบอ่อนที่พัฒนาขึ้น.....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.4 โปรแกรมสำหรับติดต่อกับผู้ใช้ที่พัฒนาขึ้น.....	26
3.5 อัลกอริทึมสำหรับดึงคุณสมบัติที่สนใจออกจากผลการทดลองที่พัฒนาขึ้น.....	35
3.5.1 อัลกอริทึมสำหรับหาแอมพลิจูดเชิงสัมบูรณ์.....	35
3.5.2 อัลกอริทึมสำหรับหาแอมพลิจูดเชิงสัมพัทธ์.....	37
3.5.3 อัลกอริทึมสำหรับหาค่าคงที่เวลา.....	39
3.5.4 อัลกอริทึมสำหรับหาความชัน.....	40
3.5.5 อัลกอริทึมสำหรับนอร์มัลไลเซชัน.....	41
3.6 การวิเคราะห์ห้วงค์ประกอบหลักโดยใช้โปรแกรม MATLAB.....	43
3.6.1 การวิเคราะห์ห้วงค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม.....	43
3.6.2 การวิเคราะห์ห้วงค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์.....	45
บทที่ 4 การทดลองและวิเคราะห์ผล.....	49
4.1 การทดลองวัดคลื่นที่มีอัตราการระเหยแตกต่างกัน.....	49
4.1.1 การทดลองวัดคลื่นที่มีอัตราการระเหยปานกลาง.....	50
4.1.2 การทดลองวัดคลื่นที่มีอัตราการระเหยต่ำ.....	52
4.2 การแยกแยะคลื่น.....	54
4.2.1 การแยกแยะคลื่นที่มีอัตราการระเหยปานกลาง.....	54
4.2.1.1 ผลการวิเคราะห์ห้วงค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม.....	55
4.2.1.2 ผลการวิเคราะห์ห้วงค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์.....	56
4.2.2 การแยกแยะคลื่นที่มีอัตราการระเหยต่ำ.....	60
4.2.2.1 ผลการวิเคราะห์ห้วงค์ประกอบหลักโดยเมทริกซ์ความแปรปรวนร่วม.....	60
4.2.2.2 ผลการวิเคราะห์ห้วงค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์.....	62
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	67
5.1 ผลการแยกแยะคลื่นที่มีอัตราการระเหยปานกลาง.....	67
5.2 ผลการแยกแยะคลื่นที่มีอัตราการระเหยต่ำ.....	67
5.3 ข้อเสนอแนะ.....	67
เอกสารอ้างอิง.....	68
ภาคผนวก.....	72
ประวัติผู้เขียน.....	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 อัตราความถูกต้องในการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง.....	60
4.2 อัตราความถูกต้องในการแยกแยะกลิ่นที่มีอัตราการระเหยต่ำ.....	66



สารบัญรูป

รูปที่	หน้า
2.1 ลักษณะของตัวรับคลื่นที่เกี่ยวข้องกับคลื่นแต่ละประเภท (ขนาดในหน่วย Angstrom) [2]	4
2.2 ระบบรับรู้คลื่นของมนุษย์ [3]	6
2.3 บล็อกไดอะแกรมของระบบแยกแยะคลื่น	6
2.4 กำแพงศักย์ที่เกิดขึ้นระหว่างอนุภาคของเมทัลออกไซด์ชนิดเอ็น (ก) การเกิดกำแพงศักย์เมื่อสัมผัสกับอากาศที่มีออกซิเจน (ข) การเกิดกำแพงศักย์เมื่อสัมผัสกับอากาศที่มีแก๊สรีดิวส์	9
2.5 เซ็นเซอร์ชนิดเมทัลออกไซด์ที่มีขายในเชิงพาณิชย์ [17]	9
2.6 ตัวอย่างวงจรใช้งานของเซ็นเซอร์ชนิดเมทัลออกไซด์ TGS2600 ของบริษัท Figaro Engineering [17]	9
2.7 ภาพตัดขวางของเซ็นเซอร์ชนิดควอร์ตคริสตัลไมโครบาลานซ์ [18]	10
2.8 วงจรสมมูลทางไฟฟ้าของเซ็นเซอร์คลื่นชนิดควอร์ตคริสตัลไมโครบาลานซ์ [18]	10
2.9 ตัวอย่างการต่อวงจรรวมสำหรับใช้งานเซ็นเซอร์ชนิดควอร์ตคริสตัลไมโครบาลานซ์ [19]	11
2.10 ฐานรองสำหรับสร้างเซ็นเซอร์ชนิดโพลิเมอร์นำไฟฟ้า [18]	13
2.11 วงจรขยายสัญญาณที่เชื่อมต่อกับหนดของพีเอ็ลลกลางคืน [18]	14
2.12 ตัวอย่างโพลาร์โรสพล็อตจากข้อมูลเซ็นเซอร์ 6 ตัว	15
2.13 ตัวอย่างการวิเคราะห์คลัสเตอร์ตามลำดับชั้น [18]	16
2.14 ตัวอย่างการแปลงแบบคาฮูเนนโลเอฟ [18]	18
2.15 ตัวอย่างผลการวิเคราะห์องค์ประกอบหลักมิติของข้อมูลเหลือ 2 มิติ	20
3.1 บล็อกไดอะแกรมของระบบรับรู้คลื่น	22
3.2 ภาพตัวเครื่องของระบบรับรู้คลื่น [1]	22
3.3 วงจรไฟฟ้าที่เชื่อมต่อกับเซ็นเซอร์คลื่นชนิดเมทัลออกไซด์ [1]	24
3.4 องค์ประกอบของระบบประมวลผล	25
3.5 โปรแกรมบันทึกค่าแรงดันไฟฟ้าจากเซ็นเซอร์	26
3.6 โพล์ชาร์ตของฟังก์ชัน Connect	27
3.7 โพล์ชาร์ตของฟังก์ชัน Start	27
3.8 โพล์ชาร์ตของฟังก์ชัน Serial port received	28
3.9 โพล์ชาร์ตของฟังก์ชัน Timer1 tick	29
3.10 โพล์ชาร์ตของฟังก์ชัน Data processing	30
3.11 โพล์ชาร์ตของฟังก์ชัน Update data and graph	31
3.12 โพล์ชาร์ตของฟังก์ชัน Timer2 tick	31
3.13 โพล์ชาร์ตของฟังก์ชัน Stop and save	32
3.14 โพล์ชาร์ตของฟังก์ชัน Save data and graph	32

สารบัญญรูป (ต่อ)

รูปที่	หน้า
3.15 โฟลว์ชาร์ตของฟังก์ชัน Clear.....	33
3.16 โฟลว์ชาร์ตของฟังก์ชัน Setting.....	34
3.17 โฟลว์ชาร์ตของฟังก์ชัน Disconnect.....	34
3.18 ตัวอย่างผลตอบสนองจากเซ็นเซอร์กลิ่น.....	35
3.19 อัลกอริทึมสำหรับคำนวณหาแอมพลิจูดเชิงสัมบูรณ์.....	36
3.20 อัลกอริทึมสำหรับคำนวณหาแอมพลิจูดเชิงสัมพัทธ์.....	37
3.21 อัลกอริทึมสำหรับคำนวณหาค่าคงที่เวลา.....	39
3.22 อัลกอริทึมสำหรับคำนวณหาความชัน.....	41
3.23 อัลกอริทึมสำหรับการนอร์มัลไลเซชัน.....	42
3.24 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม.....	43
3.25 ตัวอย่างของเมทริกซ์ความแปรปรวนร่วม (S).....	44
3.26 ตัวอย่างของ (ก) เวกเตอร์ลักษณะเฉพาะ (ข) ค่าลักษณะเฉพาะ.....	44
3.27 ตัวอย่างของเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่.....	45
3.28 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์.....	46
3.29 ตัวอย่างของเมทริกซ์สหสัมพันธ์ (R).....	47
3.30 ตัวอย่างของ (ก) เวกเตอร์ลักษณะเฉพาะ (ข) ค่าลักษณะเฉพาะ.....	47
3.31 ตัวอย่างของเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่.....	48
4.1 ระบบการทดลองวัดกลิ่น.....	49
4.2 ผลตอบสนองของเซ็นเซอร์ TGS2600 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง.....	50
4.3 ผลตอบสนองของเซ็นเซอร์ TGS2602 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง.....	51
4.4 ผลตอบสนองของเซ็นเซอร์ TGS2611 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง.....	51
4.5 ผลตอบสนองของเซ็นเซอร์ TGS2620 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง.....	52
4.6 ผลตอบสนองของเซ็นเซอร์ TGS2600 ต่อกลิ่นที่มีอัตราการระเหยต่ำ.....	52
4.7 ผลตอบสนองของเซ็นเซอร์ TGS2602 ต่อกลิ่นที่มีอัตราการระเหยต่ำ.....	53
4.8 ผลตอบสนองของเซ็นเซอร์ TGS2611 ต่อกลิ่นที่มีอัตราการระเหยต่ำ.....	53
4.9 ผลตอบสนองของเซ็นเซอร์ TGS2620 ต่อกลิ่นที่มีอัตราการระเหยต่ำ.....	54
4.10 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลแอมพลิจูดเชิงสัมบูรณ์ โดยนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	55
4.11 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลแอมพลิจูดเชิงสัมพัทธ์ โดยนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	55

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	56
4.13 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	56
4.14 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	57
4.15 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	57
4.16 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	58
4.17 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	58
4.18 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง.....	59
4.19 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	61
4.20 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	61
4.21 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	62
4.22 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	62
4.23 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	63
4.24 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	63
4.25 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	64
4.26 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น.....	64
4.27 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยต่ำ.....	65

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในอุตสาหกรรมที่เกี่ยวข้องกับกลั่นนั้น โดยทั่วไปจะใช้บุคคลที่ได้รับการฝึกมาเป็นพิเศษในการประเมินคุณภาพของกลั่นที่สนใจ อย่างไรก็ตามความสามารถในการรับรู้กลิ่นของมนุษย์นั้นผันแปรไปตามปัจจัยต่าง ๆ เช่น วัย สุขภาพ และประสบการณ์ เป็นต้น ด้วยเหตุนี้ระบบแยกแยะกลิ่นจึงถูกพัฒนาขึ้นโดยมีเป้าหมายเพื่อใช้แทนทรัพยากรบุคคล โดยทั่วไประบบแยกแยะกลิ่นจะประกอบด้วยเซ็นเซอร์กลิ่น เช่น เซ็นเซอร์ชนิดเมทัลออกไซด์ เซ็นเซอร์ชนิดควอร์ตคริสตัลไมโครบาลานซ์ ที่มีความสามารถในการตอบสนองต่อกลิ่นต่าง ๆ กันหลายตัวต่อรวมกันแบบอเรียรี โดยระบบจะสามารถแยกแยะกลิ่นได้ด้วยการวิเคราะห์ผลตอบสนองของเซ็นเซอร์ไม่ว่าจะเป็นด้วยการใช้การเรียนรู้แบบไม่มีผู้สอน เช่น วิธีการวิเคราะห์ข้อมูลหลายตัวแปร หรือการเรียนรู้แบบมีผู้สอน เช่น วิธีโครงข่ายประสาทเทียม อย่างไรก็ตามกลิ่นที่ใช้ในอุตสาหกรรมส่วนใหญ่จะเป็นกลิ่นที่เกิดจากสารต้นกำเนิดกลิ่นที่มีความสามารถในการระเหย (Volatility) ต่ำ ทำให้มีความเข้มข้นของกลิ่นในอากาศน้อย การแยกแยะกลิ่นจึงทำได้ยาก

1.2 วัตถุประสงค์ของการวิจัย

ทำการพัฒนาระบบรับรู้กลิ่นโดยใช้เซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์ต่อกันในลักษณะอเรียรี จากนั้นใช้ระบบดังกล่าวในการศึกษาความแตกต่างของผลตอบสนองที่ได้จากการวัดกลิ่นที่เกิดจากต้นกำเนิดกลิ่นที่มีความสามารถในการระเหยปานกลางและต่ำ และศึกษาความเป็นไปได้ในการใช้วิธีการวิเคราะห์องค์ประกอบหลักที่เป็นวิธีหนึ่งในวิธีการวิเคราะห์ข้อมูลหลายตัวแปรในการแยกแยะกลิ่นที่เกิดจากต้นกำเนิดกลิ่นที่มีความสามารถในการระเหยปานกลางและต่ำในเชิงเปรียบเทียบ

1.3 ขอบเขตของการวิจัย

1. พัฒนาหน่วยประมวลผลแบบอ่อนสำหรับทำงานร่วมกับอเรียรีของเซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์ และวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้องที่พัฒนาขึ้นก่อนหน้านี้ [1] เป็นระบบรับรู้กลิ่น
2. ทำการพัฒนาโปรแกรมเชื่อมต่อกับผู้ใช้บนคอมพิวเตอร์ สำหรับควบคุมและรับค่าจากระบบรับรู้กลิ่น
3. ศึกษาการนำคุณสมบัติต่าง ๆ ที่ได้จากข้อมูลที่วัด ไปทำการแยกแยะกลิ่นโดยใช้วิธีการวิเคราะห์องค์ประกอบหลักบน MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการทดลองวัดกลิ่นที่เกิดจากสารต้นกำเนิดกลิ่นที่มีความสามารถในการระเหยปานกลาง และต่ำ จากนั้นนำผลการทดลองที่ได้มาทำการแยกแยะด้วยวิธีการวิเคราะห์องค์ประกอบหลักที่พัฒนาขึ้น

1.4 ผลที่คาดว่าจะได้รับ

โดยทั่วไปแล้วการแยกแยะกลิ่นที่เกิดจากสารตั้งต้นที่มีความสามารถในการระเหยต่ำนั้นทำได้ยาก เนื่องจากมีความเข้มข้นของกลิ่นในอากาศน้อยมาก อย่างไรก็ตามด้วยการคัดเลือกคุณสมบัติที่ได้จาก ข้อมูลที่วัดมาทำการวิเคราะห์องค์ประกอบหลักอย่างเหมาะสม คาดว่าจะสามารถทำให้แยกแยะกลิ่น ดังกล่าวถูกพัฒนาขึ้นได้อีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีพื้นฐานที่เกี่ยวข้องกับการออกแบบระบบแยกแยะกลิ่น ตั้งแต่ทฤษฎีพื้นฐานเกี่ยวกับกลิ่น การรับรู้กลิ่นของมนุษย์ ไปจนถึงเซ็นเซอร์กลิ่นชนิดต่าง ๆ เทคนิคการแยกแยะกลิ่น และ FPGA ที่เป็นอุปกรณ์ที่ใช้ในการพัฒนาหน่วยประมวลผลแบบอ่อนในวิทยานิพนธ์ฉบับนี้

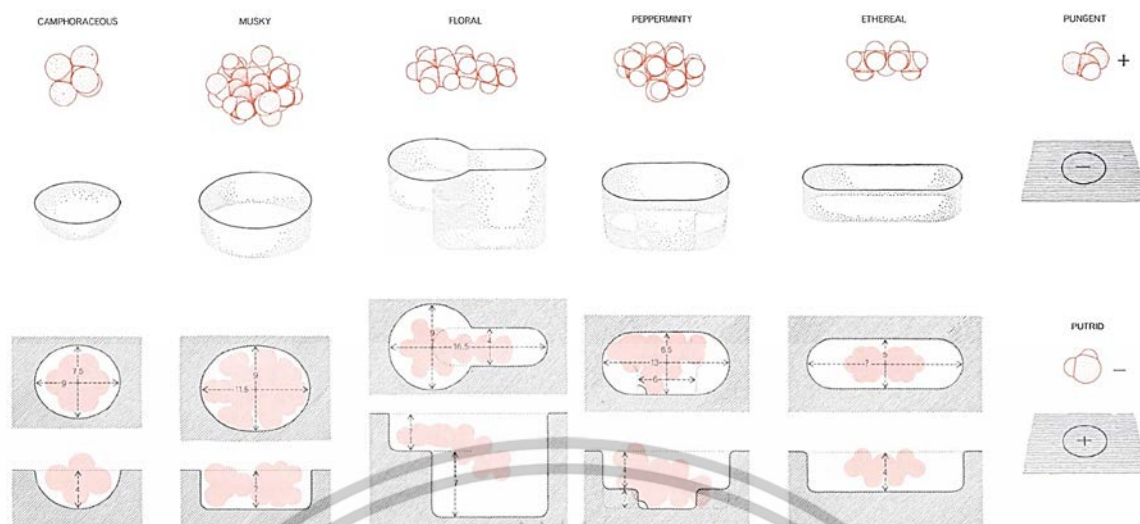
2.1 กลิ่น

กลิ่น (Odor) คือสารประกอบทางเคมีที่สามารถระเหยได้ (Volatile chemical compound) ซึ่งสามารถรับรู้ได้ผ่านทางประสาทรับกลิ่นของสิ่งมีชีวิต การรับรู้กลิ่นที่นาฬิกาหัวใจหรือไม่นาฬิกาหัวใจช่วยให้สิ่งมีชีวิตรู้ถึงข้อมูลที่จำเป็นต่อการดำรงชีวิต เช่น การหาอาหาร การหาคู่ การหลีกเลี่ยงอันตราย เป็นต้น อย่างไรก็ตามกลไกการทำงานของกลิ่น และการรับรู้กลิ่นถูกศึกษาและอธิบายเกิดเป็นทฤษฎีจำนวนหนึ่ง ยกตัวอย่างทฤษฎีที่โดดเด่นเช่น

ทฤษฎีการสั่นของไรท์ (Wright's vibrational theory) เสนอเมื่อปี 1954 เป็นการอธิบายทฤษฎีที่นำเสนอครั้งแรกโดย Malcolm Dyson ในปี 1928 ทฤษฎีกล่าวว่าคุณภาพที่สำคัญของกลิ่น (Essential quality of odor) ขึ้นอยู่กับการเคลื่อนที่โดยการสั่น (Vibrational movement) ของโมเลกุลของกลิ่นเป็นหลัก โดยความถี่ของกลิ่นขึ้นอยู่กับปัจจัยที่สำคัญคือ อัตราการระเหย ความสามารถในการละลายในน้ำหรือไขมัน และความสามารถในการดูดซับ จากสมมติฐานระบุว่า การสั่นของโมเลกุลกลิ่นจะมีความถี่ต่ำกว่าช่วง $300-400\text{ cm}^{-1}$ แต่จะไม่ต่ำกว่าช่วง $50-60\text{ cm}^{-1}$ จากสมมติฐานนี้จะแบ่งชนิดของกลิ่นหลักได้ประมาณ 25 กลิ่น

ทฤษฎีทางสเตอริโอเคมีของอาโมเรย์ (Amoore's stereochemical theory) ทฤษฎีนี้ถูกเสนอในปี 1964 ระบุว่าคุณภาพของกลิ่นจะถูกกำหนดโดยขนาดของโมเลกุล ดังนั้นกลิ่นที่คล้ายกันจะมีแนวโน้มที่ขนาดรูปร่างคล้ายคลึงกัน และความแตกต่างจะมีเพียงในระดับโมเลกุลเท่านั้น ทฤษฎีนี้ยังเสนอว่ากลิ่นที่มีความแตกต่างกันสามารถแบ่งออกได้เป็น 7 ประเภท คือ กลิ่นอีเทอร์ (Ethereal), กลิ่นการบูร (Camphoraceous), กลิ่นชะมด (Musky), กลิ่นดอกไม้ (Floral), กลิ่นสะระแหน่ (Minty), กลิ่นฉุน (Pungent) และ กลิ่นเน่า (Putrid) กลไกนี้เรียกว่า กุญแจและลูกกุญแจ (Lock-And-Key) กล่าวคือรูปร่างของโมเลกุลกลิ่นและตัวรับกลิ่นภายในประสาทรับกลิ่นจะมีรูปร่างที่สอดคล้องกัน ลักษณะของตัวรับกลิ่นที่เกี่ยวข้องกับกลิ่นแต่ละประเภทแสดงดังรูปที่ 2.1 ส่วนสองกลิ่นสุดท้ายจะไม่ขึ้นอยู่กับการรูปร่างแต่จะขึ้นอยู่กับการสถานะทางไฟฟ้าของโมเลกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ลักษณะของตัวรับกลิ่นที่เกี่ยวข้องกับกลิ่นแต่ละประเภท (ขนาดในหน่วย Angstrom) [2]

ทฤษฎีทางสเปกโทรสโกปีของตูริน (Turin's spectroscopic theory) ทฤษฎีนี้ถูกเสนอในปี 1996 อธิบายว่าการรับกลิ่นไม่ตอบสนองต่อรูปร่างของโมเลกุล แต่ตอบสนองต่อการสั่นสะเทือนของโมเลกุล (Vibrations of a molecule) ทฤษฎีนี้มีความแตกต่างจากทฤษฎีของไรท์ ในเรื่องการอธิบายกลไกการถ่ายโอนพลังงานทางชีวภาพของการสั่นสะเทือนของโมเลกุล ตัวรับที่ตอบสนองต่อช่วงที่ต่างกันของการสั่นสะเทือนมีหลายประเภท อาจจะมีบางช่วงของการสั่นสะเทือนที่ตัวรับไม่สามารถรับได้จึงทำให้ไม่มีกลิ่น นอกจากนี้ขนาดของโมเลกุลที่ใหญ่เกินไปสำหรับตัวรับจะไม่มีกลิ่นถึงแม้ว่าโมเลกุลนั้นจะมีการสั่นสะเทือนก็ตาม ทฤษฎีนี้มีการนำเสนอหลักฐานที่โดดเด่นสองประการดังนี้

1. โมเลกุลของสารอินทรีย์ส่วนใหญ่มีอะตอมของไฮโดรเจนและคาร์บอน โดยไฮโดรเจนอยู่ในรูปแบบไอโซเมอร์ (Isomeric form) โดยในนิวเคลียสจะประกอบด้วยโปรตอนและอิเล็กตรอน โดยโปรตอนเป็นมวลส่วนใหญ่ ซึ่งโปรตอนมีมวลประมาณ 1,800 เท่าของอิเล็กตรอน อย่างไรก็ตามขนาดของอะตอมไฮโดรเจนจะถูกกำหนดโดยวงโคจรของอิเล็กตรอน รูปแบบที่สองของอะตอมไฮโดรเจนเรียกว่า ดิวทีเรียม (Deuterium) มีนิวตรอนและโปรตอนอยู่ภายในนิวเคลียส ขนาดและรูปร่างของอะตอมดิวทีเรียมเหมือนกันกับอะตอมไฮโดรเจน แต่มีมวลประมาณสองเท่าของอะตอมไฮโดรเจน ถ้าอะตอมไฮโดรเจนในโมเลกุลถูกแทนที่ด้วยอะตอมดิวทีเรียมแล้วรูปร่างโมเลกุลจะไม่ได้รับผลกระทบ แต่ความถี่การสั่นสะเทือนของโมเลกุลจะลดลงอย่างมีนัยสำคัญ และเป็นผลทำให้กลิ่นมีความแตกต่างจากโมเลกุลเดิม

2. มีการค้นพบว่าโมเลกุลของกลิ่นสองกลิ่นที่มีการสั่นสะเทือนเดียวกันแม้มีรูปร่างของโมเลกุลแตกต่างกันจะมีกลิ่นเดียวกัน กำมะถันกับสารประกอบของโบรอน มีการสั่นสะเทือนเดียวกันนั้นก็มีกลิ่นเดียวกัน แม้ว่ารูปร่างของโมเลกุลจะแตกต่างกันอย่างสิ้นเชิงก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

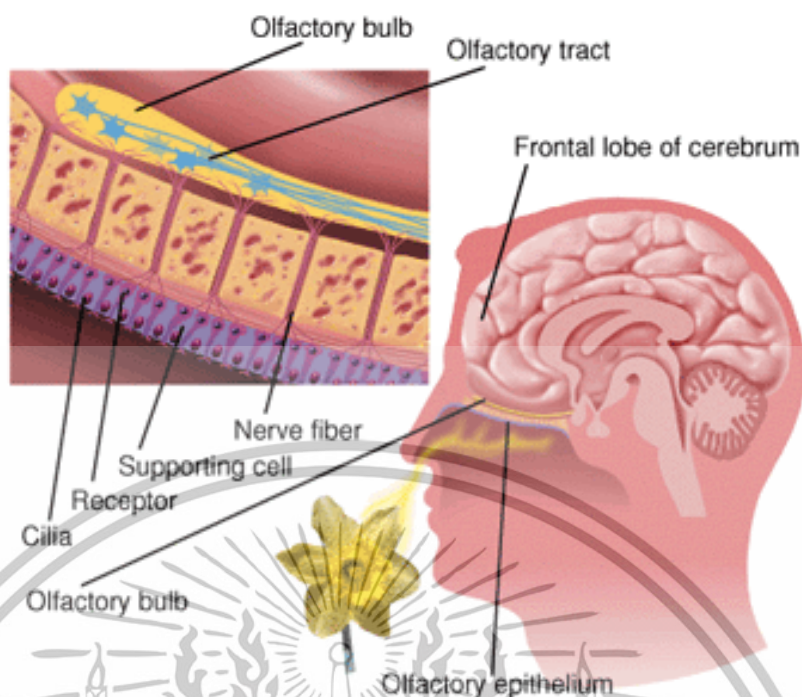
2.2 การรับรู้กลิ่นของมนุษย์

เมื่อมนุษย์หายใจเข้าผ่านทางอวัยวะรับกลิ่นหรือจมูก จะรับกลิ่นเข้าไปพร้อมอากาศด้วยดังแสดงในรูปที่ 2.2 อากาศที่ถูกลอยเข้าจะไปผ่านเซลล์ประสาทรับกลิ่น (Olfactory receptor cell) ซึ่งอยู่บริเวณเพดานภายในช่องจมูก เซลล์ประสาทรับกลิ่นประกอบด้วยเซลล์ 3 ประเภทได้แก่ เซลล์ประสาทรับความรู้สึกมีลักษณะเป็นขน (Ciliated sensory neuron) เซลล์ค้ำจุน (Supporting cell) และเซลล์พื้นฐาน (Basal cell) ซึ่งเป็นเซลล์ที่เรียงตัวเป็นแถวเดียวอยู่ชั้นล่างสุดของเยื่อจมูกดังแสดงในรูปที่ 2.2 เมื่อเซลล์ขนรับรู้ถึงโมเลกุลของกลิ่นจำนวนมากที่ละลายในเยื่อเมือก (Olfactory mucus membrane) ที่อยู่บนเยื่อบุเซลล์ (Nasal epithelium) ภายในช่องจมูก ทั้งนี้เยื่อบุเซลล์บริเวณนี้ของมนุษย์มีขนาด 2-5 ตารางเซนติเมตร หลังจากนั้นเยื่อเมือกที่มีโปรตีนซึ่งรวมตัวกับโมเลกุลของกลิ่นแล้ว (Odorant binding protein) จะไปจับกับเซลล์รับกลิ่นซึ่งจะไปกระตุ้น Adenyl cyclase ซึ่งเป็นเอนไซม์ที่ฝังตัวอยู่ในเยื่อบุเซลล์ขน เพื่อกระตุ้นให้เปลี่ยนพลังงาน ATP เป็น Cyclic AMP ภายในไซโตซอล (Cytosol) จากนั้น Cyclic AMP จะทำให้ช่องโซเดียม (Sodium channels) เปิดออกเป็นผลให้โซเดียมเข้าสู่เซลล์ ทำให้มีการเปลี่ยนแปลงความต่างศักย์เกิดเป็นกระแสประสาทส่งไปตามเส้นประสาทสมองคู่ที่ 1 (Olfactory nerve) ไปยังสมองเพื่อแปลสัญญาณกลิ่นที่ได้รับ เมื่อกระแสประสาทถูกส่งผ่านแอกซอนของเซลล์รับกลิ่นผ่านเพดานโพรงจมูกเข้าไปในป่องรับกลิ่น (Olfactory bulb) ผ่านไปตาม Olfactory tract จากนั้นจะแยกการทำงานของสมองเป็น 2 ส่วนคือ บริเวณรับกลิ่นด้านข้างส่วนต้น (Primary lateral olfactory area) ในสมองกลีบข้าง (Temporal lobe) ทำหน้าที่ในการรับรู้กลิ่น และบริเวณรับกลิ่นตรงกลางและด้านใน (Intermediate and medial olfactory area) ในสมองกลีบหน้า (Frontal lobe) ทำหน้าที่เกี่ยวกับความสัมพันธ์ระหว่างกลิ่นกับอารมณ์

เซลล์รับกลิ่นของมนุษย์มียีนที่เฉพาะเจาะจงต่อกลิ่น 1 กลิ่นเพียง 1 ยีนเท่านั้น แต่เซลล์รับกลิ่นเหล่านี้มีอยู่เป็นจำนวนมากและมีหลายชนิด ทำให้มนุษย์สามารถแยกความแตกต่างของกลิ่นต่าง ๆ ได้มากถึงประมาณ 10,000 กลิ่น โดยกลิ่นที่กระตุ้นเซลล์รับกลิ่นได้ดีมีคุณสมบัติดังนี้

1. ระเหยได้ในอากาศ เพื่อที่จะสามารถสูดผ่านเข้าจมูกได้
2. ละลายได้ดีในน้ำ เพื่อผ่านเยื่อจมูกไปสู่เซลล์รับกลิ่นได้
3. ละลายได้ดีในไขมัน เนื่องจากเซลล์รับกลิ่นมีองค์ประกอบเป็นสารไขมัน

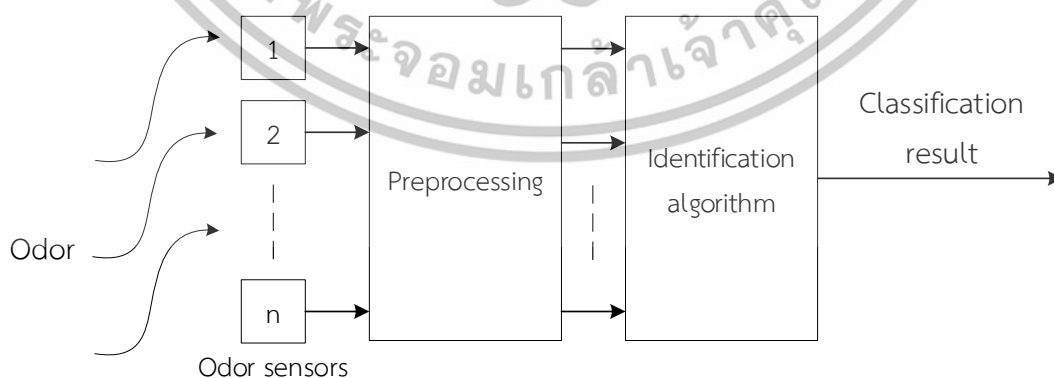
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ระบบรับรู้กลิ่นของมนุษย์ [3]

2.3 ระบบแยกแยะกลิ่น

ระบบแยกแยะกลิ่น หรือจมูกอิเล็กทรอนิกส์คือระบบที่สามารถแยกแยะกลิ่นได้เช่นเดียวกับประสาทรับกลิ่นของมนุษย์ ระบบแยกแยะกลิ่นสามารถถูกประยุกต์ใช้งานอย่างหลากหลายในงานที่การประเมินคุณภาพของกลิ่นมีความสำคัญเช่น การตรวจสอบคุณภาพของอากาศ [4-5] การตรวจสอบคุณภาพของอาหารและเครื่องดื่ม [6-7] อุตสาหกรรมเครื่องสำอาง [8-9] เกษตรกรรม [10-11] การดูแลสุขภาพ [12-13] และทางการทหาร [14-15] เป็นต้น



รูปที่ 2.3 บล็อกไดอะแกรมของระบบแยกแยะกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแยกแยะกลิ่นโดยทั่วไปมีบล็อกไดอะแกรมดังแสดงในรูปที่ 2.3 เมื่อกลิ่นลอยไปสัมผัสกับเซ็นเซอร์กลิ่น (Odor sensor) คุณสมบัติทางไฟฟ้าของเซ็นเซอร์กลิ่นจะเปลี่ยนแปลงไปตามความเข้มข้นของกลิ่น โดยการใช้เซ็นเซอร์กลิ่นจำนวนหนึ่งที่มีความสามารถในการรับรู้ (Sensitivity) กลิ่นต่าง ๆ แตกต่างกันในรูปแบบของอาเรย์ จะได้ชุดเอาต์พุตต่อกลิ่นแต่ละกลิ่นในรูปแบบของแพทเทิร์น (Pattern) แพทเทิร์นดังกล่าวอาจถูกประมวลผลเบื้องต้น (Preprocessing) ซึ่งมีหลายวิธีโดยการใช้จะแตกต่างกันไปตามชนิดของเซ็นเซอร์ที่ใช้ เช่น กระบวนการทำให้ผลตอบสนองของเซ็นเซอร์เป็นเชิงเส้นสำหรับเซ็นเซอร์บางชนิดที่มีความไม่เป็นเชิงเส้น และกระบวนการนอร์มัลไลเซชัน (Normalization) หลังจากนั้นจะใช้อัลกอริทึมที่เหมาะสมในการแยกแยะแพทเทิร์นของผลตอบสนองที่ได้จากเซ็นเซอร์ ซึ่งการพัฒนาให้ระบบสามารถแยกแยะแพทเทิร์นดังกล่าวได้นี้ สามารถทำได้ทั้งโดยวิธีการใช้การเรียนรู้แบบไม่มีผู้สอน เช่น วิธีการวิเคราะห์ข้อมูลหลายตัวแปร หรือการเรียนรู้แบบมีผู้สอน เช่น วิธีโครงข่ายประสาทเทียม เป็นต้น

2.4 เซ็นเซอร์กลิ่น

เซ็นเซอร์กลิ่นคือเซ็นเซอร์ที่สามารถวัดความเข้มข้นของกลิ่นได้ ซึ่งมีหลายชนิด ดังจะอธิบายต่อไป คุณสมบัติของเซ็นเซอร์กลิ่นที่ดีมีดังต่อไปนี้

- มีความสามารถในการรับรู้ (Sensitivity) สูง กล่าวคือ สามารถรับรู้ถึงการมีอยู่ของกลิ่นที่ความเข้มข้นต่ำ ๆ ได้
- มีความเที่ยงตรง (Precision) สูง กล่าวคือ ให้ผลตอบสนองลักษณะเดียวกันเมื่อเซ็นเซอร์ได้รับกลิ่นที่มีคุณสมบัติไม่ต่างจากเดิม
- มีระยะเวลาในการตอบสนอง (Response time) สั้น กล่าวคือ ให้ผลตอบสนองอย่างรวดเร็วหลังจากที่เซ็นเซอร์ได้รับกลิ่น
- มีระยะเวลาคืนตัว (Recovery time) สั้น กล่าวคือ ผลตอบสนองของเซ็นเซอร์ลดลงอย่างรวดเร็วหลังจากกลิ่นที่เซ็นเซอร์ได้รับหายไป
- มีความสามารถในการทำซ้ำ (Reproducibility) สูง กล่าวคือ สามารถทำงานได้อย่างมีประสิทธิภาพ และได้ผลการวัดที่ยังคงความน่าเชื่อถือ แม้จะผ่านการใช้งานหลายครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

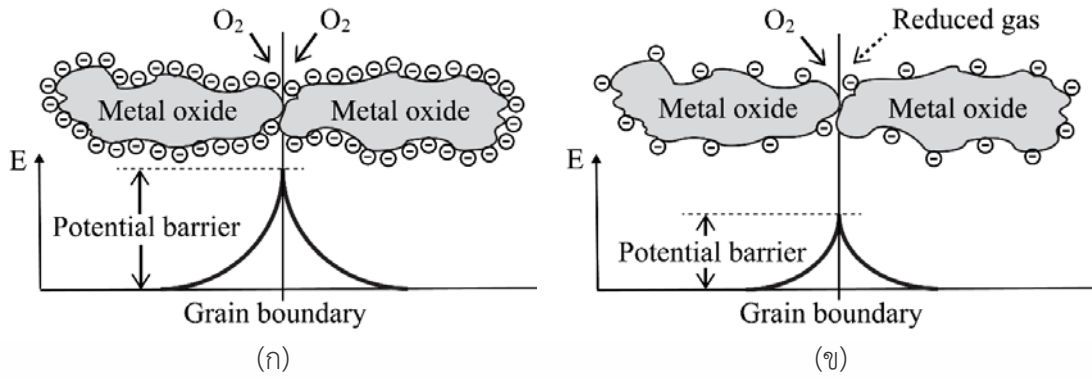
2.4.1 เซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์

เซ็นเซอร์ชนิดเมทัลออกไซด์ (Metal oxide sensor) เป็นเซ็นเซอร์ที่ใช้ประโยชน์จากปฏิกิริยาออกซิเดชันและรีดักชันของสารเมทัลออกไซด์ ที่เป็นสารประกอบที่เกิดจากโลหะและออกซิเจน แบ่งออกได้เป็น 2 ประเภท ได้แก่

1. เมทัลออกไซด์ชนิดเอ็น (N type) เช่น ดีบุกออกไซด์ (SnO_2), ไทเทเนียมออกไซด์ (TiO_2), ทังสเตนออกไซด์ (WO_3), ซิงค์ออกไซด์ (ZnO) เป็นต้น
2. เมทัลออกไซด์ชนิดพี (P type) เช่น คอปเปอร์ออกไซด์ (CuO), โครเมียมออกไซด์ (Cr_2O_3) เป็นต้น

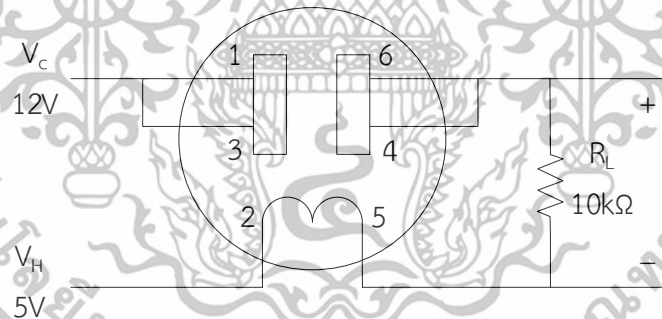
ในกรณีของสารประกอบเมทัลออกไซด์ชนิดเอ็นที่อยู่ในอากาศที่ไม่มีออกซิเจน อิเล็กตรอนอิสระสามารถไหลผ่านระหว่างอนุภาคของเมทัลออกไซด์ได้อย่างอิสระ เมื่อเมทัลออกไซด์สัมผัสกับอากาศที่มีออกซิเจน ออกซิเจนจะเกาะบนพื้นผิวของอนุภาคเมทัลออกไซด์และสร้างเป็นชั้นของออกซิเจนขึ้น ซึ่งชั้นของออกซิเจนนี้จะดักจับอิเล็กตรอนอิสระ ส่งผลทำให้เกิดกำแพงศักย์ขึ้นระหว่างอนุภาคของเมทัลออกไซด์ กำแพงศักย์นี้จะไปขัดขวางเส้นทางการไหลของอิเล็กตรอนอิสระส่งผลให้คุณสมบัติความต้านทานไฟฟ้าของเมทัลออกไซด์เพิ่มสูงขึ้น ดังแสดงในรูปที่ 2.4 (ก) และเมื่อเมทัลออกไซด์ชนิด N สัมผัสกับอากาศที่มีแก๊สรีดิวซ์ซึ่งเป็นแก๊สที่มีเลขออกซิเดชันต่ำ แก๊สชนิดนี้จะทำให้ความเข้มข้นของออกซิเจนบริเวณผิวของเมทัลออกไซด์นั้นลดลง เป็นผลให้กำแพงศักย์ระหว่างอนุภาคของเมทัลออกไซด์ลดลง อิเล็กตรอนอิสระจะสามารถไหลได้ง่ายขึ้น ส่งผลให้คุณสมบัติความต้านทานไฟฟ้าของเมทัลออกไซด์นั้นลดลง ดังแสดงในรูปที่ 2.4 (ข) สำหรับการเกิดปฏิกิริยาของเมทัลออกไซด์ชนิดพีนั้นจะตรงกันข้ามกับเมทัลออกไซด์ชนิดเอ็น

เซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์มีขายในเชิงพาณิชย์ และได้รับความนิยมเนื่องจากราคาถูก มีขนาดกระทัดรัด ใช้งานง่าย ถูกนำไปประยุกต์ใช้ในงานต่าง ๆ เช่น การประเมินคุณภาพของอากาศภายในห้องปฏิบัติการทางวิทยาศาสตร์ [4] การพัฒนาหุ่นยนต์เคลื่อนที่เพื่อใช้ในการประเมินคุณภาพของอากาศในสิ่งแวดล้อม [5] การแยกแยะสารประกอบในน้ำหอม [9] หุ่นยนต์กรองอากาศอัตโนมัติ [16] ภาพตัวถังของเซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์ที่มีขายในเชิงพาณิชย์ที่ได้รับความนิยมผลิตโดยบริษัท Figaro Engineering และวงจรไฟฟ้าที่เกี่ยวข้องแสดงในรูปที่ 2.5 และ 2.6 ตามลำดับ



รูปที่ 2.4 กำแพงศักย์ที่เกิดขึ้นระหว่างอนุภาคของเมทัลออกไซด์ชนิดอื่น (ก) การเกิดกำแพงศักย์เมื่อสัมผัสกับอากาศที่มีออกซิเจน (ข) การเกิดกำแพงศักย์เมื่อสัมผัสกับอากาศที่มีแก๊สรีดิวซ์

รูปที่ 2.5 เซ็นเซอร์ชนิดเมทัลออกไซด์ที่มีขายในเชิงพาณิชย์ [17]

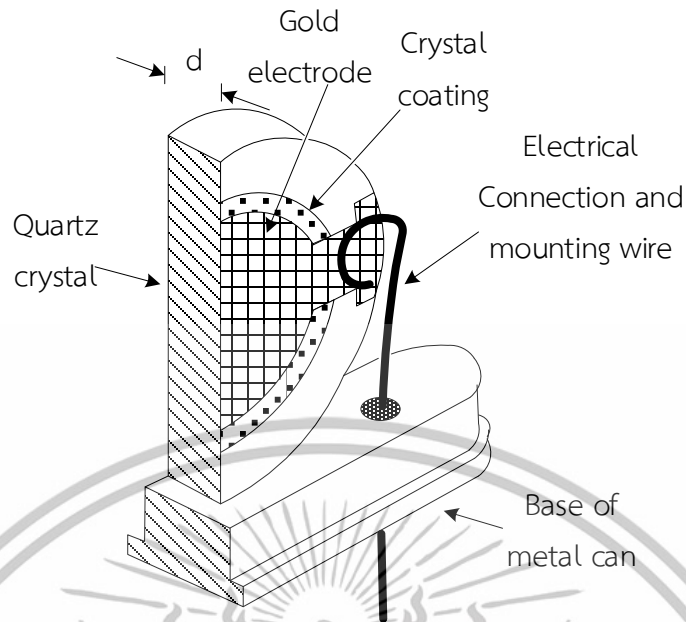


รูปที่ 2.6 ตัวอย่างวงจรใช้งานของเซ็นเซอร์ชนิดเมทัลออกไซด์ TGS2600 ของบริษัท Figaro Engineering [17]

2.4.2 เซ็นเซอร์กลืนชนิดควอ์ตคริสตัลไมโครบาลานซ์

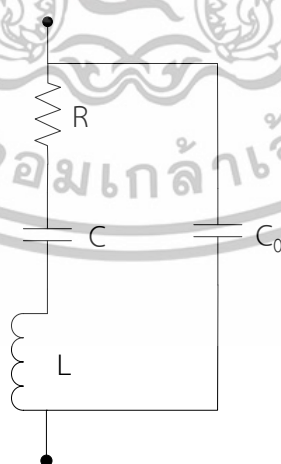
เซ็นเซอร์กลืนชนิดควอ์ตคริสตัลไมโครบาลานซ์ (Quartz Crystal Microbalance sensor: QCM sensor) เซ็นเซอร์ชนิดนี้ประกอบด้วยผลึกควอตซ์และขั้วไฟฟ้าที่ทำจากโลหะ เช่น ทองหรือ เงิน มีภาพตัดขวางดังแสดงในรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



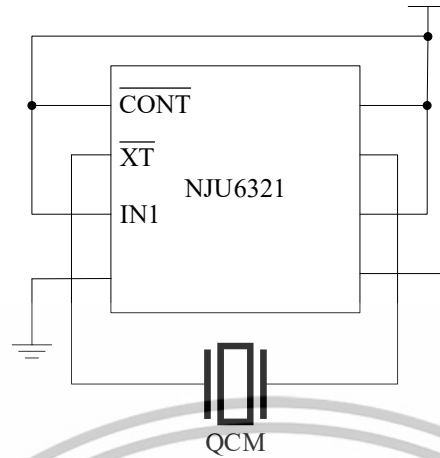
รูปที่ 2.7 ภาพตัดขวางของเซ็นเซอร์ชนิดควอ์ตคริสตัลไมโครบาลานซ์ [18]

วงจรสมมูลทางไฟฟ้าของเซ็นเซอร์คลื่นชนิดควอ์ตคริสตัลไมโครบาลานซ์ แสดงในรูปที่ 2.8 ความจุไฟฟ้า C แทนความยืดหยุ่น และความเหนียวไฟฟ้า L แทนมวลในระบบการสั่นเชิงกล ความต้านทานไฟฟ้า R แทนพลังงานสูญเสียที่เกิดจากแรงเสียดทานและการกระชากทางกล ในการใช้งานเซ็นเซอร์จะต้องต่อวงจรไฟฟ้าภายนอกเพื่อจ่ายพลังงานให้เกิดการออสซิลเลท (Oscillate) โดยจะมีความจุไฟฟ้าแฝงแบบพาราซิติค C_0 ที่เกิดจากขั้วไฟฟ้าและการเชื่อมต่อสายไฟฟ้ารวมอยู่ในวงจรสมมูลด้วย ตัวอย่างการต่อเซ็นเซอร์กับวงจรรวมสำหรับสร้างความถี่แสดงดังรูปที่ 2.9



รูปที่ 2.8 วงจรสมมูลทางไฟฟ้าของเซ็นเซอร์คลื่นชนิดควอ์ตคริสตัลไมโครบาลานซ์ [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตัวอย่างการต่อวงจรรวมสำหรับใช้งานเซ็นเซอร์ชนิดควอ์ตคริสตัลไมโครบาลานซ์ [19]

มวลที่เพิ่มขึ้นมาบนพื้นผิวของผลึกควอตซ์ส่งผลให้ความถี่เรโซแนนซ์เปลี่ยนแปลงไป ความหนาของผลึกควอตซ์แทนด้วย d มีค่าเท่ากับครึ่งหนึ่งของความยาวคลื่น λ ดังสมการ

$$d = \frac{\lambda}{2} \quad (2.1)$$

ความเร็ว v ของคลื่นเสียงของผลึกควอตซ์จะสัมพันธ์กับความถี่ f และความยาวคลื่นของตัวเอง ดังสมการ

$$v = f\lambda \quad (2.2)$$

จากสมการที่ 2.1 และ 2.2 จะได้ว่า

$$f = \frac{v}{2d} \quad (2.3)$$

ถ้าความหนาของหนึ่งหน่วยพื้นที่ของผลึกควอตซ์เพิ่มขึ้นเท่ากับ Δd จะส่งผลทำให้ความถี่เรโซแนนซ์นั้นเปลี่ยนแปลงไปเท่ากับ Δf

$$(f + \Delta f) = \frac{v}{2(d + \Delta d)} \quad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าสมมติว่าความหนาของผลึกควอตซ์ที่เพิ่มขึ้นนี้มีขนาดเล็กมากเมื่อเทียบกับความหนาเดิมของผลึก และจากความสัมพันธ์ที่ว่ามวลของผลึกเป็นสัดส่วนกับความหนาของผลึกจะได้ว่า

$$\frac{\Delta f}{f} = -\frac{\Delta m}{m} \quad (2.5)$$

เมื่อ m คือมวลต่อหนึ่งหน่วยพื้นที่ของผลึกควอตซ์

Δm คือมวลที่เพิ่มขึ้นต่อหนึ่งหน่วยพื้นที่ของผลึกควอตซ์

จะสังเกตว่าการเพิ่มมวลของผลึกควอตซ์จะส่งผลให้ความถี่เรโซแนนซ์ลดลง กล่าวคือ เมื่อโมเลกุลของกลิ่นที่ลอยอยู่ในอากาศไปสัมผัสกับเซ็นเซอร์ โมเลกุลของกลิ่นนี้จะทำให้มวลของผลึกเพิ่มขึ้น ส่งผลให้ความถี่เรโซแนนซ์ลดลง เมื่อแทนค่าสมการที่ 2.3 ลงในสมการที่ 2.5 จะสังเกตได้ว่ามวลต่อหนึ่งหน่วยพื้นที่จะมีค่าเท่ากับความหนาแน่นของผลึก ρ คูณกับความหนาของผลึก d กล่าวคือความถี่เรโซแนนซ์จะเปลี่ยนแปลงก็ต่อเมื่อความหนาของผลึกเพิ่มขึ้น ดังสมการ

$$\Delta f = \frac{2f^2 \Delta m}{\rho v} \quad (2.6)$$

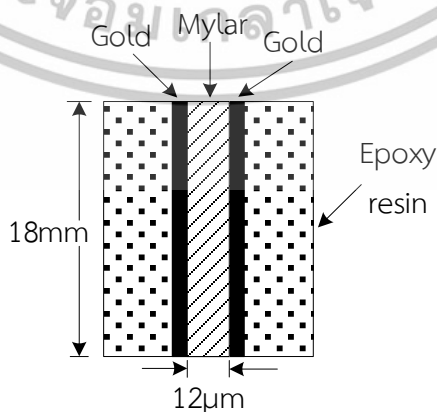
จากสมการที่ 2.6 จะสังเกตได้ว่าความไวของเซ็นเซอร์นั้นเพิ่มขึ้นตามค่าความถี่ยกกำลังสอง และการตอบสนองของเซ็นเซอร์นั้นเป็นสัดส่วนต่อความเข้มข้นของสารเคมี โดยที่ปริมาณสารเคมีที่ถูกดูดซับเป็นสัดส่วนต่อความเข้มข้นของสารเคมีในอากาศโดยรอบ กระบวนการตัดผลึกควอตซ์ที่จะทำให้เกิดผลึกที่มีค่าสัมประสิทธิ์อุณหภูมิต่ำที่สุด เรียกว่า AT cut ซึ่งจะทำให้ได้ผลึกที่มีความถี่เรโซแนนซ์ระหว่าง 0.5 - 300 MHz ดังนั้นการวัดความเปลี่ยนแปลงของความถี่เรโซแนนซ์จะทำให้ทราบถึงความเข้มข้นของกลิ่นรอบ ๆ เซ็นเซอร์ได้ โดยเซ็นเซอร์กลิ่นชนิดควอตซ์คริสตัลไมโครบาลานซ์นี้ มีขนาดเล็กกระทัดรัด มีความไวในการตอบสนองต่อโมเลกุลของกลิ่นที่ดี สามารถทำงานที่อุณหภูมิห้องได้ แต่สามารถเกิดสัญญาณรบกวนขึ้นได้ ถ้าหากทำงานในที่ที่มีความชื้นสูง มีการนำเซ็นเซอร์ชนิดนี้ไปประยุกต์ใช้ในงานต่าง ๆ เช่น การแยกแยะกลิ่นของผลไม้ [6] การแยกแยะกลิ่นของน้ำหอม [8] การตรวจหาไอระเหยของสารประกอบวัฏธูระเบิด [15] การตรวจหาสารประกอบในกลิ่นของไวน์ [20]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 เซ็นเซอร์กลืนชนิดโพลิเมอร์นำไฟฟ้า

โพลิเมอร์นำไฟฟ้า (Conducting polymer) ได้รับการคิดค้นขึ้นโดยบังเอิญที่ศูนย์วิจัย Hideki Shirakawa แห่งสถาบันเทคโนโลยีโตเกียว ตั้งแต่นั้นมาโพลิเมอร์นำไฟฟ้าได้รับการพัฒนาอย่างต่อเนื่อง ตัวอย่างของโพลิเมอร์นำไฟฟ้า เช่น พอลิไพร์โรล พอลิไทโอฟิน และพอลิอินิซีน วัสดุเหล่านี้ถูกนำมาใช้สร้างแบตเตอรี่น้ำหนักเบา ทรานซิสเตอร์สนามไฟฟ้า และยังสามารถนำมาใช้สร้างเซ็นเซอร์เคมีได้อีกด้วย เพื่อให้เซ็นเซอร์มีการตอบสนองที่รวดเร็ว จะต้องทำให้โพลิเมอร์นำไฟฟ้าเป็นฟิล์มบาง จึงจะทำให้โมเลกุลของกลืนแทรกซึมได้อย่างรวดเร็ว เพื่อชดเชยค่าความต้านทานที่สูงของฟิล์มบางดังกล่าว จะต้องทำให้เซ็นเซอร์สั้นในทิศทางที่นำไฟฟ้า และกว้างเพื่อให้เส้นทางนำไฟฟ้าที่ขนานกันมาก โพลิเมอร์จะถูกขยายออกบนฐานรอง (Substrate) ที่ประกอบด้วยขั้วไฟฟ้าที่ทำจากทองคำ 2 ชั้นที่มีระยะใกล้กันมากดังแสดงในรูปที่ 2.10

จะสามารถสร้างพอลิไพร์โรลด้วยกระบวนการโพลิเมอร์ไรเซชันในทางไฟฟ้าเคมี (Electrochemical polymerization) จากไพร์โรลในอะซิโตนไตรลโดยควบคุมการจ่ายไฟฟ้ากระแสสลับเข้าสู่ขั้วไฟฟ้าอย่างระมัดระวัง ถึงแม้ว่าจะควบคุมกระบวนการผลิตอย่างรัดกุม เซ็นเซอร์ที่สร้างจากกระบวนการเดียวกันนี้จะมีช่วงของคุณสมบัติความต้านทานไฟฟ้าเริ่มต้นในช่วงที่กว้าง มีการเปลี่ยนแปลงของคุณสมบัติความต้านทานไฟฟ้าระหว่าง 3 กิโลโอห์ม ถึง 30 กิโลโอห์ม เซ็นเซอร์ชนิดนี้เป็นเซ็นเซอร์ที่ใช้งานง่าย ราคาถูก คุณสมบัตินี้ความต้านทานไฟฟ้าของเซ็นเซอร์ที่เปลี่ยนแปลงเมื่อได้รับกลืน เป็นรูปแบบการเปลี่ยนแปลงที่สามารถถูกประมวลผลทางดิจิทัลได้ง่าย อย่างไรก็ตามความเสถียรของเซ็นเซอร์ยังต้องได้รับการปรับปรุงให้ดีขึ้น รวมถึงปัญหาด้านกระบวนการผลิตที่จะต้องทำให้เซ็นเซอร์ที่ผ่านกระบวนการผลิตมีคุณสมบัติที่เหมือนกันให้มากที่สุด เซ็นเซอร์โพลิเมอร์นำไฟฟ้านี้ถูกนำไปใช้ในงานต่าง ๆ เช่น การวิเคราะห์ภาวะหายใจของผู้ป่วยโรคเบาหวาน [21] การตรวจหากรดโคสภายในร่างกาย [22] การระบุคุณลักษณะของน้ำมันมะกอก [23] การระบุกลิ่นที่ผิดปกติกจากลูกแพร์ [24]



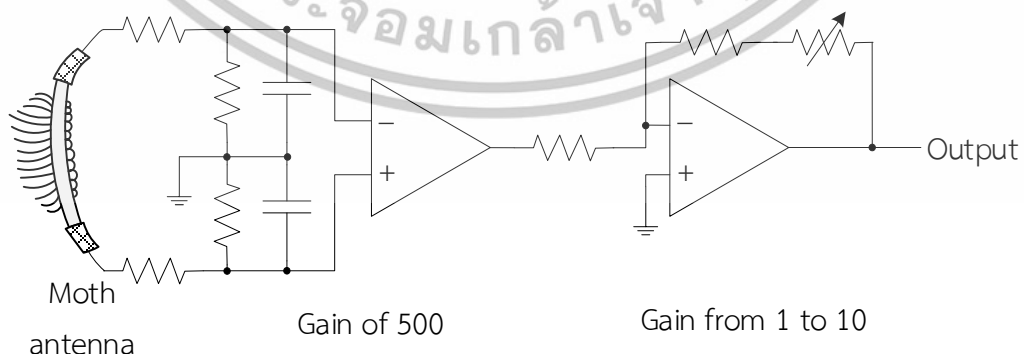
รูปที่ 2.10 ฐานรองสำหรับสร้างเซ็นเซอร์ชนิดโพลิเมอร์นำไฟฟ้า [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 เซ็นเซอร์กลิ่นที่ใช้หมวดของแมลง

เซ็นเซอร์ชนิดนี้ใช้หมวดของผีเสื้อกลางคืนตัวผู้ที่เอาออกมาสด ๆ และดำเนินการเชื่อมต่อปลายหมวดทั้งสองข้างเพื่อตรวจจับกิจกรรมทางไฟฟ้า หมวดของมันมีขนสำหรับตรวจจับกลิ่นประมาณ 2,000 เส้นซึ่งประมาณครึ่งหนึ่งจะอ่อนไหวต่อฟีโรโมนของตัวเมียที่เรียกว่า บอมไบคอลล (Bombykol) เท่านั้น ดังนั้น เซ็นเซอร์ชนิดนี้จึงเป็นเซ็นเซอร์ที่อ่อนไหวต่อฟีโรโมนชนิดนี้มากที่สุด เพื่อที่จะสร้างการเชื่อมต่อทางไฟฟ้ากับหมวดของผีเสื้อกลางคืน ส่วนปลายของหมวดทั้งสองด้านจะถูกตัดออกและนำไปเชื่อมต่อกับหลอดแก้วที่บรรจุส่วนผสมของสารละลายน้ำเกลือ น้ำเกลือสร้างการเชื่อมต่อทางไฟฟ้าและยังช่วยป้องกันไม่ให้หมวดหลุดออกอีกด้วย หลังจากนั้นสายไฟฟ้าที่ทำมาจากเงินและถูกเคลือบด้วยซิลเวอร์คอลลอยด์จะถูกแทรกเข้าไปในหลอดแก้วแต่ละหลอดเพื่อให้เกิดการเชื่อมต่อทางไฟฟ้าระหว่างสารละลายน้ำเกลือถึงปลายทั้งสองข้างของหมวดผีเสื้อกลางคืน การตอบสนองของหมวดนี้สามารถมองเห็นได้โดยใช้ออสซิลโลสโคปโดยขยายสัญญาณกว่า 100 เท่า

ทีมมหาวิทยาลัยโตเกียวได้ทำการดัดแปลง เซ็นเซอร์ชนิดนี้เพื่อสร้างเป็นเซ็นเซอร์กลิ่นสำหรับหุ่นยนต์เคลื่อนที่ น้ำเกลือถูกแทนด้วยสารละลายของ Ringer เพื่อสร้างสภาพแวดล้อมที่เหมาะสมกับหมวดของผีเสื้อกลางคืนนี้ สารละลายนี้มีอยู่ด้วยกันหลายสูตรแต่ละสูตรได้รับการปรับแต่งสำหรับประเภทของเนื้อเยื่อที่แตกต่างกัน ตัวอย่างสูตรของ Ringer เช่น NaCl 7.5 g, KCl 0.1 g, CaCl 0.2 g, NaHCO 0.2 g (ส่วนผสมต่อน้ำกลั่น 1,000 ml) หลอดแก้วถูกตัดให้สั้นและปิดผนึกด้วยซีเมนต์ เซ็นเซอร์มีความไวต่อฟีโรโมนบอมไบคอลล และสามารถคงอยู่ได้ประมาณ 1 ชม. การเชื่อมต่อวงจรไฟฟ้ากับหมวดของผีเสื้อกลางคืนแสดงดังในรูปที่ 2.11 มีการนำเซ็นเซอร์ชนิดนี้ไปใช้ในการศึกษาพฤติกรรมของแมลงที่มีต่อสารระเหยจากพืช [25] การศึกษาการตอบสนองของเซ็นเซอร์กลิ่นที่ใช้หมวดของแมลงที่มีต่อฟีโรโมนของผีเสื้อราชินี [26]



รูปที่ 2.11 วงจรขยายสัญญาณที่เชื่อมต่อกับหมวดของผีเสื้อกลางคืน [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การแยกแยะกลั่น

ข้อมูลผลตอบสนองจากเซ็นเซอร์ที่มีต่อกลิ่นที่ได้ทำการวัดนั้นจำเป็นที่จะต้องถูกประมวลผลและแสดงข้อมูลในรูปแบบที่สามารถประเมินผลของการวัดได้ ซึ่งข้อมูลจากเซ็นเซอร์ที่ไม่ได้ผ่านการประมวลผลจะเป็นข้อมูลที่ยากต่อการเห็นความแตกต่างของผลตอบสนองที่มีต่อกลิ่นซึ่งแตกต่างกัน เพื่อที่จะสามารถแยกแยะกลิ่นที่ทำการวัดออกจากกันจำเป็นที่ข้อมูลจะต้องผ่านการประมวลผลในรูปแบบที่เหมาะสม

2.5.1 โพลาร์โรสพล็อต

โพลาร์โรสพล็อต (Polar rose plot) เป็นรูปแบบการนำเสนอข้อมูลผลตอบสนองจากเซ็นเซอร์ที่ง่ายและไม่ซับซ้อน ข้อมูลจะถูกพล็อตอยู่ที่มุมของแผนภาพซึ่งจะมีระยะห่างที่เท่ากัน ขนาดของข้อมูลจะถูกวัดรัศมีจากจุดศูนย์กลางที่มุมที่เหมาะสม และจุดเหล่านี้จะถูกเชื่อมต่อถึงกันทั้งหมด กระบวนการนี้จะแสดงข้อมูลผลตอบสนองจากเซ็นเซอร์ในรูปแบบที่มนุษย์สามารถประเมินความเหมือนของชุดข้อมูลที่แตกต่างกันได้ ข้อมูลผลตอบสนองจากเซ็นเซอร์จะถูกนำมาพิจารณาเพื่อทำให้ข้อมูลเต็มสเกลมีค่าเท่ากับข้อมูลที่มีค่ามากที่สุดที่เซ็นเซอร์อ่านได้ ตัวอย่างของการพล็อตข้อมูลจากเซ็นเซอร์ 6 ตัวแสดงในรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างโพลาร์โรสพล็อตจากข้อมูลเซ็นเซอร์ 6 ตัว

2.5.2 การวิเคราะห์คลัสเตอร์ตามลำดับชั้น

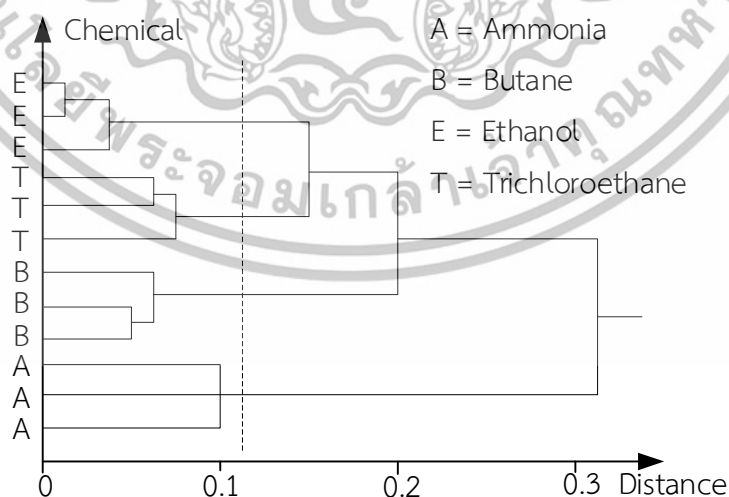
การวิเคราะห์คลัสเตอร์ตามลำดับชั้น (Hierarchical cluster analysis) เป็นรูปแบบการนำเสนอข้อมูลเป็นลำดับชั้น ซึ่งมีชื่อเรียกอีกอย่างหนึ่งว่าเดนโดรแกรม (Dendrogram) เป็นแผนภาพที่ช่วยแสดงการจัดกลุ่มตามธรรมชาติของข้อมูล สำหรับระบบแยกแยะกลิ่นที่ประกอบด้วยเซ็นเซอร์จำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

n ตัว แต่ละกลุ่มของข้อมูลจะสามารถแทนได้ด้วยจุดภายในพื้นที่ n มิติ และถ้าแต่ละจุดของข้อมูลถูกขยายด้วยอัตราเดียวกัน แผนภาพนี้จะแสดงลำดับซึ่งจุดที่ขยายตัวเหล่านั้นเชื่อมต่อกันกลายเป็นกลุ่มที่มีขนาดใหญ่กว่า ตัวอย่างอัลกอริทึมในการสร้างแผนภาพการวิเคราะห์คลัสเตอร์ตามลำดับชั้นมีดังนี้

เงื่อนไขเริ่มต้นคือแต่ละจุดของข้อมูลจะถูกพิจารณาเป็น 1 กลุ่ม และมีทั้งหมด n กลุ่มที่ตรงกันกับข้อมูล n จุด หลังจากนั้นทำการคำนวณระยะทางแบบยูคลิด (Euclidean distance) ระหว่างกลุ่มสองกลุ่มที่ใกล้กันมากที่สุด และรวมทั้งสองกลุ่มนั้นให้เป็นกลุ่มเดียวกัน จากนั้นเมื่อแต่ละกลุ่มบรรจุจุดของข้อมูลมากกว่า 1 จุด ระยะห่างที่สั้นที่สุดระหว่าง 2 กลุ่มจะถูกนำมาพิจารณาและกลายเป็นระยะห่างที่สั้นที่สุดระหว่างทุก ๆ สมาชิกภายในกลุ่มนั้น และทุกสมาชิกของกลุ่มอื่น ๆ (จะพิจารณาทุกคู่ที่เป็นไปได้) ตอนนี้มีกลุ่มทั้งหมด n-1 กลุ่ม การรวมกลุ่มที่ใกล้กันมากที่สุดจะถูกทำซ้ำจนกว่าจุดของข้อมูลทั้งหมดจะถูกรวมกันอยู่ในกลุ่ม ๆ เดียว

ตัวอย่างของแผนภาพการวิเคราะห์คลัสเตอร์ตามลำดับชั้นแสดงในรูปที่ 2.13 ค่าที่อ่านได้จากเซ็นเซอร์สามารถแบ่งออกได้เป็น 2 กลุ่มได้แก่ กลุ่มที่มีความคล้ายคลึงกัน และกลุ่มที่แตกต่างกัน ด้วยการกำหนดเส้นแนวตั้งที่ตัดผ่านแผนภาพ กลุ่มใดที่อยู่บริเวณด้านซ้ายของเส้นแนวตั้งนี้แสดงว่ามีความคล้ายคลึงกัน ซึ่งก็คือค่าที่อ่านได้จากการวัดสารเคมีที่เหมือนกันจะถูกจัดกลุ่มรวมกัน และกลุ่มใดที่อยู่บริเวณด้านขวาแสดงว่ามีความแตกต่างกัน ซึ่งก็คือค่าที่อ่านได้จากการวัดสารเคมีต่างชนิดกันนั่นเอง แผนภาพนี้ถูกนำไปใช้ในการแสดงข้อมูลของการศึกษาการแยกแยะกลิ่นในสัตว์เลี้ยงลูกด้วยนม [27] การวิเคราะห์กลิ่นภายในห้องโดยสารรถยนต์ [28] การประเมินคุณภาพของกลิ่น [29] การเข้ารหัสแทนกลิ่นอาหารในสมองของแมลงหวี่ [30]



รูปที่ 2.13 ตัวอย่างการวิเคราะห์คลัสเตอร์ตามลำดับชั้น [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 การแปลงแบบคาฮูเนนโลเอฟ

การแปลงแบบคาฮูเนนโลเอฟ (Karhunen-Loeve transformation) กระบวนการนี้เป็นการประมาณข้อมูล n มิติสร้างเป็นข้อมูลที่มีมิติต่ำกว่า หรือมักจะประมาณเป็นข้อมูลใน 2 มิติ โดยจะยังคงเก็บรักษาไว้ซึ่งความแปรปรวนของข้อมูลเดิมไว้ให้ได้มากที่สุด ดังนั้นการจัดกลุ่มตามธรรมชาติของข้อมูลจะยังคงถูกเก็บรักษาไว้ได้ ระบบแยกแยะกลิ่นที่ประกอบด้วยเซ็นเซอร์จำนวน n ตัวและทำให้ได้ชุดของข้อมูลที่สามารถอ่านได้ทั้งหมด m ชุด ข้อมูลทั้งหมดจะถูกเฉลี่ย เวกเตอร์ค่าเฉลี่ยของข้อมูลที่สามารถอ่านได้จากเซ็นเซอร์แต่ละตัว \bar{X}_j คือ

$$\bar{X}_j = \frac{1}{m} \sum_{i=1}^m X_{ij} \quad (2.7)$$

และใช้ค่าเฉลี่ยของข้อมูลทั้งหมดจากเอาท์พุทของแต่ละเซ็นเซอร์ที่สามารถอ่านได้ในการคำนวณเมตริกความแปรปรวนร่วม (Covariance matrix : C_{kl}) ได้ดังนี้

$$C_{kl} = \sum_{i=1}^m (X_{ki} - \bar{X}_k)(X_{li} - \bar{X}_l) \quad (2.8)$$

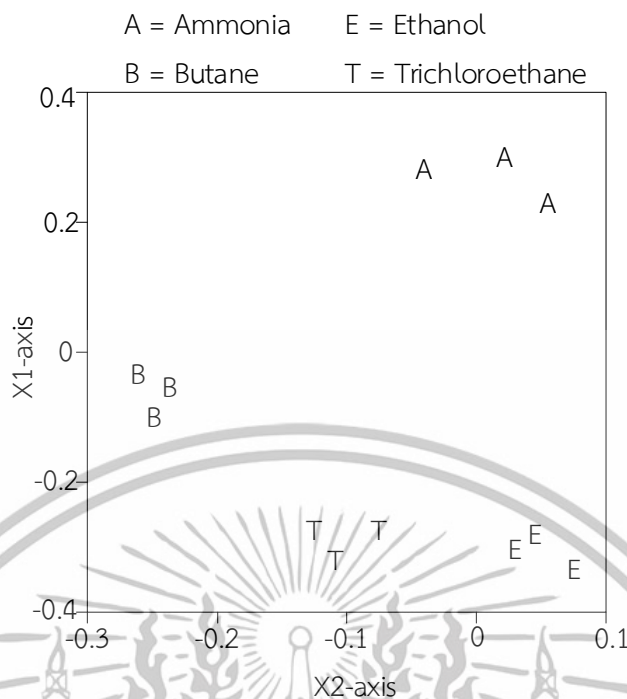
คำนวณหาค่าลักษณะเฉพาะ (Eigenvalue) λ_i และเวกเตอร์ลักษณะเฉพาะ (Eigenvector) V_i ของเมตริกความแปรปรวนร่วมได้ดังนี้

$$CV_i = \lambda_i V_i \quad (2.9)$$

เพื่อที่จะทำให้สามารถแสดงข้อมูลจากเซ็นเซอร์ใน 2 มิติ เวกเตอร์ลักษณะเฉพาะจะต้องสอดคล้องกันกับ 2 ค่าที่มากที่สุดของค่าลักษณะเฉพาะ โดยค่าลักษณะเฉพาะแต่ละจำนวนเป็นส่วนต่อการเปลี่ยนแปลงของข้อมูลจากเซ็นเซอร์ในตำแหน่งที่สอดคล้องกันกับเวกเตอร์ลักษณะเฉพาะ ดังนั้นการเลือกเวกเตอร์ลักษณะเฉพาะที่สอดคล้องกันกับ 2 ค่าที่มากที่สุดของค่าลักษณะเฉพาะจะทำให้ได้จำนวนที่มากที่สุดของความแปรปรวนในข้อมูลจากเซ็นเซอร์ การคำนวณสัดส่วนของความแปรปรวนดั้งเดิม RV ด้วยการประมาณได้ดังนี้

$$RV = \frac{\lambda_1 + \lambda_2}{\sum_{i=1}^n \lambda_i} \times 100\% \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ตัวอย่างการแปลงแบบค่าฐานโลเอฟ [18]

การโปรเจกชันรูปแบบเวกเตอร์ไปยังหนึ่งใน Eigenvector สามารถทำได้โดยใช้กระบวนการคูณภายใน (Inner product) อย่างง่าย ตัวอย่างแผนภาพนี้ดังแสดงในรูปที่ 2.14 โดยการแปลงแบบค่าฐานโลเอฟจะเก็บรักษาความแปรปรวนในข้อมูลต้นฉบับไว้ได้มากกว่า 99% จะเห็นได้ว่าจุดข้อมูลที่เป็นผลมาจากสารเคมีเดียวกันจะถูกจัดกลุ่มอยู่รวมกัน และกลุ่มแต่ละกลุ่มที่เป็นสารเคมีต่างชนิดกันจะถูกจัดกลุ่มแยกออกจากกัน การแปลงแบบค่าฐานโลเอฟถูกนำไปใช้ในการแสดงข้อมูลของการวิเคราะห์ประสิทธิภาพของยาระงับกลิ่น [31] การศึกษาความสามารถที่แท้จริงของประสาทรับรู้กลิ่น [32]

2.5.4 การวิเคราะห์องค์ประกอบหลัก

การวิเคราะห์องค์ประกอบหลัก (Principal component analysis: PCA) เป็นวิธีการทางสถิติที่มีประโยชน์มากในการแยกแยะและแสดงรูปแบบของข้อมูล สามารถแสดงความเหมือนหรือความแตกต่างของข้อมูลให้เห็นอย่างเด่นชัด เนื่องจากการแยกแยะและแสดงรูปแบบของข้อมูลที่มีหลายมิติทำได้ยาก วิธีการนี้ทำให้สามารถลดจำนวนมิติของข้อมูลได้โดยสูญเสียรายละเอียดของข้อมูลน้อยมาก โดยมีวัตถุประสงค์คือสร้างองค์ประกอบหลักที่มีจำนวนน้อยที่สุดที่สามารถอธิบายความแปรปรวนของข้อมูลให้ได้มากที่สุดเท่าที่จะเป็นไปได้ การคำนวณหาองค์ประกอบหลักสามารถทำได้ 2 วิธีได้แก่ การคำนวณหาองค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม (Variance-covariance matrix) และการคำนวณหาองค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ (Correlation matrix)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาองค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมจะมีกระบวนการคำนวณที่คล้ายกันกับการแปลงแบบคาชูเนนโลเอฟที่ได้อธิบายในหัวข้อที่แล้ว ข้อมูลผลตอบสนองจากเซ็นเซอร์จำนวน n ตัว ประกอบไปด้วยชุดของข้อมูลที่อ่านได้ทั้งหมด m ชุด เวกเตอร์ค่าเฉลี่ยของข้อมูลที่อ่านได้จากเซ็นเซอร์แต่ละตัว \bar{X}_j สามารถคำนวณได้ดังสมการที่ (2.7) และใช้ค่าเฉลี่ยของข้อมูลทั้งหมดจากเอาท์พุทของแต่ละเซ็นเซอร์ที่อ่านได้ในการคำนวณ เมทริกซ์ความแปรปรวนร่วม S ได้ดังนี้

$$S = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})' \quad (2.11)$$

หลังจากนั้นทำการคำนวณค่าลักษณะเฉพาะ และเวกเตอร์ลักษณะเฉพาะของเมทริกซ์ความแปรปรวนร่วม และเลือกเวกเตอร์ลักษณะเฉพาะที่สอดคล้องกันกับค่าที่มากที่สุดของค่าลักษณะเฉพาะตามจำนวนขององค์ประกอบหลักที่ต้องการ จะทำให้ได้จำนวนที่มากที่สุดของความแปรปรวนในข้อมูลจากเซ็นเซอร์ สามารถคำนวณหาองค์ประกอบหลักได้ดังนี้

$$PC_m = w_{m1}X_1 + w_{m2}X_2 + \dots + w_{mp}X_p \quad (2.12)$$

เมื่อ w คือ เวกเตอร์ลักษณะเฉพาะของเมทริกซ์ความแปรปรวนร่วม

การคำนวณหาองค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์นั้นจะต้องคำนวณหาเมทริกซ์ความแปรปรวนร่วมของข้อมูลจากเซ็นเซอร์โดยใช้สมการที่ (2.11) เพื่อใช้ในการคำนวณหาค่ามาตรฐานของข้อมูลดิบ (Standardized raw data) Z ได้ดังนี้

$$Z_i = \frac{X_i - \bar{X}_i}{\sqrt{S_{ii}}} \quad (2.13)$$

หลังจากนั้นทำการคำนวณหาเมทริกซ์สหสัมพันธ์ของข้อมูลจากเซ็นเซอร์ทั้งหมดได้ดังนี้

$$r_{jk} = \frac{\sum_{i=1}^n (X_{ij} - \bar{X}_j)(X_{ik} - \bar{X}_k)}{\sqrt{\sum_{i=1}^n (X_{ij} - \bar{X}_j)^2} \sqrt{\sum_{i=1}^n (X_{ik} - \bar{X}_k)^2}} \quad (2.14)$$

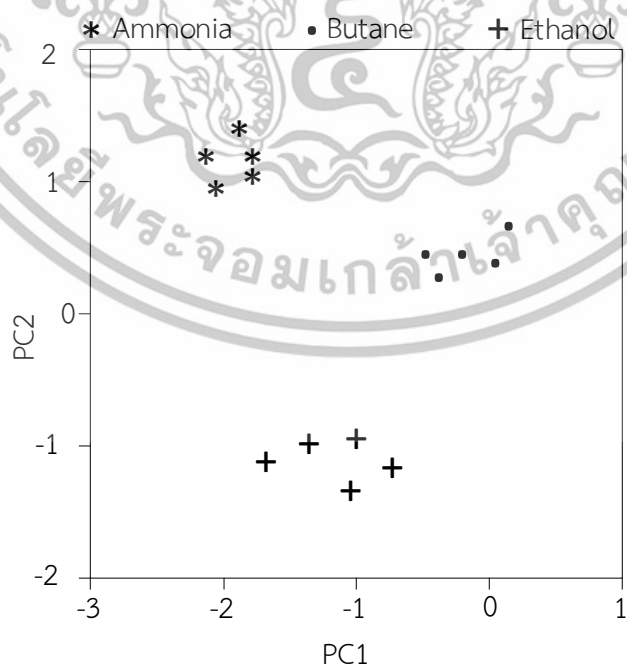
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นทำการคำนวณค่าลักษณะเฉพาะ และเวกเตอร์ลักษณะเฉพาะของเมทริกซ์สหสัมพันธ์และเลือก เวกเตอร์ลักษณะเฉพาะที่สอดคล้องกันกับค่าที่มากที่สุดของค่าลักษณะเฉพาะตามจำนวนขององค์ประกอบหลักที่ต้องการ จะทำให้ได้จำนวนที่มากที่สุดของความแปรปรวนในข้อมูลจากเซ็นเซอร์สามารถคำนวณหาองค์ประกอบหลักอันดับที่ m ได้ดังนี้

$$PC_m = w_{m1}Z_1 + w_{m2}Z_2 + \dots + w_{mp}Z_p \quad (2.15)$$

เมื่อ w คือ เวกเตอร์ลักษณะเฉพาะของเมทริกซ์สหสัมพันธ์

โดยองค์ประกอบหลักอันดับที่ 1 หรือ PC_1 จะประกอบด้วยความแปรปรวนของตัวแปรเดิมมากที่สุด องค์ประกอบหลักอันดับที่ 2 หรือ PC_2 จะประกอบด้วยความแปรปรวนของตัวแปรเดิมที่เหลือเกือบทั้งหมดที่ไม่ได้อธิบายในองค์ประกอบหลักอันดับที่ 1 และองค์ประกอบหลักอันดับรองลงมาจะประกอบด้วยความแปรปรวนของตัวแปรเดิมน้อยลงไปเรื่อยๆ ๑ ซึ่งองค์ประกอบหลักทุกตัวจะไม่มีความสัมพันธ์กับองค์ประกอบหลักอื่น ๆ หรือกล่าวได้ว่าทุกองค์ประกอบหลักตั้งฉากซึ่งกันและกัน ตัวอย่างการพล็อตกราฟจากองค์ประกอบหลักที่ 1 และองค์ประกอบหลักที่ 2 แสดงดังรูปที่ 2.15 เป็นการลดจำนวนมิติของข้อมูลเดิมลงเหลือ 2 มิติ ซึ่งยังคงเก็บรักษาความแปรปรวนของข้อมูลเดิมไว้ได้มากที่สุด และทำให้สามารถแยกแยะและแสดงรูปแบบของข้อมูลออกมาได้ ข้อมูลจากสารเคมีเดียวกันจะถูกจัดกลุ่มอยู่รวมกัน และสารเคมีต่างชนิดกันจะถูกจัดกลุ่มแยกออกจากกัน



รูปที่ 2.15 ตัวอย่างผลการวิเคราะห์องค์ประกอบหลักลดมิติของข้อมูลเหลือ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์องค์ประกอบหลักถูกนำไปใช้ในงานต่าง ๆ เช่น การประเมินราคาพืชผลทางการเกษตรด้วยการใช้กลิน [11] การพัฒนารองเท้าเพื่อสุขภาพด้านท่าทางในการเดินและปัญหากรลินเท้า [12] การศึกษาการแยกแยะตัวบุคคลผ่านทางกลินกาย [33] การศึกษาเกี่ยวกับรูปแบบทางชีวเคมีขององค์ประกอบกลินในปัสสาวะของผู้ป่วยโรคมะเร็งกระเพาะปัสสาวะ [34]

2.6 FPGA

FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ลอจิกที่โปรแกรมได้ชนิดหนึ่ง ซึ่งประกอบด้วยบล็อกของลอจิกที่สามารถโปรแกรมให้ทำงานเป็นลอจิกเกตพื้นฐานต่าง ๆ และมีโครงข่ายการเชื่อมต่อภายในแบบแมตริกซ์ สามารถโปรแกรมการเชื่อมต่อระหว่างบล็อกต่าง ๆ เข้าด้วยกันเป็นฟังก์ชันเชิงจัดหมู่ (Combinational function) ที่ซับซ้อนตามที่ผู้ออกแบบต้องการได้ โดยการเขียนข้อกำหนด (Configuration) ขึ้นมาอธิบายการทำงานของวงจรที่ต้องการออกแบบ โดยทั่วไปสามารถเขียนอธิบายวงจรที่จะสร้างได้โดยใช้ภาษา HDL (Hardware Description Language) เช่น VHDL, Verilog, AHDL, JHDL เป็นต้น ซึ่งคำอธิบายใน HDL จะถูกสังเคราะห์เป็นวงจรดิจิทัลด้วยเครื่องมือ EDA (Electronic Design Automation) ซึ่งผู้ผลิตได้นำเสนอให้กับผู้ออกแบบ เช่น โปรแกรม Quartus II ของบริษัท Intel ที่ประกอบด้วยเครื่องมือในการออกแบบวงจรดิจิทัลมากมาย และประกอบด้วย IP (Intellectual property) ให้เลือกใช้งานอย่างหลากหลาย เช่น DSP, Processors, Interface protocols, Memory interfaces, Memory controllers เป็นต้น ทำให้สามารถออกแบบฮาร์ดแวร์ที่ทำงานเฉพาะทางภายใต้สภาพแวดล้อมการออกแบบที่ยืดหยุ่นเป็นอย่างมาก ข้อดีของ FPGA ที่สำคัญ คือสามารถโปรแกรมได้ในภาคสนามหรือที่หน้างาน จึงเป็นที่มาของคำว่า "Field programmable" แม้จะส่งมอบงานแล้วก็ยังสามารถแก้ไขวงจรได้โดยง่าย ดังนั้นการพัฒนาวงจร หรือผลิตภัณฑ์โดยใช้ FPGA จะใช้เวลาในการพัฒนาน้อย อย่างไรก็ตาม FPGA เป็นอุปกรณ์ที่เก็บข้อมูลได้ชั่วคราว (Volatile) กล่าวคือวงจรที่พัฒนาขึ้นจะหายไปหลังจากหยุดจ่ายไฟเลี้ยง จึงต้องใช้หน่วยความจำถาวร (Nonvolatile memory) ทำหน้าที่เก็บรักษาข้อมูลดังกล่าวไว้ ซึ่งสามารถออกแบบให้ FPGA อ่านข้อมูลนั้นโดยอัตโนมัติหลังจากได้รับไฟเลี้ยงอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

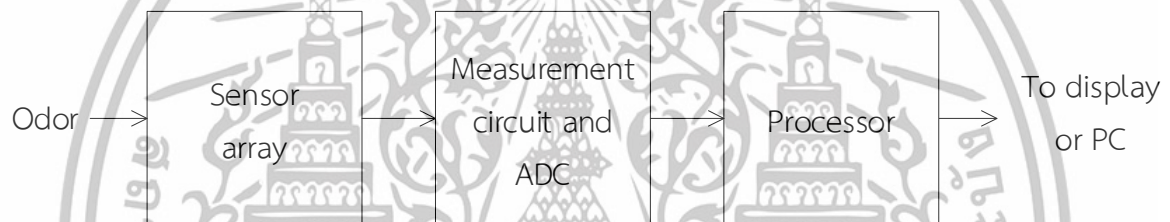
บทที่ 3

วิธีการดำเนินการวิจัย

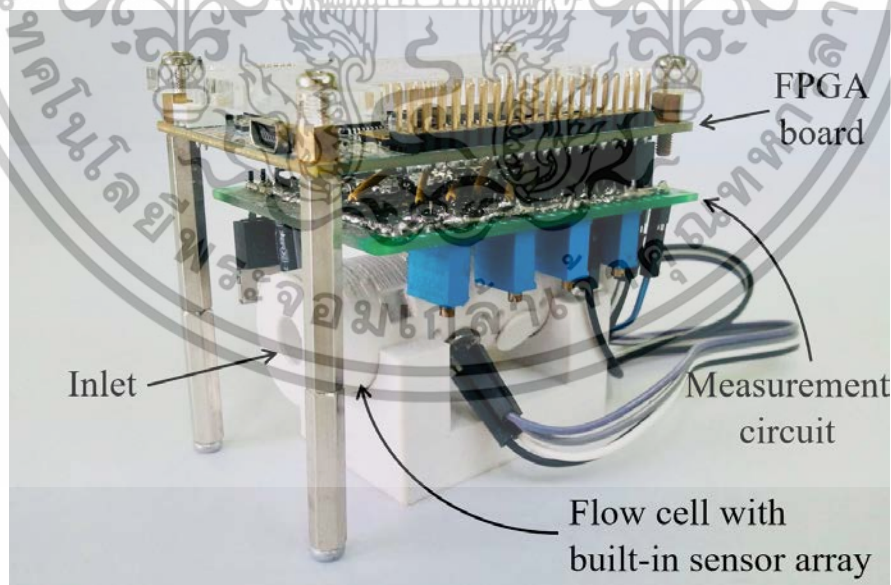
บทนี้จะกล่าวถึงระบบรับรู้กลิ่น และวงจรสำหรับใช้งานเซ็นเซอร์ หน่วยประมวลผลแบบอ่อน โปรแกรมติดต่อกับผู้ใช้ อัลกอริทึมสำหรับดึงค่าคุณสมบัติที่สนใจออกจากผลการทดลอง และวิธีการวิเคราะห์องค์ประกอบหลักที่พัฒนาขึ้นในงานวิจัยนี้

3.1 ระบบรับรู้กลิ่น

ระบบรับรู้กลิ่นที่พัฒนาขึ้นในงานวิจัยนี้มีบล็อกไดอะแกรมดังแสดงในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบรับรู้กลิ่น



รูปที่ 3.2 ภาพตัวเครื่องของระบบรับรู้กลิ่น [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบดังกล่าวจะใช้เซ็นเซอร์กลืนชนิดเมทัลออกไซด์ ที่ทำงานร่วมกันเป็นเซ็นเซอร์อาร์เรย์จำนวน 4 ตัว ได้แก่ TGS2600, TGS2602, TGS2611 และ TGS2620 ซึ่งเป็นเซ็นเซอร์ที่ถูกพัฒนาขึ้นสำหรับวัดคาร์บอนมอนอกไซด์ แอมโมเนียกับโทลูอิน มีเทนกับไอโซบิวทีน และเอทานอล ตามลำดับ จึงมีความไวในการตอบสนองต่อกลิ่นต่าง ๆ แตกต่างกันไป เซ็นเซอร์ทั้ง 4 ตัวถูกติดตั้งอยู่ภายในเซ็นเซอร์เซลล์ (Sensor cell) เพื่อทำให้อัตราการไหลมีเสถียรภาพดังแสดงในรูปที่ 3.2 เซ็นเซอร์เซลล์ดังกล่าวถูกสร้างด้วยวัสดุ ABS (Acrylonitrile butadiene styrene) โดยใช้เครื่องพิมพ์ 3 มิติ มีเส้นผ่าศูนย์กลางภายในท่อลม 8.4 mm และมีความยาว 68 mm

3.2 วงจรสำหรับใช้งานเซ็นเซอร์

วงจรอิเล็กทรอนิกส์ที่ใช้กับอาร์เรย์ของเซ็นเซอร์ชนิดเมทัลออกไซด์ในงานวิจัยนี้แสดงดังรูปที่ 3.3 เซ็นเซอร์กลืนชนิดเมทัลออกไซด์จำนวน 4 ตัว ถูกให้ความร้อนผ่านทางตัวต้านทาน R_h ที่ต้องการแรงดันไฟเลี้ยง 5 โวลต์ แหล่งจ่ายแรงดันไฟฟ้ากระแสตรง 5 โวลต์ จึงถูกใช้เป็นแหล่งพลังงานหลักของวงจร ตัวต้านทานขนาน R_L ที่เป็นตัวต้านทานปรับค่าได้ถูกนำมาต่ออนุกรมแบ่งแรงกับเซ็นเซอร์ ดังนั้นเมื่อเซ็นเซอร์ได้รับกลิ่น ความต้านทานของเซ็นเซอร์ R_s จะเปลี่ยนแปลงไป แรงดันไฟฟ้าที่ตกคร่อม R_L หรือ V_L จึงเปลี่ยนแปลงตามไปด้วย ซึ่งสามารถคำนวณได้ดังนี้

$$V_L = \frac{V_{out}}{\frac{R_s}{R_L} + 1} \quad (3.1)$$

โดย $V_{out} = 3.3V$

R_L = ความต้านทานโหลด

R_s = ความต้านทานไฟฟ้าของเซ็นเซอร์

V_L = แรงดันไฟฟ้าที่ตกคร่อม R_L

โดยค่าความต้านทาน R_L ถูกปรับให้เหมาะสม เป็นค่าดังต่อไปนี้

$R_{L1} = 9.87$ กิโลโอห์ม

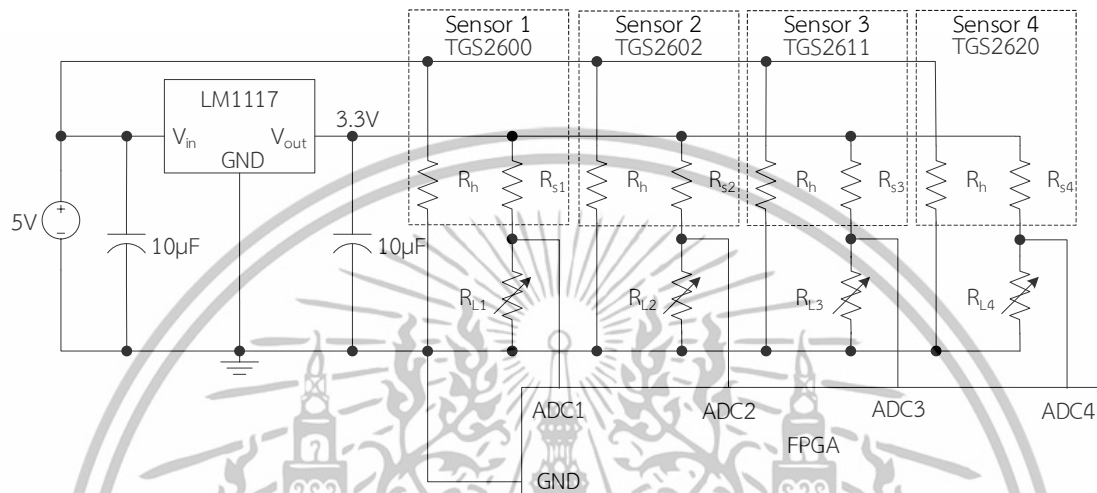
$R_{L2} = 19.83$ กิโลโอห์ม

$R_{L3} = 12.55$ กิโลโอห์ม

$R_{L4} = 5.58$ กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งนี้เนื่องจากค่าที่ V_L จะถูกส่งต่อไปแปลงเป็นสัญญาณดิจิทัลเพื่อการประมวลผลต่อไป และโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิทัลที่มีอยู่บนบอร์ด FPGA ที่ใช้ (จะอธิบายในหัวข้อ 3.3 ต่อไป) สามารถรับแรงดันแอนะล็อกอินพุตได้สูงสุด 3.3 โวลต์ วงจรรวมรักษาระดับแรงดัน LM1117 จึงถูกใช้เพื่อรักษาระดับแรงดันไว้ที่ 3.3 โวลต์



รูปที่ 3.3 วงจรไฟฟ้าที่เชื่อมต่อกับเซ็นเซอร์กลิ่นชนิดเมทัลออกไซด์ [1]

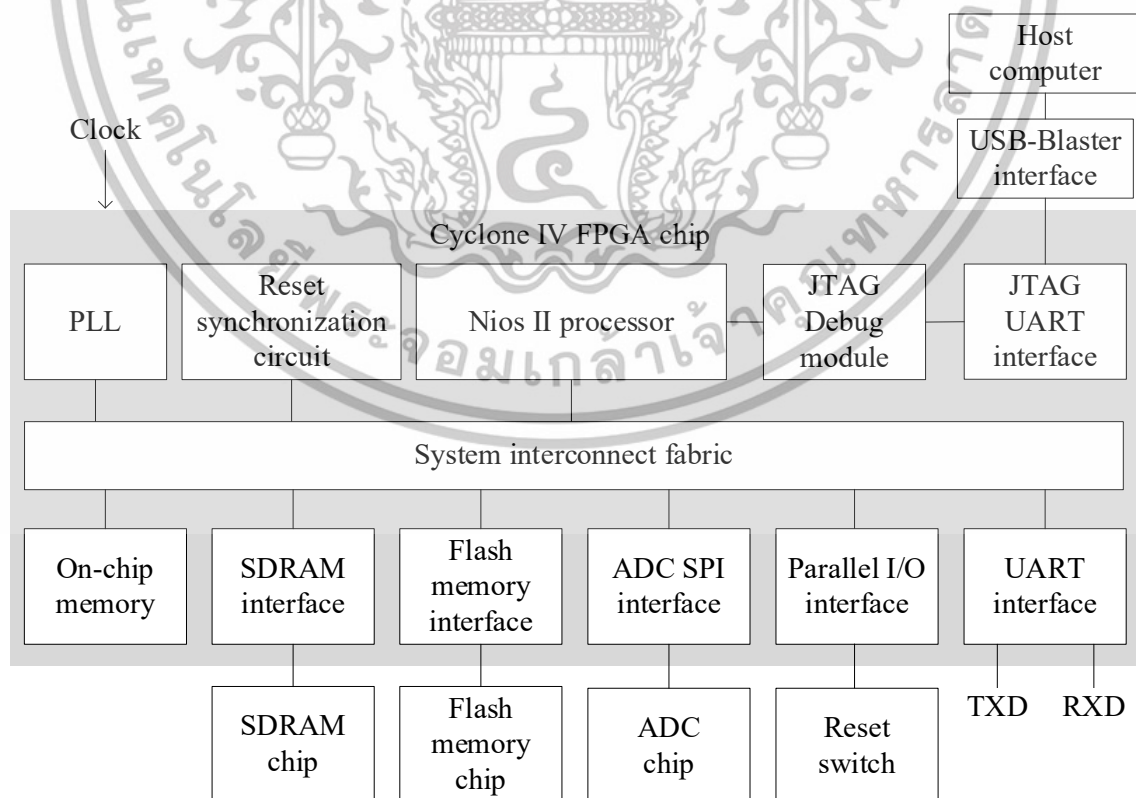
โดยในส่วนของเซ็นเซอร์อาร์เรย์ เซ็นเซอร์เซลล์ และวงจรสำหรับใช้งานเซ็นเซอร์ที่อธิบายในหัวข้อ 3.1 และ 3.2 นี้ นำมาจากงานวิจัยก่อนหน้า คือ “การศึกษาผลตอบสนองของเซ็นเซอร์ชนิดเมทัลออกไซด์” [1]

3.3 หน่วยประมวลผลแบบอ่อนที่พัฒนาขึ้น

ในงานวิจัยนี้หน่วยประมวลผลถูกพัฒนาขึ้นเพื่อรับข้อมูลผลตอบสนองเซ็นเซอร์ที่วัดได้ที่ถูกแปลงให้อยู่ในรูปสัญญาณดิจิทัลแล้วและแปลงให้อยู่ในรูปแบบ UART เพื่อส่งต่อไปทำการประมวลผลเพิ่มเติมบนคอมพิวเตอร์ ทั้งนี้หน่วยประมวลผลที่ใช้ในงานวิจัยนี้เป็นหน่วยประมวลผลแบบอ่อนที่เรียกว่า Nios II ที่ถูกพัฒนาขึ้นภายใน FPGA (Cyclone IV EP4CE22F17C6N, Altera) บนบอร์ด FPGA (DE0 Nano, Terasic) เพื่อความยืดหยุ่น โดยใช้โปรแกรม Quartus II 14.1, Qsys 14.1 และโปรแกรม Nios II 14.1 Software Build Tools for Eclipse ในการออกแบบฮาร์ดแวร์ของหน่วยประมวลผลแบบอ่อน ซอฟต์แวร์ที่ทำงานในหน่วยประมวลผลถูกพัฒนาด้วยภาษา C โปรแกรมทั้งหมดทำงานบน Windows 10 64 บิต

องค์ประกอบของฮาร์ดแวร์ของระบบหน่วยประมวลผลแบบอ่อนที่พัฒนาขึ้น แสดงในรูปที่ 3.4 ระบบทำงานด้วยสัญญาณนาฬิกาความถี่ 50 MHz จากคริสตัลอสซิลเลเตอร์บนบอร์ด ตัวหน่วยประมวลผลแบบอ่อนถูกเชื่อมต่อกับองค์ประกอบอื่นในระบบผ่าน System interconnect fabric ทั้งนี้เนื่องจากหน่วยความจำภายในชิป (On-chip memory) มีขนาดไม่เพียงพอสำหรับเก็บซอฟต์แวร์ภาษา C เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับแปลงให้อยู่ในรูปแบบ UART เพื่อส่งต่อที่ได้พัฒนาขึ้น จึงต้องใช้ชิป SDRAM (IS42S16160G, ISSI) เป็นหน่วยความจำหลักสำหรับเก็บซอฟต์แวร์ดังกล่าว ทั้งนี้ชิป SDRAM ดังกล่าวต้องการสัญญาณนาฬิกาที่มีเฟสเลื่อนไปจากสัญญาณนาฬิกาของ Nios II -54 องศา เฟสล็อกคูลูป (Phase Lock Loop : PLL) จึงถูกใช้ทำหน้าที่สร้างสัญญาณนาฬิกาดังกล่าวจากสัญญาณนาฬิกาหลัก นอกจากนี้ ชิปแฟลชเมโมรี่ (Flash memory chip) (FL064PIF, Spansion) ถูกใช้งานเป็นหน่วยความจำถาวร (Non-volatile memory) สำหรับเก็บข้อมูลฮาร์ดแวร์วงจรและระบบทั้งหมด เพื่อให้วงจรที่สร้างขึ้นไม่หายไปเมื่อตัดไฟเลี้ยง โดย SDRAM Interface และ Flash memory interface ถูกใช้เป็นสื่อกลางในการสื่อสารระหว่าง Nios II กับชิป SDRAM และแฟลชเมโมรี่ตามลำดับ JTAG debug module และ JTAG UART interface ทำหน้าที่เป็นเส้นทางเชื่อมต่อในการดาวน์โหลดวงจรการทำงานทั้งหมดของระบบจากคอมพิวเตอร์เข้าสู่หน่วยความจำผ่านทาง USB-Blaster แรงดันไฟฟ้าจากเซ็นเซอร์จะถูกแปลงเป็นสัญญาณดิจิทัลด้วยตัวแปลงสัญญาณแอนะล็อกเป็นดิจิทัล (ADC128S022, Texas Instruments) ข้อมูลจะถูกส่งไปแสดงผลยังคอมพิวเตอร์ผ่านทาง การสื่อสารแบบอนุกรมโดยใช้ UART interface วงจรเข้าจังหวะการรีเซ็ต (Reset synchronization) ถูกสร้างขึ้นเพิ่มเติมด้วยภาษา VHDL ทำหน้าที่สร้างสัญญาณรีเซ็ตที่เหมาะสมสำหรับองค์ประกอบต่าง ๆ ภายในระบบเพื่อป้องกันปัญหา Metastability ระบบหน่วยประมวลผลทั้งหมดที่สร้างขึ้นใช้ลอจิกอิเลิเมนต์ 2,639 ตัว จากทั้งหมด 26,320 ตัว ใช้รีจิสเตอร์ 1,679 ตัว ใช้หน่วยความจำภายในชิปทั้งหมด 76,800 บิต จากทั้งหมด 59,400 บิต และใช้ PLL 1 ตัว จากทั้งหมด 4 ตัว

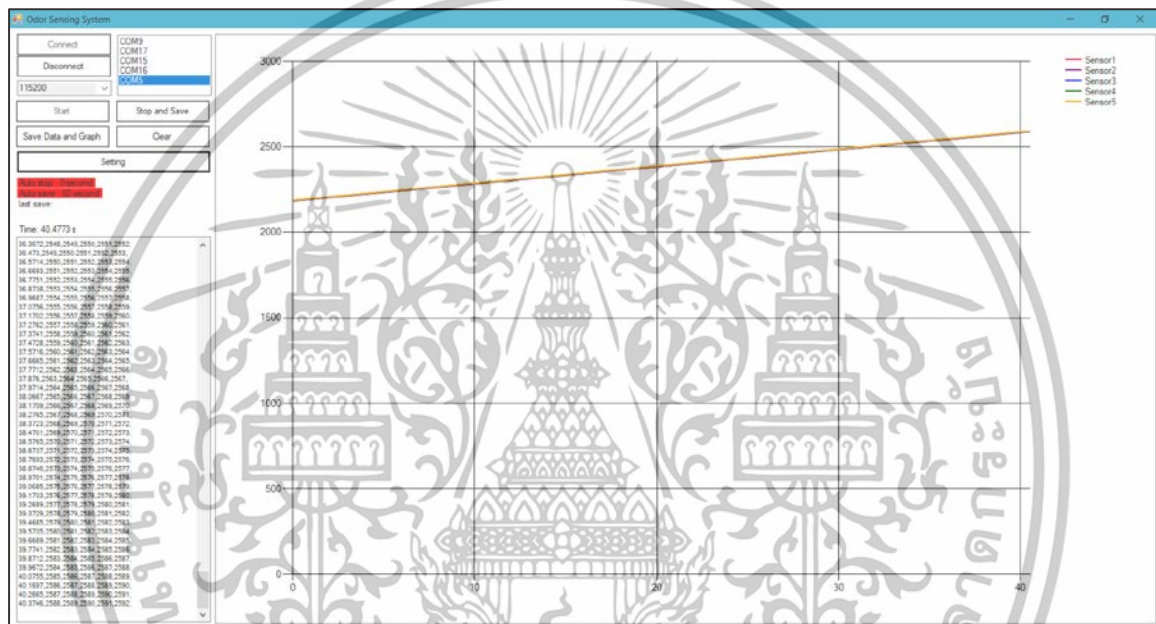


รูปที่ 3.4 องค์ประกอบของระบบประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โปรแกรมสำหรับติดต่อกับผู้ใช้ที่พัฒนาขึ้น

ได้พัฒนาโปรแกรมเชื่อมต่อกับผู้ใช้สำหรับใช้งานบนคอมพิวเตอร์ ที่จะทำหน้าที่พล็อตกราฟและเก็บบันทึกข้อมูลผลตอบสนองของเซ็นเซอร์ ที่ถูกส่งเข้ามาจากหน่วยประมวลผลแบบอ่อน ในรูปแบบของไฟล์ .csv ทำการพัฒนาโปรแกรมดังกล่าวด้วยภาษา Visual C# โดยใช้โปรแกรม Visual Studio 2017 บน Windows 10 64 บิต หน้าต่างส่วนติดต่อผู้ใช้งานของโปรแกรมแสดงในรูปที่ 3.5 ประกอบด้วยปุ่มควบคุมการทำงานของโปรแกรม ส่วนแสดงผลค่าแรงดันไฟฟ้าเป็นตัวเลข และส่วนแสดงกราฟการเปลี่ยนแปลงของแรงดันไฟฟ้า ฟังก์ชันการทำงานของโปรแกรมหาดังนี้

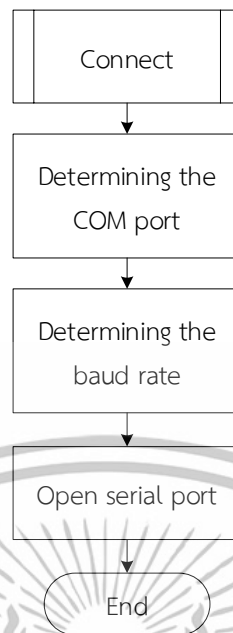


รูปที่ 3.5 โปรแกรมบันทึกค่าแรงดันไฟฟ้าจากเซ็นเซอร์

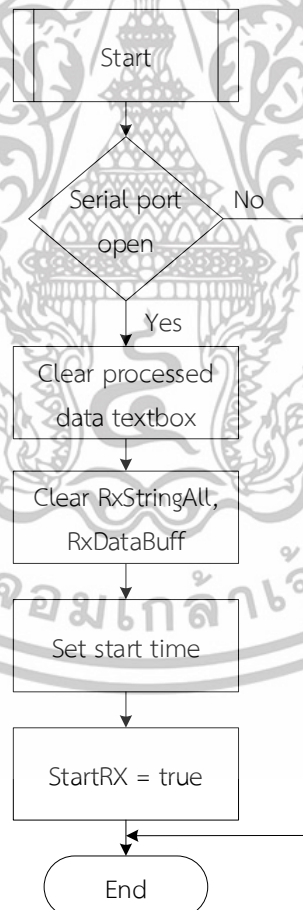
- **Connect** ฟังก์ชันการเชื่อมต่อ Serial port โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.6 ทำหน้าที่เริ่มต้นการเชื่อมต่อระหว่างคอมพิวเตอร์กับระบบรับรู้กลิ่น การทำงานเริ่มต้นด้วยการกำหนด COM port และกำหนดความเร็วในการสื่อสารข้อมูล (Baud rate) หลังจากนั้นโปรแกรมจะทำการเชื่อมต่อตามการตั้งค่าดังกล่าว

- **Start** ฟังก์ชันเริ่มต้นการรับข้อมูลจากระบบรับรู้กลิ่น โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.7 เริ่มต้นด้วยการตรวจสอบการเชื่อมต่อกับพอร์ตอนุกรม (Serial port) ถ้ามีการเชื่อมต่อเรียบร้อยแล้ว จะทำการลบข้อมูลภายในกล่องข้อความ Processed data textbox เพื่อเป็นการเตรียมกล่องข้อความให้พร้อมรับข้อมูลจากระบบรับรู้กลิ่น หลังจากนั้นจะทำการลบข้อมูลภายในตัวแปรที่ทำหน้าที่เก็บข้อมูลที่ได้รับมาจากระบบรับรู้กลิ่นทั้งหมดหรือ RxStringAll และทำการลบข้อมูลภายในตัวแปรที่ทำหน้าที่เก็บข้อมูลหลังจากการประมวลผลหรือ RxDataBuff หลังจากนั้นทำการบันทึกเวลาในขณะที่เริ่มต้นการรับข้อมูล และกำหนดให้ตัวแปร StartRX มีค่าเท่ากับ True เพื่อเริ่มต้นการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 โฟลว์ชาร์ตของฟังก์ชัน Connect



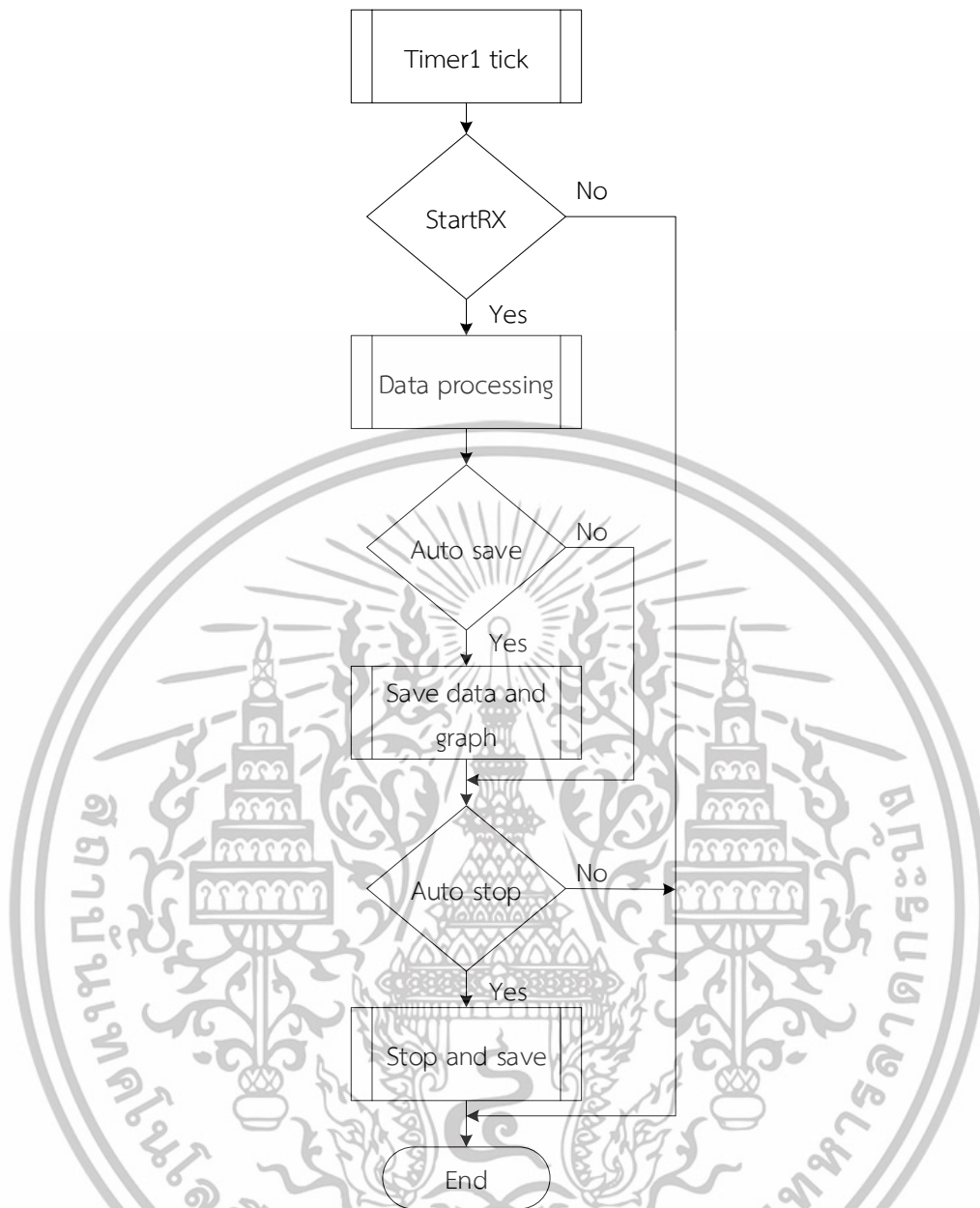
รูปที่ 3.7 โฟลว์ชาร์ตของฟังก์ชัน Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Serial port received** ฟังก์ชันรับข้อมูลจากระบบรับรู้กลืนผ่านทาง Serial port ข้อมูลที่ถูกส่งมาจากระบบรับรู้กลืนมีลักษณะดังนี้ “Data1*Data2*Data3*Data4*d\n\r” โดย Data1 ถึง Data 4 คือข้อมูลแรงดันไฟฟ้าจากเซ็นเซอร์ตัวที่ 1 ถึงตัวที่ 4 แต่ละข้อมูลคั่นด้วยสัญลักษณ์ ‘*’ ตัวอักษร ‘d’ เป็นไบต์สิ้นสุดของข้อมูล ‘\n’ เป็นการขึ้นบรรทัดใหม่ และ ‘\r’ คือ Carriage return ที่ถูกใช้ระบุจุดสิ้นสุดของชุดข้อมูลนั้น ๆ ในงานวิจัยนี้ ข้อมูลจะถูกประมวลผลด้วยฟังก์ชันที่แสดงดังรูปที่ 3.8 ข้อมูลที่ได้รับมาทั้งหมดจะถูกเก็บไว้ภายในตัวแปร RxStringAll หลังจากนั้นข้อมูลจะถูกตรวจสอบว่าประกอบด้วยตัวอักษร ‘d’ หรือไม่ ถ้าเป็นจริงจะทำการบันทึกเวลาในขณะที่ได้รับข้อมูลเก็บไว้ใน Time list



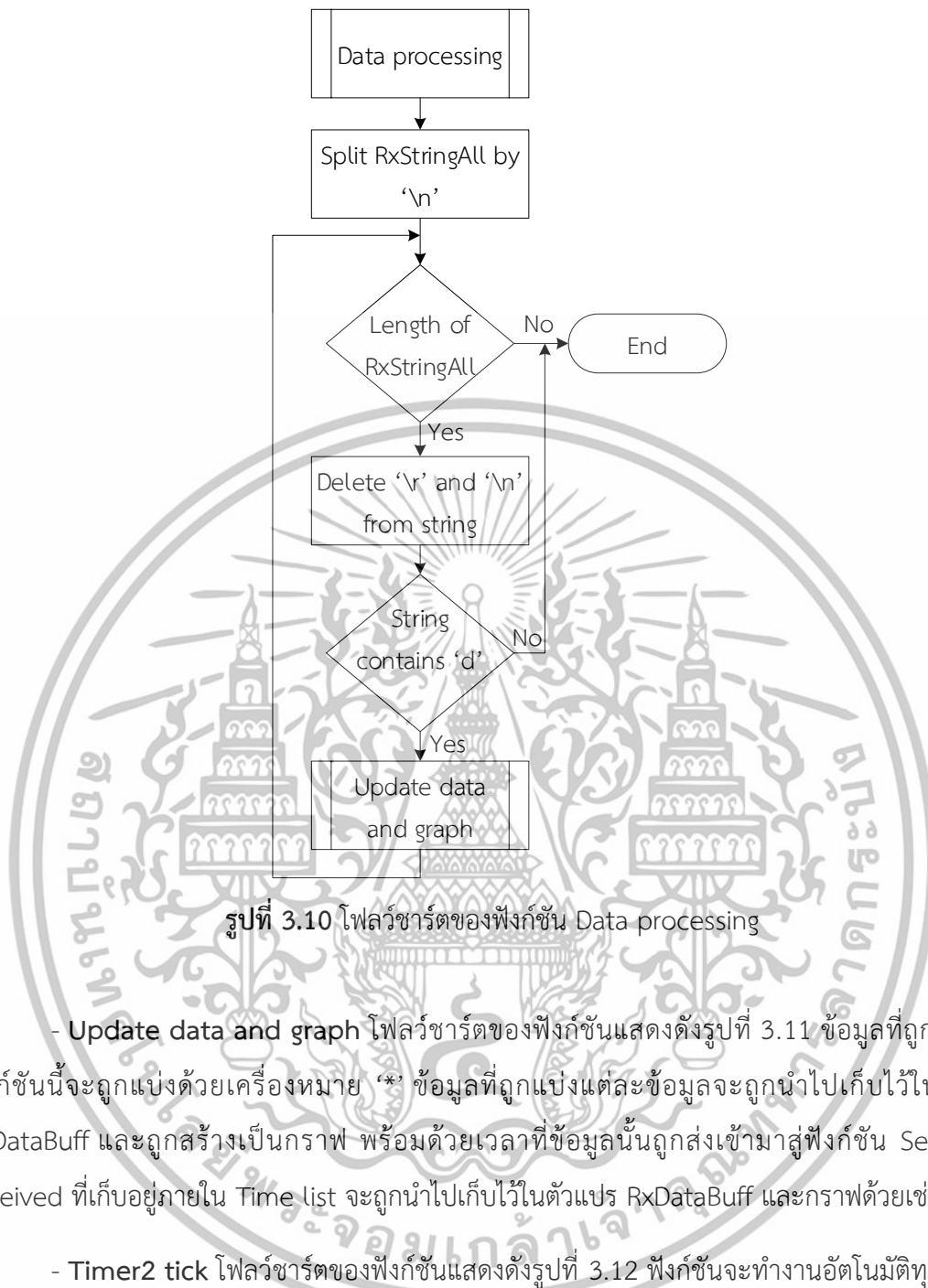
- **Timer1 tick** โฟลว์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.9 ฟังก์ชันจะทำงานอัตโนมัติทุก ๆ 100 มิลลิวินาที ทำหน้าที่เรียกใช้ฟังก์ชัน Data processing เพื่อประมวลผลข้อมูลที่ฟังก์ชัน Serial port received ได้รับมา นอกจากนี้ยังทำหน้าที่ตรวจสอบการตั้งค่าการบันทึกข้อมูลอัตโนมัติ และการหยุดรับข้อมูลอัตโนมัติ



รูปที่ 3.9 โฟลว์ชาร์ตของฟังก์ชัน Timer1 tick

- Data processing โฟลว์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.10 ข้อมูลที่ได้รับมาทั้งหมดที่เก็บไว้ในในตัวแปร RxStringAll ถูกนำมาแยกข้อมูลแต่ละบรรทัดออกจากกัน โดยการแบ่งข้อมูลด้วยเครื่องหมาย ‘\n’ ข้อมูลแต่ละบรรทัดจะถูกนำมาลบเครื่องหมายที่ไม่ต้องการออกได้แก่ ‘\r’ และ ‘\n’ หลังจากนั้นทำการตรวจสอบว่าข้อมูลประกอบด้วยตัวอักษร ‘d’ หรือไม่ ถ้าเป็นจริงจะไปประมวลผลต่อที่ฟังก์ชัน Update data and graph

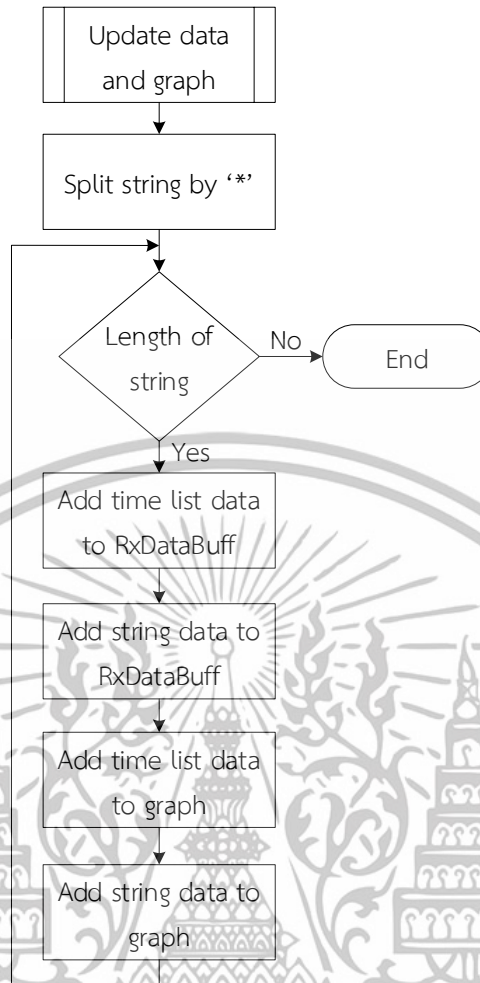
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



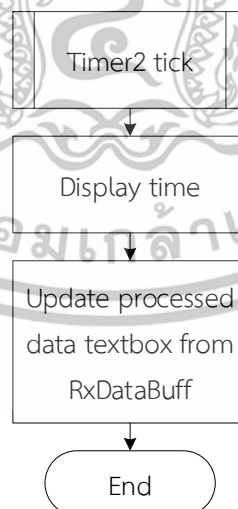
รูปที่ 3.10 โพล์ชาร์ตของฟังก์ชัน Data processing

- **Update data and graph** โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.11 ข้อมูลที่ถูกส่งมายังฟังก์ชันนี้จะถูกแบ่งด้วยเครื่องหมาย ‘*’ ข้อมูลที่ถูกแบ่งแต่ละข้อมูลจะถูกนำไปเก็บไว้ในตัวแปร RxDataBuff และถูกสร้างเป็นกราฟ พร้อมกับเวลาที่ข้อมูลนั้นถูกส่งเข้ามาสู่ฟังก์ชัน Serial port received ที่เก็บอยู่ภายใน Time list จะถูกนำไปเก็บไว้ในตัวแปร RxDataBuff และกราฟด้วยเช่นกัน

- **Timer2 tick** โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.12 ฟังก์ชันจะทำงานอัตโนมัติทุก ๆ 500 มิลลิวินาที ทำหน้าที่แสดงข้อมูลเวลา และแสดงข้อมูลหลังจากประมวลผลแล้ว ที่เก็บไว้ในตัวแปร RxDataBuff ในกล่องข้อความ Processed data textbox



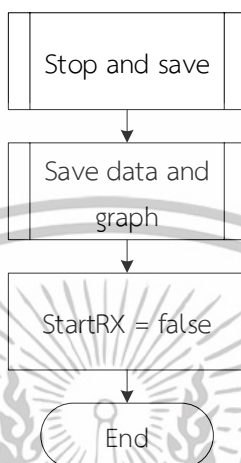
รูปที่ 3.11 โฟลว์ชาร์ตของฟังก์ชัน Update data and graph



รูปที่ 3.12 โฟลว์ชาร์ตของฟังก์ชัน Timer2 tick

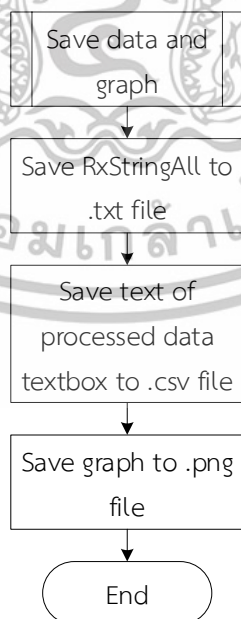
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Stop and save** ฟังก์ชันสิ้นสุดการรับข้อมูลจากระบบรับรู้กลิ่น และบันทึกข้อมูลที่ได้รับทั้งหมด โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.13 เริ่มต้นด้วยการเรียกใช้ฟังก์ชัน Save data and graph หลังจากนั้นทำการกำหนดให้ตัวแปร StartRX มีค่าเท่ากับ false เพื่อสิ้นสุดการรับข้อมูล



รูปที่ 3.13 โพล์ชาร์ตของฟังก์ชัน Stop and save

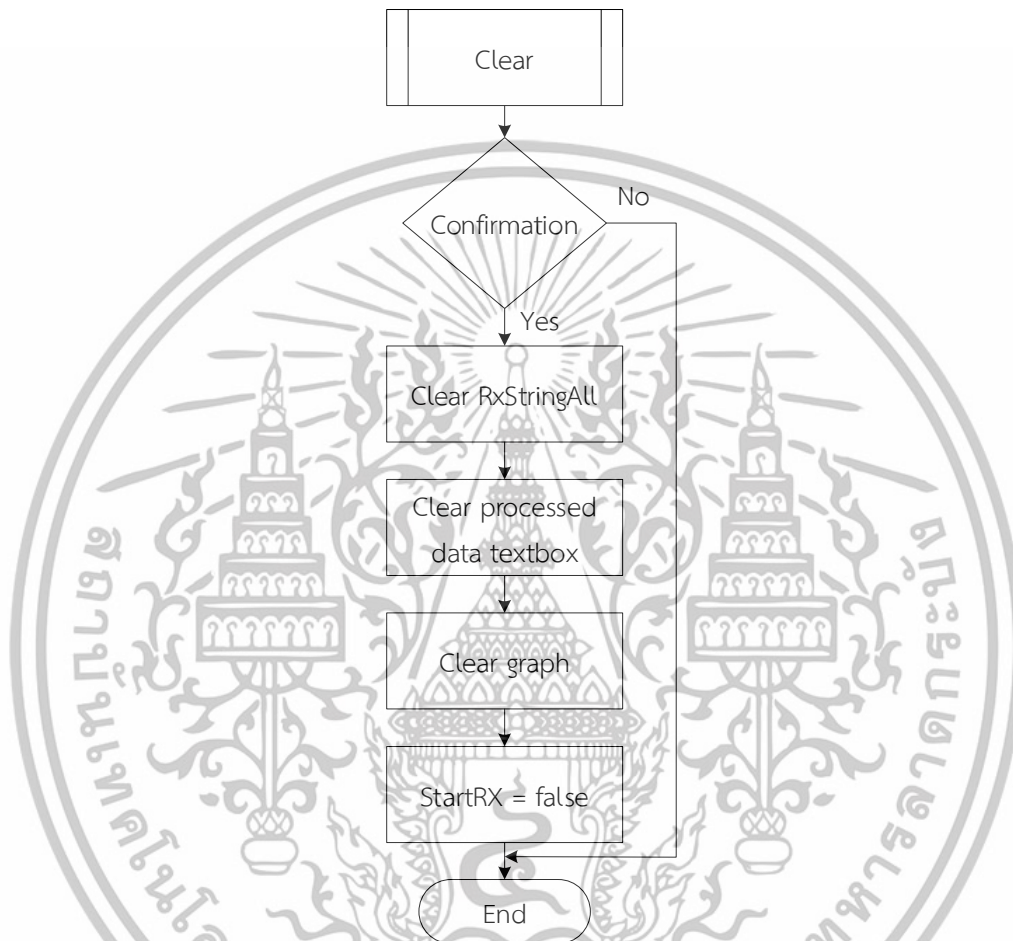
- **Save data and graph** ฟังก์ชันบันทึกข้อมูล และกราฟ โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.14 ทำการบันทึกข้อมูลภายในตัวแปร RxStringAll เป็นไฟล์นามสกุล .txt หลังจากนั้นทำการบันทึกข้อมูลภายในกล่องข้อความ Processed data textbox เป็นไฟล์นามสกุล .csv และบันทึกกราฟเป็นไฟล์ภาพนามสกุล .png



รูปที่ 3.14 โพล์ชาร์ตของฟังก์ชัน Save data and graph

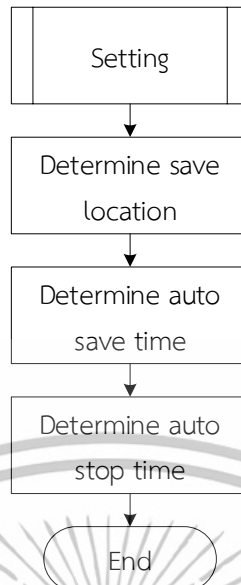
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Clear** ฟังก์ชันลบข้อมูล และกราฟ โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.15 เริ่มต้นด้วยการยืนยันการลบข้อมูล หลังจากยืนยันเรียบร้อยแล้วจะทำการลบข้อมูลที่เก็บอยู่ในตัวแปร RxStringAll หลังจากนั้นทำการลบข้อมูลที่อยู่ในกล่องข้อความ Processed data textbox หลังจากนั้นทำการลบกราฟ และกำหนดให้ตัวแปร StartRX มีค่าเท่ากับ false เพื่อสิ้นสุดการรับข้อมูล



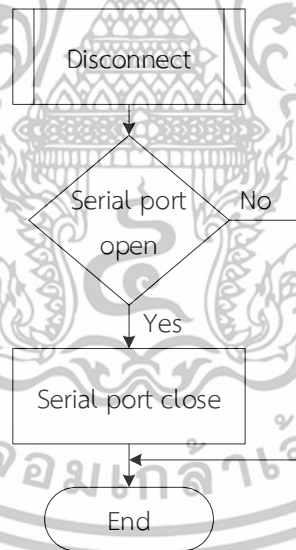
รูปที่ 3.15 โพล์ชาร์ตของฟังก์ชัน Clear

- **Setting** ฟังก์ชันตั้งค่าโปรแกรม โพล์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.16 ประกอบด้วยการตั้งค่าตำแหน่งการบันทึกข้อมูล และกราฟ การตั้งค่าเวลาในการบันทึกข้อมูลอัตโนมัติ และการตั้งค่าเวลาในการหยุดรับข้อมูลอัตโนมัติ



รูปที่ 3.16 โฟลว์ชาร์ตของฟังก์ชัน Setting

- Disconnect ฟังก์ชันยกเลิกการเชื่อมต่อกับพอร์ตอนุกรม โฟลว์ชาร์ตของฟังก์ชันแสดงดังรูปที่ 3.17 ทำหน้าที่ยกเลิกการเชื่อมต่อระหว่างคอมพิวเตอร์กับระบบรับรู้กลิ่น



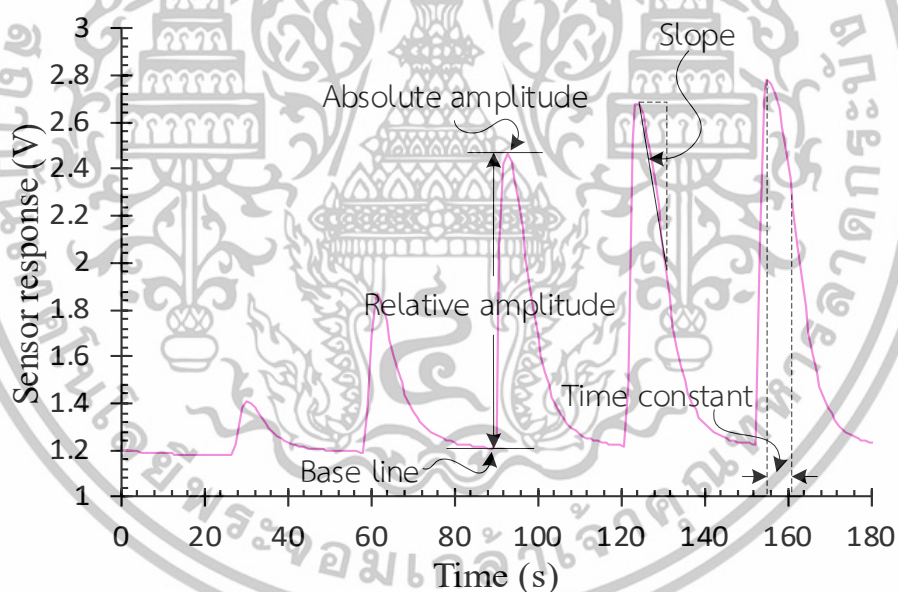
รูปที่ 3.17 โฟลว์ชาร์ตของฟังก์ชัน Disconnect

ทั้งนี้โค้ดทั้งหมดของโปรแกรมเชื่อมต่อกับผู้ใช้ที่พัฒนาขึ้นนี้ถูกใส่ไว้ในภาคผนวกท้ายเล่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 อัลกอริทึมสำหรับดึงคุณสมบัติที่สนใจออกจากผลการทดลองที่พัฒนาขึ้น

หลังจากทำการวัดผลตอบสนองของเซ็นเซอร์ ในหัวข้อนี้จะเป็นการนำผลตอบสนองจากเซ็นเซอร์ที่วัดได้มาประมวลผลเพื่อหาคุณสมบัติที่สำคัญโดยใช้อัลกอริทึมที่พัฒนาขึ้นมาสำหรับคำนวณหาคุณสมบัติต่าง ๆ ได้แก่ แอมพลิจูดเชิงสัมบูรณ์ (Absolute amplitude) แอมพลิจูดเชิงสัมพัทธ์ (Relative amplitude) ค่าคงที่เวลา (Time constant) ความชัน (Slope) และอัลกอริทึมสำหรับการนอร์มัลไลเซชัน (Normalization) ผลตอบสนองจากเซ็นเซอร์ที่วัดได้จากการทดลองนั้นจะมีลักษณะดังแสดงในรูปที่ 3.18 จะมีการตอบสนองในส่วนที่เป็นเบสไลน์ (Base line) คือการตอบสนองต่ออากาศ และหลังจากเซ็นเซอร์สัมผัสกับกลิ่นผลตอบสนองจะเปลี่ยนแปลงไป จากภาพผลตอบสนองมีการเปลี่ยนแปลงสูงขึ้น จนการเปลี่ยนแปลงถึงค่าสูงที่สุดแล้วผลตอบสนองค่อย ๆ คืบกลับสู่ค่าเบสไลน์อีกครั้ง ค่าสูงที่สุดนี้เรียกว่าแอมพลิจูดเชิงสัมบูรณ์ และความสัมพันธ์ระหว่างค่าเบสไลน์กับค่าสูงสุดเรียกว่าแอมพลิจูดเชิงสัมพัทธ์ ส่วนค่าคงที่เวลาคือช่วงเวลาที่มีผลตอบสนองค่อย ๆ ลดกลับลงมาเท่ากับ 37% ของค่าแอมพลิจูดเชิงสัมพัทธ์ และความชันคือค่าความชันในช่วงเวลาที่ผลตอบสนองค่อย ๆ ลดกลับสู่ค่าเบสไลน์



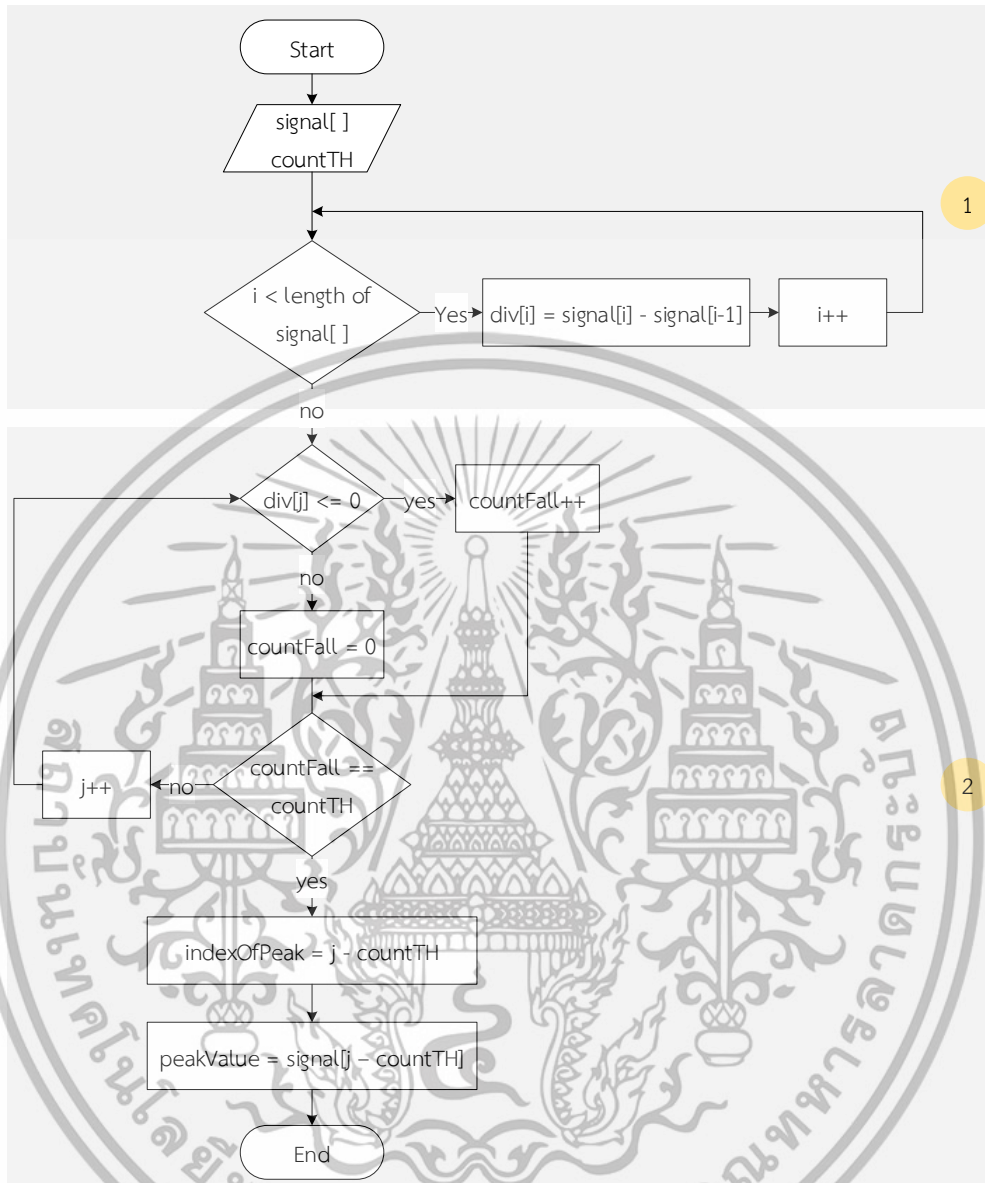
รูปที่ 3.18 ตัวอย่างผลตอบสนองจากเซ็นเซอร์กลิ่น

3.5.1 อัลกอริทึมสำหรับหาแอมพลิจูดเชิงสัมบูรณ์

อัลกอริทึมสำหรับคำนวณแอมพลิจูดเชิงสัมบูรณ์ ดังแสดงในรูปที่ 3.19 ขั้นตอนที่ 1 คือการนำผลตอบสนองจากเซ็นเซอร์ที่เก็บไว้ในตัวแปร `signal[]` นำมาคำนวณอนุพันธ์อันดับหนึ่ง (First derivative) หรือ `div[i]` คือการนำข้อมูลในตัวแปร `signal[]` ในอินเด็กซ์นั้น `signal[i]` มาลบด้วยข้อมูลในอินเด็กซ์ก่อนหน้า `signal[i-1]` ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{div}[i] = \text{signal}[i] - \text{signal}[i - 1] \quad (3.2)$$



รูปที่ 3.19 อัลกอริทึมสำหรับคำนวณหาแอมพลิจูดเชิงสัมบูรณ์

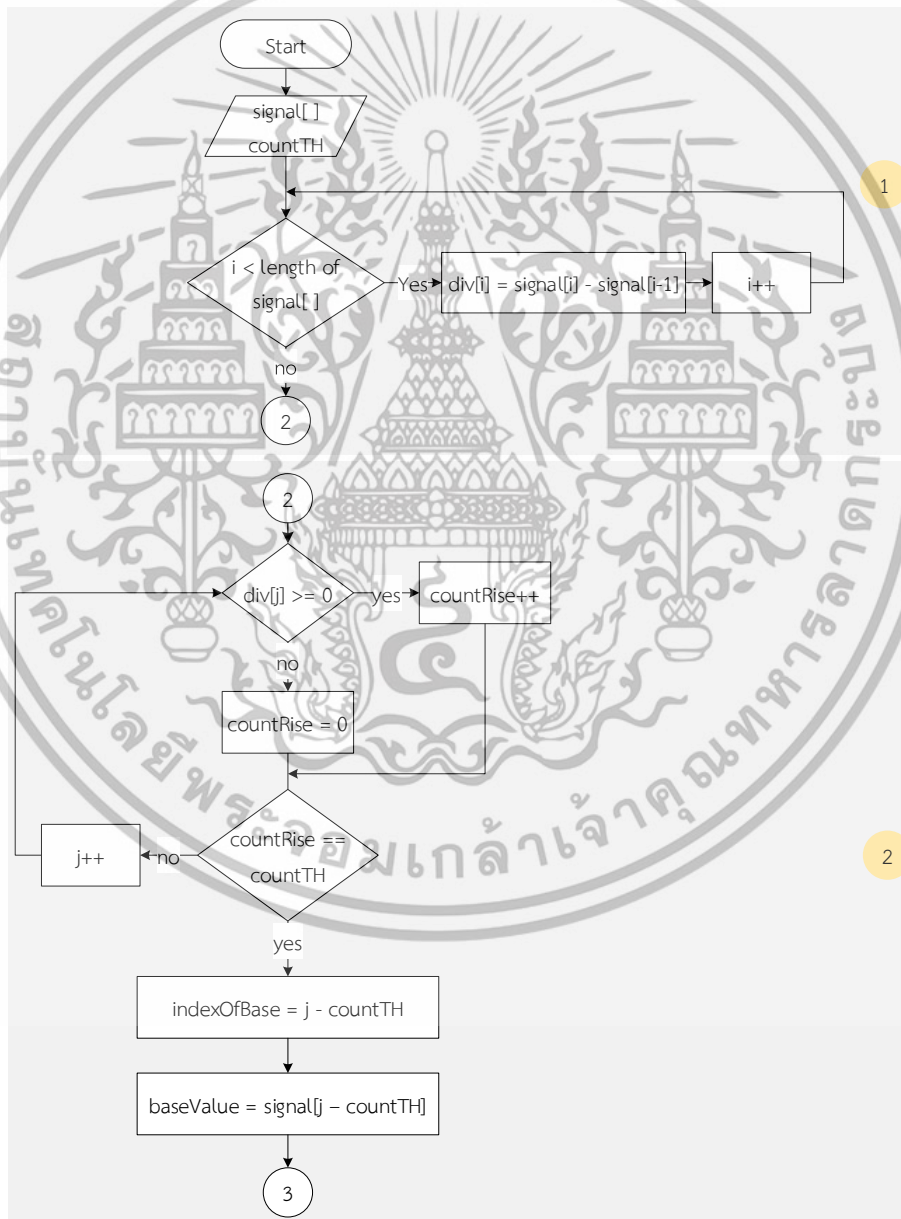
การคำนวณอนุพันธ์อันดับหนึ่งจะทำให้สามารถรู้แนวโน้มการเพิ่มขึ้นหรือลดลงของสัญญาณได้ ถ้าสัญญาณมีแนวโน้มเพิ่มขึ้นค่าที่คำนวณได้ในตัวแปร $\text{div}[i]$ จะมีค่าเป็นบวก แต่ถ้าหากสัญญาณมีแนวโน้มลดลงค่าที่คำนวณได้จะมีค่าเป็นศูนย์หรือเป็นลบ ขั้นตอนที่ 2 เป็นการตรวจสอบแนวโน้มของสัญญาณด้วยการตรวจสอบค่าในตัวแปร $\text{div}[i]$ ว่าจะน้อยกว่าหรือเท่ากับศูนย์หรือไม่ ถ้าเงื่อนไขเป็นจริงแสดงว่าสัญญาณกำลังมีค่าลดลง จึงทำการเพิ่มค่าให้กับตัวแปร countFall หนึ่งค่า และทำการตรวจสอบต่อไปเรื่อย ๆ จนค่าในตัวแปร countFall มีค่าเท่ากับค่าเทรชโฮลด์หรือ countTH แสดงว่าสัญญาณมีการลดลงอย่างต่อเนื่อง จึงทำให้ได้ค่าที่เป็นจุดสูงสุดของสัญญาณคือค่าในตำแหน่ง j

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

countTH ผลลัพธ์ที่ได้จากกระบวนการทำงานนี้คือ อินเด็กซ์ indexOfPeak และขนาด peakValue ของสัญญาณที่ตำแหน่งสูงสุด ซึ่งขนาดของสัญญาณที่ตำแหน่งสูงสุดก็คือแอมพลิจูดเชิงสัมบูรณ์

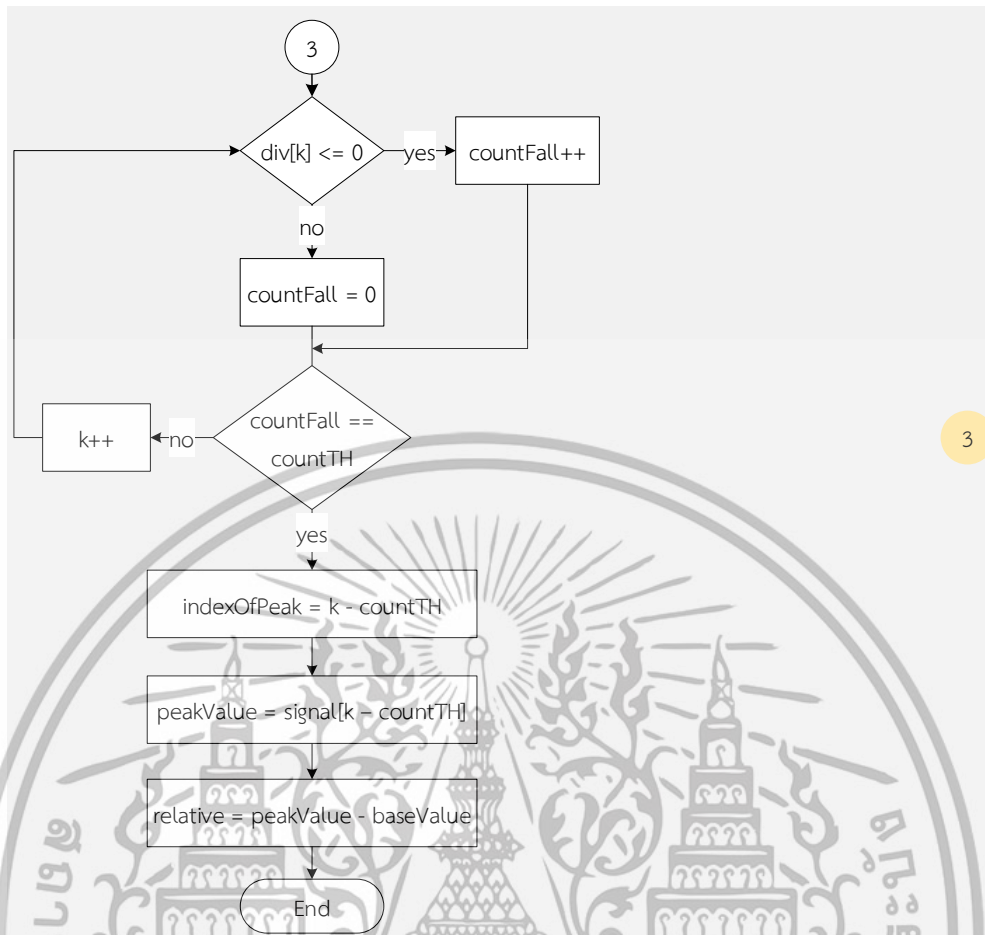
3.5.2 อัลกอริทึมสำหรับหาแอมพลิจูดเชิงสัมพัทธ์

อัลกอริทึมสำหรับคำนวณแอมพลิจูดเชิงสัมพัทธ์ ดังแสดงในรูปที่ 3.20 ขั้นตอนที่ 1 คือการนำผลตอบสนองจากเซ็นเซอร์ที่เก็บไว้ในตัวแปร $signal[]$ มาคำนวณอนุพันธ์อันดับหนึ่ง หรือ $div[i]$ คือการนำข้อมูลในตัวแปร $signal[]$ ในอินเด็กซ์นั้น $signal[i]$ มาลบด้วยข้อมูลในอินเด็กซ์ก่อนหน้า $signal[i-1]$ ดังสมการ (3.2) ซึ่งจะเหมือนกับขั้นตอนที่ 1 ของการคำนวณแอมพลิจูดเชิงสัมบูรณ์



รูปที่ 3.20 อัลกอริทึมสำหรับคำนวณหาแอมพลิจูดเชิงสัมพัทธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 อัลกอริทึมสำหรับคำนวณหาแอมพลิจูดเชิงสัมพัทธ์ (ต่อ)

ขั้นตอนที่ 2 เป็นการตรวจสอบแนวโน้มของสัญญาณว่ามีแนวโน้มที่จะเพิ่มขึ้นหรือไม่ เพื่อทำการหาค่าเบสไลน์ด้วยการตรวจสอบค่าในตัวแปร $div[i]$ ว่ามากกว่าหรือเท่ากับศูนย์หรือไม่ ถ้าเงื่อนไขเป็นจริงแสดงว่าสัญญาณกำลังมีค่าเพิ่มขึ้น จึงทำการเพิ่มค่าให้กับตัวแปร $countRise$ หนึ่งค่า และทำการตรวจสอบต่อไปเรื่อย ๆ จนค่าในตัวแปร $countRise$ มีค่าเท่ากับค่าเทรชโฮลด์หรือ $countTH$ แสดงว่าสัญญาณมีการเพิ่มขึ้นอย่างต่อเนื่อง จึงทำให้ได้ค่าเบสไลน์ของสัญญาณคือค่าในตำแหน่ง $j - countTH$ ผลลัพธ์ที่ได้จากกระบวนการทำงานนี้คือ อินเด็กซ์ $indexOfBase$ และขนาด $baseValue$ ของสัญญาณที่ตำแหน่งเบสไลน์

ขั้นตอนที่ 3 เป็นการตรวจสอบแนวโน้มของสัญญาณว่ามีแนวโน้มที่จะลดลงหรือไม่ เพื่อทำการหาค่าสูงสุดของสัญญาณ ด้วยการตรวจสอบค่าในตัวแปร $div[i]$ ว่าน้อยกว่าหรือเท่ากับศูนย์หรือไม่ ถ้าเงื่อนไขเป็นจริงแสดงว่าสัญญาณกำลังมีค่าลดลงจึงทำการเพิ่มค่าให้กับตัวแปร $countFall$ หนึ่งค่า และทำการตรวจสอบต่อไปเรื่อย ๆ จนค่าในตัวแปร $countFall$ มีค่าเท่ากับค่าเทรชโฮลด์หรือ $countTH$ แสดงว่าสัญญาณมีการลดลงอย่างต่อเนื่อง จึงทำให้ได้ค่าที่เป็นจุดสูงสุดของสัญญาณคือค่าในตำแหน่ง $k - countTH$ ผลลัพธ์ที่ได้จากกระบวนการทำงานนี้คือ อินเด็กซ์ $indexOfPeak$ และขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

peakValue ของสัญญาณที่ตำแหน่งสูงสุด หลังจากนั้นทำการคำนวณหาค่าแอมพลิจูดเชิงสัมพันธ์ดังสมการ

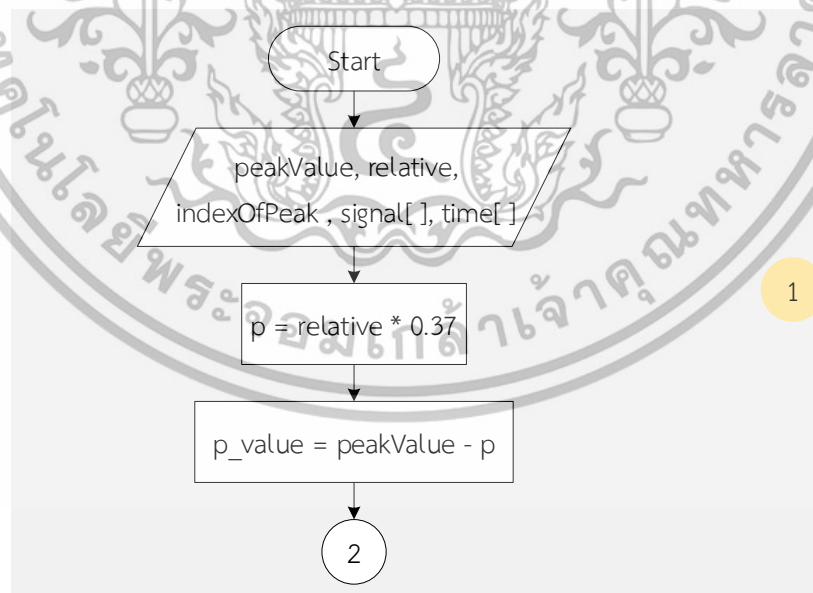
$$\text{relative} = \text{peakValue} - \text{baseValue} \quad (3.3)$$

3.5.3 อัลกอริทึมสำหรับหาค่าคงที่เวลา

อัลกอริทึมสำหรับคำนวณค่าคงที่เวลา ดังแสดงในรูปที่ 3.21 ซึ่งจำเป็นต้องใช้ข้อมูลแอมพลิจูดเชิงสัมพันธ์ และข้อมูล indexOfPeak สามารถคำนวณได้โดยใช้อัลกอริทึมในหัวข้อที่ 3.5.1 และข้อมูลแอมพลิจูดเชิงสัมพันธ์ สามารถคำนวณได้โดยใช้อัลกอริทึมในหัวข้อที่ 3.5.2 ขั้นตอนที่ 1 คำนวณหาค่า 37% ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ เก็บไว้ในตัวแปร p หลังจากนั้นนำข้อมูลแอมพลิจูดเชิงสัมพันธ์ลบด้วยค่าในตัวแปร p เก็บผลลัพธ์ที่ได้ไว้ในตัวแปร p_value

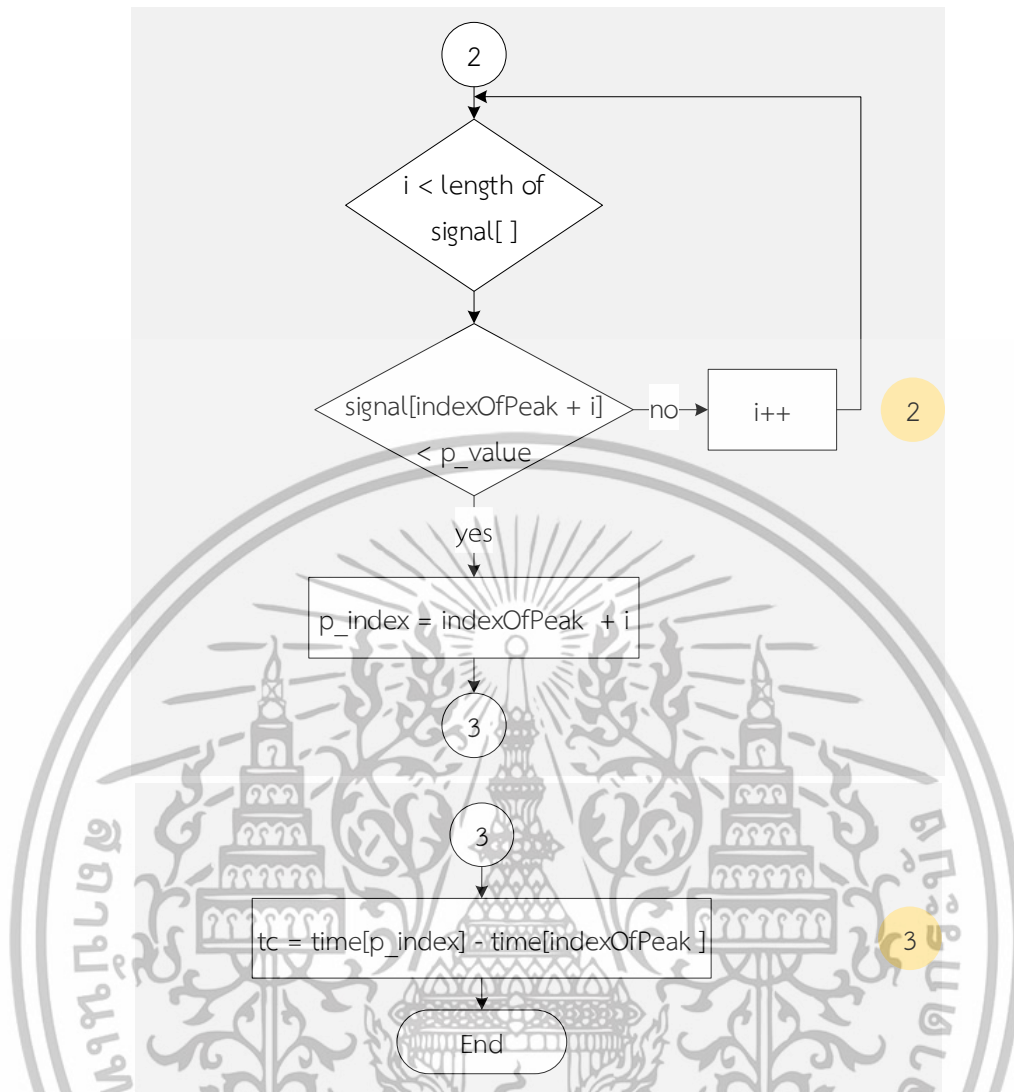
ขั้นตอนที่ 2 นำผลตอบสนองจากเซ็นเซอร์ที่เก็บไว้ในตัวแปร signal[] เปรียบเทียบกับค่าในตัวแปร p_value เพื่อหาค่าตำแหน่งที่ผลตอบสนองจากเซ็นเซอร์ลดลง 37% และเก็บตำแหน่งไว้ในตัวแปร p_index

ขั้นตอนที่ 3 นำข้อมูลเวลาที่เก็บไว้ในตัวแปร time[] ในตำแหน่งที่ p_index ลบด้วยข้อมูลในตำแหน่งที่ indexOfPeak เพื่อคำนวณหาค่าคงที่เวลา



รูปที่ 3.21 อัลกอริทึมสำหรับคำนวณหาค่าคงที่เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 อัลกอริทึมสำหรับคำนวณหาค่าคงที่เวลา(ต่อ)

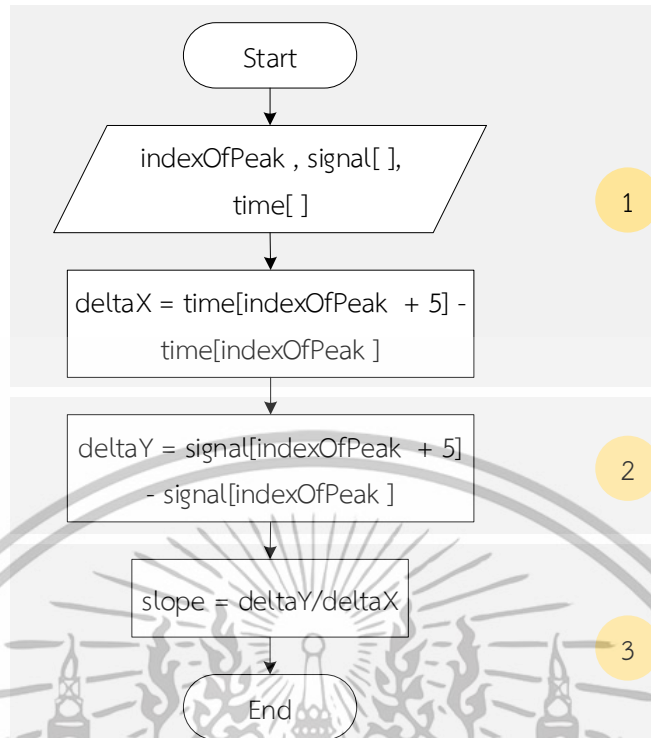
3.5.4 อัลกอริทึมสำหรับหาค่าความชัน

อัลกอริทึมสำหรับคำนวณความชัน ดังแสดงในรูปที่ 3.22 ซึ่งจำเป็นต้องใช้ข้อมูล `indexOfPeak` สามารถคำนวณได้โดยใช้อัลกอริทึมในหัวข้อที่ 3.5.1 ขั้นตอนที่ 1 คือการคำนวณค่าการเปลี่ยนแปลงในแกน X หรือแกนเวลา โดยการนำข้อมูลเวลาที่เก็บไว้ในตัวแปร `time[]` ในตำแหน่งที่ `indexOfPeak + 5` ลบด้วยข้อมูลในตำแหน่งที่ `indexOfPeak` ผลลัพธ์ที่ได้เก็บไว้ในตัวแปร `deltaX`

ขั้นตอนที่ 2 คือการคำนวณค่าการเปลี่ยนแปลงในแกน Y โดยการนำผลตอบสนองจากเซ็นเซอร์ที่เก็บไว้ในตัวแปร `signal[]` ในตำแหน่งที่ `indexOfPeak + 5` ลบด้วยข้อมูลในตำแหน่งที่ `indexOfPeak` ผลลัพธ์ที่ได้เก็บไว้ในตัวแปร `deltaY`

ขั้นตอนที่ 3 คำนวณหาความชันด้วยการหารค่าการเปลี่ยนแปลงในแกน Y หรือ `deltaY` ด้วยค่าการเปลี่ยนแปลงในแกน X หรือ `deltaX`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 อัลกอริทึมสำหรับคำนวณหาความชัน

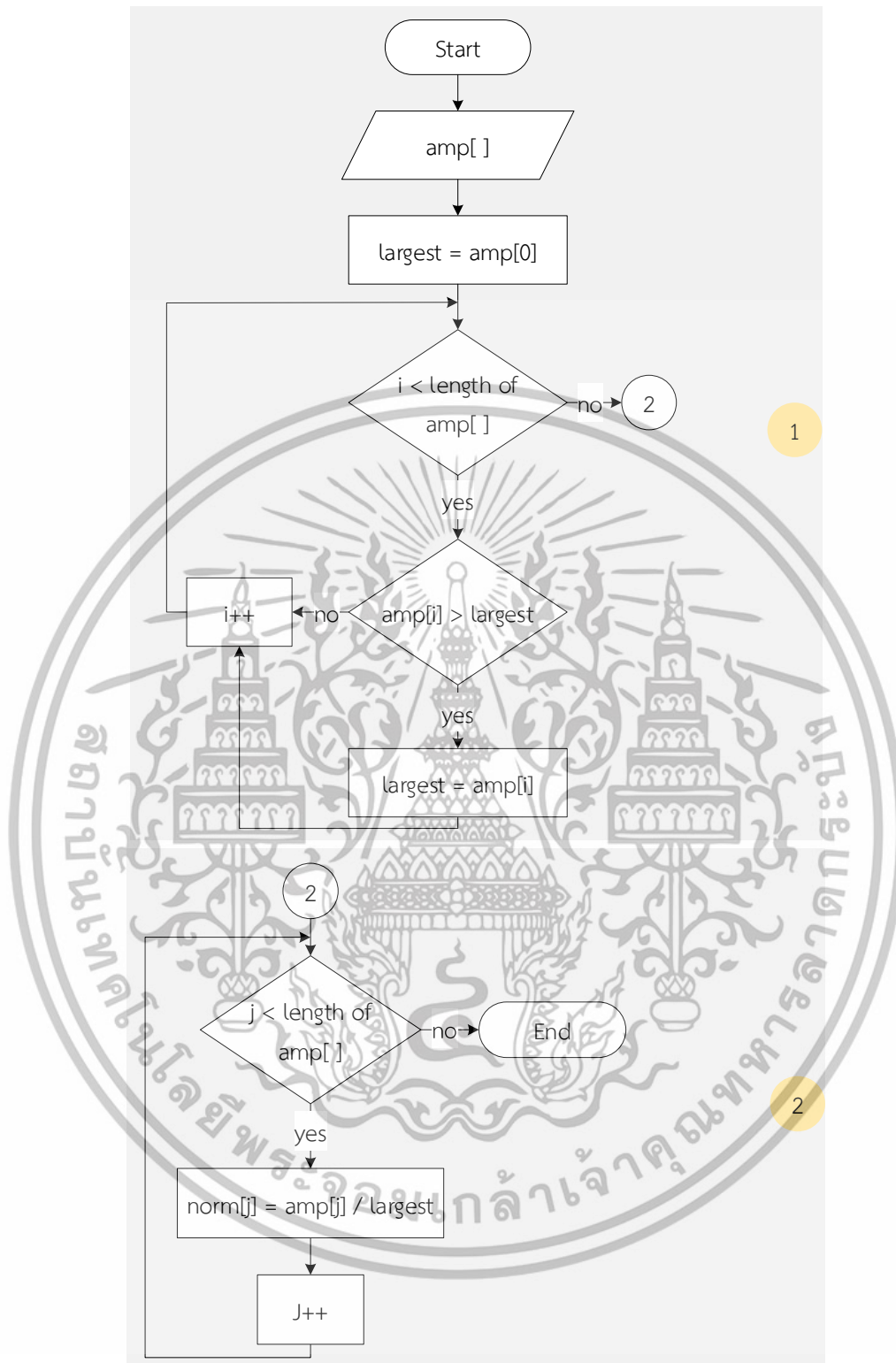
3.5.5 อัลกอริทึมสำหรับนอร์มัลไลเซชัน

หลังจากทำการคำนวณหาคุณสมบัติของผลตอบสนองจากเซ็นเซอร์คือ แอมพลิจูดเชิงสัมบูรณ์ แอมพลิจูดเชิงสัมพันธ์ ค่าคงที่เวลา และความชันเป็นที่เรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการนำคุณสมบัติต่าง ๆ มาทำการนอร์มัลไลเซชัน (Normalization) โดยมีกระบวนการทั้งหมด 2 ขั้นตอนคือการหาค่าสูงสุด และการนำค่าสูงสุดไปหารค่าอื่น ๆ ทั้งหมด อัลกอริทึมสำหรับการนอร์มัลไลเซชันดังแสดงในรูปที่ 3.23

ขั้นตอนที่ 1 กระบวนการหาจำนวนที่มีค่าสูงสุดจากค่าที่เก็บไว้ในตัวแปร amp[] เริ่มต้นด้วยการกำหนดให้ตัวแปร largest มีค่าเท่ากับข้อมูลในตัวแปร amp[0] คือเป็นการกำหนดเป็นข้อมูลที่มีค่าสูงสุด และทำการเปรียบเทียบจำนวนทั้งหมดในตัวแปร amp[] กับข้อมูลในตัวแปร largest ถ้าข้อมูลในตัวแปร amp[i] มีค่ามากกว่า largest จะกำหนดให้ตัวแปร largest มีค่าเท่ากับข้อมูลในตัวแปร amp[i] และทำการเปรียบเทียบต่อไปจนครบทุกจำนวนในตัวแปร amp[] จะทำให้ได้จำนวนที่มีค่าสูงสุดเก็บอยู่ในตัวแปร largest

ขั้นตอนที่ 2 กระบวนการนอร์มัลไลเซชัน คือการนำข้อมูลที่มีค่าสูงสุดที่เก็บไว้ในตัวแปร largest ไปหารจำนวนอื่นทั้งหมดในตัวแปร amp[] และเก็บผลลัพธ์การหารที่ได้ไว้ในตัวแปร norm[] ซึ่งก็คือผลลัพธ์จากการนอร์มัลไลเซชันนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 อัลกอริทึมสำหรับการนอร์มัลไลเซชัน

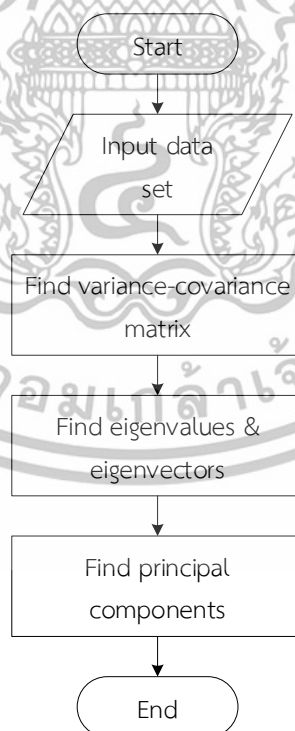
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การวิเคราะห์องค์ประกอบหลักโดยใช้โปรแกรม MATLAB

การคำนวณหาคุณสมบัติที่สำคัญของผลตอบสนองจากเซ็นเซอร์ในหัวข้อที่ 3.5 ทำให้เราทราบคุณสมบัติ 4 อย่างได้แก่ แอมพลิจูดเชิงสัมบูรณ์ แอมพลิจูดเชิงสัมพัทธ์ ค่าคงที่เวลา และความชัน ซึ่งเป็นคุณสมบัติที่สามารถนำมาใช้ในกระบวนการแยกแยะกลิ่นได้ หลังจากนั้นนำคุณสมบัติต่าง ๆ ผ่านกระบวนการนอร์มัลไลเซชัน ในขั้นตอนนี้เป็นการนำผลลัพธ์ที่ได้มาเข้าสู่กระบวนการคำนวณการวิเคราะห์องค์ประกอบหลัก เพื่อผลลัพธ์ในการแยกแยะกลิ่น โดยจะเป็นการวิเคราะห์องค์ประกอบหลักจากผลตอบสนองจากเซ็นเซอร์กลิ่นโดยใช้โปรแกรม MATLAB การคำนวณจะทำตามขั้นตอนทางคณิตศาสตร์ที่ได้อธิบายไว้ในหัวข้อ 2.5.4 โดยจะอธิบายการวิเคราะห์องค์ประกอบหลักทั้งสองวิธีได้แก่ การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม และการวิเคราะห์องค์ประกอบหลักโดยใช้ เมทริกซ์สหสัมพันธ์

3.6.1 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม

โปรแกรม MATLAB มีฟังก์ชันสำเร็จรูปสำหรับการคำนวณทางคณิตศาสตร์ เพื่อรองรับความต้องการของผู้ใช้งานมากมาย เพื่อผลลัพธ์ที่ต้องการคำนวณในแต่ละขั้นตอนจะใช้ฟังก์ชันสำเร็จรูปต่าง ๆ แต่จะมีการจัดเรียงข้อมูลบางส่วนใหม่ เพื่อให้ตรงกับทฤษฎีทางคณิตศาสตร์ ขั้นตอนการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมดังแสดงในรูปที่ 3.24



รูปที่ 3.24 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนแรกในการวิเคราะห์องค์ประกอบหลักคือการนำข้อมูลอินพุต (X) มาคำนวณหาเมทริกซ์ความแปรปรวนร่วม (S) ด้วยคำสั่งดังนี้

```
S = cov(X, 'omitrows');
```

'omitrows' คือคำสั่งให้ยกเว้นการคำนวณแถวที่มีค่าเป็น NaN ทั้งหมด ตัวอย่างของเมทริกซ์ความแปรปรวนร่วม (S) ที่คำนวณได้ดังแสดงในรูปที่ 3.25

0.0307	0.0158	-0.0132	0.0312
0.0158	0.0674	-0.0204	0.0410
-0.0132	-0.0204	0.0174	-0.0232
0.0312	0.0410	-0.0232	0.0508

รูปที่ 3.25 ตัวอย่างของเมทริกซ์ความแปรปรวนร่วม (S)

หลังจากนั้นทำการคำนวณหาค่าลักษณะเฉพาะ (Eigenvalues) และเวกเตอร์ลักษณะเฉพาะ (Eigenvectors) ของเมทริกซ์ความแปรปรวนร่วม (S) ผลลัพธ์ที่ได้ประกอบด้วย V คือเวกเตอร์ลักษณะเฉพาะและ D คือค่าลักษณะเฉพาะ ตัวอย่างที่คำนวณได้ดังแสดงในรูปที่ 3.26 ด้วยการใช้คำสั่งดังนี้

```
[V, D] = eig(S);
```

0.4986	0.4727	0.6343	0.3545	0.00365	0	0	0
0.1975	0.2426	-0.6968	0.6455	0	0.00739	0	0
-0.4419	0.8385	-0.1104	-0.2991	0	0	0.03115	0
-0.7192	-0.1209	0.3162	0.6068	0	0	0	0.12399

(ก)

(ข)

รูปที่ 3.26 ตัวอย่างของ (ก) เวกเตอร์ลักษณะเฉพาะ (ข) ค่าลักษณะเฉพาะ

หลังจากนั้นจะทำการจัดเรียงข้อมูลในเวกเตอร์ลักษณะเฉพาะเสียใหม่ โดยคอลัมภ์ที่สอดคล้องกันกับค่าที่มากที่สุดของค่าลักษณะเฉพาะจะถูกจัดเรียงไว้ทางด้านซ้ายของตาราง และคอลัมภ์ที่สอดคล้องกับค่าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้อยกว่าจะถูกจัดเรียงเป็นคอลัมป์ถัดมา จำนวนคอลัมป์จะตรงกันกับจำนวนขององค์ประกอบหลักที่ต้องการ ตัวอย่างการจัดเรียงที่ได้ดังแสดงในรูปที่ 3.27 เวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่ในชื่อว่า `coeff` ด้วยคำสั่งดังนี้

```
for i=1: size(X,2) % number of column
    coeff(:,i) = V(:,size(X,2)-(i-1));
end
```

0.35451	0.63432	0.47266	0.49856
0.64551	-0.6968	0.24258	0.19751
-0.2991	-0.1104	0.83853	-0.4419
0.6068	0.31621	-0.1209	-0.7192

รูปที่ 3.27 ตัวอย่างของเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่

ขั้นตอนสุดท้ายคือการคำนวณหาองค์ประกอบหลัก สามารถทำได้โดยการคูณข้อมูลอินพุต (X) ด้วยเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่นั้นคือ `coeff` จะทำให้ได้ผลลัพธ์คือองค์ประกอบหลักที่ 1, 2, 3 และ 4 ได้แก่ PC1, PC2, PC3 และ PC4 ตามลำดับ สามารถคำนวณด้วยการใช้คำสั่งดังนี้

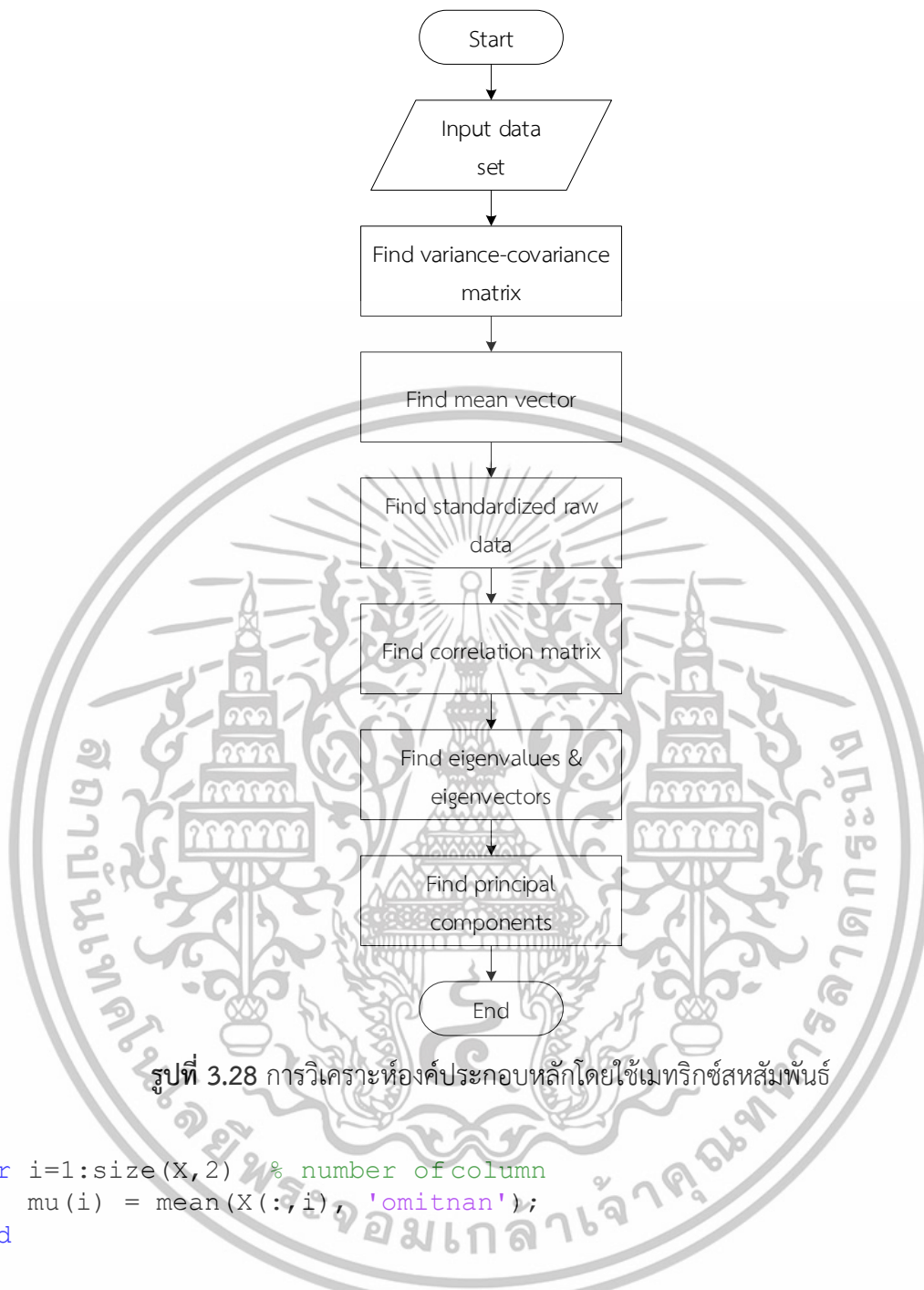
```
for j=1:size(coeff,2) % number of column
    for i=1:size(X,1) % number of rows
        PC(i,j) = sum(X(i,:).*(coeff(:,j)'));
    end
end
PC1 = PC(:,1);
PC2 = PC(:,2);
PC3 = PC(:,3);
PC4 = PC(:,4);
```

3.6.2 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์

การวิเคราะห์องค์ประกอบหลักด้วยวิธีนี้มีขั้นตอนมากกว่าที่ได้อธิบายในหัวข้อที่

3.6.1 ขั้นตอนในการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ดังแสดงในรูปที่ 3.28 ขั้นตอนแรกในการวิเคราะห์องค์ประกอบหลักคือการนำข้อมูลอินพุต (X) มาคำนวณหาเมทริกซ์ความแปรปรวนร่วม (S) โดยมีกระบวนการเช่นเดียวกับที่ได้อธิบายในหัวข้อที่ 3.6.1 สามารถใช้คำสั่งต่าง ๆ ที่กล่าวมาแล้วได้ หลังจากนั้นนำข้อมูลอินพุต (X) มาคำนวณหาเวกเตอร์ของค่าเฉลี่ย สามารถหาเวกเตอร์ของค่าเฉลี่ย μ ได้ด้วยคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์

```

for i=1:size(X,2) % number of column
    mu(i) = mean(X(:,i), 'omitnan');
end
  
```

หลังจากนั้นทำการคำนวณเพื่อหาค่ามาตรฐานของข้อมูลดิบ (Standardized raw data) (Z) จากข้อมูลอินพุต (X) เวกเตอร์ของค่าเฉลี่ย μ และเมทริกซ์ความแปรปรวนร่วม (S) ด้วยคำสั่งดังนี้

```

for i=1:size(X,2) % number of column
    Z(:,i) = (X(:,i)-mu(i))/sqrt(S(i,i));
End
  
```

ขั้นตอนต่อไปเป็นการนำข้อมูลอินพุต (X) มาทำการคำนวณหาเมทริกซ์สหสัมพันธ์ (R) ด้วยคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[R] = corrcoef(X, 'Rows', 'complete')
```

คำสั่ง 'Rows', 'complete' คือคำสั่งให้ยกเว้นการคำนวณแถวที่มีค่าเป็น NaN ทั้งหมด ตัวอย่างของเมทริกซ์สหสัมพันธ์ (R) ดังแสดงในรูปที่ 3.29

1	0.34786	-0.5717	0.79045
0.34786	1	-0.5949	0.7006
-0.5717	-0.5949	1	-0.7804
0.79045	0.7006	-0.7804	1

รูปที่ 3.29 ตัวอย่างของเมทริกซ์สหสัมพันธ์ (R)

หลังจากนั้นทำการคำนวณหาค่าลักษณะเฉพาะ และเวกเตอร์ลักษณะเฉพาะของเมทริกซ์สหสัมพันธ์ (R) ผลลัพธ์ที่ได้ประกอบด้วย V คือเวกเตอร์ลักษณะเฉพาะ และ D คือค่าลักษณะเฉพาะตัวอย่างที่คำนวณได้ ดังแสดงในรูปที่ 3.30 ด้วยการใช้คำสั่งดังนี้

```
[V,D] = eig(R)
```

-0.4541	0.3239	0.6860	0.4672	0.08785	0	0	0
-0.3171	0.4264	-0.7177	0.4500	0	0.33839	0	0
0.2001	0.8323	0.0865	-0.5097	0	0	0.65884	0
0.8082	0.1433	0.0824	0.5652	0	0	0	2.91492

(ก)

(ข)

รูปที่ 3.30 ตัวอย่างของ (ก) เวกเตอร์ลักษณะเฉพาะ (ข) ค่าลักษณะเฉพาะ

หลังจากนั้นจะทำการจัดเรียงข้อมูลในเวกเตอร์ลักษณะเฉพาะเสียใหม่ โดยคอลัมภ์ที่สอดคล้องกันกับค่าที่มากที่สุดของค่าลักษณะเฉพาะจะถูกจัดเรียงไว้ทางด้านซ้ายของตาราง และคอลัมภ์ที่สอดคล้องกับค่าน้อยกว่าจะถูกจัดเรียงเป็นคอลัมภ์ถัดมา จำนวนคอลัมภ์จะตรงกันกับจำนวนขององค์ประกอบหลักที่ต้องการ ตัวอย่างการจัดเรียงที่ได้ดังแสดงในรูปที่ 3.31 เวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่ในชื่อว่า coeff ด้วยคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for i=1:size(X,2) % number of column
    coeff(:,i) = V(:,size(X,2)-(i-1));
end

```

0.4672	0.6860	0.3239	-0.4541
0.4500	-0.7177	0.4264	-0.3171
-0.5097	0.0865	0.8323	0.2001
0.5652	0.0824	0.1433	0.8082

รูปที่ 3.31 ตัวอย่างของเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่

ขั้นตอนสุดท้ายคือการคำนวณหองค์ประกอบหลัก สามารถทำได้โดยการคูณค่ามาตรฐานของข้อมูลดิบ (z) ด้วยเวกเตอร์ลักษณะเฉพาะที่ถูกจัดเรียงใหม่นั้นคือ `coeff` จะทำให้ได้ผลลัพธ์คือองค์ประกอบหลักที่ 1, 2, 3 และ 4 ได้แก่ PC1, PC2, PC3 และ PC4 ตามลำดับ สามารถคำนวณด้วยการใช้คำสั่งดังนี้

```

for j=1:size(coeff,2) % number of column
    for i=1:size(Z,1) % number of rows
        PC(i,j) = sum(Z(i,:) .* (coeff(:,j)'));
    end
end
PC1 = PC(:,1);
PC2 = PC(:,2);
PC3 = PC(:,3);
PC4 = PC(:,4);

```

การวิเคราะห์องค์ประกอบหลักโดยใช้โปรแกรม MATLAB นี้ทำให้สามารถคำนวณค่าน้ำหนักหรือ `coeff` และองค์ประกอบหลักของตัวแปรเดิมคือ อินพุต (X) ซึ่งก็คือข้อมูลผลตอบสนองจากเซ็นเซอร์ที่มีต่อกลิ่นเป้าหมายที่ได้จากการทดลอง ผลลัพธ์เหล่านี้จะถูกใช้เป็นฐานข้อมูลในกระบวนการแยกแยะกลิ่น ผลตอบสนองของเซ็นเซอร์ที่มีต่อกลิ่นที่ถูกวัดครั้งใหม่จะทำให้ได้ข้อมูลอินพุต (X) ชุดใหม่ และคูณด้วยค่าน้ำหนักเดิมที่เป็นฐานข้อมูล จะทำให้สามารถสร้างองค์ประกอบหลักของข้อมูลใหม่นี้ได้ และทำการเปรียบเทียบองค์ประกอบหลักใหม่กับองค์ประกอบหลักเดิมที่เป็นฐานข้อมูลของกลิ่นต่าง ๆ ที่ได้จัดทำเตรียมไว้ จะทำให้สามารถรู้ได้ว่าเป็นกลิ่นเดิมหรือไม่

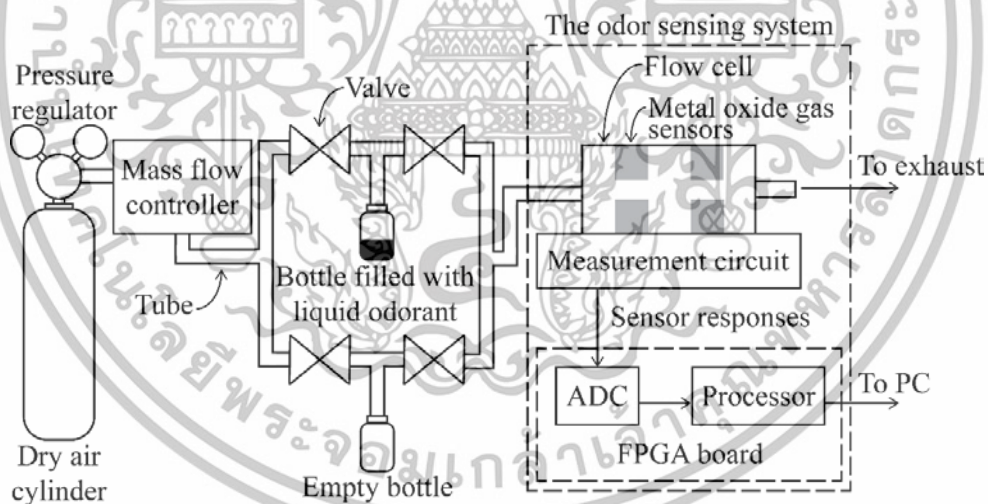
บทที่ 4

การทดลองและวิเคราะห์ผล

บทนี้จะกล่าวถึงการทดลองเพื่อประเมินผลระบบแยกแยะกลิ่นที่ได้ทำการพัฒนาขึ้น แบ่งออกเป็น การทดลองวัดกลิ่นที่มีอัตราการระเหยแตกต่างกันโดยใช้ระบบรับรู้กลิ่น การนำผลตอบสนองของเซ็นเซอร์ที่ได้มาวิเคราะห์ด้วยอัลกอริทึมเพื่อหาคุณสมบัติต่าง ๆ และคำนวณการวิเคราะห์องค์ประกอบหลัก เพื่อแยกแยะกลิ่นจากการทดลอง

4.1 การทดลองวัดกลิ่นที่มีอัตราการระเหยแตกต่างกัน

ในเบื้องต้นได้ทำการทดลองวัดกลิ่นด้วยระบบรับรู้กลิ่นที่พัฒนาขึ้นดังอธิบายในบทที่ 3 โดยทำการทดลองในระบบปิดเพื่อป้องกันผลกระทบที่เกิดจากความชื้นในอากาศ และความแปรปรวนของลม ห้องทดลองมีเครื่องปรับอากาศที่ควบคุมอุณหภูมิให้อยู่ที่ประมาณ 25°C โดยทำการวัดกลิ่นที่โมเลกุลมีอัตราการระเหยปานกลาง และมีอัตราการระเหยต่ำ ดังจะอธิบายในหัวข้อ 4.1.1 และ 4.1.2 ตามลำดับ ระบบการทดลองแสดงในรูปที่ 4.1



รูปที่ 4.1 ระบบการทดลองวัดกลิ่น

ในการปล่อยกลิ่น อากาศแห้ง (Zero ambient air) จากถังอากาศถูกจ่ายให้กับโซลินอยวาวด์ (LHDB0352115H, Lee) ที่เชื่อมต่ออยู่กับขวดบรรจุสารที่มีกลิ่นที่ใช้ในการทดลองและขวดเปล่า อัตราการไหลของอากาศในระบบถูกควบคุมด้วยแมสโฟลวคอนโทรลเลอร์โดยใช้ (RO7020, Tylan) โซลินอยวาวด์จะทำงานในลักษณะของสวิตช์ที่เลือกช่องทางไหลออกของอากาศจากถังอากาศ ไปยังขวดบรรจุสารที่มีกลิ่นหรือขวดเปล่าอย่างใดอย่างหนึ่ง โดยการสับสวิตช์ของโซลินอยวาวด์ จะถูกควบคุมโดยใช้สัญญาณพัลส์วิดท์มอดูเลชัน (Pulse Width Modulation : PWM) ที่ปล่อยออกจาก FPGA (ไม่ได้แสดงในรูป) ที่

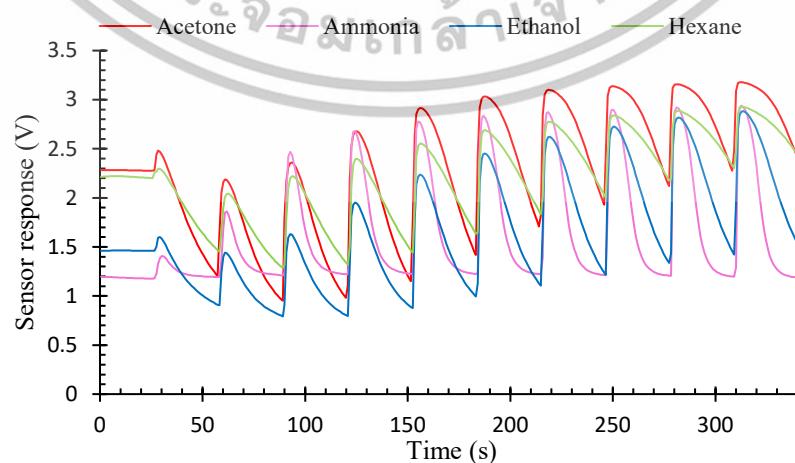
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชื่อมต่อกับซอฟต์แวร์เชื่อมต่อกับผู้ใช้บนคอมพิวเตอร์ (ไม่ได้แสดงในรูป) โดยการปรับค่าดิวตี้ไซเคิล (Duty cycle) ของสัญญาณ PWM จะทำให้สามารถควบคุมความเข้มข้นของกลิ่นได้ด้วยการผสมอากาศที่ถูกปล่อยจากขวดเปล่ากับกลิ่นที่ถูกปล่อยจากขวดที่บรรจุสารที่มีกลิ่นด้วยอัตราส่วนตามที่ระบุในซอฟต์แวร์เชื่อมต่อกับผู้ใช้ โดยโซลินอยด์วาล์วที่ใช้ในการทดลองนี้มีระยะเวลาในการสวิตช์เร็วที่สุดที่เป็นไปได้คือ 4 ms

ในการวัดกลิ่น กลิ่นที่ถูกปล่อยจะไหลผ่านไปยังเซ็นเซอร์เซลล์ที่บรรจุเซ็นเซอร์ชนิดเมทัลออกไซด์ 4 ตัว คือ TGS2600, TGS2602, TGS2611 และ TGS2620 ซึ่งเป็นเซ็นเซอร์ที่ถูกพัฒนาขึ้นเพื่อวัดคาร์บอนมอนอกไซด์ แอมโมเนียกับโทลูอิน มีเทนกับไอโซบิวทีน และเอทานอล ตามลำดับ ดังอธิบายในหัวข้อที่ 3.1 เซ็นเซอร์จะถูกต่ออยู่กับวงจรวัดที่อธิบายในหัวข้อที่ 3.2 ผลตอบสนองจากเซ็นเซอร์จะถูกส่งไปยังบอร์ด FPGA (DE0 Nano, Terasic) ที่มีวงจรแปลงแอนะล็อกเป็นดิจิทัล แปลงข้อมูลให้อยู่ในรูปแบบของสัญญาณดิจิทัลและส่งต่อไปยังหน่วยประมวลผลแบบอ่อนที่อธิบายในหัวข้อที่ 3.3 ที่จะส่งข้อมูลผ่าน UART ไปยังคอมพิวเตอร์ ข้อมูลจะถูกแสดงผลและถูกบันทึกให้อยู่ในรูปแบบของไฟล์ .csv ทุก ๆ 1 วินาที ด้วยโปรแกรมเชื่อมต่อกับผู้ใช้บนคอมพิวเตอร์ที่อธิบายในหัวข้อที่ 3.4 โดยตลอดการทดลองความดันอากาศถูกตั้งค่าไว้ที่ 8 psi และอัตราการไหลของอากาศถูกตั้งค่าไว้ที่ 250 mL/min

4.1.1 การทดลองวัดกลิ่นที่มีอัตราการระเหยปานกลาง

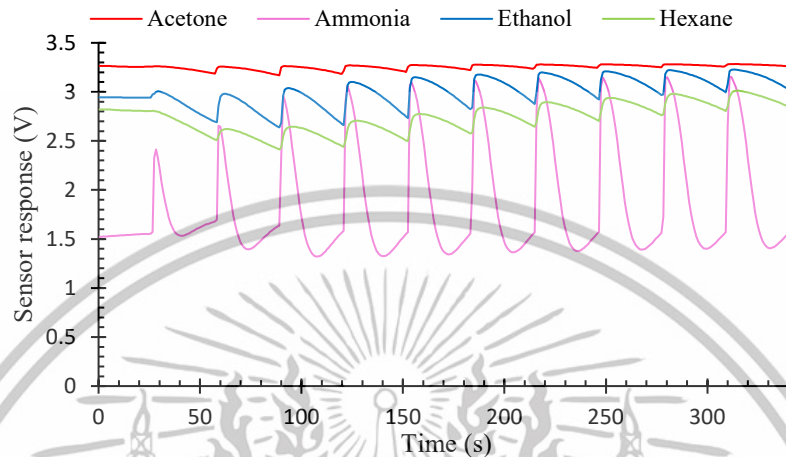
ในการทดลองวัดกลิ่นที่มีอัตราการระเหยปานกลาง เลือกใช้สารมีกลิ่นในการทดลอง คือ อะซีโตน แอมโมเนีย เอทานอล และเฮกเซน เนื่องจากเป็นสารบริสุทธิ์ที่จะช่วยลดความซับซ้อนในการตีความผลที่ได้จากการทดลอง โดยในการทดลองอากาศเปล่าและกลิ่นจะถูกปล่อยเข้าสู่เซ็นเซอร์เซลล์เป็นเวลา 30 วินาทีและ 1 วินาทีตามลำดับ โดยปล่อยสลับกันไปเรื่อย ๆ ความเข้มข้นของกลิ่นถูกเปลี่ยนแปลงจาก 10% ถึง 100% ด้วยอัตราการเพิ่มครั้งละ 10% ถึงขั้นตอนนี้ผลตอบสนองจากเซ็นเซอร์ที่ได้รับมีจำนวนทั้งหมด 160 ลูกคลื่น ดังแสดงในรูปที่ 4.2 ถึงรูปที่ 4.5 การทดลองดำเนินการซ้ำเช่นนี้รวมทั้งหมด 10 ครั้ง ทำให้ได้ผลตอบสนองจากเซ็นเซอร์รวมทั้งหมดจำนวน 1,600 ลูกคลื่น



รูปที่ 4.2 ผลตอบสนองของเซ็นเซอร์ TGS2600 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง

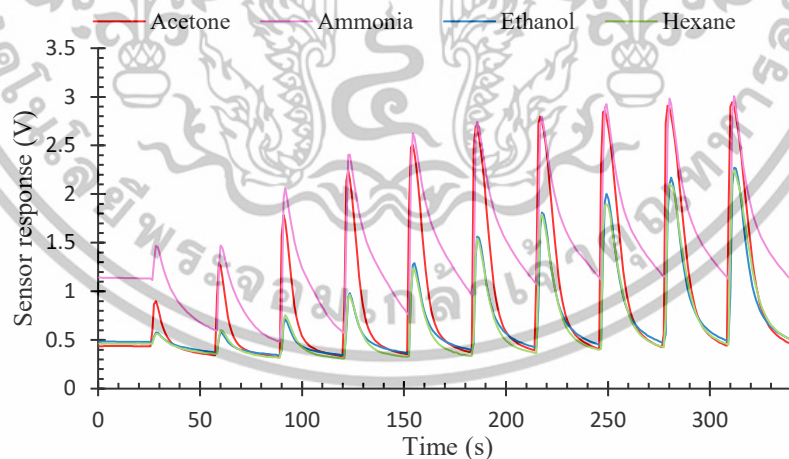
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แสดงให้เห็นว่าผลตอบสนองต่อกลิ้นที่มีอัตราการระเหยปานกลางที่ได้จากเซ็นเซอร์ TGS2600 มีความชันในช่วงขาขึ้นใกล้เคียงกัน แต่มีแอมพลิจูดและความชันในช่วงขาลงต่างกัน โดยตอบสนองต่อกลิ้นของเฮกเซนน้อยกว่ากลิ่นอื่น ๆ



รูปที่ 4.3 ผลตอบสนองของเซ็นเซอร์ TGS2602 ต่อกลิ้นที่มีอัตราการระเหยปานกลาง

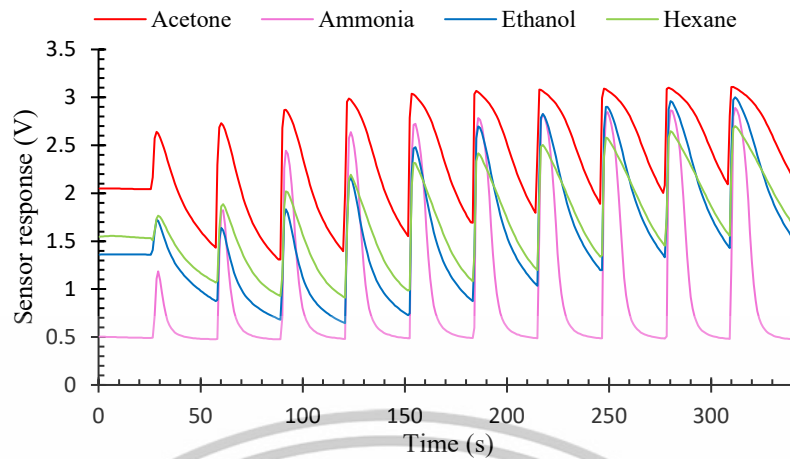
รูปที่ 4.3 แสดงให้เห็นว่าผลตอบสนองต่อกลิ้นที่มีอัตราการระเหยปานกลางที่ได้จากเซ็นเซอร์ TGS2602 มีความชันในช่วงขาขึ้น แอมพลิจูดและความชันในช่วงขาลงต่างกัน โดยตอบสนองต่อกลิ้นของแอมโมเนียมากที่สุด และตอบสนองต่อกลิ้นของอะซิโตนน้อยมาก



รูปที่ 4.4 ผลตอบสนองของเซ็นเซอร์ TGS2611 ต่อกลิ้นที่มีอัตราการระเหยปานกลาง

รูปที่ 4.4 แสดงให้เห็นว่าผลตอบสนองต่อกลิ้นที่มีอัตราการระเหยปานกลางที่ได้จากเซ็นเซอร์ TGS2611 มีความชันในช่วงขาขึ้นใกล้เคียงกัน มีแอมพลิจูดของอะซิโตนและแอมโมเนียกับเอทานอลและเฮกเซนใกล้เคียงกัน นอกเหนือจากแอมโมเนีย สารอื่น ๆ มีความชันในช่วงขาลงใกล้เคียงกัน โดยในภาพรวมแทบจะแยกความแตกต่างของผลตอบสนองต่อเอทานอลและเฮกเซนไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

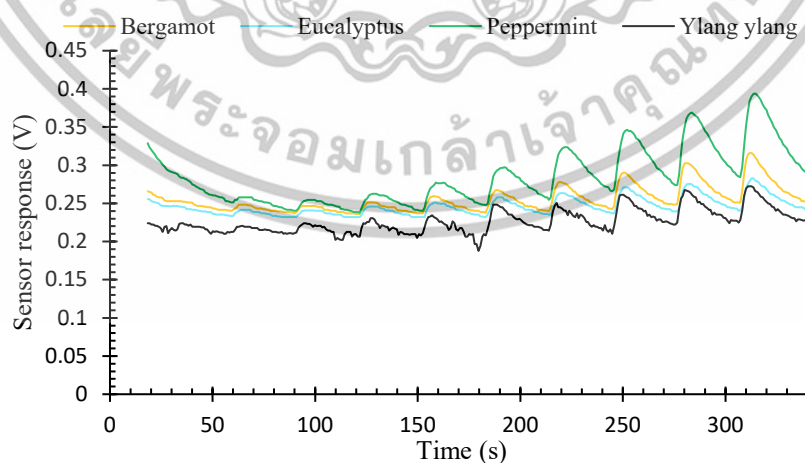


รูปที่ 4.5 ผลตอบสนองของเซ็นเซอร์ TGS2620 ต่อกลิ่นที่มีอัตราการระเหยปานกลาง

รูปที่ 4.5 แสดงให้เห็นว่าผลตอบสนองต่อกลิ่นที่มีอัตราการระเหยปานกลางที่ได้จากเซ็นเซอร์ TGS2620 มีความชันในช่วงขาขึ้นใกล้เคียงกัน มีแอมพลิจูดต่างกันเล็กน้อย เอทานอลและเฮกเซน มีความชันในช่วงขาลงใกล้เคียงกัน

4.1.2 การทดลองวัดกลิ่นที่มีอัตราการระเหยต่ำ

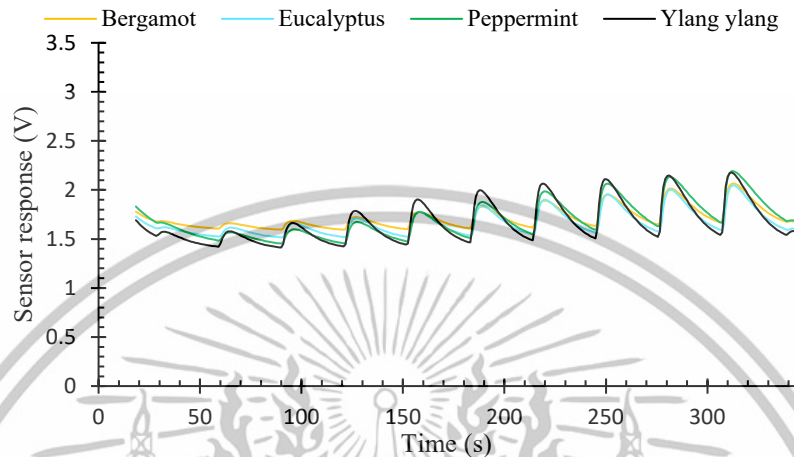
ในการทดลองวัดกลิ่นที่มีอัตราการระเหยต่ำ เลือกใช้สารมีกลิ่นในการทดลอง คือ น้ำมันหอมระเหยจากมะกรูด (Bergamot), ยูคาลิปตัส (Eucalyptus), สะระแหน่ (Peppermint) และกระดังงา (Ylang ylang) โดยทำการทดลองเช่นเดียวกับที่ระบุในหัวข้อ 4.1.1 ผลตอบสนองจากเซ็นเซอร์ที่ได้รับมีจำนวนทั้งหมด 160 ลูกคลื่น ดังแสดงในรูปที่ 4.6 ถึงรูปที่ 4.9 การทดลองดำเนินการซ้ำเช่นนี้รวมทั้งหมด 10 ครั้ง ทำให้ได้ผลตอบสนองจากเซ็นเซอร์รวมทั้งหมดจำนวน 1,600 ลูกคลื่น



รูปที่ 4.6 ผลตอบสนองของเซ็นเซอร์ TGS2600 ต่อกลิ่นที่มีอัตราการระเหยต่ำ

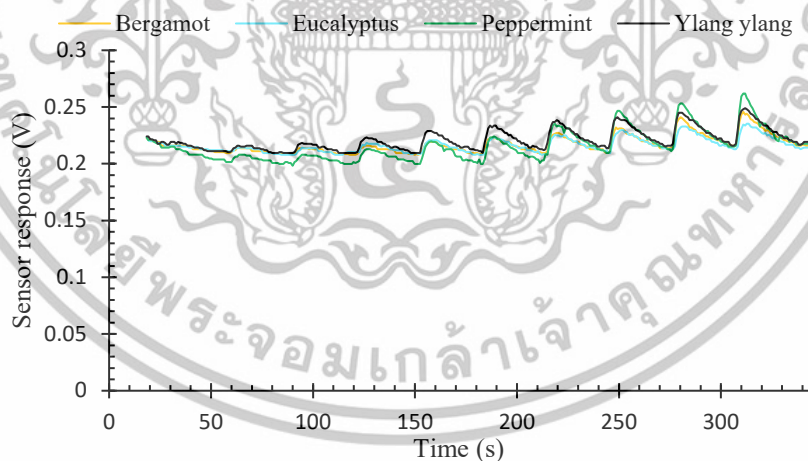
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 แสดงให้เห็นว่าผลตอบสนองต่อกลิ่นที่มีอัตราการระเหยต่ำที่ได้จากเซ็นเซอร์ TGS2600 มีความชันในช่วงขาขึ้นและความชันในช่วงขาลงใกล้เคียงกัน แต่มีแอมพลิจูดต่างกันเล็กน้อย โดยให้ผลตอบสนองต่อสภาวะแห้งชัดเจนที่สุด



รูปที่ 4.7 ผลตอบสนองของเซ็นเซอร์ TGS2602 ต่อกลิ่นที่มีอัตราการระเหยต่ำ

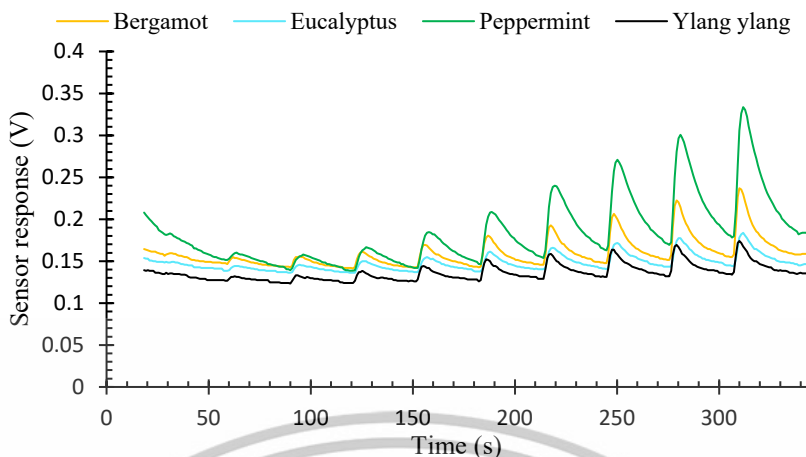
รูปที่ 4.7 แสดงให้เห็นว่าผลตอบสนองต่อกลิ่นที่มีอัตราการระเหยต่ำที่ได้จากเซ็นเซอร์ TGS2602 มีความชันในช่วงขาขึ้นใกล้เคียงกัน แต่มีแอมพลิจูดและความชันในช่วงขาลงต่างกันเล็กน้อย



รูปที่ 4.8 ผลตอบสนองของเซ็นเซอร์ TGS2611 ต่อกลิ่นที่มีอัตราการระเหยต่ำ

รูปที่ 4.8 แสดงให้เห็นว่าผลตอบสนองต่อกลิ่นที่มีอัตราการระเหยต่ำที่ได้จากเซ็นเซอร์ TGS2611 มีความภาพรวมใกล้เคียงกันทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลตอบสนองของเซ็นเซอร์ TGS2620 ต่อกลิ่นที่มีอัตราการระเหยต่ำ

รูปที่ 4.9 แสดงให้เห็นว่าผลตอบสนองต่อกลิ่นที่มีอัตราการระเหยต่ำที่ได้จากเซ็นเซอร์ TGS2620 มีความชันในช่วงขาขึ้นใกล้เคียงกัน มีความชันในช่วงขาลงต่างกันเล็กน้อย แต่มีแอมพลิจูดต่างกัน

4.2 การแยกแยะกลิ่น

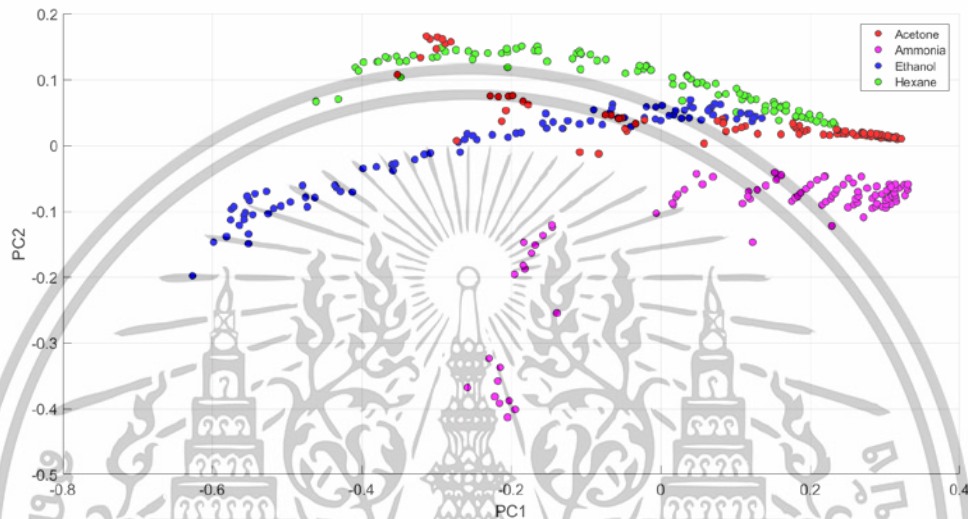
ผลตอบสนองของเซ็นเซอร์ที่ได้จากการทดลองในหัวข้อที่ 4.1 ถูกนำมาใช้ในการแยกแยะกลิ่นในหัวข้อนี้ ด้วยการตั้งคุณสมบัติที่สำคัญจากผลตอบสนองที่ได้มาทำการวิเคราะห์องค์ประกอบหลัก ทั้งนี้เมื่อพิจารณาจากรูปที่ 4.2 ถึง 4.9 จะเลือกใช้คุณสมบัติที่จะนำไปวิเคราะห์องค์ประกอบหลักต่อ คือแอมพลิจูดเชิงสัมบูรณ์ แอมพลิจูดเชิงสัมพัทธ์ ค่าคงที่เวลา และความชันโดยวิธีการตั้งคุณสมบัติเหล่านี้ออกจากข้อมูลที่วัดได้ได้อธิบายในหัวข้อ 3.5.1 ถึง 3.5.4 ทั้งนี้เมื่อตั้งคุณสมบัติดังกล่าวออกมาได้แล้ว จะต้องทำการนอร์มัลไลซ์ข้อมูลเพื่อให้ค่าทั้งหมดอยู่ในสเกลเดียวกัน โดยการนอร์มัลไลซ์นั้นมีวิธีที่เป็นไปได้ 3 วิธี คือนอร์มัลไลซ์ข้อมูลทั้งหมด นอร์มัลไลซ์แยกความเข้มข้นรวมกันทุกกลิ่น และนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่น อย่างไรก็ตาม เนื่องจากในการใช้งานวัดกลิ่นจริงนั้น ความเข้มข้นจะมีค่าเพียงค่าเดียว และผู้วัดจะไม่ทราบล่วงหน้าว่ากลิ่นที่วัดอยู่นั้นเป็นกลิ่นอะไร ด้วยเหตุนี้การนอร์มัลไลซ์จะต้องใช้เป็นแบบแยกความเข้มข้นแยกกลิ่น ข้อมูลที่ได้จากการนอร์มัลไลซ์จะถูกนำไปวิเคราะห์องค์ประกอบหลักซึ่งทำได้ 2 วิธี คือการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม และการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ท้ายที่สุด จะทำการเปรียบเทียบผลลัพธ์ที่ได้จากการแยกแยะกลิ่นแต่ละเงื่อนไขและสรุปรูปแบบการแยกแยะกลิ่นที่สามารถแยกแยะกลิ่นที่ใช้ในการทดลองได้ดีที่สุด

4.2.1 การแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง

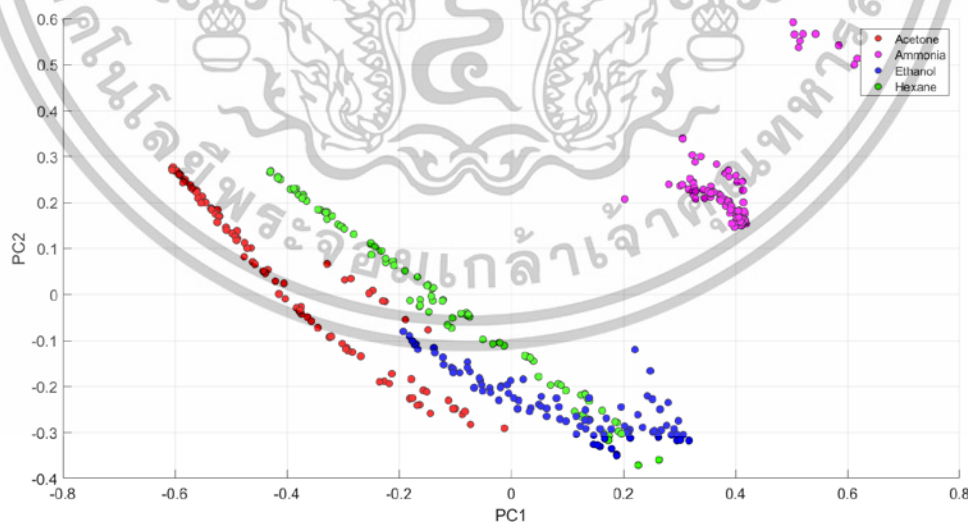
การแยกแยะกลิ่นจากของเหลว 4 ชนิดได้แก่ อะซีโตน แอมโมเนีย เอทานอล และเฮกเซน หลังจากที่ได้ดำเนินการคำนวณหาคุณสมบัติต่าง ๆ ของผลตอบสนองจากเซ็นเซอร์เรียบร้อยแล้ว ข้อมูลทั้งหมดถูกนอร์มัลไลซ์ และคำนวณการวิเคราะห์องค์ประกอบหลักทั้ง 2 วิธีดังนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.1 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม

ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ทำการพล็อตกราฟจากองค์ประกอบหลักที่ 1 และองค์ประกอบหลักที่ 2 จากคุณสมบัติผลตอบสนองของเซ็นเซอร์ได้แก่ แอมพลิจูดเชิงสัมบูรณ์ดังแสดงในรูปที่ 4.10 แอมพลิจูดเชิงสัมพัทธ์ดังแสดงในรูปที่ 4.11 ค่าคงที่เวลาดังแสดงในรูปที่ 4.12 และความชันดังแสดงในรูปที่ 4.13

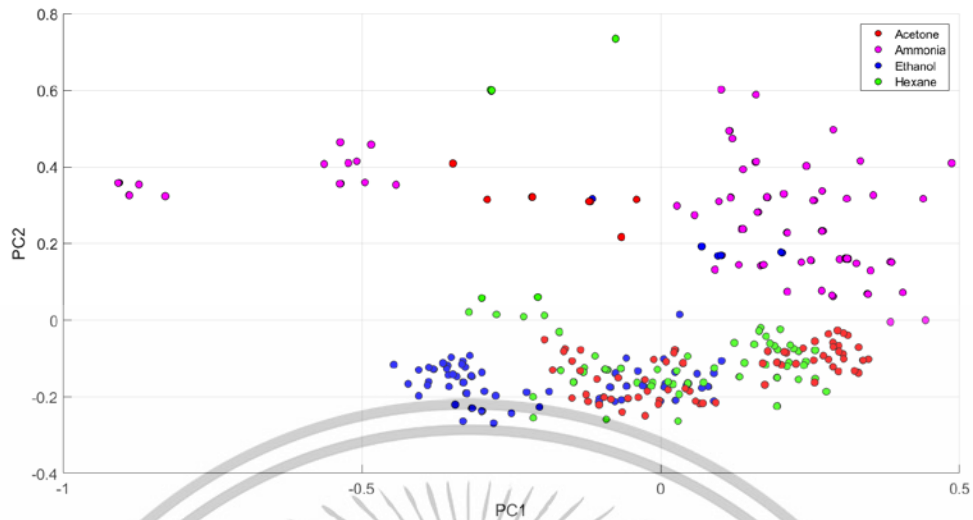


รูปที่ 4.10 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลแอมพลิจูดเชิงสัมบูรณ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

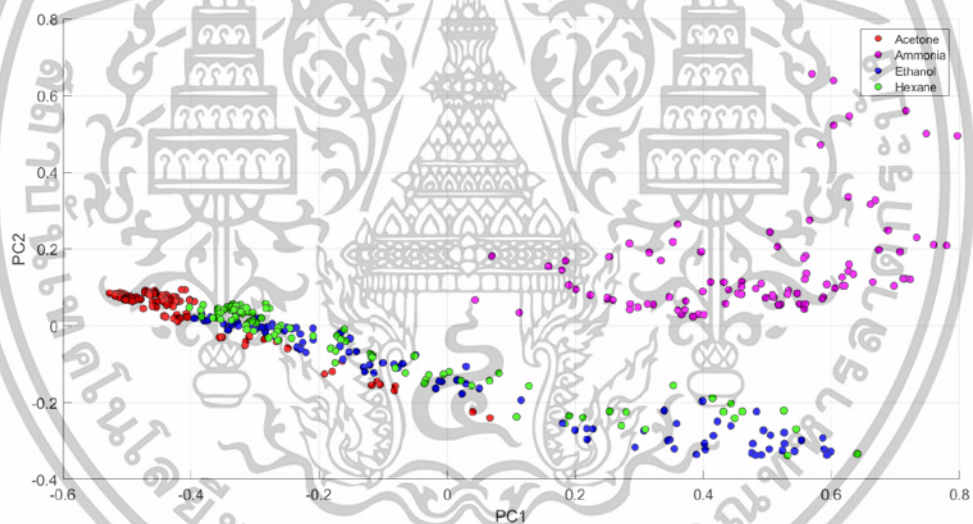


รูปที่ 4.11 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลแอมพลิจูดเชิงสัมพัทธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกัน

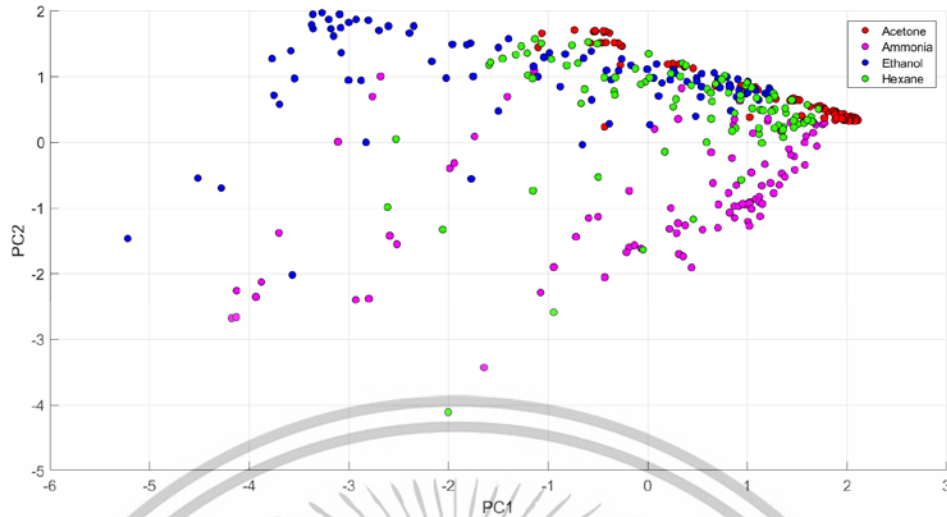


รูปที่ 4.13 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกัน

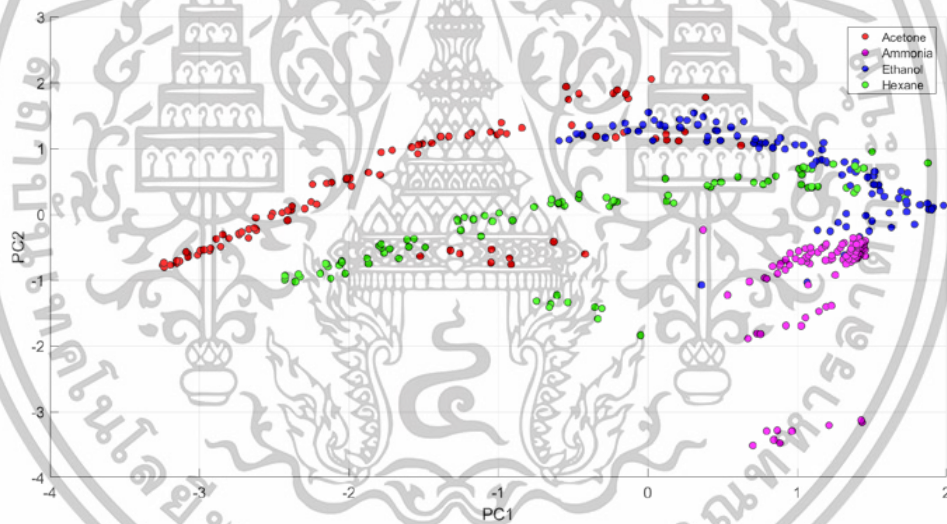
4.2.1.2 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์

ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ทำการพล็อตกราฟจากองค์ประกอบหลักที่ 1 และองค์ประกอบหลักที่ 2 จากคุณสมบัติผลตอบสนองของเซ็นเซอร์ได้แก่ แอมพลิจูดเชิงสัมพันธ์ดังแสดงในรูปที่ 4.14 แอมพลิจูดเชิงสัมพันธ์ดังแสดงในรูปที่ 4.15 ค่าคงที่เวลาดังแสดงในรูปที่ 4.16 และความชื้นแสดงในรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

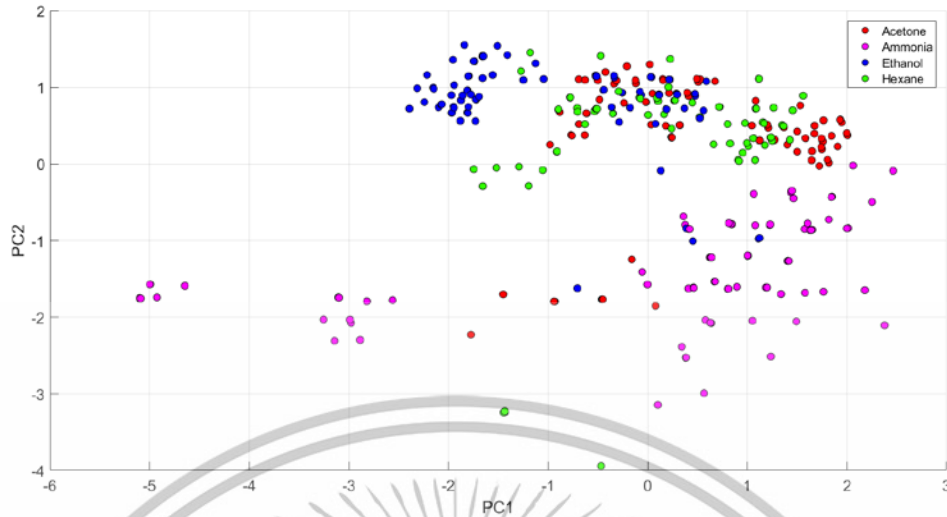


รูปที่ 4.14 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอัมไลซ์แยกความเข้มข้นแยกกลิ่น

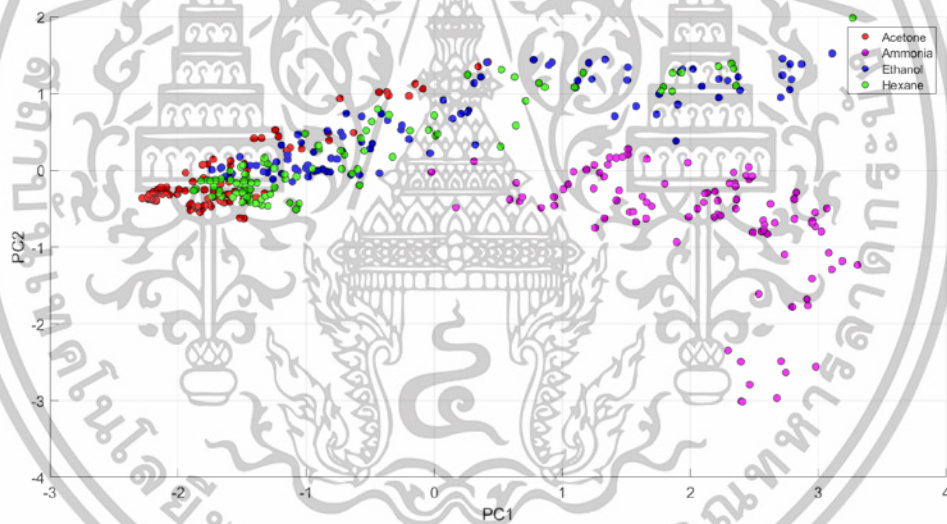


รูปที่ 4.15 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอัมไลซ์แยกความเข้มข้นแยกกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกัน

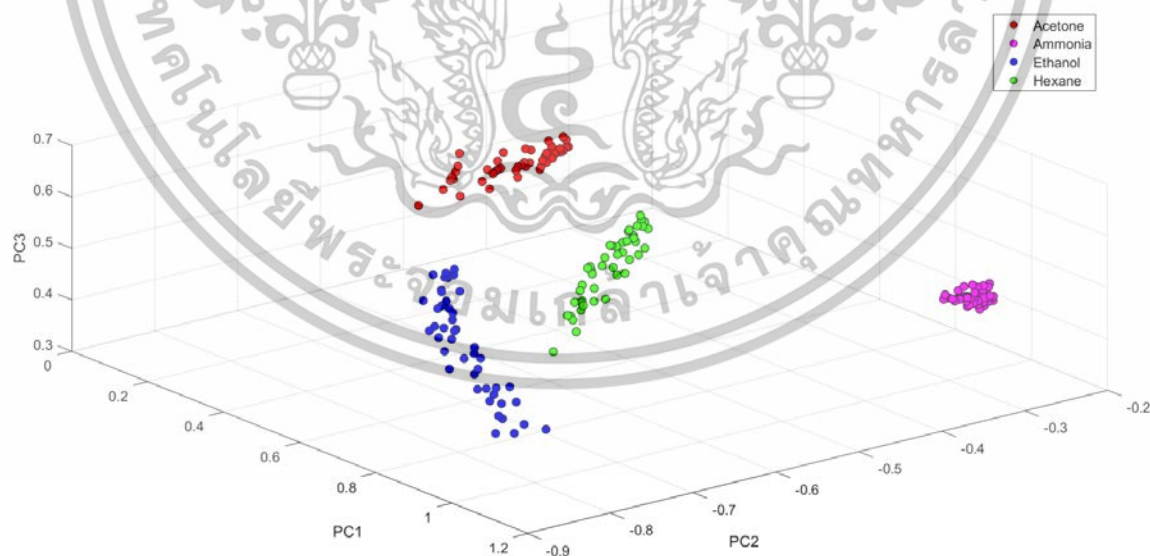


รูปที่ 4.17 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการวิเคราะห์องค์ประกอบหลักทั้งหมดของกลิ่นที่มีอัตราการระเหยปานกลางดังแสดงในรูปที่ 4.10 ถึงรูปที่ 4.17 จากการสังเกตของผู้วิจัยพบว่า ผลลัพธ์การวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่นที่แสดงในรูปที่ 4.11 ให้ผลลัพธ์ที่ดีที่สุด มีข้อมูลที่กระจายตัวออกนอกกลุ่ม ซึ่งเป็นข้อมูลจากการทดลองปล่อยกลิ่นที่ความเข้มข้นต่ำตั้งแต่ 10% ถึง 30% และที่ความเข้มข้น 40% ถึง 50% จะมีบางส่วนที่กระจายตัวออกนอกกลุ่ม จึงเลือกข้อมูลจากความเข้มข้นของกลิ่นที่ 60% ถึง 100% จากการทดลองครั้งที่ 2-10 นำมาพล็อตขึ้นใหม่จากองค์ประกอบหลักที่ 1, 2 และ 3 ดังแสดงในรูปที่ 4.18 และถูกใช้เป็นฐานข้อมูลของกลิ่น นอกจากนี้ค่าน้ำหนักที่คำนวณได้จากการวิเคราะห์องค์ประกอบหลัก จะถูกใช้เป็นค่าน้ำหนักอ้างอิง เพื่อกระบวนการแยกแยะกลิ่นในขั้นตอนถัดไป

การทดลองวัดกลิ่นที่มีอัตราการระเหยปานกลางได้แก่ อะซีโตน แอมโมเนีย เอทานอล และเฮกเซน ถูกดำเนินการอีกจำนวน 9 ครั้ง ด้วยการปล่อยกลิ่นที่มีความเข้มข้น 60% ถึง 100% ผลตอบสนองของเซ็นเซอร์ที่วัดได้ถูกนำไปผ่านกระบวนการนอร์มัลไลเซชัน และคูณด้วยค่าน้ำหนักอ้างอิงเพื่อสร้างองค์ประกอบหลักชุดใหม่แล้วทำการเปรียบเทียบกับฐานข้อมูลกลิ่นในรูปที่ 4.18 จุดทั้งหมดภายในรูปจะถูกคำนวณระยะทางแบบยูคลิด (Euclidean distances) กับทุกจุดขององค์ประกอบหลักชุดใหม่นี้ ผลลัพธ์จากการคำนวณหาค่าเฉลี่ยของระยะทางแบบยูคลิดที่มีค่าน้อยที่สุดจะเป็นตัวระบุว่า องค์ประกอบหลักชุดใหม่เป็นกลิ่นอะไร ผลลัพธ์จากการแยกแยะกลิ่นทั้ง 4 กลิ่น สามารถคำนวณเป็นเปอร์เซ็นต์ความถูกต้องได้ดังแสดงในตารางที่ 4.1



รูปที่ 4.18 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.1 แสดงให้เห็นว่าระบบสามารถแยกแยะกลิ่นที่มีอัตราการระเหยปานกลางได้อย่างถูกต้อง อย่างไรก็ตามในกรณีของเฮกเซน อัตราความถูกต้องค่อนข้างต่ำที่ความเข้มข้นของกลิ่นเท่ากับ 60% โดยระบบระบุผิดพลาดว่าเป็นกลิ่นของเอทานอล

ตารางที่ 4.1 อัตราความถูกต้องในการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง

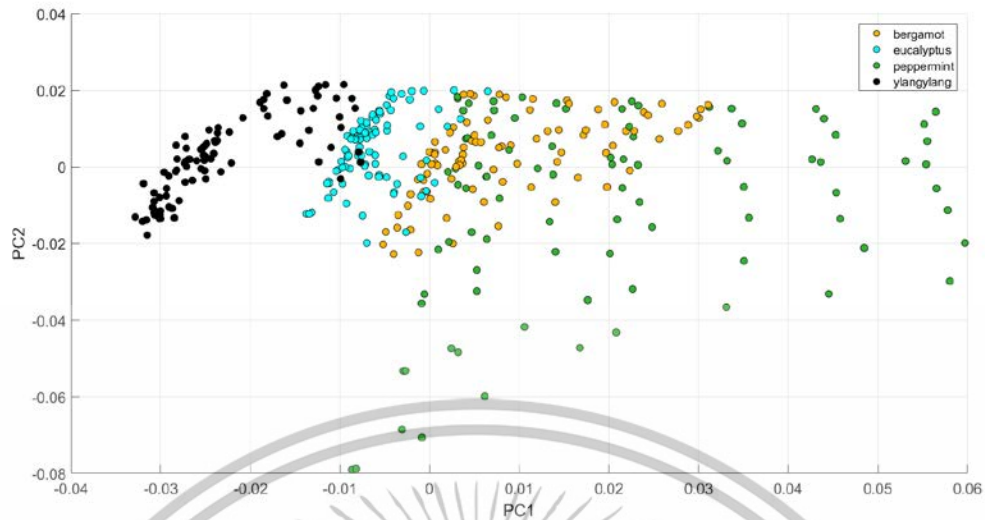
Discrimination rate				
Intensity (%)	Acetone (%)	Ammonia (%)	Ethanol (%)	Hexane (%)
60	100.00	100.00	100.00	11.11
70	100.00	100.00	100.00	88.89
80	100.00	100.00	100.00	100.00
90	100.00	100.00	100.00	100.00
100	100.00	100.00	100.00	100.00
Average	100.00	100.00	100.00	80.00

4.2.2 การแยกแยะกลิ่นที่มีอัตราการระเหยต่ำ

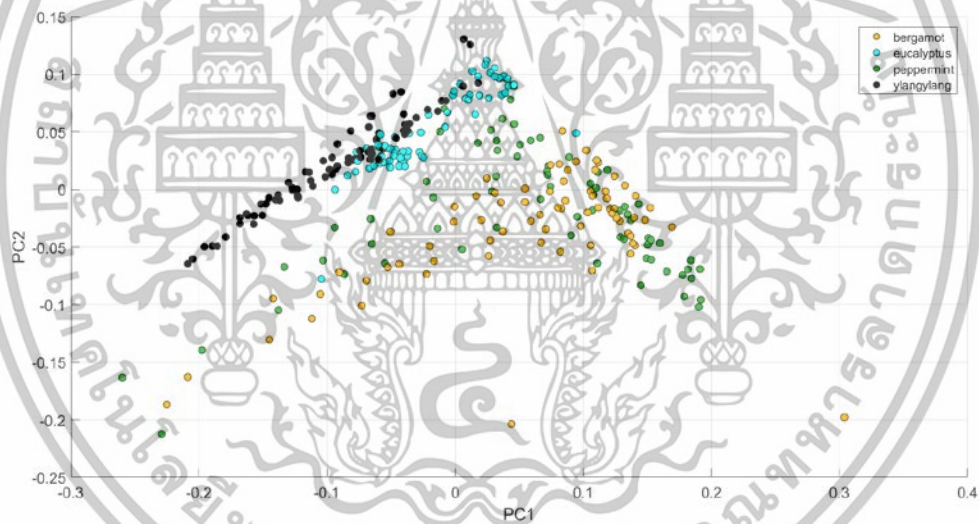
การแยกแยะกลิ่นจากของเหลว 4 ชนิดได้แก่ มะกรูด ยูคาลิปตัส สะระแหน่และกระดังงา หลังจากที่ได้ดำเนินการคำนวณหาคุณสมบัติต่าง ๆ ของผลตอบสนองจากเซ็นเซอร์เรียบร้อยแล้ว ข้อมูลทั้งหมดจะถูกนอร์มัลไลเซชัน และคำนวณการวิเคราะห์องค์ประกอบหลักทั้ง 2 วิธีดังนี้

4.2.2.1 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม

ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วม ทำการพล็อตกราฟจากองค์ประกอบหลักที่ 1 และองค์ประกอบหลักที่ 2 จากคุณสมบัติผลตอบสนองของเซ็นเซอร์ได้แก่ แอมพลิจูดเชิงสัมบูรณ์ดังแสดงในรูปที่ 4.19 แอมพลิจูดเชิงสัมพัทธ์ดังแสดงในรูปที่ 4.20 ค่าคงที่เวลาดังแสดงในรูปที่ 4.21 และความชื้นดังแสดงในรูปที่ 4.22

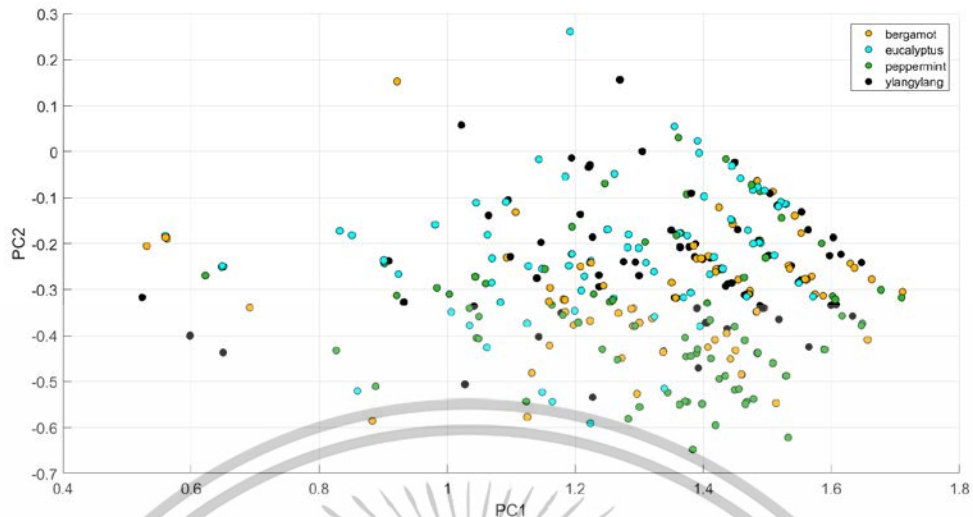


รูปที่ 4.19 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น



รูปที่ 4.20 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

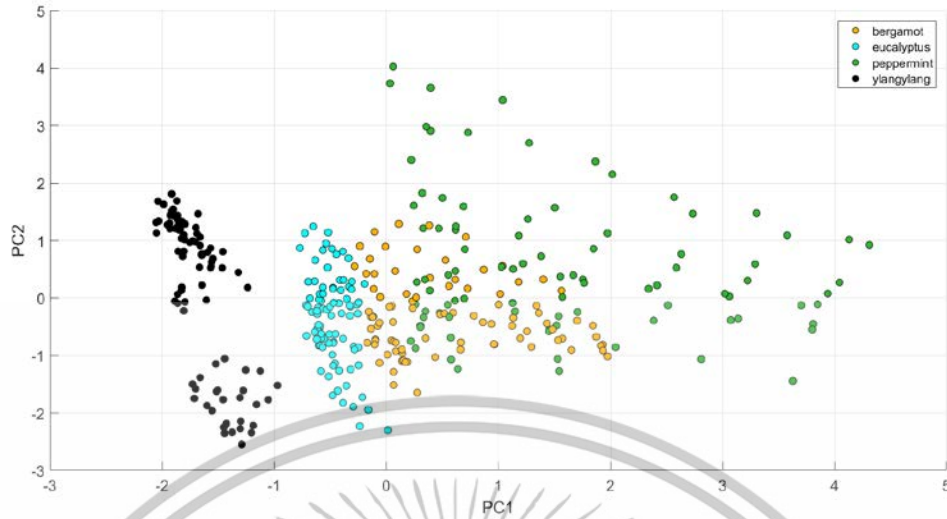


รูปที่ 4.22 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์ความแปรปรวนร่วมของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

4.2.2.2 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์

ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์ ทำการพล็อตกราฟจากองค์ประกอบหลักที่ 1 และองค์ประกอบหลักที่ 2 จากคุณสมบัติผลตอบสนองของเซ็นเซอร์ได้แก่อัมพลิจูดเชิงสัมพันธ์ดังแสดงในรูปที่ 4.23 อัมพลิจูดเชิงสัมพันธ์ดังแสดงในรูปที่ 4.24 ค่าคงที่เวลาดังแสดงในรูปที่ 4.25 และความชื้นดังแสดงในรูปที่ 4.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

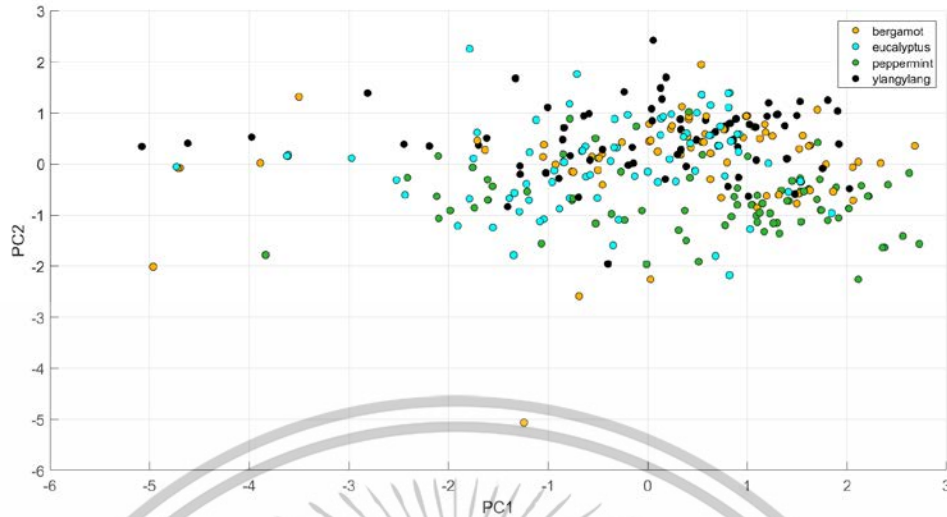


รูปที่ 4.23 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่น



รูปที่ 4.24 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลแอมพลิจูดเชิงสัมพันธ์ โดยนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลค่าคงที่เวลา โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น



รูปที่ 4.26 ผลการวิเคราะห์องค์ประกอบหลักโดยใช้เมทริกซ์สหสัมพันธ์
ของข้อมูลความชื้น โดยนอมัลไลซ์แยกความเข้มข้นแยกกลิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการวิเคราะห์องค์ประกอบหลักทั้งหมดของกลิ่นที่มีอัตราการระเหยต่ำ ดังแสดงในรูปที่ 4.19 ถึงรูปที่ 4.26 จากการสังเกตของผู้วิจัยพบว่า ผลลัพธ์การวิเคราะห์องค์ประกอบหลัก โดยใช้เมทริกซ์สหสัมพันธ์ของข้อมูลแอมพลิจูดเชิงสัมบูรณ์ โดยนอร์มัลไลซ์แยกความเข้มข้นแยกกลิ่นที่แสดงในรูปที่ 4.23 ให้ผลลัพธ์ที่ดีที่สุด มีข้อมูลที่กระจายตัวออกนอกกลุ่ม ซึ่งเป็นข้อมูลจากการทดลองปล่อยกลิ่นที่ความเข้มข้นตั้งแต่ 10% ถึง 60% จึงเลือกข้อมูลความเข้มข้นของกลิ่นที่ 70% ถึง 100% จากการทดลองครั้งที่ 2-10 นำมาพล็อตขึ้นใหม่จากองค์ประกอบหลักที่ 1, 2 และ 3 ดังแสดงในรูปที่ 4.27 และถูกใช้เป็นฐานข้อมูลของกลิ่น นอกจากนี้ค่าน้ำหนักที่คำนวณได้จากการวิเคราะห์องค์ประกอบหลักจะถูกใช้เป็นค่าน้ำหนักอ้างอิงเพื่อกระบวนการแยกแยะกลิ่นในขั้นตอนถัดไป



รูปที่ 4.27 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยต่ำ

การทดลองวัดกลิ่นที่มีอัตราการระเหยต่ำได้แก่ มะกรูด ยูคาลิปตัส สะระแหน่ และกระดังงา ถูกดำเนินการอีกจำนวน 9 ครั้ง ด้วยการปล่อยกลิ่นที่มีความเข้มข้น 70% ถึง 100% ผลตอบสนองของเซ็นเซอร์ที่วัดได้ถูกนำไปผ่านกระบวนการนอร์มัลไลเซชัน และคูณด้วยค่าน้ำหนักอ้างอิงเพื่อสร้างองค์ประกอบหลักชุดใหม่ แล้วทำการเปรียบเทียบกับฐานข้อมูลกลิ่นในรูปที่ 4.27 จุดทั้งหมดภายในรูปจะถูกคำนวณระยะทางแบบยุคลิดกับทุกจุดขององค์ประกอบหลักชุดใหม่ ผลลัพธ์จากการคำนวณหาค่าเฉลี่ยของระยะทางแบบยุคลิดที่มีค่าน้อยที่สุดจะเป็นตัวระบุว่าองค์ประกอบหลักชุดใหม่เป็นกลิ่นอะไร ผลลัพธ์จากการแยกแยะกลิ่นทั้ง 4 กลิ่นสามารถคำนวณเป็นเปอร์เซ็นต์ความถูกต้องได้ดังแสดงในตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 อัตราความถูกต้องในการแยกแยะกลิ่นที่มีอัตรากระเหยต่ำ

Discrimination rate				
Intensity (%)	Bergamot (%)	Eucalyptus (%)	Peppermint (%)	Ylang ylang (%)
70	100	0	22.22	11.11
80	100	100	55.56	22.22
90	100	100	44.44	44.44
100	100	100	55.56	55.56
Average	100.00	75.00	44.44	33.33

จากตารางที่ 4.2 แสดงให้เห็นว่าระบบสามารถแยกแยะกลิ่นของมะกรูดได้อย่างถูกต้อง อย่างไรก็ตามในกรณีของยูคาลิปตัส อัตราความถูกต้องเท่ากับ 0% ที่ความเข้มข้นของกลิ่นเท่ากับ 70% โดยระบบระบุผิดพลาดว่าเป็นกลิ่นของมะกรูด ในกรณีของสะระแหน่ความผิดพลาดทั้งหมดเกิดจากระบบระบุว่าเป็นกลิ่นของมะกรูดเช่นกัน และกรณีของกระดังงาความผิดพลาดทั้งหมดเกิดจากระบบระบุว่าเป็นกลิ่นของยูคาลิปตัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลาง

จากผลการวิเคราะห์องค์ประกอบหลักของกลิ่นที่มีอัตราการระเหยปานกลาง ทำให้สามารถสร้างกราฟจากองค์ประกอบหลักที่ 1, 2 และ 3 ของความเข้มข้นกลิ่นตั้งแต่ 60% ถึง 100% ซึ่งข้อมูลของแต่ละกลิ่นสามารถแยกออกจากกันอย่างชัดเจน ข้อมูลกลิ่นชุดใหม่ที่ความเข้มข้นเดียวกันถูกนำมาคำนวณหาองค์ประกอบหลัก และเปรียบเทียบระยะทางแบบยูคลิดทำให้ได้ผลการแยกแยะกลิ่นที่มีอัตราการระเหยปานกลางได้แก่ อะซีโตน แอมโมเนีย เอทานอล และเฮกเซน มีอัตราความถูกต้องเฉลี่ยเท่ากับ 100%, 100%, 100% และ 80% ตามลำดับ

5.2 ผลการแยกแยะกลิ่นที่มีอัตราการระเหยต่ำ

จากผลการวิเคราะห์องค์ประกอบหลักของกลิ่นที่มีอัตราการระเหยต่ำ ทำให้สามารถสร้างกราฟจากองค์ประกอบหลักที่ 1, 2 และ 3 ของความเข้มข้นกลิ่นตั้งแต่ 70% ถึง 100% ซึ่งข้อมูลของแต่ละกลิ่นสามารถแยกออกจากกันอย่างชัดเจน ข้อมูลกลิ่นชุดใหม่ที่ความเข้มข้นเดียวกันถูกนำมาคำนวณหาองค์ประกอบหลัก และเปรียบเทียบระยะทางแบบยูคลิดทำให้ได้ผลการแยกแยะกลิ่นที่มีอัตราการระเหยต่ำได้แก่ มะกรูด ยูคาลิปตัส สระระแห่น และกระดังงา มีอัตราความถูกต้องเฉลี่ยเท่ากับ 100%, 75%, 44.44% และ 33.33% ตามลำดับ

5.3 ข้อเสนอแนะ

อัตราความถูกต้องของการแยกแยะกลิ่นอาจสามารถปรับปรุงได้ด้วยการเพิ่มจำนวนเซ็นเซอร์กลิ่นที่มีความสามารถในการรับรู้กลิ่นต่าง ๆ แตกต่างกัน

เอกสารอ้างอิง

- [1] ภาณุพงศ์ อังคสุวรรณ, รวมรัตน์ ได้ผลชัยญา "การศึกษาผลตอบสนองของเซ็นเซอร์ชนิดเมทัลออกไซด์" วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2559.
- [2] John E. Amooore, James W. Johnston. Jr., Martin Rubin. "The Stereochemical Theory of Odor" **Scientific American**, vol. 210, 1964, pp. 42-49.
- [3] Swatantra K.S. Kushwaha, Ravi Kumar Keshari, A.K. Rai. "Advances in nasal transmucosal drug delivery" **Journal of Applied Pharmaceutical Science**, vol. 1, 2011, pp. 21-28.
- [4] T. Pogfay, N. Watthanawisuth, W. Pimpao, A. Wisitsoraat, S. Mongpraneet, T. Lomas, M. Sangworasil, A. Tuantranont. "Development of Wireless Electronic Nose for Environment Quality Classification" **ECTI 2010**, Chiang Mai, Thailand, May 2010.
- [5] Nyayu Latifah Husni, Ade Silvia Handayani, Siti Nurmaini, Irsyadi Yani. "Odor Classification Using Support Vector Machine" **ICECOS 2017**, pp. 71-76, Palembang, Indonesia, August 2017.
- [6] Fumikazu Yoshino, Takamichi Nakamoto. "Odor Recognition System Using Embedded Learning Vector Quantization Circuit" **Sensors and Materials**, vol. 26, 2014, pp. 137-147.
- [7] Mahdi Ghasemi-Varnamkhasti, Seyed Saeid Mohtasebi, Maryam Siadat. "Biomimetic-based odor and taste sensing systems to food quality and safety characterization: An overview on basic principles and recent achievements" **Journal of Food Engineering**, vol. 100, 2010, pp. 377-387.
- [8] T. Nakamoto, A. Fukuda, T. Moriizumi. "Perfume and flavor identification by odor sensing system using quartz-resonator sensor array and neuralnetwork pattern recognition" **Solid-State Sensors and Actuators 1991**, pp. 355-358, San Francisco, CA, USA, August 2002.
- [9] Andrea Branca, Paola Simonian, Marcela Ferrante, Elena Novas, R. Mart'in Negri. "Electronic nose based discrimination of a perfumery compound in a fragrance" **Sensors and Actuators B: Chemical**, vol. 92, 2003, pp. 222-227.
- [10] Alphus D. Wilson. "Diverse Applications of Electronic-Nose Technologies in Agriculture and Forestry" **Sensors (Basel)**, vol. 13, 2013, pp. 2295-2348.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [11] Hyung GiByun, Krishna C. Persaud, Soad Mohialdin Khaffaf, Philip J. Hobbs, Tom H. Misselbrook. "Application of unsupervised clustering methods to the assessment of malodour in agriculture using an array of conducting polymer odour sensors" **Computers and Electronics in Agriculture**, vol. 17, 1997, pp. 233-247.
- [12] Treenet Thepudom, Thara Seesaard, Wathang Donkrajang, Teerakiat Kerdcharoen. "Healthcare shoe system for gait monitoring and foot odor detections" **GCCE 2013**, pp. 81-82, Tokyo, Japan, October 2013.
- [13] Roberto F. Machado, Daniel Laskowski, Olivia Deffenderfer, Timothy Burch, Shuo Zheng, Peter J. Mazzone, Tarek Mekhail, Constance Jennings, James K. Stoller, Jacqueline Pyle, Jennifer Duncan, Raed A. Dweik, Serpil C. Erzurum. "Detection of Lung Cancer by Sensor Array Analyses of Exhaled Breath" **AMERICAN JOURNAL OF RESPIRATORY AND CRITICAL CARE MEDICINE**, vol. 171, 2005, pp. 1286-1291.
- [14] Natchanon Ratchapakorn, Yossiri Ariyakul. "Development of a Low-cost Explosive Vapor Detector Using Metal Oxide Gas Sensors" **ICSEC 2017**, pp. 121-124, Bangkok, Thailand, November 2017.
- [15] Christelle Barthet, Pierre Montméat, Nathalie Eloy, Philippe Prené. "Detection of explosives vapours using a multi-quartz crystal microbalance system" **Procedia Engineering**, vol. 5, 2010, pp. 472-475.
- [16] Jung Y. Kim, Sung W. Kang, Tae Z. Shin, Myung K. Yang, Kyu S. Lee. "Design of a Smart Gas Sensor System for Room Air-Cleaner of Automobile -Thick - Film Metal Oxide Semiconductor Gas Sensor" **2006 International Forum on Strategic Technology**, pp. 72-75, Ulsan, South Korea, February 2007.
- [17] FIGARO Engineering, "TGS 2600 - for the detection of Air Contaminants", TGS2600 datasheet
- [18] R. Andrew Russell, "Odour detection by mobile robots", **World scientific in robotics and intelligent systems**, vol. 22, 1998.
- [19] JRC, "Quartz crystal oscillator", NJU6321 datasheet
- [20] Claudia Di Nucci, Ada Fort, Santina Rocchi, Luca Tondi, Valerio Vignoli, Fabio Di Francesco, M. Belén Serrano Santos. "A measurement system for odor classification based on the dynamic response of QCM sensors" **IEEE Transactions on Instrumentation and Measurement**, vol. 52, 2003, pp. 1079-7086.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [21] Joon-Boo Yu, Hyung-Gi Byun, Myung-Suk So, Jeung-Soo Huh. "Analysis of diabetic patient's breath with conducting polymer sensor array" **Sensors and Actuators B: Chemical**, vol. 108, 2005, pp. 305-308.
- [22] Erica S. Forzani, Haiqian Zhang, Larry A. Nagahara, Ishamshah Amlani, Raymond Tsui, and Nongjian Tao "A Conducting Polymer Nanojunction Sensor for Glucose Detection" **NANO LETTERS**, vol. 4, 2004, pp. 1785-1788.
- [23] Rita Stella, Joseph N. Barisci, Giorgio Serra, Gordon G. Wallace, Danilo De Rossi. "Characterisation of olive oil by an electronic nose based on conducting polymer sensors" **Sensors and Actuators B: Chemical**, vol. 63, 2000, pp.1-9.
- [24] S. Oshita, K. Shima, T. Haruta, Y. Seo, Y. Kawagoe, S. Nakayama, H. Takahara. "Discrimination of odors emanating from 'La France' pear by semi-conducting polymer" **Computers and Electronics in Agriculture**, vol. 26, 2000, pp. 209-216.
- [25] Douglas M. Light, Eric B. Jang, Joseph C. Dickens. "Electroantennogram responses of the mediterranean fruit fly, *Ceratitis capitata*, to a spectrum of plant volatiles" **Journal of Chemical Ecology**, vol. 14, 1988, pp. 159-180.
- [26] D. Schneider, Uta Seibt. "Sex Pheromone of the Queen Butterfly: Electroantennogram Responses" **Science**, vol. 164, 1969, pp. 1173-1174.
- [27] Bettina Malnic, Junzo Hirono, Takaaki Sato, Linda B. Buck. "Combinatorial Receptor Codes for Odors" **Cell**, vol. 96, 1999, pp. 713-723.
- [28] Kiyomi Sakakibara, Kazumi Kaitani, Chie Hamada, Shigeyuki Sato, Miho Matsuo. "Analysis of odor in car cabin" **JSAE Review**, vol. 20, 1999, pp. 237-241.
- [29] Paul M. Wise, Mats J. Olsson, William S. Cain. "Quantification of Odor Quality" **Chemical Senses**, vol. 25, 2000, pp. 429-443.
- [30] Jennifer Beshel, Yi Zhong. "Graded Encoding of Food Odor Value in the *Drosophila* Brain" **Journal of Neuroscience**, vol. 33, 2013, pp. 15693-15704.
- [31] Susan S. Schiffman, Bahram G. Kermani, H.Troy Nagle. "Analysis of Medication Off-odors Using an Electronic Nose" **Chemical Senses**, vol. 22, 1997, pp. 119-128.
- [32] James R. McGill, Bruce R. Kowalski. "Intrinsic Dimensionality of Smell" **ANALYTICAL CHEMISTRY**, vol. 49, 1977, pp. 596-602.
- [33] Chatchawal Wongchoosuk, Mario Lutz, Teerakiat Kerdcharoen. "Detection and Classification of Human Body Odor Using an Electronic Nose" **Sensors**, vol. 9, 2009, pp. 7234-7249.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [34] Kohei Jobu, Changhai Sun, Saburo Yoshioka, Junko Yokota, Masahide Onogawa, Chiaki Kawada, Keiji Inoue, Taro Shuin, Toshiaki Sendo, Mitsuhiko Miyamura. "Metabolomics Study on the Biochemical Profiles of Odor Elements in Urine of Human with Bladder Cancer" **Biological and Pharmaceutical Bulletin**, vol. 35, 2012, pp. 639–642.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Diagnostics;
using System.IO;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

```

```

namespace W1_Odor_Sensing_System_Plot
{
    public partial class Form1 : Form
    {
        int uartDelay = 40;
        int savedCount = 1;
        int processEnable = 0;
        string RxString;
        string RxStringAll;

        double timeCount;
        DateTime startTime;
        DateTime lastSave;
        bool createChartDone = false;

        byte[] commandForCalibrate = new Byte[] { 0xFB }; //for nios
        byte[] commandForMeasure = new Byte[] { 0x00 }; //for arm

        int rxDataLineCount;
        int rxCount;

        string rxTimeS;
        string rxDataBuff;
        List<double> rxTimeList = new List<double>();

        bool prepareData;
        public static string labProgressText;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmbBaudRate.Items.Add("9600");
cmbBaudRate.Items.Add("14400");
cmbBaudRate.Items.Add("19200");
cmbBaudRate.Items.Add("28800");
cmbBaudRate.Items.Add("38400");
cmbBaudRate.Items.Add("57600");
cmbBaudRate.Items.Add("76800");
cmbBaudRate.Items.Add("115200");
cmbBaudRate.Items.Add("230400");
cmbBaudRate.Items.Add("250000");
cmbBaudRate.SelectedIndex = 7;

btnStartRX.Enabled = false;
btnStopRX.Enabled = false;
btnReset.Enabled = false;
btnSaveDataAndGraph.Enabled = false;

prepareData = false;

////////// load setting data //////////
try
{
    string[] settingData = File.ReadAllLines("setting.ini");
    globalVariable.autoSaveEnable = Convert.ToInt16(settingData[0]);
    globalVariable.savedTime = Convert.ToInt16(settingData[1]);
    globalVariable.saveDirectory = settingData[2];
    for(int i=0; i<8; i++)
    {
        globalVariable.referecne[i] = Convert.ToInt16(settingData[i + 3]);
    }

    for (int i = 0; i < 8; i++)
    {
        globalVariable.divider[i] = Convert.ToInt16(settingData[i + 11]);
    }

    globalVariable.systemData = "" + globalVariable.referecne[0] + "*" +
    globalVariable.referecne[1] + "*" + globalVariable.referecne[2] + "*" +
    globalVariable.referecne[3] + "*" + globalVariable.referecne[4] + "*" +
    globalVariable.referecne[5] + "*" + globalVariable.referecne[6] + "*" +
    globalVariable.referecne[7] + "*" + globalVariable.divider[0] + "*" +
    globalVariable.divider[1] + "*" + globalVariable.divider[2] + "*" +
    globalVariable.divider[3] + "*" + globalVariable.divider[4] + "*" +
    globalVariable.divider[5] + "*" + globalVariable.divider[6] + "*" +
    globalVariable.divider[7] + "*";
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    catch (Exception) { }
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    var result = MessageBox.Show("Do you want to exit?", "Confirmation",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);

    e.Cancel = (result == DialogResult.No);

    if (result == DialogResult.Yes)
    {
        if (serialPort1.IsOpen) serialPort1.Close();
    }
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (serialPort1.IsOpen) serialPort1.Close();
}

int intlen = 0;
private void timer1_Tick(object sender, EventArgs e)
{
    if (startRX) processData2();

    // check setting status
    check_status();

    // check conected com port
    string[] ports = SerialPort.GetPortNames();
    if (intlen != ports.Length)
    {
        lsbComPort.Items.Clear();
        intlen = ports.Length;
        for (int j = 0; j < intlen; j++)
        {
            lsbComPort.Items.Add(ports[j]);
            lsbComPort.SelectedIndex = 0;
        }
    }
}

// resize tool
chart1.Width = this.Width - 294;
chart1.Height = this.Height - 52;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

txbData.Height = this.Height - 325;

// auto save data
if (globalVariable.autoSaveEnable == 1 && startRX == true)
{
    if ((Math.Round(timeCount, 0) > ((globalVariable.savedTime * savedCount) +
1)))
    {
        saveData();
        savedCount++;
    }
}

// auto stop
if (globalVariable.stopTimeEnable == 1 && startRX == true)
{
    if (Math.Round(timeCount, 0) > ((globalVariable.stopTime) + 1))
    {
        //// stop and save ////
        startRX = false;
        timeCount = 0;
        savedCount = 1;
        saveData();
        btnStartRX.Enabled = true;
        btnStopRX.Enabled = false;
        btnReset.Enabled = true;
        return;
    }
}

private void buttonStart_Click(object sender, EventArgs e)
{
    try
    {
        serialPort1.PortName = lsbComPort.Text;
        serialPort1.BaudRate = Convert.ToInt32(cmbBaudRate.Text);

        serialPort1.Open();
        if (serialPort1.IsOpen)
        {
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            btnStartRX.Enabled = true;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    catch (SystemException ex)
    {
        MessageBox.Show(string.Format("Error Connection: {0}", ex.Message));
    }
}

private void buttonStop_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        serialPort1.Close();
        buttonStart.Enabled = true;
        buttonStop.Enabled = false;

        btnStartRX.Enabled = false;
        btnStopRX.Enabled = false;
        btnReset.Enabled = false;

        startRX = false;
        savedCount = 1;
    }
    else
    {
        buttonStart.Enabled = true;
        buttonStop.Enabled = false;
    }
    backgroundWorker1.CancelAsync();
}

private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    RxString = serialPort1.ReadExisting();
    RxStringAll += RxString;

    if (RxString.Contains("d"))
    {
        rxTimeList.Add(timeCount);
    }
}

private void processData2()
{
    try
    {
        string[] splitLine = RxStringAll.Split(new Char[] { '\n' });

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int lineCount = splitLine.Length;

for (int i = rxDataLineCount; i < lineCount; i++)
{
    splitLine[i] = splitLine[i].Replace("\r", "");
    splitLine[i] = splitLine[i].Replace("\n", "");

    if ((splitLine[i].Contains("d"))) //splitLine[i] != "" &&
    {
        if (!createChartDone) createChart2(splitLine[i]);
        rxDataBuff += rxTimeList[rxCount] + ",";
        addChartData(splitLine[i], rxTimeList[rxCount]);
        rxCount++;
    }
    else
    {
        rxDataLineCount--;
    }
    rxDataLineCount++;
}
}
catch (Exception)
{
}
}

private void createChart2(string rxData)
{
    string seriesName;
    string seriesLabel;

    string[] split = rxData.Split(new Char[] { '*' });
    int dataLength = ((split.Length) - 1);

    chart1.ChartAreas[0].AxisX.Minimum = 0;

    for (int i = 0; i < dataLength; i++)
    {
        try
        {
            seriesName = "Series" + (i + 1);
            seriesLabel = "Sensor" + (i + 1);
            chart1.Series.Add(seriesName);
            chart1.Series[seriesName].ChartType = SeriesChartType.Line;
            chart1.Series[seriesName].Name = seriesLabel;
            chart1.Series[i].BorderWidth = 2;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        rxDataBuff += "0,";
    }
    catch (Exception) { }
}
rxDataBuff += "0,";
rxDataBuff += System.Environment.NewLine;
createChartDone = true;
}

int dataCount2;
private void addChartData(string rxData, double rxTime)
{
    string[] split = rxData.Split(new Char[] { '*' });

    foreach (string s in split)
    {
        //////////////////////////////////////
        if (s.Trim() == "end" || s.Trim() == "d" // && s.Trim().Contains("\r"))//s.Trim() !=
"\n" && s.Trim() != "\r")
        {
            Array.Clear(split, 0, split.Length);
            rxDataBuff += System.Environment.NewLine;
            dataCount2 = 0;
            return;
        }
        else if (s.Trim() != "" && s.Trim() != "end" || s.Trim() != "d")
        {
            String sBuff = s.Replace("\n", "");
            sBuff = sBuff.Replace("\r", "");
            rxDataBuff += (sBuff);
            rxDataBuff += (",");
            if(!prepareData) this.chart1.Series[dataCount2].Points.AddXY(rxTime, sBuff);
            dataCount2++;
        }
    }
}
}
}

```

```

private void DisplayText(object sender, EventArgs e)
{
    if (RxString.Contains("*end") || RxString.Contains("*d"))
    {
        processEnable = 1;
    }
    if (processEnable == 1 && startRX == true)
    {
        txbRX.AppendText(RxString);
        try

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (createChartDone == false)
            createChart();
        processData();
    }
    catch (Exception) { }
}
}

int dataCount;
private void createChart()
{
    string seriesName;
    string seriesLabel;
    string[] split = txbRX.Text.Split(new Char[] { '*' });
    int dataLength = ((split.Length) - 1);
    chart1.ChartAreas[0].AxisX.Minimum = 0;
    if (chart1.Series.Count() > 0) return;
    for (int i=0; i<dataLength; i++)
    {
        try
        {
            seriesName = "Series" + (i+1);
            seriesLabel = "Sensor" + (i+1);
            chart1.Series.Add(seriesName);
            chart1.Series[seriesName].ChartType = SeriesChartType.Line;
            chart1.Series[seriesName].Name = seriesLabel;
            chart1.Series[i].BorderWidth = 2;

            txbData.Text += "0,";
        }
        catch (Exception) { }
    }
    txbData.Text += "0,";
    txbData.AppendText(System.Environment.NewLine);
    createChartDone = true;
}

private void processData()
{
    txbData.AppendText(Math.Round(timeCount, 2) + ",");
    string words = txbRX.Text; //words is RxString
    string[] split = words.Split(new Char[] { '*' });
    foreach (string s in split)
    {
        //////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (s.Trim() == "end" || s.Trim() == "d") // && s.Trim().Contains("\r")//s.Trim() !=
"\n" && s.Trim() != "\r")
        {
            Array.Clear(split, 0, split.Length);
            txbRX.Clear();
            txbData.AppendText(System.Environment.NewLine);
            dataCount = 0;
            //secondCount++;
            return;
        }
        else if (s.Trim() != "" && s.Trim() != "end" || s.Trim() != "d")
        {
            String sBuff = s.Replace("\n", "");
            sBuff = sBuff.Replace("\r", "");
            txbData.AppendText(sBuff);
            txbData.AppendText(",");
            this.chart1.Series[dataCount].Points.AddXY(timeCount, sBuff);
            dataCount++;
        }
    }
}

private void btnSplit_Click(object sender, EventArgs e)
{
    createChart();
    txbData.AppendText(timeCount + ",");

    string words = txbRX.Text; //words is RxString
    string[] split = words.Split(new Char[] { '*' });

    foreach (string s in split)
    {
        //////////////////////////////////////
        if (s.Trim() == "d")
        {
            Array.Clear(split, 0, split.Length);
            txbRX.Clear();
            txbData.AppendText(System.Environment.NewLine);
            dataCount = 0;
            return;
        }
        else if (s.Trim() != "" && s.Trim() != "d")
        {
            txbData.AppendText(s);
            txbData.AppendText(",");
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        this.chart1.Series[dataCount].Points.AddXY(timeCount, s);
        dataCount++;
    }
    //////////////////////////////////////
}
}

int clickCount;
private void btnTestPlot_Click(object sender, EventArgs e)
{
    chart1.ChartAreas[0].AxisX.Minimum = 0;

    chart1.Series.Add("Series1");
    chart1.Series["Series1"].ChartType = SeriesChartType.Line;
    chart1.Series["Series1"].Name = "Sensor 1";

    chart1.Series.Add("Series2");
    chart1.Series["Series2"].ChartType = SeriesChartType.Line;
    chart1.Series["Series2"].Name = "Sensor 2";

    chart1.Series.Add("Series3");
    chart1.Series["Series3"].ChartType = SeriesChartType.Line;
    chart1.Series["Series3"].Name = "Sensor 3";

    chart1.Series.Add("Series4");
    chart1.Series["Series4"].ChartType = SeriesChartType.Line;
    chart1.Series["Series4"].Name = "Sensor 4";

    try
    {
        string[] testData1 = { "1", "2", "3", "4"};
        string[] testData2 = { "3", "5", "7", "9"};
        if (clickCount == 0)
        {
            for (int i = 0; i < testData1.Length; i++)
            {
                this.chart1.Series[i].Points.AddXY(clickCount, testData1[i]);
            }
        }
        else if (clickCount == 1)
        {
            for (int i = 0; i < testData1.Length; i++)
            {
                this.chart1.Series[i].Points.AddXY(clickCount, testData2[i]);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        clickCount++;
    }

    catch (Exception)
    {
        MessageBox.Show("something is wrong");
    }
}

```

```

private void clearData()
{
    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }
}

```

```

int chartSeries = chart1.Series.Count();
label1.Text = chartSeries.ToString();

```

```

for (int i = 0; i < chartSeries; i++)
{
    chart1.Series.RemoveAt(0);
}

```

```

txbData.Clear();
txbRX.Clear();
timeCount = 0;
rxDataLineCount = 0;
rxTimeList.Clear();
RxString = string.Empty;
RxStringAll = string.Empty;
rxTimeS = string.Empty;
rxCount = 0;

```

```

rxDataBuff = string.Empty;
labRecordTime.Text = "Time:";
}

```

```

private void clearChart()
{
    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }
    rxDataBuff = string.Empty;
    txbData.Clear();
    rxDataLineCount = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        rxCount = 0;
    }

    private void clearRxProgress()
    {
        rxDataLineCount = 0;
        rxCount = 0;
    }

    private void btnReset_Click(object sender, EventArgs e)
    {
        DialogResult result = MessageBox.Show("Do you want to clear all data?",
"Confirmation", MessageBoxButtons.YesNo);
        if (result == DialogResult.Yes)
        {
            clearData();
            startRX = false;
            timeCount = 0;
            savedCount = 1;

            btnStartRX.Enabled = true;
            btnStopRX.Enabled = false;
            btnReset.Enabled = true;
            btnSaveDataAndGraph.Enabled = false;
            createChartDone = false;
            txbAllRxData.Text = "";
            backgroundWorker1.CancelAsync();
        }
    }

    private void btnSaveCSV_Click(object sender, EventArgs e)
    {
        string filename = "Result" + DateTime.Now.ToString("yyyyMMdd_HHmms");

        // pass in true to cause writes to be appended
        StreamWriter sw = new StreamWriter("D:\\\" + filename + ".csv", true);
        // use writeline so we get a newline character written
        sw.WriteLine(txbData.Text);
        // Ensure data is written to disk
        sw.Close();

        this.chart1.SaveImage("D:\\\" + filename + ".png", ChartImageFormat.Png);
    }

    bool startRX = false;

    private void btnStartRX_Click(object sender, EventArgs e)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if (!serialPort1.IsOpen) return;
    if (backgroundWorker1.IsBusy != true) // && startRX == true
    {
        // Start the asynchronous operation.
        backgroundWorker1.RunWorkerAsync();
    }

    clearData();

    btnStartRX.Enabled = false;
    btnStopRX.Enabled = true;
    btnReset.Enabled = true;
    btnSaveDataAndGraph.Enabled = true;

    // update start time
    startTime = DateTime.Now;
    startRX = true;
}

private void btnStopRX_Click(object sender, EventArgs e)
{
    if (!serialPort1.IsOpen) return;
    startRX = false;
    timeCount = 0;
    savedCount = 1;
    saveData();
    btnStartRX.Enabled = true;
    btnStopRX.Enabled = false;
    btnReset.Enabled = true;
    createChartDone = false;
    txbAllRxData.Text = "";
    // disable background worker
    // Cancel the asynchronous operation.
    backgroundWorker1.CancelAsync();
}

private void saveData()
{
    try
    {
        string filename = "Result" + DateTime.Now.ToString("yyyyMMdd_HH:mm:ss");
        // pass in true to cause writes to be appended
        StreamWriter sw2 = new StreamWriter(globalVariable.saveDirectory + filename
+ "data.txt", true);
        // use writeline so we get a newline character written
        sw2.WriteLine(RxStringAll);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Ensure data is written to disk
sw2.Close();

if (!Directory.Exists(globalVariable.saveDirectory))
{
    Directory.CreateDirectory(globalVariable.saveDirectory);
}

// pass in true to cause writes to be appended
StreamWriter sw = new StreamWriter(globalVariable.saveDirectory + filename
+ ".csv", true);
// use writeline so we get a newline character written
sw.WriteLine(rxDataBuff);
// Ensure data is written to disk
sw.Close();

this.chart1.SaveImage(globalVariable.saveDirectory + filename + ".png",
ChartImageFormat.Png);
lastSave = DateTime.Now;
lbSaveStatus.Text = "last save: " + lastSave + "\n@" +
globalVariable.saveDirectory;
}
catch (Exception ex)
{
    MessageBox.Show(string.Format("Error Connection: {0}", ex.Message));
}
}

private void btnSaveDataAndGraph_Click(object sender, EventArgs e)
{
    saveData();
}

private void btnSetting_Click(object sender, EventArgs e)
{
    Form2 frm2 = new Form2(serialPort1);
    frm2.ShowDialog();
}

private void btnCalibrate_Click(object sender, EventArgs e)
{
    this.Cursor = Cursors.WaitCursor;
    serialPort1.Write(commandForCalibrate, 0, 1);
    serialPort1.Write("\n\r");
    Thread.Sleep(3000);
    serialPort1.Write("c"); //for de0 soc
    serialPort1.Write("\n\r");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Thread.Sleep(1000);

clearData();
startRX = false;
timeCount = 0;

btnStartRX.Enabled = true;
btnStopRX.Enabled = false;
btnReset.Enabled = true;
btnSaveDataAndGraph.Enabled = false;
this.Cursor = Cursors.Default;
}

private void btnSent0xFD_Click(object sender, EventArgs e)
{
    byte[] commandForMixOut = new Byte[] { 0xFD };
    serialPort1.Write(commandForMixOut, 0, 1);
    Thread.Sleep(uartDelay);
}

private void button1_Click(object sender, EventArgs e)
{
    serialPort1.Write("root\n");
}

private void button2_Click(object sender, EventArgs e)
{
    serialPort1.Write("./fcountsoc\n");
}

private void check_status()
{
    if(globalVariable.autoSaveEnable == 1)
    {
        labAutoSaveStatus.BackColor = System.Drawing.Color.Green;
        labAutoSaveStatus.Text = "Auto save : " + globalVariable.savedTime + "
second";
    }
    else
    {
        labAutoSaveStatus.BackColor = System.Drawing.Color.Red;
        labAutoSaveStatus.Text = "Auto save : " + globalVariable.savedTime + "
second";
    }

    if(globalVariable.stopTimeEnable == 1)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

labAutoStopStatus.BackColor = System.Drawing.Color.Green;
labAutoStopStatus.Text = "Auto stop : " + globalVariable.stopTime + "
second";
    }
    else
    {
        labAutoStopStatus.BackColor = System.Drawing.Color.Red;
        labAutoStopStatus.Text = "Auto stop : " + globalVariable.stopTime + "
second";
    }
}

private void lbSaveStatus_DoubleClick(object sender, EventArgs e)
{
    Process.Start("explorer.exe", @"\" + globalVariable.saveDirectory);
}

private void btnStartbackgroundWorker1_Click(object sender, EventArgs e)
{
    if (backgroundWorker1.IsBusy != true)
    {
        // Start the asynchronous operation.
        backgroundWorker1.RunWorkerAsync();
    }
}

private void backgroundWorker1_DoWork(object sender,
System.ComponentModel.DoWorkEventArgs e)
{
    while (true)
    {
        TimeSpan elapsed = DateTime.Now - startTime;
        timeCount = Math.Round(elapsed.TotalSeconds, 4);
        Thread.Sleep(10);
    }
}

private void backgroundWorker1_RunWorkerCompleted(object sender,
System.ComponentModel.RunWorkerCompletedEventArgs e)
{
}

static string MinifyB(string p)
{
    StringBuilder b = new StringBuilder(p);
    b.Replace(" ", "");
    b.Replace("end", string.Empty);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        b.Replace("d", string.Empty);
        return b.ToString();
    }

    private void btnCancelbtbackgroundWorker1_Click(object sender, EventArgs e)
    {
        if (backgroundWorker1.WorkerSupportsCancellation == true)
        {
            // Cancel the asynchronous operation.
            backgroundWorker1.CancelAsync();
        }
    }

    private void btnLoop_Click(object sender, EventArgs e)
    {
        for(int i = 0; i < 20; i++)
        {
            Thread.Sleep(1000);
        }
    }

    private void btnPlot2test_Click(object sender, EventArgs e)
    {
        processData2();
    }

    private void timer2_Tick(object sender, EventArgs e)
    {
        if (startRX)//startRX
        {
            try
            {
                uint timeIndex = Convert.ToUInt32(rxTimeList.Count());
                labRecordTime.Text = "Time: " + rxTimeList[Convert.ToInt32(timeIndex - 1)]
                + " s";
            }
            catch (Exception) { }
            txbData.Text = rxDataBuff;
            txbData.SelectionStart = txbData.Text.Length;
            txbData.ScrollToCaret();
        }
    }

    private void btnShowRxTimeList_Click(object sender, EventArgs e)
    {
        textBox2.Clear();
        foreach (double time in rxTimeList)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        textBox2.Text += time + System.Environment.NewLine;
    }
}

private void btnShowRxStringList_Click(object sender, EventArgs e)
{
    textBox2.Clear();
    clearChart();
}

private void btnShowRxData_Click(object sender, EventArgs e)
{
    txbAllRxData.Text = RxStringAll;
}

private void btnPrepareData_Click(object sender, EventArgs e)
{
    prepareData = true;
    clearRxProgress();
    processData2();
    prepareData = false;
}

private void btnProgressFrm_Click(object sender, EventArgs e)
{
    labProgressText = "In progress";
    using (progress_frm progressFrm = new progress_frm(Finalprogress))
    {
        progressFrm.ShowDialog(this);
    }
}

void Finalprogress()
{
    prepareData = true;
    clearRxProgress();
    processData2();
    prepareData = false;
}

private void button3_Click(object sender, EventArgs e)
{
    textBox1.Text = rxDataBuff;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

using System;

using System.IO;
using System.IO.Ports;
using System.Windows.Forms;

namespace W1_Odor_Sensing_System_Plot
{
    public partial class Form2 : Form
    {
        SerialPort serialPort1;
        int uartDelay = 40;
        byte[] commandForDeltaF = new Byte[] { 0xFC };
        byte[] stopByte = new Byte[] { 0xFF };
        int[] sensorIndex = new int[] { 0, 0, 0, 0, 0, 0, 0, 0 };

        public Form2(SerialPort port)
        {
            InitializeComponent();
            serialPort1 = port;
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            //////////// load sensor index ////////////
            try
            {
                string[] readSensorIndex = File.ReadAllLines("sensorIndex.ini");
                for(int i=0; i<8; i++)
                {
                    sensorIndex[i] = Convert.ToInt16(readSensorIndex[i]);
                }
            }
            catch (Exception) { }

            //////////// load setting data ////////////
            try
            {
                string[] setData = File.ReadAllLines("setting.ini");
                globalVariable.autoSaveEnable = Convert.ToInt16(setData[0]);
                globalVariable.savedTime = Convert.ToInt16(setData[1]);
                globalVariable.saveDirectory = setData[2];
                txbSaveDirectory.Text = globalVariable.saveDirectory;

                if (globalVariable.autoSaveEnable == 1) chbAutoSave.Checked = true;
                else chbAutoSave.Checked = false;

                if (globalVariable.stopTimeEnable == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        chbStopTimeEnable.Checked = true;
        numericUpDown1.Value = globalVariable.stopTime;
    }
    else chbStopTimeEnable.Checked = false;
}
catch (Exception) { }

cmbAutoSaveTime.Items.Add("1 minuts");
cmbAutoSaveTime.Items.Add("5 minuts");
cmbAutoSaveTime.Items.Add("10 minuts");
cmbAutoSaveTime.Items.Add("30 minuts");
cmbAutoSaveTime.Items.Add("60 minuts");
switch (globalVariable.savedTime)
{
    case 60: cmbAutoSaveTime.SelectedIndex = 0; break;
    case 300: cmbAutoSaveTime.SelectedIndex = 1; break;
    case 600: cmbAutoSaveTime.SelectedIndex = 2; break;
    case 1800: cmbAutoSaveTime.SelectedIndex = 3; break;
    case 3600: cmbAutoSaveTime.SelectedIndex = 4; break;
}
}

private void Form2_FormClosed(object sender, FormClosedEventArgs e)
{
}

private void saveSetting()
{
    ///////////Save listBox1 to .ini file//////////
    using (System.IO.StreamWriter file = new
System.IO.StreamWriter("sensorIndex.ini"))
    {
        foreach (var item in sensorIndex)
        {
            file.WriteLine(item.ToString());
        }
    }

    ///////////Save setting//////////
    using (System.IO.StreamWriter file = new System.IO.StreamWriter("setting.ini"))
    {
        if (chbAutoSave.Checked)
            file.WriteLine("1");
        else
            file.WriteLine("0");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

file.WriteLine(globalVariable.savedTime);

file.WriteLine(globalVariable.saveDirectory);

foreach (var item in globalVariable.referecne)
{
    file.WriteLine(item.ToString());
}
foreach (var item in globalVariable.divider)
{
    file.WriteLine(item.ToString());
}
}
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnOK_Click(object sender, EventArgs e)
{
    try
    {
        switch (cmbAutoSaveTime.SelectedIndex)
        {
            case 0:
                globalVariable.savedTime = 60; //60 seconds
                break;
            case 1:
                globalVariable.savedTime = 300; //300 seconds
                break;
            case 2:
                globalVariable.savedTime = 600; //600 seconds
                break;
            case 3:
                globalVariable.savedTime = 1800; //1800 seconds
                break;
            case 4:
                globalVariable.savedTime = 3600; //3600 seconds
                break;
        }

        if(chbAutoSave.Checked)
        {
            globalVariable.autoSaveEnable = 1;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    globalVariable.autoSaveEnable = 0;
}

if(chbStopTimeEnable.Checked)
{
    globalVariable.stopTimeEnable = 1;
    globalVariable.stopTime = (Int16) numericUpDown1.Value;
}
else
{
    globalVariable.stopTimeEnable = 0;
    globalVariable.stopTime = 0;
}

saveSetting();
this.Close();
}
catch (Exception) { MessageBox.Show("Something Wrong"); };
}

private void btnFolderBrowserDialog_Click(object sender, EventArgs e)
{
    var fbd = new FolderBrowserDialog();
    DialogResult result = fbd.ShowDialog();
    globalVariable.saveDirectory = fbd.SelectedPath + '\\';
    txbSaveDirectory.Text = globalVariable.saveDirectory;
}
}
}

```

```

using System;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace W1_Odor_Sensing_System_Plot
{
    public partial class progress_frm : Form
    {
        public Action Worker { get; set; }
        public progress_frm(Action worker)
        {
            InitializeComponent();

            this.progressBar1.Name = "progressBar1";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.progressBar1.TabIndex = 0;
this.Controls.Add(this.progressBar1);

this.labProgress.Name = "labProgress";
this.progressBar1.TabIndex = 1;
this.Controls.Add(this.labProgress);

if (worker == null)
    throw new ArgumentNullException();
Worker = worker;
}

protected override void OnLoad(EventArgs e)
{
    base.OnLoad(e);
    Task.Factory.StartNew(Worker).ContinueWith(t => { this.Close(); },
TaskScheduler.FromCurrentSynchronizationContext());
}

private void progress_frm_Load(object sender, EventArgs e)
{
    labProgress.Text = Form1.labProgressText;
}

private void progress_frm_Shown(object sender, EventArgs e)
{
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายธิติวุฒิ ทองอยู่
วัน เดือน ปีเกิด	7 พฤศจิกายน 2535
ที่อยู่	141 หมู่ 4 ต.ชุมโค อ.ปะทิว จ.ชุมพร 86160
ประวัติการศึกษา	พ.ศ.2555 ประกาศนียบัตรวิชาชีพชั้นสูง สาขาวิชาคอมพิวเตอร์ธุรกิจ โรงเรียนชุมพรบริหารธุรกิจ พ.ศ.2558 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ (เกียรตินิยมอันดับ1) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์ Tel. 082-8080629 Email. titiwoot.to@gmail.com

ติดต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้