

เครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้ว
ที่มีการส่งสัญญาณด้วยบลูทูธ
BLUETOOTH FINGERTIP PULSE OXIMETER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้ว
ที่มีการส่งสัญญาณด้วยบลูทูธ
BLUETOOTH FINGERTIP PULSE OXIMETER



นางสาวมนัสวี รุ่งแกร 61010860
นางสาวสุภัทสรวิทย์ วิริยธนชิต 61011142

อาจารย์ที่ปรึกษา
รศ.ดร. ปราโมทย์ วาดเขียน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดความอิ่มตัวออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ

BLUETOOTH FINGERTIP PULSE OXIMETER

ผู้จัดทำ

1. นางสาวณัสนีย์ รุ่งแกร 61010860
2. นางสาวสุภัทสรวิญญ์ วิริยธนชิต 61011142

ปรีชาเมธี

อาจารย์ที่ปรึกษา

(รศ.ดร. ปรีชาเมธี วาดเขียน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปริญญานิพนธ์เรื่อง “เครื่องวัดความอืดตัวออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความอนุเคราะห์อย่างดียิ่งจาก รศ.ดร.ปราโมทย์ วาดเขียน ที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้โครงการนี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดทำโครงการ

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษามีค่ายิ่งแก่ผู้จัดทำ

นางสาวมนัสวี รุ่งแกร
นางสาวสุภัทสรุา วิริยธนชิต
ผู้จัดทำ

เครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่มีการส่ง
สัญญาณด้วยบลูทูธ

BLUETOOTH FINGERTIP PULSE OXIMETER

โดย นางสาวมนัสวี รุ่งแกร 61010860

นางสาวสุภัทสรวิญญิต 61011142

อาจารย์ที่ปรึกษา รศ.ดร. ปราโมทย์ วาดเขียน

บทคัดย่อ

สถานการณ์การแพร่ระบาดของโรคติดเชื้อไวรัสโคโรนา 2019 ที่เกิดขึ้นในปัจจุบันส่งผลกระทบต่อการใช้ชีวิตของผู้คนเป็นอย่างมาก ทำให้ผู้คนต่างตื่นตัวในเรื่องของการดูแลสุขภาพ ปฏิญาณนิพนธ์นี้จึงได้ทำการออกแบบและสร้างเครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ ซึ่งจะเป็นอุปกรณ์ที่ช่วยในการตรวจสอบ และเฝ้าระวังอาการเบื้องต้นของโรคต่าง ๆ โดยที่ด้านบนของหัววัดความอิ่มตัวของออกซิเจนในเลือด จะมีการขับแหล่งกำเนิดแสงสีแดงด้วยคลื่นความถี่ 80 เฮิร์ต สำหรับวัดออกซิฮีโมโกลบิน และขับแหล่งกำเนิดแสงอินฟราเรดด้วยคลื่นความถี่ 190 เฮิร์ต สำหรับวัดออกซิฮีโมโกลบิน ทั้งสองคลื่นนี้จะส่องผ่านทะลุนิ้วมือในเวลาเดียวกันไปยังตัวรับแสงที่อยู่ด้านล่างของหัววัดความอิ่มตัวของออกซิเจนในเลือด เพื่อวัดปริมาณของฮีโมโกลบินทั้งสองในเลือด และนำข้อมูลที่ได้มาคำนวณหาปริมาณร้อยละของออกซิเจนในเลือดในช่วงร้อยละ 80-100 โดยการประมวลผลทั้งหมดถูกกระทำบนไมโครคอนโทรลเลอร์ ESP32 จากนั้นส่งข้อมูลที่ได้ผ่านบลูทูธไปแสดงผลที่หน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ พบว่าได้ค่าความเที่ยงตรงของเครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วเท่ากับ 98.34% และได้ค่าแม่นยำเท่ากับ $96.33 \pm 0.91\%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

The current situation of the coronavirus disease 2019 epidemic has affected people's lives. People are awake to take care of their health. This project will present a Bluetooth Fingertip Pulse Oximeter. This device will help to detect the initial symptoms of various diseases. At the top of the Pulse oximeter probe, A red light source is activated at a frequency of 80 Hz to measure oxyhemoglobin. And an infrared light source is activated at a frequency of 1 9 0 Hz to measure deoxyhemoglobin. Both waves pass simultaneously through the finger to the photodiode (located under the Pulse oximeter probe) to measure the amount of hemoglobin in the blood. Then use the data to calculate the percentage of oxygen in the blood in the range of 80–100 percent. Finally, send the data via Bluetooth on the ESP32 microcontroller to display on the mobile application. It was found that the accuracy of the Bluetooth fingertip pulse oximeter was 98.34%, and the precision was $96.33 \pm 0.91\%$.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	X
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 หลักการดูดซับแสงของออกซิอีโมโกลบิน และดีออกซิอีโมโกลบิน	3
2.2 หัววัดความอิมิตัวออกซิเจนในเลือด	5
2.3 วงจรบัฟเฟอร์	6
2.4 วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า	6
2.5 LM324	7
2.6 การ์ดเสียง	7
2.7 ไมโครคอนโทรลเลอร์ ESP32	8
2.8 การมอดูเลต	9
2.9 การดีมอดูเลต	11
2.10 การมัลติเพล็กซ์สัญญาณ	13
2.11 ตัวกรองสัญญาณเชิงเลข	14
2.12 ประสิทธิภาพการทำงานของเครื่องมือวัด	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3	
การออกแบบและการจัดทำปฏิญญานิพนธ์	21
3.1 การออกแบบ	21
3.2 เครื่องมือที่ใช้ในการทดลอง	42
3.3 การจัดเก็บผลการทดลอง	44
บทที่ 4	48
ผลการทดลอง	
4.1 ผลการทดสอบการทำงานของโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิร์ต	48
4.2 ผลการทดสอบวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดออกซิเจน	48
4.3 การประมวลผลสัญญาณที่ได้รับบนไมโครคอนโทรลเลอร์ ESP32	52
4.4 การทดสอบการรับและส่งข้อมูลผ่านบลูทูธ	54
4.5 การทดสอบระบบรวมทั้งหมด	56
4.6 ผลการทดลองเทียบกับเครื่องมาตรฐาน	57
บทที่ 5	63
สรุปผลและข้อเสนอแนะ	
5.1 สรุปผล	63
5.2 ข้อเสนอแนะ	64
บรรณานุกรม	65
ภาคผนวก ก	
คำสั่งการทำงานในโปรแกรม MATLAB	67
ภาคผนวก ข	
คำสั่งการทำงานในโปรแกรม ARDUINO IDE	72

สารบัญรูป

รูปที่	หน้า
2.1 คุณสมบัติการดูดซับแสงของออกซีไอโมโกลบินและดีออกซีไอโมโกลบิน	3
2.2 การดูดซับแสงของช่วงที่มีการไหลของเลือด และช่วงที่ไม่มีไหลของเลือด	4
2.3 สัญลักษณ์ทางไฟฟ้าของแอลอีดี	5
2.4 สัญลักษณ์ทางไฟฟ้าของเซนเซอร์ตัวรับแสง	5
2.5 วงจรบัฟเฟอร์	6
2.6 วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า	6
2.7 ลักษณะภายนอก และโครงสร้างภายในของ LM324	7
2.8 การ์ดเสียง	7
2.9 ไมโครคอนโทรลเลอร์ ESP WROOM-32 CHIP รุ่น 36 ขา	8
2.10 ตำแหน่งและหน้าที่ขาของไมโครคอนโทรลเลอร์ ESP WROOM-32 CHIP	8
2.11 การมอดูเลตเชิงขนาด	9
2.12 การติมอดูเลตแบบตรวจจับขอบ	12
2.13 การติมอดูเลตแบบแบบโคฮีเรนต์ดีเทคชัน	12
2.14 การมัลติเพล็กซ์และติมัลติเพล็กซ์ทางความถี่ของสัญญาณเอเอ็ม	14
2.15 โครงสร้างระบบแบบผลตอบสนองอิมพัลส์ไม่จำกัด	14
2.16 กราฟแสดงฟังก์ชันการส่งผ่านกับความถี่ของวงจรกรองความถี่แถบผ่าน	15
2.17 ผลการตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน	17
3.1 แผนภาพการทำงาน	21
3.2 วงจรขับตัวให้กำเนิดแสงสีแดง และแสงอินฟราเรด	22
3.3 วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความเข้มตัวออกซิเจนในเลือด	22
3.4 วงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงรวมเข้ากับหัววัดความเข้มตัวออกซิเจนในเลือด	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 PCB LAYOUT ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และ วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจน ในเลือด	25
3.6 SCHEMATIC ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และ วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจน ในเลือด	25
3.7 ลายวงจรพิมพ์ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และ วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจน ในเลือด	26
3.8 วงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณ จากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือดที่สร้างเสร็จ แล้ว	26
3.9 แผนภาพการทำงานการสร้าง และเก็บสัญญาณไชน์ความถี่ 80 เฮิร์ต และ 190 เฮิร์ต บนไมโครคอนโทรลเลอร์ ESP32	27
3.10 คำสั่งในการสร้างและเก็บสัญญาณไชน์ความถี่ 80 และ 190 เฮิร์ต	28
3.11 แผนภาพการทำงานการประมวลผลในโปรแกรม MATLAB	29
3.12 คำสั่งในการสร้างตัวกรองแถบความถี่ผ่าน, การตีมอดูเลต และตัวกรอง ความถี่ต่ำผ่านในโปรแกรม MATLAB	30
3.13 แผนภาพขั้นตอนการประมวลผลสัญญาณ และคำนวณหาค่าความอืดตัว ของออกซิเจนในเลือด	31
3.14 แผนภาพการทำงานของการตีมอดูเลตสัญญาณ	36
3.15 แผนภาพการหาค่าจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีทั้งสอง	37
3.16 แผนภาพการหาค่าความอืดตัวของออกซิเจนในเลือด	38
3.17 แผนภาพการรับและส่งข้อมูลผ่านบลูทูธบนไมโครคอนโทรลเลอร์ ESP32	39
3.18 คำสั่งในการรับและส่งข้อมูลผ่านบลูทูธ	39

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 แผนภาพการทำงานของแอปพลิเคชัน	40
3.20 บล็อกการทำงานของแอปพลิเคชัน	40
3.21 แผนภาพการทำงานการรวมระบบบนไมโครคอนโทรลเลอร์ ESP32	41
3.22 เครื่องออสซิลโลสโคป	42
3.23 เครื่องจ่ายไฟ	42
3.24 เครื่องสร้างสัญญาณ	43
3.25 โลโก้โปรแกรม ARDUINO IDE	43
3.26 โลโก้โปรแกรม PROTEUS	44
3.27 โลโก้โปรแกรม MATLAB	44
3.28 การติดตั้งการทดสอบการทำงานของโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต	45
3.29 การติดตั้งวงจรสำหรับการทดสอบวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือด	45
3.30 การติดตั้งสำหรับการประมวลผลสัญญาณที่ได้รับบนไมโครคอนโทรลเลอร์ ESP32	46
3.31 การติดตั้งและสภาพแวดล้อมสำหรับการทดลอง	47
4.1 สัญญาณไซน์ที่ได้บนหน้าจอเครื่องออสซิลโลสโคป	48
4.2 สัญญาณเอพดีเอ็มที่โปรแกรม MATLAB รับเข้ามา	49
4.3 สัญญาณเอพดีเอ็มที่ผ่านการลบด้วยค่าเฉลี่ย	49
4.4 สัญญาณเอเอ็มของคลื่นแสงสีแดง	50
4.5 สัญญาณเอเอ็มของคลื่นอินฟราเรด	50
4.6 สัญญาณเอเอ็มยกกำลังสองของคลื่นแสงสีแดง	50
4.7 สัญญาณเอเอ็มยกกำลังสองของคลื่นอินฟราเรด	50

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.8	สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรด	51
4.9	สัญญาณเอพดีเอ็มที่รับเข้ามาพล็อตแสดงผลบนโปรแกรม MATLAB	52
4.10	สัญญาณเอเอ็มของคลื่นแสง และคลื่นอินฟราเรดบนโปรแกรม MATLAB	53
4.11	กราฟสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดบนโปรแกรม MATLAB	53
4.12	อัตราส่วนการดูดซับแสง และความอิมพัลส์ของออกซิเจนในเลือดบนซีเรียลมอนิเตอร์	54
4.13	บาร์โค้ดสำหรับการติดตั้งแอปพลิเคชัน และไอคอนของแอปพลิเคชัน	54
4.14	ผลทดสอบการรับและส่งข้อมูลผ่านบลูทูธบนซีเรียลมอนิเตอร์	55
4.15	ผลทดสอบการรับและส่งข้อมูลผ่านบลูทูธบนหน้าจอแอปพลิเคชัน	56
4.16	ผลทดสอบระบบรวมบนซีเรียลมอนิเตอร์	56
4.17	ผลทดสอบระบบรวมบนหน้าจอแอปพลิเคชัน	57

สารบัญตาราง

ตารางที่	หน้า
4.1 ค่าเปอร์เซ็นต์ความอิมตัวออกซิเจนในเลือดที่ได้จากการทดลอง (หญิง)	58
4.2 ค่าเปอร์เซ็นต์ความอิมตัวออกซิเจนในเลือดที่ได้จากการทดลอง (ชาย)	59
4.3 ค่าความแม่นยำ และความเที่ยงตรงที่ได้จากการคำนวณ (หญิง)	61
4.4 ค่าความแม่นยำ และความเที่ยงตรงที่ได้จากการคำนวณ (ชาย)	62



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากสถานการณ์การแพร่ระบาดของโรคติดเชื้อไวรัสโคโรนา 2019 (โควิด-19) ที่เกิดขึ้นทั้งในประเทศไทย และต่างประเทศ ทำให้เกิดผลกระทบขึ้นมากมายทั้งทางด้านเศรษฐกิจ การท่องเที่ยว การแพทย์ รวมไปถึงการใช้ชีวิตของผู้คนในปัจจุบัน ซึ่งเหตุการณ์ที่เกิดขึ้นนี้ทำให้ผู้คนจำนวนมากตื่นตัวในเรื่องของการดูแลสุขภาพมากขึ้น

ปัญหานี้จึงได้ทำการออกแบบและสร้างเครื่องวัดความอิ่มตัวออกซิเจนในเลือดปลายนิ้วมือ ซึ่งเป็นอุปกรณ์ที่ช่วยในการตรวจสอบ และเฝ้าระวังอาการเบื้องต้นของโรคต่าง ๆ ได้ โดยที่ด้านบนของหัววัดความอิ่มตัวออกซิเจนในเลือด (Pulse oximeter probe) จะปล่อยคลื่นแสงที่มีความถี่แตกต่างกันสองคลื่น คือ คลื่นสีแดงสำหรับวัดออกซีฮีโมโกลบิน (Oxyhemoglobin) และคลื่นอินฟราเรดสำหรับวัดดีออกซีฮีโมโกลบิน (Deoxyhemoglobin) ทั้งสองคลื่นนี้จะส่องผ่านทะลุนิ้วมือในเวลาเดียวกันไปยังตัวรับแสงที่อยู่ด้านล่างของหัววัดความอิ่มตัวออกซิเจนในเลือด เพื่อวัดปริมาณของฮีโมโกลบินทั้งสองในเลือด และนำข้อมูลที่ได้มาคำนวณหาปริมาณร้อยละของออกซิเจนในเลือด โดยการประมวลผลทั้งหมดถูกกระทำบนไมโครคอนโทรลเลอร์ ESP32 จากนั้นส่งข้อมูลที่ได้ผ่านบลูทูธ ไปแสดงผลที่หน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile device)

1.2 วัตถุประสงค์

- 1) ออกแบบและสร้างเครื่องวัดความอิ่มตัวออกซิเจนในเลือดปลายนิ้วที่สามารถทำงานได้ภายใต้สภาวะแสงที่แตกต่างกัน
- 2) สร้างแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ เพื่อแสดงผลปริมาณร้อยละของออกซิเจนในเลือด

1.3 ขอบเขตของปฏิญญานิพนธ์

ปฏิญญานิพนธ์นี้ได้ทำการออกแบบและสร้างเครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้ว โดยจะมีการขับแหล่งกำเนิดแสงสีแดงด้วยคลื่นความถี่ 80 เฮิร์ต และขับแหล่งกำเนิดแสงอินฟราเรดด้วยคลื่นความถี่ 190 เฮิร์ต เพื่อทำให้เกิดการรวมของสัญญาณพีพีจี (Photoplethysmography: PPG) ที่สร้างจากไมโครคอนโทรลเลอร์ ESP32 และมีการแสดงผลปริมาณร้อยละของออกซิเจนในเลือดในช่วงร้อยละ 80-100 ผ่านบลูทูธไปยังแอปพลิเคชันบนอุปกรณ์เคลื่อนที่



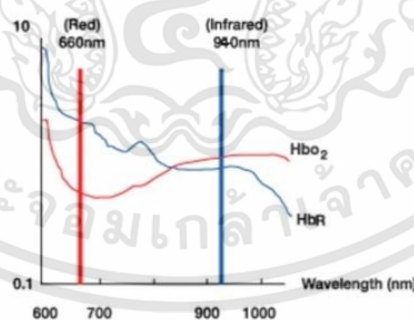
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญาณีพนธ์เรื่อง “เครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ” ได้ทำการออกแบบส่วนของวงจรจับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิ่มตัวของออกซิเจนในเลือด ซึ่งอุปกรณ์นี้สามารถวัดปริมาณของฮีโมโกลบิน และนำข้อมูลที่ได้ออกมาคำนวณหาปริมาณร้อยละของออกซิเจนในเลือดได้ ดังนั้นปริญาณีพนธ์ที่น่าเสนอจึงมีหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 หลักการดูดซับแสงของออกซิฮีโมโกลบิน และดีออกซิฮีโมโกลบิน

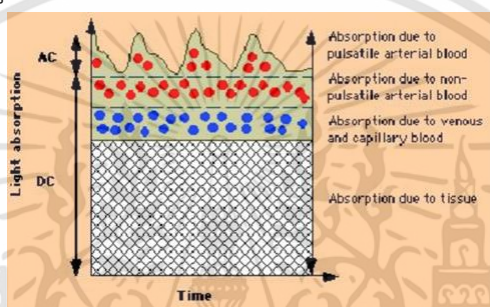
การวัดค่าความอิ่มตัวของออกซิเจนในเลือดจำเป็นต้องใช้หลักการดูดซับแสงของฮีโมโกลบิน 2 ชนิด คือ ฮีโมโกลบินที่จับตัวกับออกซิเจน (Oxyhemoglobin) และฮีโมโกลบินที่ไม่จับตัวกับออกซิเจน (Deoxyhemoglobin) โดยฮีโมโกลบินทั้งสองชนิดนี้จะมีการดูดซับแสงในช่วงความยาวคลื่นที่แตกต่างกัน นั่นคือออกซิฮีโมโกลบินจะสามารถดูดซับคลื่นแสงได้ดีในช่วงความยาวคลื่น 600–750 นาโนเมตร และดีออกซิฮีโมโกลบินสามารถดูดซับคลื่นแสงได้ดีในช่วงความยาวคลื่น 850–1000 นาโนเมตร แสดงคุณสมบัติการดูดซับแสงของออกซิฮีโมโกลบินและดีออกซิฮีโมโกลบินในเลือดได้ดังรูปที่ 2.1



รูปที่ 2.1 คุณสมบัติการดูดซับแสงของออกซิฮีโมโกลบินและดีออกซิฮีโมโกลบิน [1]

การวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วจะแบ่งออกเป็น 2 ส่วนหลัก คือ ส่วนการสร้างสัญญาณเพื่อควบคุมการทำงานของแหล่งกำเนิดแสงในหัววัดความอิ่มตัวของออกซิเจนในเลือด ซึ่งจะสร้างสัญญาณไซน์ที่มีความถี่แตกต่างกัน 2 ความถี่ และส่วนหัววัดความอิ่มตัวของออกซิเจน

ในเลือดที่จะทำหน้าที่ในการปล่อยและรับคลื่นแสง เมื่อปล่อยคลื่นแสงจากแหล่งกำเนิดแสง คลื่นแสงจะเดินทางผ่านปลายนิ้วมายังเซนเซอร์ตัวรับแสง ซึ่งเซนเซอร์ตัวรับแสงนี้จะสามารถรับคลื่นแสงได้ 2 ช่วง ได้แก่ ช่วงที่มีการไหลของเลือด (Alternating Current Light: AC) ในช่วงนี้จะเป็นการดูดซับคลื่นแสงของหลอดเลือดดำและหลอดเลือดแดงที่มีการไหลของเลือดมาจากการบีบคลายตัวของหัวใจ และช่วงที่ไม่มีมีการไหลของเลือด (Direct Current Light: DC) ช่วงนี้จะเป็นการดูดซับคลื่นแสงของเนื้อเยื่อและกระดูก สามารถแสดงการดูดซับแสงของช่วงที่มีการไหลของเลือด และช่วงที่ไม่มีมีการไหลของเลือดดังรูปที่ 2.2



รูปที่ 2.2 การดูดซับแสงของช่วงที่มีการไหลของเลือด และช่วงที่ไม่มีมีการไหลของเลือด [1]

เมื่อทราบค่าการดูดซับแสงในแต่ละช่วงสามารถนำมาคำนวณหาอัตราส่วนการดูดซับคลื่นแสง (R) ได้ดังสมการต่อไปนี้

$$R = \frac{\frac{AC_{RED}}{DC_{RED}}}{\frac{AC_{IR}}{DC_{IR}}} \quad (2.1)$$

โดยที่

- R คือ อัตราส่วนการดูดซับคลื่นแสง
- AC_{RED} คือ การดูดซับคลื่นแสงสีแดงในช่วงที่มีการไหลของเลือด
- DC_{RED} คือ การดูดซับคลื่นแสงสีแดงในช่วงที่ไม่มีมีการไหลของเลือด
- AC_{IR} คือ การดูดซับคลื่นอินฟราเรดในช่วงที่มีการไหลของเลือด
- DC_{IR} คือ การดูดซับคลื่นอินฟราเรดในช่วงที่ไม่มีมีการไหลของเลือด

และเมื่อคำนวณหาอัตราส่วนการดูดซับคลื่นแสงแล้ว สามารถนำมาคำนวณหาค่าความอิ่มตัวของออกซิเจนในเลือดในช่วงที่มีเลือดไหลผ่าน ($\%SpO_2$) ได้ดังสมการต่อไปนี้

$$SpO_2 = 110 - (25 \times R) \quad (2.2)$$

2.2 หัววัดความอิ่มตัวออกซิเจนในเลือด (Pulse Oximeter Probe)

ภายในของหัววัดความอิ่มตัวออกซิเจนในเลือดยี่ห้อ Datex-Ohmeda DB9 จะประกอบไปด้วยการทำงาน 2 ส่วน ดังต่อไปนี้

2.2.1 แอลอีดี (Light Emitting Diode)

แอลอีดี หรือไดโอดเปล่งแสง คือ อุปกรณ์สารกึ่งตัวนำที่สามารถเปล่งแสงได้ ติดตั้งอยู่ที่ด้านบนของหัววัดความอิ่มตัวออกซิเจนในเลือด ใช้เป็นแหล่งกำเนิดแสงสำหรับในการวัดความอิ่มตัวของออกซิเจนในเลือด เมื่อมีการจ่ายแรงดันไฟฟ้าให้กับแอลอีดี โดยจ่ายไฟขั้วบวกเข้ากับขั้วแอโนด (Anode) และไฟขั้วลบเข้ากับขั้วแคโทด (Cathode) จะทำให้อิเล็กตรอนได้รับพลังงานที่สูงขึ้น เกิดการข้ามรอยต่อมารวมกับโฮลที่สารพี ระดับพลังงานของอิเล็กตรอนเกิดการเปลี่ยนแปลง จึงทำให้คายพลังงานออกมาเป็นคลื่นแสง แสดงสัญลักษณ์ทางไฟฟ้าได้ดังรูปที่ 2.3



รูปที่ 2.3 สัญลักษณ์ทางไฟฟ้าของแอลอีดี

2.2.2 เซนเซอร์ตัวรับแสง (Photodiode)

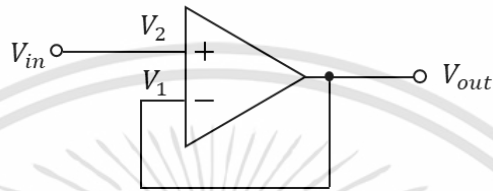
เซนเซอร์ตัวรับแสง คือ อุปกรณ์สารกึ่งตัวนำสองชั้นพีเอ็น ติดตั้งอยู่ที่ด้านล่างของหัววัดความอิ่มตัวออกซิเจนในเลือด ใช้สำหรับรับสัญญาณแสงที่ส่องมาจากแอลอีดีผ่านทะลุนิ้วมือลงมาเปลี่ยนให้เป็นสัญญาณไฟฟ้า โดยจะทำงานเมื่อมีแสงตกกระทบอิเล็กตรอนที่ยึดติดอยู่กับอะตอม จึงทำให้เกิดอิเล็กตรอนอิสระและโฮลอิสระขึ้น จากนั้นอิเล็กตรอนและโฮลเหล่านี้จะเคลื่อนที่เข้าไปในเขตปลอดพาหะ เกิดเป็นกระแสไฟฟ้าที่แปรเปลี่ยนตามปริมาณความเข้มแสงเรียกว่า กระแสโฟโต (Photocurrent: I_p) แสดงสัญลักษณ์ทางไฟฟ้าได้ดังรูปที่ 2.4



รูปที่ 2.4 สัญลักษณ์ทางไฟฟ้าของเซนเซอร์ตัวรับแสง

2.3 วงจรบัฟเฟอร์ (Buffer)

วงจรบัฟเฟอร์เป็นวงจรที่ไม่มีการลดทอนค่าแรงดันไฟฟ้า เนื่องจากมีค่าอิมพีแดนซ์สูง นิยมใช้สำหรับเชื่อมต่อระหว่างสองวงจร เพื่อช่วยให้แรงดันไฟฟ้ามีค่าคงที่ แสดงลักษณะวงจรบัฟเฟอร์ที่สร้างจากออปแอมป์ได้ดังรูปที่ 2.5



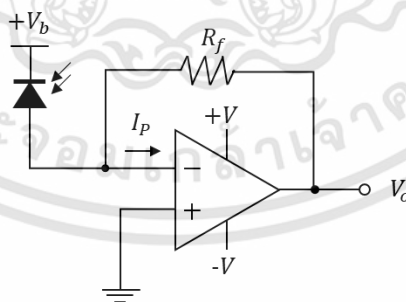
รูปที่ 2.5 วงจรบัฟเฟอร์

จากวงจรดังรูป 2.5 สามารถเขียนสมการอธิบายความสัมพันธ์ได้ดังสมการที่ 2.3

$$V_{out} = V_{in} \quad (2.3)$$

2.4 วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า (Current to Voltage Converter)

การแปลงกระแสไฟฟ้าที่เกิดจากเซนเซอร์ตัวรับแสงให้อยู่ในรูปของแรงดันไฟฟ้า จะใช้วงจรขยายทรานซิมพีแดนซ์ (Transimpedance Amplifier) ซึ่งเป็นวงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า ในวงจรมีการใช้งานออปแอมป์โดยที่ลักษณะของวงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้าสามารถแสดงได้ดังรูปที่ 2.6



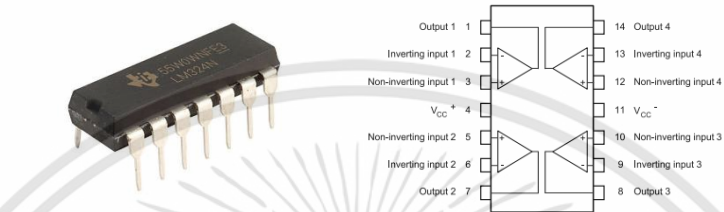
รูปที่ 2.6 วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า

จากวงจรดังรูป 2.6 สามารถเขียนสมการอธิบายความสัมพันธ์ได้ดังสมการที่ 2.4

$$V_{out} = -R_f \times I_p \quad (2.4)$$

2.5 LM324

LM324 เป็นไอซีที่ประกอบด้วยออปแอมป์ที่เป็นอิสระต่อกันถึง 4 ตัว ใช้สำหรับขยายสัญญาณทางไฟฟ้าต่าง ๆ และมีการออกแบบให้สามารถใช้งานได้โดยใช้ไฟเลี้ยงเพียงค่าเดียวในช่วงแรงดันไฟฟ้าที่หลากหลาย แสดงลักษณะภายนอก และโครงสร้างภายในของ LM324 ได้ดังรูป 2.7



(ก) ลักษณะภายนอก (ข) โครงสร้างภายใน
รูปที่ 2.7 ลักษณะภายนอก [3] และโครงสร้างภายในของ LM324 [4]

2.6 การ์ดเสียง (Sound Card USB)

การ์ดเสียงเป็นอุปกรณ์ที่ใช้สำหรับเชื่อมต่อกับคอมพิวเตอร์ ซึ่งมีหลักการทำงานโดยเริ่มจากรับข้อมูลจากภายนอกเข้ามาในรูปแบบสัญญาณแอนะล็อก เมื่อสัญญาณถูกส่งมาที่ภาค ADC (Analog to Digital Converter) จะทำการแปลงสัญญาณให้อยู่ในรูปของฐานสอง (Binary) จากนั้นข้อมูลนี้จะถูกส่งต่อไปที่ภาคดีเอสพี (Digital Signal Processor: DSP) เพื่อทำการบีบอัดข้อมูลให้มีขนาดเล็กแล้วค่อยส่งต่อไปให้กับซีพียู เพื่อทำการประมวลผลตามคำสั่งต่อไป มีลักษณะดังรูปที่ 2.8



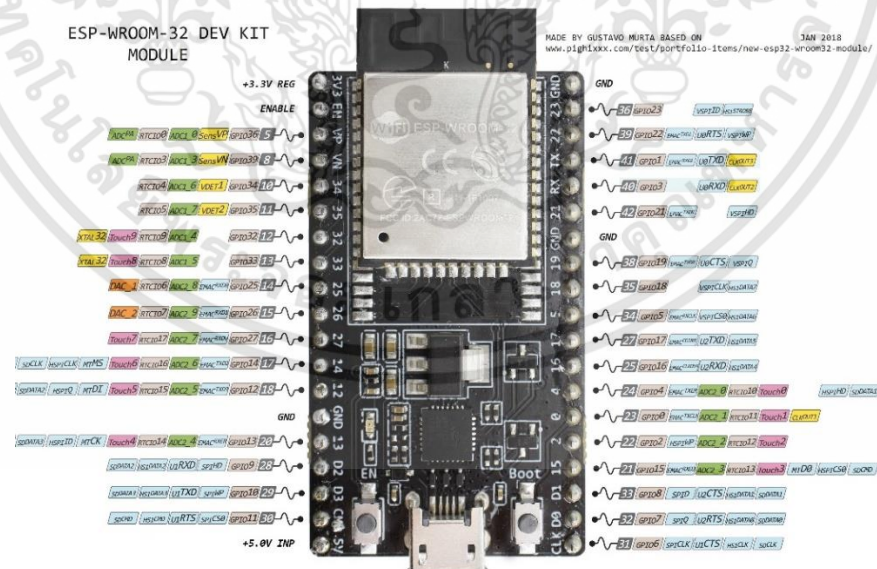
รูปที่ 2.8 การ์ดเสียง [5]

2.7 ไมโครคอนโทรลเลอร์ ESP32

ไมโครคอนโทรลเลอร์ ESP32 เป็นบอร์ดพัฒนาสำเร็จรูปที่รองรับการเชื่อมต่อไวไฟ (Wi-Fi) และบลูทูธรุ่น 4.2 (Bluetooth 4.2 BLE) ผลิตโดยบริษัท Espressif จากประเทศจีน รองรับ การเขียนโปรแกรมโดยใช้โปรแกรม Arduino IDE มีการใช้งานง่าย และราคาถูก บอร์ดชนิดนี้มีการ ผลิตออกมาให้ใช้งานหลากหลายรุ่นทั้ง ESP32-DevKitC, ESP-WROVER-KIT และ ESP32-PICO-KIT เป็นต้น โดยบอร์ดที่เป็นที่นิยมใช้งานในปัจจุบันคือ ESP WROOM-32 Chip ซึ่งเป็นบอร์ดรุ่น แรกที่ถูกผลิตขึ้น มีการทำงานแบบสองคอร์ (Dual Core) ที่มีความเร็ว 240 เมกะเฮิร์ต มีขนาดของ แรม 512 กิโลไบต์ และมีให้เลือกใช้งานทั้งรุ่น 30 ขา และ 36 ขา สำหรับเชื่อมต่อกับอุปกรณ์ ภายนอก มีลักษณะดังรูปที่ 2.9 ในการใช้งานไมโครคอนโทรลเลอร์ ESP32 จำเป็นต้องจ่ายไฟเลี้ยง 3.3 โวลต์ และแต่ละขาจะมีหน้าที่ในการใช้งานที่แตกต่างกันแสดงได้ดังรูปที่ 2.10



รูปที่ 2.9 ไมโครคอนโทรลเลอร์ ESP WROOM-32 Chip รุ่น 36 ขา [7]



รูปที่ 2.10 ตำแหน่งและหน้าที่ขาของไมโครคอนโทรลเลอร์ ESP WROOM-32 Chip [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

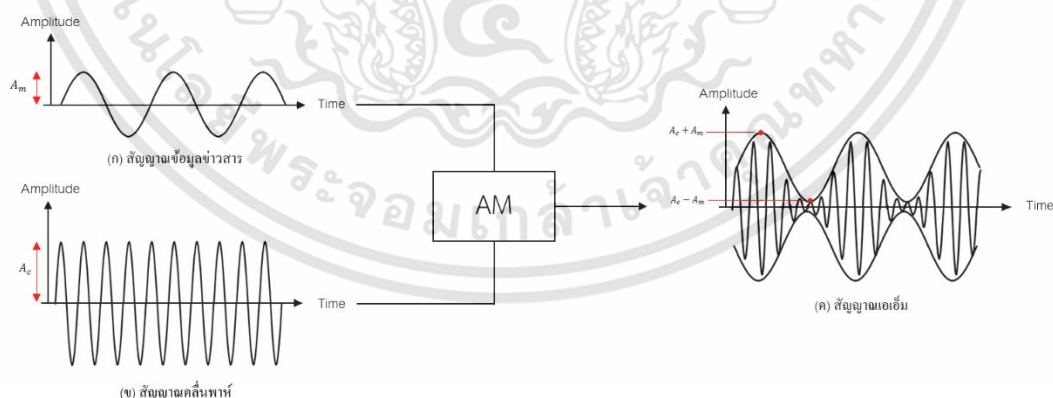
จากรูปที่ 2.10 จะแสดงตำแหน่งของขาสำหรับการใช้งาน โดยในแต่ละขาจะมีการใช้งานที่แตกต่างกัน ซึ่งมีขา GPIO จำนวน 32 ช่อง, ขาที่รองรับ UART จำนวน 3 ช่อง, ขาที่รองรับ SPI จำนวน 3 ช่อง, ขาที่รองรับ I2C จำนวน 2 ช่อง, ขาที่รองรับ ADC จำนวน 12 ช่อง, ขาที่รองรับ DAC จำนวน 2 ช่อง, ขาที่รองรับ I2S จำนวน 2 ช่อง, ขาที่รองรับ PWM/Timer ทุกช่อง และขารองรับการเชื่อมต่อกับ SD-Card

2.8 การมอดูเลต (Modulation)

การมอดูเลตเป็นกระบวนการเคลื่อนย้ายย่านความถี่ของสัญญาณข้อมูลข่าวสาร หรือสัญญาณแบนด์มูลฐาน (Baseband Signal) ไปกับสัญญาณคลื่นพาห์ (Carrier Signal) ซึ่งจะทำให้พารามิเตอร์ต่าง ๆ ของสัญญาณคลื่นพาห์มีการเปลี่ยนแปลงไปตามสัญญาณข้อมูลข่าวสาร โดยในที่นี้จะขอกกล่าวถึงเพียงการมอดูเลตเชิงขนาด (Amplitude Modulation: AM)

2.8.1 การมอดูเลตเชิงขนาด

เมื่อนำสัญญาณข้อมูลข่าวสารฝากไปกับสัญญาณคลื่นพาห์ หรือสัญญาณไซน์ (Sinusoidal Signal) ที่มีความถี่สูงกว่าสัญญาณข้อมูลข่าวสารมาก ๆ จะทำให้ค่าแอมพลิจูดของสัญญาณคลื่นพาห์นั้น ถูกควบคุมให้เปลี่ยนแปลงไปตามสัญญาณข้อมูลข่าวสารที่นำมามอดูเลต สามารถแสดงความสัมพันธ์ของสัญญาณข้อมูลข่าวสาร สัญญาณคลื่นพาห์ และสัญญาณเอเอ็มได้ดังรูปที่ 2.11



รูปที่ 2.11 การมอดูเลตเชิงขนาด

(ก) สัญญาณข้อมูลข่าวสาร, (ข) สัญญาณคลื่นพาห์ และ (ค) สัญญาณเอเอ็ม

และสามารถอธิบายการมอดูเลตเชิงขนาดเป็นสมการทางคณิตศาสตร์ได้ดังนี้

กำหนดให้สัญญาณข้อมูลข่าวสารคือ

$$m(t) = A_m \cos(\omega_m t) \quad (2.5)$$

โดยที่ $m(t)$ คือ ข้อมูลข่าวสาร

A_m คือ แอมพลิจูดของข้อมูลข่าวสาร

ω_m คือ ความถี่เชิงมุมของสัญญาณข้อมูลข่าวสาร

และสัญญาณคลื่นพาห้คือ

$$c(t) = A_c \cos(\omega_c t) \quad (2.6)$$

โดยที่ $c(t)$ คือ สัญญาณคลื่นพาห้

A_c คือ แอมพลิจูดของสัญญาณคลื่นพาห้

ω_c คือ ความถี่เชิงมุมของสัญญาณคลื่นพาห้

จากสมการสัญญาณข้อมูลข่าวสาร (2.5) และสัญญาณคลื่นพาห้ (2.6) เมื่อผ่านกระบวนการมอดูเลตเชิงขนาดแล้วจะได้สัญญาณเอเอ็มดังสมการที่ 2.7

$$\begin{aligned} \phi_{AM}(t) &= A_c \cos(\omega_c t) + A_m \cos(\omega_m t) \cos(\omega_c t) \\ \phi_{AM}(t) &= [A_c + A_m \cos(\omega_m t)] \cos(\omega_c t) \end{aligned} \quad (2.7)$$

โดยที่ $\phi_{AM}(t)$ คือ สัญญาณเอเอ็ม

2.8.2 ดัชนีการมอดูเลต

สัญญาณเอเอ็มที่ผ่านการมอดูเลตนั้นจะเกิดจากการเปลี่ยนแปลงค่าแอมพลิจูดของสัญญาณคลื่นพาห์ ซึ่งค่าแอมพลิจูดของสัญญาณคลื่นพาห์ที่เปลี่ยนแปลงไปก็ขึ้นอยู่กับอัตราส่วนระหว่างแอมพลิจูดของสัญญาณข้อมูลข่าวสารกับสัญญาณคลื่นพาห์ที่เรียกว่า “ ดัชนีการมอดูเลต (Modulation Index : m) ” โดยสามารถหาค่าดัชนีการมอดูเลตของสัญญาณเอเอ็มได้ดังสมการที่ 2.8

$$m = \frac{A_m}{A_c} \quad (2.8)$$

โดยที่ m คือ ค่าดัชนีการมอดูเลต

ดังนั้นในการพิจารณาค่าดัชนีการมอดูเลตของสัญญาณเอเอ็มที่ได้ต้องมีค่าไม่เกิน 1 จะได้ขอบของสัญญาณเอเอ็มที่มีรูปร่างเหมือนกับสัญญาณข้อมูลข่าวสาร ซึ่งจะทำให้ในการตีมอดูเลตสัญญาณกลับมาสามารถใช้วิธีการตีมอดูเลตแบบตรวจจับขอบได้ แต่หากค่าดัชนีการมอดูเลตเกิน 1 จะได้ขอบของสัญญาณเอเอ็มที่มีรูปร่างไม่เหมือนสัญญาณข้อมูลข่าวสาร ซึ่งจะส่งผลต่อการตีมอดูเลตสัญญาณกลับมาจะต้องใช้วิธีการตีมอดูเลตสัญญาณวิธีอื่นที่ไม่ใช่การตีมอดูเลตแบบตรวจจับขอบ

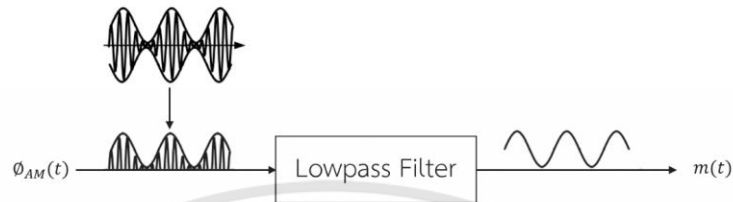
2.9 การตีมอดูเลต (Demodulation)

การตีมอดูเลตเป็นกระบวนการกู้คืนสัญญาณข้อมูลข่าวสารที่ถูกส่งมาในรูปสัญญาณเอเอ็ม ให้มีความผิดเพี้ยนของสัญญาณข้อมูลข่าวสารเดิมน้อยที่สุด โดยในที่นี้จะขอกกล่าวถึงเพียงการตีมอดูเลตเชิงขนาด (Amplitude Demodulation) ซึ่งจะประกอบไปด้วยวิธีการตีมอดูเลตแบบตรวจจับขอบ (Envelope Detection) และวิธีการตีมอดูเลตแบบตรวจจับซิงโครนัสดีเทคชัน (Synchronous Detection) หรือเรียกอีกชื่อว่าการตีมอดูเลตแบบโคฮีเรนต์ดีเทคชัน (Coherent Detection)

2.9.1 การตีมอดูเลตแบบตรวจจับขอบ

เนื่องจากสัญญาณเอเอ็มจะปรากฏขอบของสัญญาณข้อมูลข่าวสารทั้งสองด้าน ดังนั้นจึงใช้การตีมอดูเลตแบบตรวจจับขอบ โดยจะเลือกสัญญาณเอเอ็มที่ปรากฏขอบของสัญญาณข้อมูลข่าวสารเพียงด้านเดียว และขอบของสัญญาณข้อมูลข่าวสารนั้นต้องมีค่าเป็นบวก เพื่อให้ในการตีมอดูเลตสัญญาณเอเอ็มนั้นได้สัญญาณข้อมูลข่าวสารที่มีใกล้เคียงกับสัญญาณข้อมูลข่าวสารเดิมให้มากที่สุด จากนั้นนำสัญญาณเอเอ็มด้านที่เลือกไปผ่านตัวกรองความถี่ต่ำ จะได้สัญญาณ

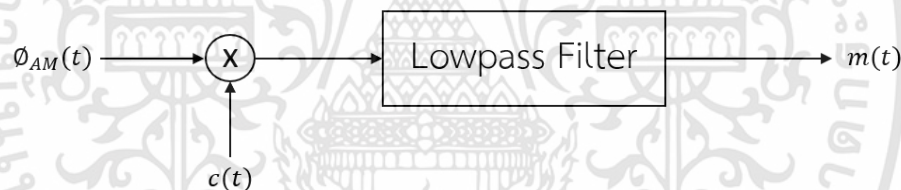
ของข้อมูลข่าวสารออกมา สามารถแสดงความสัมพันธ์ของสัญญาณเอเอ็มที่ผ่านการการดีมอดูเลตแบบตรวจจับขอบได้ดังรูปที่ 2.12



รูปที่ 2.12 การดีมอดูเลตแบบตรวจจับขอบ

2.9.2 การดีมอดูเลตแบบโคฮีเรนต์ดีเทคชัน

การดีมอดูเลตแบบโคฮีเรนต์ดีเทคชัน สามารถทำได้โดยนำสัญญาณที่มีความถี่และเฟสตรงกันกับสัญญาณของความถี่คลื่นพาห้มาคูณกับสัญญาณเอเอ็ม หรือสัญญาณของข้อมูลข่าวสารที่ผ่านการมอดูเลตแล้ว จากนั้นก็นำสัญญาณที่ผ่านการคูณกันดังกล่าวมาผ่านตัวกรองความถี่ต่ำ จะได้สัญญาณของข้อมูลข่าวสาร $m(t)$ ออกมาดังรูปที่ 2.13



รูปที่ 2.13 การดีมอดูเลตแบบโคฮีเรนต์ดีเทคชัน

และสามารถอธิบายการดีมอดูเลตแบบโคฮีเรนต์ดีเทคชันเป็นสมการทางคณิตศาสตร์ได้จากสมการที่ 2.9

$$\phi_{AM}(t)c(t) = [A_c + A_m \cos(\omega_m t)] A_c \cos^2(\omega_c t) \quad (2.9)$$

จากสมการทางคณิตศาสตร์

$$\cos^2 \theta = \frac{1 + \cos(2\theta)}{2} \quad (2.10)$$

เมื่อแทนค่าสมการที่ 2.10 ในสมการที่ 2.9 จะได้

$$\begin{aligned} \phi_{AM}(t)c(t) &= [A_c + A_m \cos(\omega_m t)] A_c \left(\frac{1 + \cos(2\omega_c t)}{2} \right) \\ &= [A_c^2 + A_m A_c \cos(\omega_m t)] \left(\frac{1 + \cos(2\omega_c t)}{2} \right) \end{aligned}$$

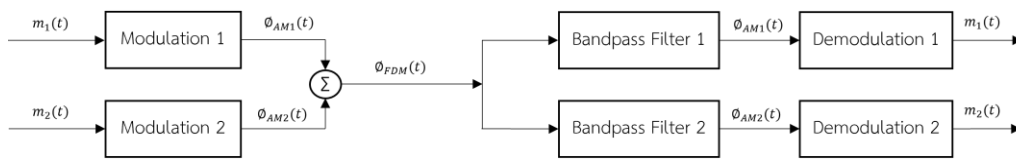
$$= \frac{A_c^2}{2} + \frac{A_c^2 \cos(2\omega_c t)}{2} + \frac{A_m A_c \cos(\omega_m t)}{2} + \frac{A_m A_c \cos(\omega_m t) \cos(2\omega_c t)}{2} \quad (2.11)$$

จากสมการที่ 2.11 คือผลจากการตีมอดูเลตที่ได้จากการคูณสัญญาณเอเอ็มกับสัญญาณคลื่นพาห์ ซึ่งจะเห็นว่าได้องค์ประกอบสัญญาณเกิดขึ้นทั้งหมด 4 ส่วน โดยเทอม $\frac{1}{2}A_m A_c \cos(\omega_m t)$ เป็นเทอมของสัญญาณข้อมูลข่าวสาร ดังนั้นเมื่อนำผลจากการตีมอดูเลตจากสมการที่ 2.11 มาผ่านตัวกรองความถี่ต่ำเทอมดังกล่าวจะได้สัญญาณข้อมูลข่าวสาร และสัญญาณเทอมที่เหลือซึ่งก็คือสัญญาณไฟตรง และสัญญาณความถี่สูงจะถูกกำจัดออกไป

2.10 การมัลติเพล็กซ์สัญญาณ (Multiplexing)

ในการสื่อสารจำเป็นต้องมีการส่งสัญญาณหลาย ๆ สัญญาณผ่านทางตัวกลางตัวเดียวกัน โดยตัวกลางนั้นจะมีความสามารถในการรับและแยกสัญญาณหลาย ๆ สัญญาณออกจากกันได้ ซึ่งการรวมของสัญญาณจะเรียกว่าการมัลติเพล็กซ์สัญญาณ และการแยกสัญญาณออกจากกันจะเรียกว่าการดีมัลติเพล็กซ์สัญญาณ (Demultiplexing) โดยในที่นี้จะขอกกล่าวถึงการมัลติเพล็กซ์ทางความถี่ (Frequency Division Multiplexing: FDM)

การมัลติเพล็กซ์ทางความถี่ คือการแบ่งช่วงความถี่ของช่องทางการสื่อสารออกเป็นช่องความถี่ย่อย ๆ (Sub Channel) โดยแต่ละสัญญาณจะผ่านการมอดูเลตกับสัญญาณคลื่นพาห์ที่มีความถี่ต่างกัน เมื่อทำการมัลติเพล็กซ์ทางความถี่จะเรียกสัญญาณที่ผ่านการมัลติเพล็กซ์ทางความถี่นี้ว่าสัญญาณเอฟดีเอ็ม สัญญาณเอฟดีเอ็มจะถูกส่งผ่านช่องทางการสื่อสารหรือตัวกลางเดียวกัน โดยภายในช่องทางการสื่อสารจะมีการ์ดแบนด์ (Guard Band) ที่ทำหน้าที่ป้องกันไม่ให้สัญญาณแต่ละความถี่เกิดการแทรกแซงสัญญาณระหว่างกัน และในส่วนการดีมัลติเพล็กซ์สัญญาณ จะนำสัญญาณเอฟดีเอ็มไปทำการแยกสัญญาณแต่ละความถี่ออกจากกันด้วยการนำสัญญาณไปผ่านตัวกรองความถี่แถบผ่าน และทำการตีมอดูเลตสัญญาณตามลำดับ โดยสามารถแสดงการมัลติเพล็กซ์และดีมัลติเพล็กซ์ทางความถี่ของสัญญาณเอเอ็มได้ดังรูปที่ 2.14



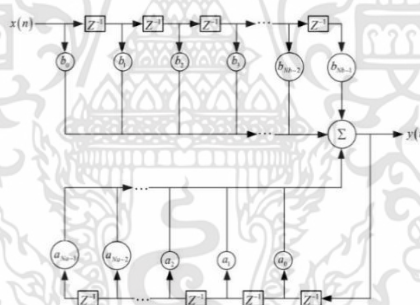
รูปที่ 2.14 การมัลติเพล็กซ์และดีมัลติเพล็กซ์ทางความถี่ของสัญญาณเอเอ็ม

2.11 ตัวกรองสัญญาณเชิงเลข (Digital Filter)

ตัวกรองสัญญาณเชิงเลข คือตัวกรองที่อาศัยกระบวนการทางคณิตศาสตร์มาใช้ในการวิเคราะห์ และประมวลผลสัญญาณ โดยในที่นี้จะขอกกล่าวถึงตัวกรองสัญญาณเชิงเลขระบบแบบผลตอบสนองอิมพัลส์ไม่จำกัด (Infinite Impulse Response System: IIR System)

2.11.1 ระบบแบบผลตอบสนองอิมพัลส์ไม่จำกัด

สามารถอธิบายโครงสร้างระบบแบบผลตอบสนองอิมพัลส์ไม่จำกัดได้ดังรูปที่ 2.15



รูปที่ 2.15 โครงสร้างระบบแบบผลตอบสนองอิมพัลส์ไม่จำกัด [11]

จากโครงสร้างระบบแบบไอโออาร์สามารถแสดงความสัมพันธ์ในทางคณิตศาสตร์ของสัญญาณขาเข้า และสัญญาณขาออกได้ดังสมการที่ 2.12

$$y(n) = \sum_{i=0}^{N_b-1} b_i x(n-i) - \sum_{j=1}^{N_a-1} a_j y(n-j) \tag{2.12}$$

โดยที่ $x(n)$ คือ สัญญาณขาเข้า

$y(n)$ คือ สัญญาณขาออก

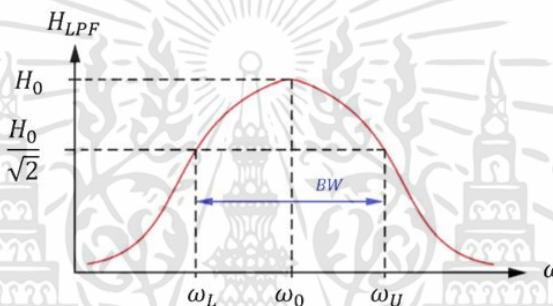
a_j, b_i คือ ค่าสัมประสิทธิ์ของระบบไอโออาร์

N_a, N_b คือ อันดับของระบบไอโออาร์

จะเห็นว่าสัญญาณขาออก $y(n)$ เกิดจากผลรวมของสัญญาณขาเข้าก่อนหน้า $x(n-i)$ คูณกับค่าสัมประสิทธิ์ของระบบไอโออาร์ b_i โดยที่ $i = 0, 1, \dots, N_b-1$ รวมกับผลรวมของสัญญาณขาออกก่อนหน้า $y(n-j)$ คูณกับค่าสัมประสิทธิ์ของระบบไอโออาร์ a_j โดยที่ $j = 1, 2, \dots, N_a-1$

2.11.2 วงจรกรองความถี่แถบผ่าน (Band Pass Filter: BPF)

วงจรกรองความถี่แถบผ่าน คือ วงจรที่ยอมให้สัญญาณที่มีความถี่ในช่วงความถี่ตัดสามารถผ่านไปได้ และหากสัญญาณนั้นมีช่วงความถี่มากกว่าหรือน้อยกว่าความถี่ตัดสัญญาณจะถูกลดทอนลงจนไม่สามารถผ่านไปได้ โดยสามารถเขียนกราฟแสดงฟังก์ชันการส่งผ่านกับความถี่ของวงจรกรองความถี่แถบผ่านได้ดังรูปที่ 2.16



รูปที่ 2.16 กราฟแสดงฟังก์ชันการส่งผ่านกับความถี่ของวงจรกรองความถี่แถบผ่าน [13]

จากรูปที่ 2.15 จะเห็นว่าจุดที่มีการตอบสนองสูงที่สุดจะอยู่ในช่วงความถี่ตัด (Cutoff Frequency: ω_0) ส่วนจุดที่มีการตอบสนองลดลงมา $\frac{H_0}{\sqrt{2}}$ จะอยู่ในช่วงต่ำกว่าความถี่ตัด (Lower Cutoff Frequency: ω_L) และในช่วงสูงกว่าความถี่ตัด (Upper Cutoff Frequency: ω_U) ซึ่งผลต่างระหว่าง ω_U และ ω_L เรียกว่า แบนด์วิดท์ (Bandwidth: BW) นอกจากนี้อัตราส่วนระหว่างความถี่ ω_0 กับแบนด์วิดท์ เรียกว่า ตัวประกอบคุณภาพ (Quality Factor: Q) โดยสามารถแสดงฟังก์ชันการส่งผ่านของวงจรกรองความถี่แถบผ่านอันดับสองได้ดังสมการที่ 2.13

$$H_{BPF}(s) = \frac{\frac{H_0 \omega_0}{Q} s}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2} \quad (2.13)$$

โดยที่ $H_{BPF}(s)$ คือ ฟังก์ชันการส่งผ่านของวงจรกรองความถี่แถบผ่าน

ω_0 คือ ความถี่ตัดเชิงมุม

Q คือ ตัวประกอบคุณภาพ

จากฟังก์ชันการส่งผ่านของวงจรความถี่แถบผ่านสามารถนำมาออกแบบสมการ
วงจรความถี่แถบผ่านอันดับสองได้ดังนี้

$$H_{\text{BPF}}(s) = \frac{\frac{H_0 \omega_0}{Q} s}{\left(s + \frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) \left(s + \frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)} \quad (2.14)$$

$$\begin{aligned} H_{\text{BPF}}(z) &= \frac{\frac{H_0 \omega_0}{Q} (1-z^{-1})}{\left(1 - e^{-\left(\frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) T_s} z^{-1}\right) \left(1 - e^{-\left(\frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) T_s} z^{-1}\right)} \\ &= \frac{\frac{H_0 \omega_0}{Q} (1-z^{-1})}{1 - e^{-\left(\frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) T_s} z^{-1}} - e^{-\left(\frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) T_s} z^{-1} + e^{-\frac{\omega_0}{Q} T_s} z^{-2}} \\ &= \frac{\frac{H_0 \omega_0}{Q} (1-z^{-1})}{1 - e^{-\frac{\omega_0}{2Q} T_s} z^{-1} 2 \left(\frac{e^{\frac{j\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}} + e^{-\frac{j\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}}}{2} \right) + e^{-\frac{\omega_0}{Q} T_s} z^{-2}} \\ &= \frac{\frac{H_0 \omega_0}{Q} (1-z^{-1})}{1 - e^{-\frac{\omega_0}{2Q} T_s} z^{-1} 2 \cos\left(\frac{\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}\right) + e^{-\frac{\omega_0}{Q} T_s} z^{-2}} \end{aligned} \quad (2.15)$$

$$\frac{y(z)}{x(z)} = \frac{\frac{H_0 \omega_0}{Q} (1-z^{-1})}{1 - e^{-\frac{\omega_0}{2Q} T_s} z^{-1} 2 \cos\left(\frac{\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}\right) + e^{-\frac{\omega_0}{Q} T_s} z^{-2}} \quad (2.16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

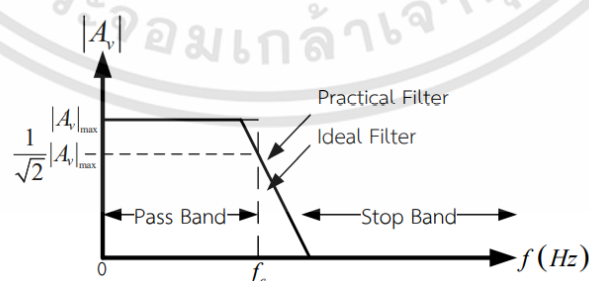
$$y(z) = x(z) \frac{H_0 \omega_0}{Q} (1-z^{-1}) + y(z) e^{-\frac{\omega_0 T_s}{2Q}} z^{-1} 2 \cos\left(\frac{\omega_0 T_s \sqrt{4 - \left(\frac{1}{Q}\right)^2}}{2}\right) - y(z) e^{-\frac{\omega_0 T_s}{Q}} z^{-2} \quad (2.17)$$

$$y(n) = \frac{H_0 \omega_0}{Q} (x(n) - x(n-1)) + 2e^{-\frac{\omega_0 T_s}{2Q}} \cos\left(\frac{\omega_0 T_s \sqrt{4 - \left(\frac{1}{Q}\right)^2}}{2}\right) y(n-1) - e^{-\frac{\omega_0 T_s}{Q}} y(n-2) \quad (2.18)$$

โดยที่ $x(n)$ คือ สัญญาณขาเข้าวงจรกรองความถี่แถบผ่าน
 $y(n)$ คือ สัญญาณขาออกวงจรกรองความถี่แถบผ่าน

2.11.3 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter: LPF)

วงจรกรองความถี่ต่ำผ่าน คือวงจรที่ยอมให้สัญญาณที่มีความถี่น้อยกว่าหรือเท่ากับความถี่ตัดสามารถผ่านไปได้ และหากสัญญาณนั้นมีความถี่มากกว่าความถี่ตัดสัญญาณจะถูกลดทอนไม่สามารถผ่านไปได้ โดยสามารถแสดงผลการตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่านได้ดังรูปที่ 2.17



รูปที่ 2.17 ผลการตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน [13]

และสามารถแสดงฟังก์ชันการส่งผ่านของวงจรกรองความถี่ต่ำผ่านอันดับสองได้ดังสมการที่ 2.19

$$H_{LPF}(s) = \frac{H_0 \omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \quad (2.19)$$

โดยที่ $H_{LPF}(s)$ คือ ฟังก์ชันการส่งผ่านของวงจรกรองความถี่ต่ำผ่าน

ω_0 คือ ความถี่ตัดเชิงมุม

Q คือ ตัวประกอบคุณภาพ

จากฟังก์ชันการส่งผ่านของวงจรกรองความถี่ต่ำผ่านสามารถนำมาออกแบบสมการวงจรกรองความถี่ต่ำผ่านอันดับสองได้ดังนี้

$$H_{LPF}(s) = \frac{H_0 \omega_0^2}{\left(s + \frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right) \left(s + \frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)} \quad (2.20)$$

$$\begin{aligned} H_{LPF}(z) &= \frac{H_0 \omega_0^2}{\left(1 - e^{-\left(\frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)T_s} z^{-1}\right) \left(1 - e^{-\left(\frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)T_s} z^{-1}\right)} \\ &= \frac{H_0 \omega_0^2}{1 - e^{-\left(\frac{\omega_0}{Q} + \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)T_s} z^{-1} - e^{-\left(\frac{\omega_0}{Q} - \sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_0^2}\right)T_s} z^{-1} + e^{-\frac{\omega_0}{Q}T_s} z^{-2}} \end{aligned} \quad (2.21)$$

$$= \frac{H_0 \omega_0^2}{1 - e^{-\frac{\omega_0}{2Q}T_s} z^{-1} 2 \left(\frac{e^{\frac{j\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}} + e^{-\frac{j\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}}}{2} \right) + e^{-\frac{\omega_0}{Q}T_s} z^{-2}}$$

$$\frac{y(z)}{x(z)} = \frac{H_0 \omega_0^2}{1 - e^{-\frac{\omega_0}{2Q}T_s} z^{-1} 2 \cos\left(\frac{\omega_0 T_s}{2} \sqrt{4 - \left(\frac{1}{Q}\right)^2}\right) + e^{-\frac{\omega_0}{Q}T_s} z^{-2}} \quad (2.22)$$

$$y(z) = \frac{x(z)H_0\omega_0^2 + y(z)e^{-\frac{\omega_0 T_s}{2Q}} z^{-1} 2\cos\left(\frac{\omega_0 T_s \sqrt{4 - \left(\frac{1}{Q}\right)^2}}{2}\right) - y(z)e^{-\frac{\omega_0 T_s}{Q}} z^{-2}}{1} \quad (2.23)$$

$$y(n) = \frac{H_0\omega_0^2 x(n) + 2e^{-\frac{\omega_0 T_s}{2Q}} \cos\left(\frac{\omega_0 T_s \sqrt{4 - \left(\frac{1}{Q}\right)^2}}{2}\right) y(n-1) - e^{-\frac{\omega_0 T_s}{Q}} y(n-2)}{1} \quad (2.24)$$

โดยที่ $x(n)$ คือ สัญญาณขาเข้าวงจรรองความถี่ต่ำผ่าน
 $y(n)$ คือ สัญญาณขาออกวงจรรองความถี่ต่ำผ่าน

2.12 ประสิทธิภาพการทำงานของเครื่องมือวัด

เครื่องมือวัดต่าง ๆ ที่ใช้ในการเก็บข้อมูลนั้นมีความสำคัญมากต่อความน่าเชื่อถือของข้อมูล ดังนั้นเครื่องมือวัดจะต้องมีการตรวจสอบคุณภาพ และประสิทธิภาพการทำงานว่ามีมากพอหรือไม่ที่จะรวบรวมข้อมูลให้มีความเที่ยงตรง และมีความเชื่อถือได้ ซึ่งเครื่องมือวัดที่ดีจะมีคุณสมบัติมากมาย แต่ในที่นี้ขอกล่าวถึงเพียง 2 คุณสมบัติ ได้แก่

2.12.1 ความเที่ยงตรง (Accuracy)

ความเที่ยงตรง คือ ค่าที่ได้จากเครื่องมือวัดนั้น จะมีค่าเข้าใกล้ค่าที่ถูกต้อง ซึ่งสามารถคำนวณหาค่าความเที่ยงตรงได้จากสมการที่ 2.25

$$\%Error = \left| \frac{\text{Measure} - \text{Standard}}{\text{Standard}} \right| \times 100$$

$$\%Accuracy = 100 - \%Error \quad (2.25)$$

โดยที่ Accuracy คือ ค่าความเที่ยงตรง
 Error คือ ค่าความผิดพลาด
 Measure คือ ค่าที่วัดได้จากเครื่องมือ
 Standard คือ ค่าที่ถูกต้อง

2.12.2 ความแม่นยำ (Precision)

ความแม่นยำ คือ ค่าที่ได้จากเครื่องมือวัดนั้น เมื่อทำการวัดหลาย ๆ รอบจะได้ค่าใกล้เคียงกัน ซึ่งสามารถคำนวณหาค่าความแม่นยำได้จากสมการที่ 2.26

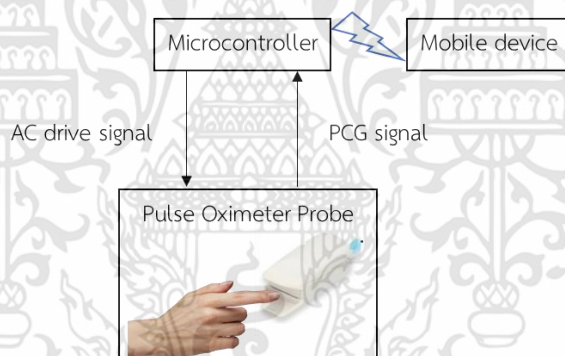
$$\text{Precision} = \mu \pm \frac{\sum |x - \mu|}{n} \quad (2.26)$$

โดยที่ x คือ ค่าที่ได้จากเครื่องมือวัดในแต่ละครั้ง
 μ คือ ค่าเฉลี่ยที่ได้จากการวัด
 n คือ จำนวนครั้งที่ทำการวัด

บทที่ 3

การออกแบบและการจัดทำปริญญาานิพนธ์

ในบทนี้จะกล่าวถึงการวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ เมื่อนำหัววัดความอิ่มตัวของออกซิเจนในเลือดมาหนีบที่ปลายนิ้วมือ ที่ด้านบนของหัววัดความอิ่มตัวของออกซิเจนในเลือดจะปล่อยคลื่นแสงที่มีความถี่แตกต่างกันสองคลื่น คือ คลื่นสีแดงสำหรับวัดออกซิฮีโมโกลบิน และคลื่นอินฟราเรดสำหรับวัดดีออกซีฮีโมโกลบิน โดยทั้งสองคลื่นนี้จะส่องผ่านทะลุนิ้วมือในเวลาเดียวกันไปยังตัวรับแสงที่อยู่ด้านล่างของหัววัดความอิ่มตัวของออกซิเจนในเลือด เพื่อวัดปริมาณของฮีโมโกลบินทั้งสองในเลือด และนำข้อมูลที่ได้มาคำนวณหาปริมาณร้อยละของออกซิเจนในเลือด โดยการประมวลผลทั้งหมดถูกกระทำบนไมโครคอนโทรลเลอร์ ESP32 จากนั้นส่งข้อมูลที่ได้ผ่านบลูทูธไปแสดงผลที่หน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ แสดงแผนภาพการทำงานดังรูปที่ 3.1



รูปที่ 3.1 แผนภาพการทำงาน

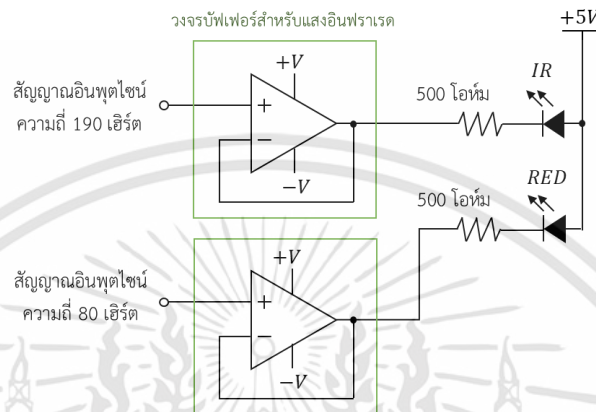
3.1 การออกแบบ

3.1.1 การออกแบบและสร้างวงจรจับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิ่มตัวของออกซิเจนในเลือด

3.1.1.1 การออกแบบวงจรจับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด

การออกแบบวงจรจับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด เราเลือกใช้วงจรบัฟเฟอร์สำหรับเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ ESP32 ที่จะเป็นตัวสร้างสัญญาณอินพุต

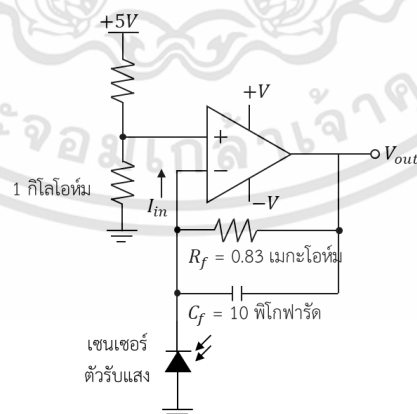
ให้กับวงจรขับตัวให้กำเนิดแสง เพื่อให้แรงดันไฟฟ้ามีค่าคงที่ และมีการต่อตัวต้านทานขนาด 500 โอห์ม เพื่อควบคุมกระแสไฟฟ้าไม่ให้ตัวกำเนิดแสงสีแดงและแสงอินฟราเรดได้รับกระแสไฟฟ้ามากเกินไป แสดงวงจรขับตัวให้กำเนิดแสงสำหรับแสงสีแดงและแสงอินฟราเรดได้ดังรูปที่ 3.2



รูปที่ 3.2 วงจรขับตัวให้กำเนิดแสงสีแดง และแสงอินฟราเรด

3.1.1.2 การออกแบบวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือด

เนื่องจากแสงที่ได้รับจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือดนั้นจะทำให้ความเข้มแสงถูกเปลี่ยนเป็นกระแสไฟฟ้า เราจึงเลือกใช้วงจรขยายทรานซิสเตอร์สำหรับนำมาขยายกระแสไฟฟ้าที่ได้รับจากเซนเซอร์ตัวรับแสงให้เป็นแรงดันไฟฟ้าแสดงได้ดังรูปที่ 3.3 โดยจะมีการไบอัสกลับตัวเซนเซอร์ตัวรับแสงด้วยแรงดันไฟฟ้าขนาด 500 มิลลิโวลต์ ที่ขาอินพุตบวกของออปแอมป์ และทำการจ่ายไฟเลี้ยง 5 โวลต์ ให้กับออปแอมป์



รูปที่ 3.3 วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือด

หาอัตราขยายของวงจรถยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัด ความอืดตัวออกซิเจนในเลือด ได้ดังสมการที่ 3.1

$$V_{out} = -R_f \times I_{in} \quad (3.1)$$

$$Gain = \frac{V_{out}}{I_{in}} = -R_f$$

$$Gain = -0.83 \times 10^6 \text{ โวลต์ต่อแอมแปร์} \quad (3.2)$$

และคำนวณหาค่าความถี่ตัดได้จากสมการที่ 3.3

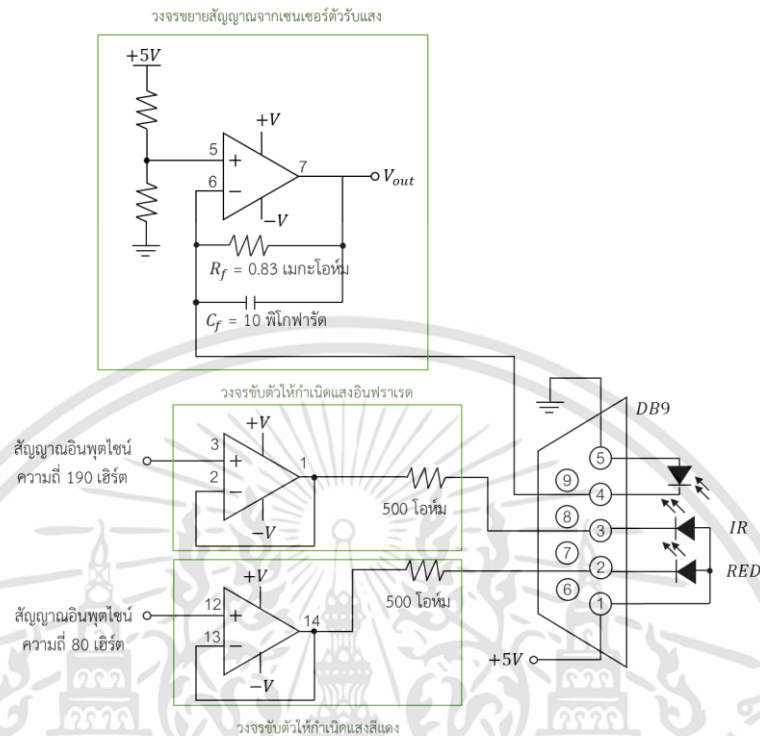
$$f_c = \frac{1}{2\pi R_f C_f}$$

$$f_c = \frac{1}{2\pi(0.83 \times 10^6)(10 \times 10^{-12})}$$

$$f_c = 19.175 \text{ กิโลเฮิร์ต} \quad (3.3)$$

3.1.1.3 การนำวงจรถับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรถยายสัญญาณจากเซนเซอร์ตัวรับแสงรวมเข้ากับหัววัดความอืดตัวออกซิเจนในเลือด

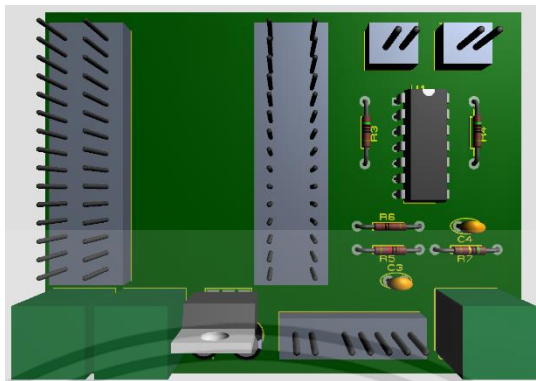
การรวมวงจรถับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรถยายสัญญาณจากเซนเซอร์ตัวรับแสงเข้ากับหัววัดความอืดตัวออกซิเจนในเลือดนั้น จะเลือกใช้หัวเชื่อมต่อ DB-9 สำหรับเชื่อมต่อวงจรถับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรดกับตัวให้กำเนิดแสง (ฝั่งส่ง) และเชื่อมต่อวงจรถยายสัญญาณจากตัวเซนเซอร์รับแสงกับเซนเซอร์ตัวรับแสง (ฝั่งรับ) ที่อยู่ภายในของหัววัดความอืดตัวออกซิเจนในเลือดได้ดังรูปที่ 3.4 และเราจะเลือกใช้ไอซี LM324 ซึ่งเป็นไอซีออปแอมป์ที่ใช้ไฟเลี้ยงเดียว และภายในของตัวไอซีนี้มีถึง 4 ออปแอมป์มาใช้ในการสร้างวงจรถับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรถยายสัญญาณจากเซนเซอร์ตัวรับแสง



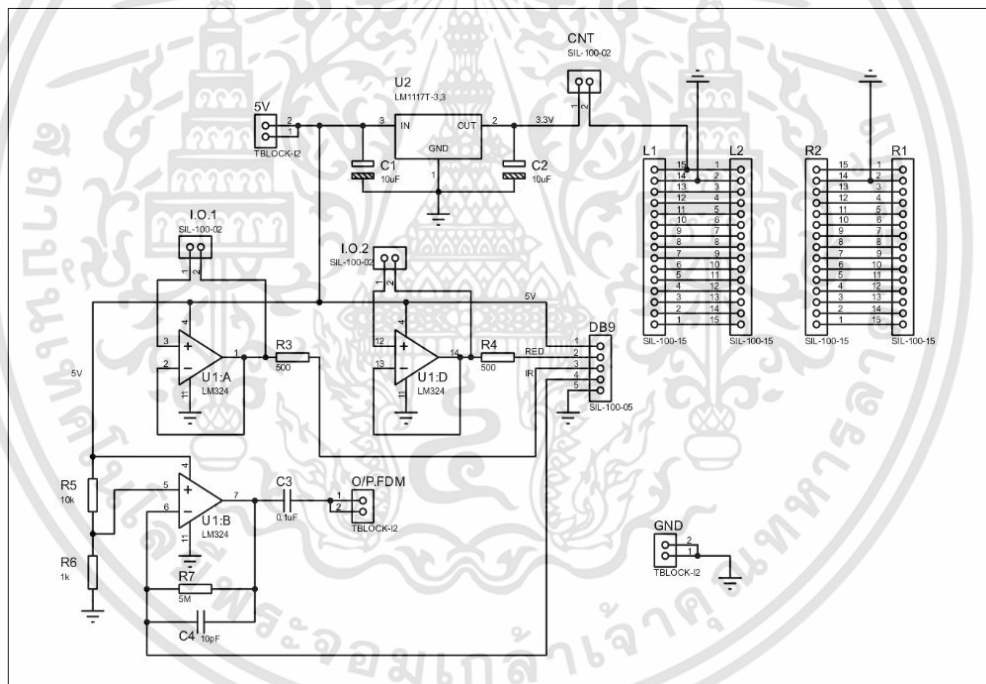
รูปที่ 3.4 วงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงรวมเข้ากับหัววัดความอิมิต์วออกซิเจนในเลือด

3.1.1.4 การสร้างวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงรวมเข้ากับหัววัดความอิมิต์วออกซิเจนในเลือด

นำวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิต์วออกซิเจนในเลือดที่ออกแบบได้ตั้งหัวข้อ 3.1.1.3 มาทำการออกแบบในโปรแกรม Proteus และทำการเพิ่มฐานรองไมโครคอนโทรลเลอร์ ESP32 โดยมีรูป PCB Layout ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิต์วออกซิเจนในเลือดดังรูปที่ 3.5 และมีรูป Schematic ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิต์วออกซิเจนในเลือดดังรูปที่ 3.6 แล้วนำลายวงจรพิมพ์ (Print Circuit Board: PCB) ที่ได้ไปทำการกัดปริ้นบนแผ่นทองแดง ซึ่งมีลายวงจรพิมพ์ดังรูปที่ 3.7 จากนั้นใส่อุปกรณ์และบัดกรีเมื่อทำตามขั้นตอนดังกล่าวเสร็จสิ้นแล้วจะได้วงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิต์วออกซิเจนในเลือดดังรูปที่ 3.8

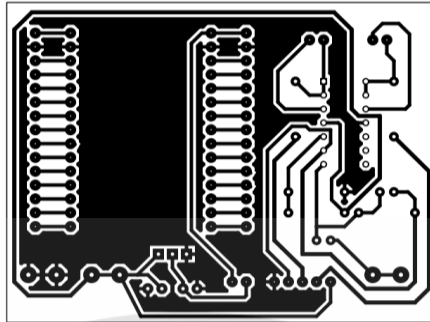


รูปที่ 3.5 PCB Layout ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือด

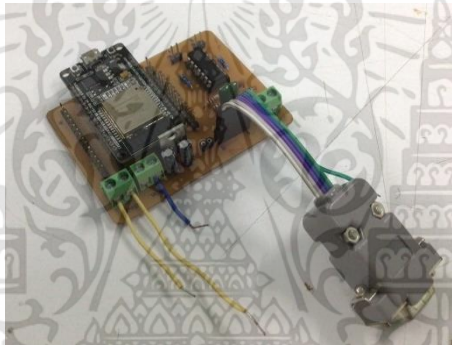


รูปที่ 3.6 Schematic ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ลายวงจรพิมพ์ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือด



รูปที่ 3.8 วงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิตัวออกซิเจนในเลือดที่สร้างเสร็จแล้ว

3.1.2 การออกแบบและเขียนโปรแกรมสร้างและเก็บสัญญาณไซน์ความถี่ 80 และ 190 เฮิร์ต บนไมโครคอนโทรลเลอร์ ESP32

เขียนโปรแกรม Arduino IDE สร้างและเก็บสัญญาณไซน์ สามารถอธิบายขั้นตอนการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32 ได้ดังรูปที่ 3.9 โดยโปรแกรมจะเริ่มจากกำหนดเงื่อนไขตัวแปร n ให้เริ่มต้นที่ 1 ไปจนถึง 8500 แล้วทำการตรวจสอบค่าตัวแปร n จากนั้นนำค่าตัวแปร n ที่ได้ไปแทนค่าในสมการสร้างสัญญาณไซน์จากสมการที่ 3.4

$$Car = 128 + (120 \sin(n\theta)) \quad (3.4)$$

โดยที่ Car คือ สัญญาณสัญญาณไซน์

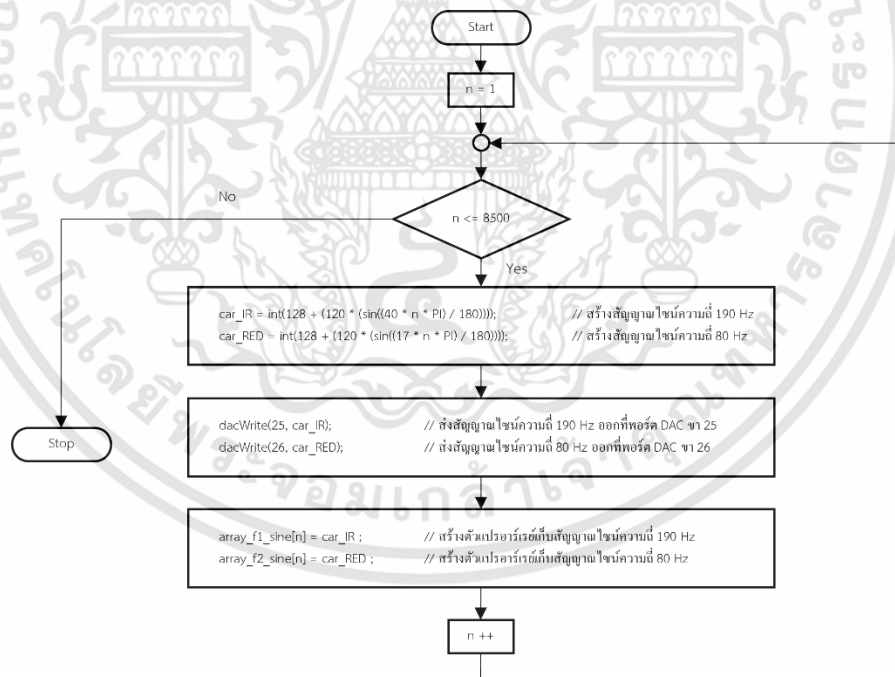
ออกแบบสัญญาณไซน์ของคลื่นแสงสีแดง โดยกำหนดให้ $\theta = \frac{17\pi}{180}$ สามารถแสดงได้ดังข้างล่างนี้

$$\text{Car}_R = 128 + \left(120 \sin \left(\frac{17\pi n}{180} \right) \right) \quad (3.5)$$

จะได้ค่าสัญญาณไซน์ความถี่ 80 เฮิร์ต และออกแบบสัญญาณไซน์ของคลื่นอินฟราเรด โดยกำหนดให้ $\theta = \frac{40\pi}{180}$ สามารถแสดงได้ดังข้างล่างนี้

$$\text{Car}_{IR} = 128 + \left(120 \sin \left(\frac{40\pi n}{180} \right) \right) \quad (3.6)$$

จะได้ค่าสัญญาณไซน์ความถี่ 190 เฮิร์ต แล้วนำค่าสัญญาณไซน์ที่เป็นตัวเลขจากสมการที่ 3.5 และ 3.6 ไปแปลงเป็นสัญญาณแอนะล็อกส่งออกทางพอร์ต DAC ที่ขา 26 และ ขา 25 ของไมโครคอนโทรลเลอร์ ESP32 ตามลำดับ จากนั้นสร้างตัวแปรอาร์เรย์จำนวน 2 ตัวแปร มาเก็บค่าสัญญาณไซน์ความถี่ 80 เฮิร์ต และ 190 เฮิร์ต เพื่อที่จะนำตัวแปรอาร์เรย์ดังกล่าวไปใช้ในการติ่มอดูเลตในส่วนการประมวลผลข้อมูลบนไมโครคอนโทรลเลอร์ ESP32 ต่อไป สามารถเขียนคำสั่งการทำงานได้ดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพการทำงานการสร้าง และเก็บสัญญาณไซน์ความถี่ 80 เฮิร์ต และ 190 เฮิร์ตบนไมโครคอนโทรลเลอร์ ESP32

```

for (n = 1; n <= loopnum; n++)
{
    car_IR = int(128 + (120 * (sin((freq1 * n * PI) / 180)))); // freq1 = 40
    car_RED = int(128 + (120 * (sin((freq2 * n * PI) / 180)))); // freq2 = 17

    digitalWrite(25, car_IR); // Sine wave1 190Hz
    digitalWrite(26, car_RED); // Sine wave2 80Hz

    array_f1_sine[n] = car_IR ;
    array_f2_sine[n] = car_RED ;

    array_l[n] = analogRead(sensor) / 5 ;

    yRED_MIN = 999999999 ;
    yRED_MAX = 0 ;
    yIR_MIN = 999999999 ;
    yIR_MAX = 0 ;

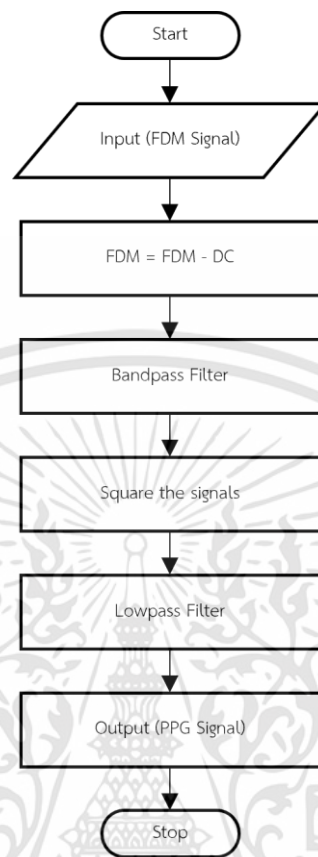
    digitalWrite(countpin, !digitalRead(countpin));
    delayMicroseconds(490); // fs = 1.7kHz
}

```

รูปที่ 3.10 คำสั่งในการสร้างและเก็บสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต

3.1.3 การออกแบบและเขียนโปรแกรมสำหรับประมวลผลข้อมูลใน MATLAB

ในส่วนด้านรับสัญญาณจำเป็นต้องทราบว่าสัญญาณที่ได้รับมา มีความถูกต้องหรือไม่ ดังนั้นจึงมีการใช้โปรแกรม MATLAB เพื่อทำการประมวลผลสัญญาณที่รับมาได้ สามารถอธิบายการทำงานของโปรแกรมได้ดังรูปที่ 3.11 โดยโปรแกรมจะเริ่มต้นจากรับสัญญาณเมื่อนำหัววัดความถี่ออกมาจากรีเลย์ในเลือดมาหนีบที่ปลายนิ้วมือ จากนั้นจะนำมาเทียบกับค่าเฉลี่ยของสัญญาณ เนื่องจากสัญญาณที่เราได้รับมานั้นมีค่าออฟเซตเพิ่มขึ้น ทำให้สัญญาณลอยขึ้นไม่สามารถนำไปทำการตีความดูเลตได้ ต่อไปนำสัญญาณที่ได้นี้ไปกรองผ่านตัวกรองแถบความถี่ผ่าน เพื่อแยกสัญญาณเอพดีเอ็มได้เป็นสัญญาณเอเอ็มของคลื่นแสงสีแดง (คลื่นพาห์ 80 เฮิรต) และสัญญาณเอเอ็มของคลื่นอินฟราเรด (คลื่นพาห์ 190 เฮิรต) จากนั้นทำการตีความดูเลตสัญญาณด้วยวิธีตรวจจับขอบ เพื่อให้ได้สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดที่สามารถนำไปคำนวณหาอัตราส่วนการดูดซับแสง และค่าความถี่ของออกซิเจนในเลือดต่อไปได้ สามารถเขียนเป็นคำสั่งการทำงานได้ดังรูปที่ 3.12



รูปที่ 3.11 แผนภาพการทำงานการประมวลผลในโปรแกรม MATLAB

```

clear all; close all; clc;

Fs=44000;
time = 10;
t = 1/Fs: 1/Fs: time;
bits = 16;
sig = audiorecorder(Fs,bits,1); % 1 channel
recordingblocking(sig, time); % 10 seconds
signal = getaudiodata(sig);
figure(1); plot(t, signal); ylim([-2 2]); grid on

FDM_SIGNAL = (signal - mean(signal)) * 1;

% Fs = Fs / downsampleRate ;
% t = 1/Fs: 1/Fs: duration;

figure(2);
subplot(311);
plot(t, FDM_SIGNAL); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('ORIGINAL FDM SIGNAL');

% Demodulation Processing
% Filter with 2 BPFs

F1pass = 395; % F1 Carrier
F2pass = 757; % F2 Carrier

guardband = 100;

% Band Pass Filter Design
[B1, A1] = butter(3, [(F1pass-guardband) (F1pass+guardband)]/(Fs/2), 'bandpass'); % 1070 Hz
[B2, A2] = butter(5, [(F2pass-guardband) (F2pass+guardband)]/(Fs/2), 'bandpass'); % 1550 Hz

AM_RED = filter(B1, A1, FDM_SIGNAL);
AM_IR = filter(B2, A2, FDM_SIGNAL);

subplot(312);
plot(t, AM_RED); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('RED AM Signal');
subplot(313);
plot(t, AM_IR); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('IR AM Signal');

% Envelope Detector
fullwave_RED = power(AM_RED, 2);
fullwave_IR = power(AM_IR, 2);

figure(3);
subplot(211);
plot(t, fullwave_RED); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('RED PPG Signal'); grid on;
subplot(212);
plot(t, fullwave_IR); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('IR PPG Signal'); grid on;

% Low Pass Filter Design
[B3, A3] = butter(5, 10/(Fs/2), 'low'); % 1070 Hz

RED = filter(B3, A3, fullwave_RED);
IR = filter(B3, A3, fullwave_IR);

RED = sqrt(RED);
IR = sqrt(IR);

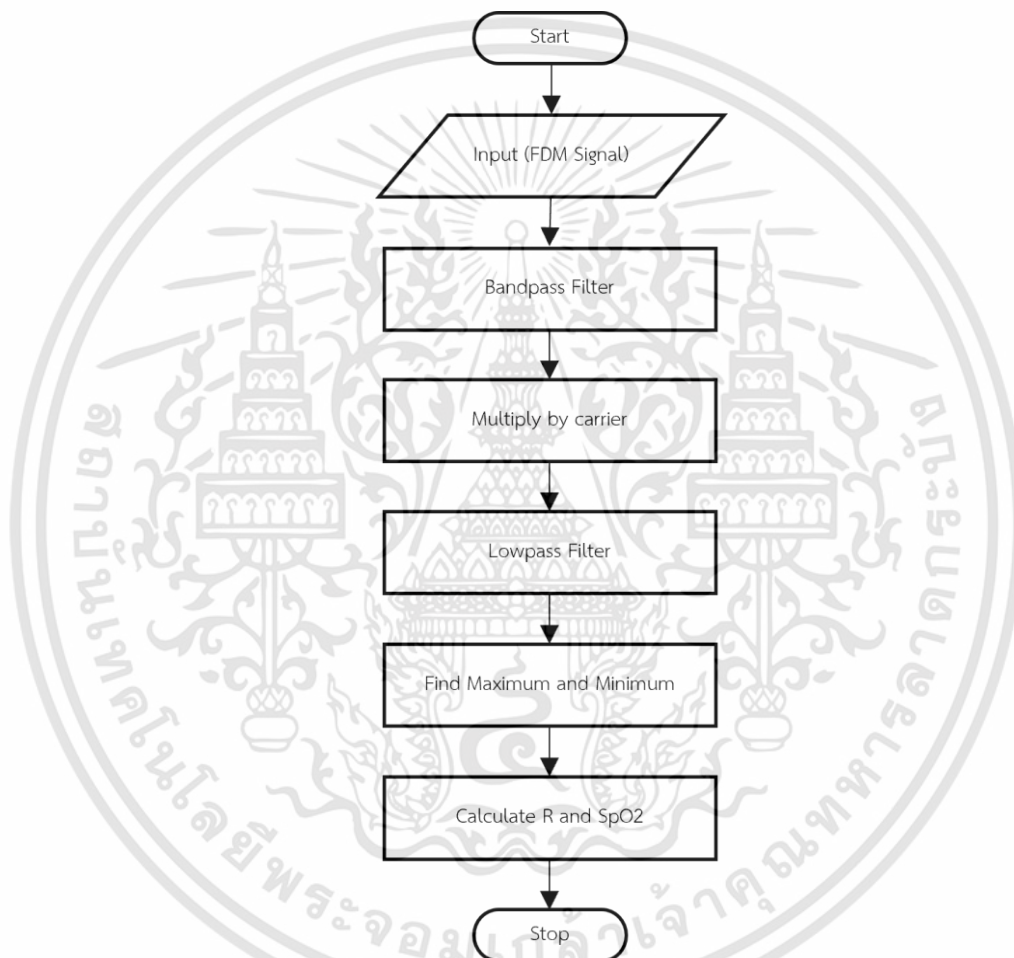
figure(4);
subplot(211);
plot(t, RED); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('RED PPG Signal'); grid on;
subplot(212);
plot(t, IR); xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('IR PPG Signal'); grid on;

```

รูปที่ 3.12 คำสั่งในการสร้างตัวกรองแถบความถี่ผ่าน, การตีมอดูเลต และตัวกรองความถี่ต่ำผ่านในโปรแกรม MATLAB

3.1.4 การออกแบบและเขียนโปรแกรมสำหรับประมวลผลข้อมูลบนไมโครคอนโทรลเลอร์ ESP32

เขียนโปรแกรม Arduino IDE ประมวลผลสัญญาณ และคำนวณหาค่าความอิ่มตัวของออกซิเจนในเลือด สามารถอธิบายขั้นตอนการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32 ได้ดังรูปที่ 3.13



รูปที่ 3.13 แผนภาพขั้นตอนการประมวลผลสัญญาณและคำนวณหาค่าความอิ่มตัวของออกซิเจนในเลือด

โดยโปรแกรมจะเริ่มจากรับสัญญาณเอฟดีเอ็ม สามารถแสดงความสัมพันธ์ทางคณิตศาสตร์ได้ดังสมการที่ 3.7

$$FDM(t) = PPG_R(t)\cos(\omega_1 t) + PPG_{IR}(t)\cos(\omega_2 t) \quad (3.7)$$

โดยที่ $PPG_R(t)$ คือ สัญญาณพีพีจีของคลื่นแสงสีแดง

$PPG_{IR}(t)$ คือ สัญญาณพีพีจีของคลื่นอินฟราเรด

$\cos(\omega_1 t)$ คือ สัญญาณคลื่นพาร์ห์ของคลื่นแสงสีแดง

$\cos(\omega_2 t)$ คือ สัญญาณคลื่นพาร์ห์ของคลื่นอินฟราเรด

เมื่อรับสัญญาณเอฟดีเอ็มที่ผ่านวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมตัวออกซิเจนในเลือด เข้ามาที่พอร์ต ADC ขา 34 ของไมโครคอนโทรลเลอร์ ESP32 แล้วนำสัญญาณเอฟดีเอ็มที่รับเข้ามาทำการผ่านตัวกรองความถี่แถบผ่านที่มีความถี่ศูนย์กลาง ω_1 จะได้สัญญาณเอเอ็มของคลื่นแสงสีแดง สามารถแสดงความสัมพันธ์ทางคณิตศาสตร์ได้ดังสมการที่ 3.8

$$AM_R(t) = PPG_R(t) \cos(\omega_1 t) \quad (3.8)$$

โดยที่ $AM_R(t)$ คือ สัญญาณเอเอ็มของคลื่นแสงสีแดง

และในทำนองเดียวกัน ถ้านำสัญญาณเอฟดีเอ็มที่รับเข้ามาทำการผ่านตัวกรองความถี่แถบผ่านที่มีความถี่ศูนย์กลาง ω_2 จะได้สัญญาณเอเอ็มของคลื่นอินฟราเรด สามารถแสดงความสัมพันธ์ทางคณิตศาสตร์ได้ดังสมการที่ 3.9

$$AM_{IR}(t) = PPG_{IR}(t) \cos(\omega_2 t) \quad (3.9)$$

โดยที่ $AM_{IR}(t)$ คือ สัญญาณเอเอ็มของคลื่นอินฟราเรด

โดยเราจะทำการออกแบบตัวกรองความถี่แถบผ่านได้จากสมการที่ 3.10

$$BPF(n) = B(x(n)-x(n-1)) + A_1 y(n-1) - A_2(n-2) \quad (3.10)$$

โดยที่ $A_1 = 2e^{-\frac{\omega_0 T_s}{2Q}} \cos\left(\frac{\omega_0 T_s \sqrt{4 - (1/Q)^2}}{2}\right)$

$$A_2 = -e^{-\frac{\omega_0 T_s}{Q}}$$

$$B = \frac{H_0\omega_0}{Q}$$

ในการออกแบบตัวกรองความถี่แถบผ่านของคลื่นแสงสีแดง กำหนดให้ $T_s = 588 \mu\text{sec}$, $Q = 2.67$, $H_0 = 2$ และ $\omega_0 = 160\pi$ สามารถแทนค่าสัมประสิทธิ์ A_1 , A_2 และ B ได้ดังสมการข้างล่าง

$$\begin{aligned} A_1 &= \frac{160\pi \times 588\mu}{2e^{-\frac{160\pi \times 588\mu}{2 \times 2.67}}} \cos\left(\frac{160\pi \times 588\mu \sqrt{4 - (1/2.67)^2}}{2}\right) \\ &= 1.8129 \end{aligned} \quad (3.11)$$

$$\begin{aligned} A_2 &= -e^{-\frac{160\pi \times 588\mu}{2.67}} \\ &= -0.8950 \end{aligned} \quad (3.12)$$

$$\begin{aligned} B &= \frac{2 \times 160\pi}{2.67} \\ &= 376.9911 \end{aligned} \quad (3.13)$$

$$\text{BPF}_R(n) = 376.9911(x(n)-x(n-1)) + 1.8129y(n-1) - 0.8950y(n-2) \quad (3.14)$$

และในการออกแบบตัวกรองความถี่แถบผ่านของคลื่นอินฟราเรด กำหนดให้ $T_s = 588 \mu\text{sec}$, $Q = 6.33$, $H_0 = 2$ และ $\omega_0 = 380\pi$ สามารถแทนค่าสัมประสิทธิ์ A_1 , A_2 และ B ได้ดังสมการข้างล่าง

$$\begin{aligned} A_1 &= \frac{380\pi \times 588\mu}{2e^{-\frac{380\pi \times 588\mu}{2 \times 6.33}}} \cos\left(\frac{380\pi \times 588\mu \sqrt{4 - (1/6.33)^2}}{2}\right) \\ &= 1.4471 \end{aligned} \quad (3.15)$$

$$\begin{aligned} A_2 &= -e^{-\frac{380\pi \times 588\mu}{6.33}} \\ &= -0.8950 \end{aligned} \quad (3.16)$$

$$\begin{aligned} B &= \frac{2 \times 380\pi}{6.33} \\ &= 376.9911 \end{aligned} \quad (3.17)$$

$$\text{BPF}_{IR}(n) = 376.9911(x(n)-x(n-1)) + 1.4471y(n-1) - 0.8950y(n-2) \quad (3.18)$$

จากนั้นนำสัญญาณสมการที่ 3.8 ไปทำการตีโมดดูเลตด้วยการคูณกับตัวแปรอาร์เรย์ที่เก็บค่าสัญญาณไซน์ที่มีความถี่ ω_1 จะได้ความสัมพันธ์ทางคณิตศาสตร์สามารถแสดงได้ดังนี้

$$\begin{aligned} AM_R(t) \times \cos(\omega_1 t) &= (PPG_R(t) \cos(\omega_1 t)) \times \cos(\omega_1 t) \\ &= PPG_R(t) \cos^2(\omega_1 t) \\ &= PPG_R(t) \left(\frac{1}{2} + \frac{1}{2} \cos(2\omega_1 t) \right) \\ &= \frac{PPG_R(t)}{2} + \frac{PPG_R(t) \cos(2\omega_1 t)}{2} \end{aligned} \quad (3.19)$$

จากสมการข้างต้นจะเห็นได้ว่ามีองค์ประกอบสัญญาณเกิดขึ้นทั้งหมด 2 ส่วน โดยเทอมที่ 1 ทางซ้ายมือคือสัญญาณพีพีจีที่เกิดจากคลื่นแสงสีแดง ส่วนอีกเทอมที่เหลือคือสัญญาณความถี่สูง ดังนั้นเมื่อนำสัญญาณที่เกิดจากการคูณดังกล่าวมาผ่านตัวกรองความถี่ต่ำผ่านจะได้ค่าเอาต์พุตเป็นพีพีจีของคลื่นแสงสีแดงเท่านั้น และในทำนองเดียวกัน ถ้านำ $\cos(\omega_2 t)$ ไปคูณสัญญาณสมการที่ 3.9 ก็จะได้ความสัมพันธ์ทางคณิตศาสตร์สามารถแสดงได้ดังนี้

$$\begin{aligned} AM_{IR}(t) \times \cos(\omega_2 t) &= (PPG_{IR}(t) \cos(\omega_2 t)) \times \cos(\omega_2 t) \\ &= PPG_{IR}(t) \cos^2(\omega_2 t) \\ &= PPG_{IR}(t) \left(\frac{1}{2} + \frac{1}{2} \cos(2\omega_2 t) \right) \\ &= \frac{PPG_{IR}(t)}{2} + \frac{PPG_{IR}(t) \cos(2\omega_2 t)}{2} \end{aligned} \quad (3.20)$$

ดังนั้นเมื่อนำสัญญาณสมการที่ 3.19 และ 3.20 ไปผ่านตัวกรองความถี่ต่ำผ่านจะได้สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรด ตามลำดับ โดยเราจะทำการออกแบบตัวกรองความถี่ต่ำผ่านได้จากสมการที่ 3.21

$$LPF(n) = Bx(n) + A_1 y(n-1) - A_2 y(n-2) \quad (3.21)$$

$$\text{โดยที่} \quad A_1 = 2e^{-\frac{\omega_0 T_s}{2Q}} \cos\left(\frac{\omega_0 T_s \sqrt{4 - (1/Q)^2}}{2}\right)$$

$$A_2 = -e^{-\frac{\omega_0 T_s}{Q}}$$

$$B = H_0 \omega_0^2$$

ออกแบบตัวกรองความถี่ต่ำผ่านของคลื่นแสงสีแดง และคลื่นอินฟราเรด โดยกำหนดให้ $T_s = 588 \mu\text{sec}$, $Q = 0.5$, $H_0 = 2$ และ $\omega_0 = 6\pi$ สามารถแทนค่าสัมประสิทธิ์ A_1, A_2 และ B ได้ดังสมการข้างล่าง

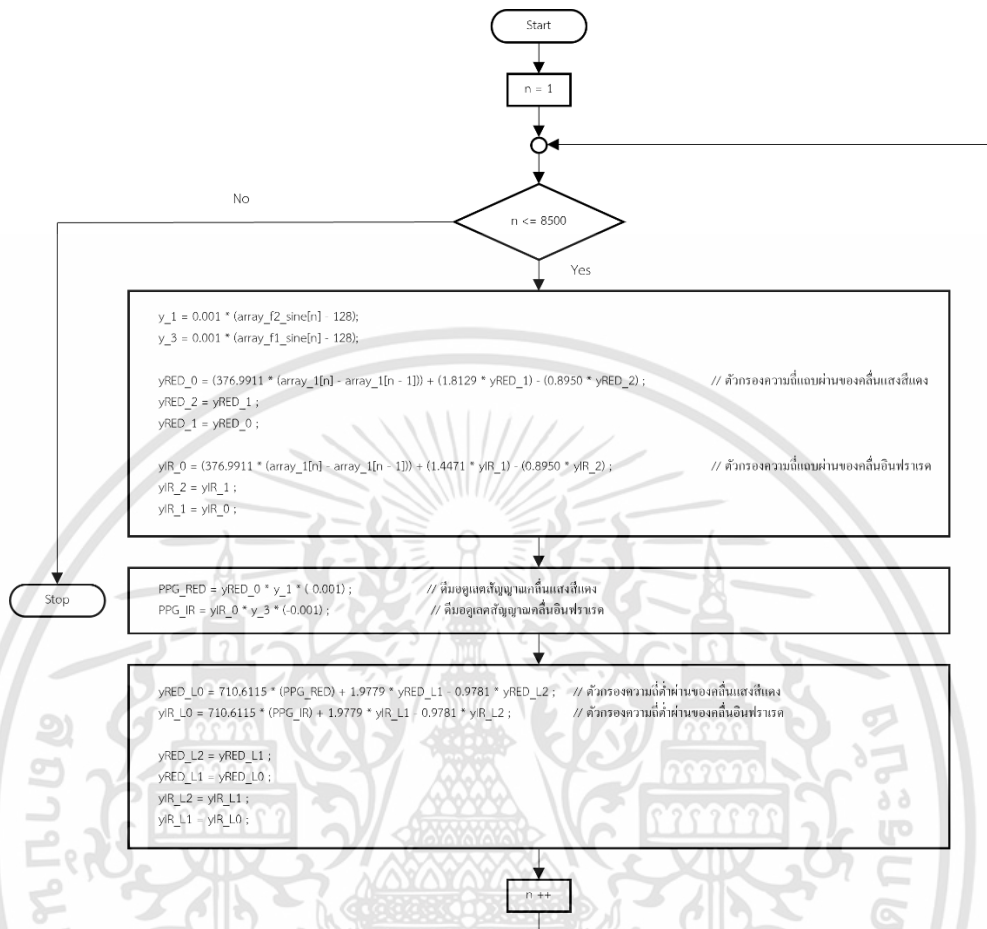
$$\begin{aligned} A_1 &= 2e^{-\frac{6\pi \times 588 \mu}{2 \times 0.5}} \cos\left(\frac{6\pi \times 588 \mu \sqrt{4 - (1/0.5)^2}}{2}\right) \\ &= 1.9779 \end{aligned} \quad (3.22)$$

$$\begin{aligned} A_2 &= -e^{-\frac{6\pi \times 588 \mu}{0.5}} \\ &= -0.9781 \end{aligned} \quad (3.23)$$

$$\begin{aligned} B &= 2 \times 36\pi^2 \\ &= 710.6115 \end{aligned} \quad (3.24)$$

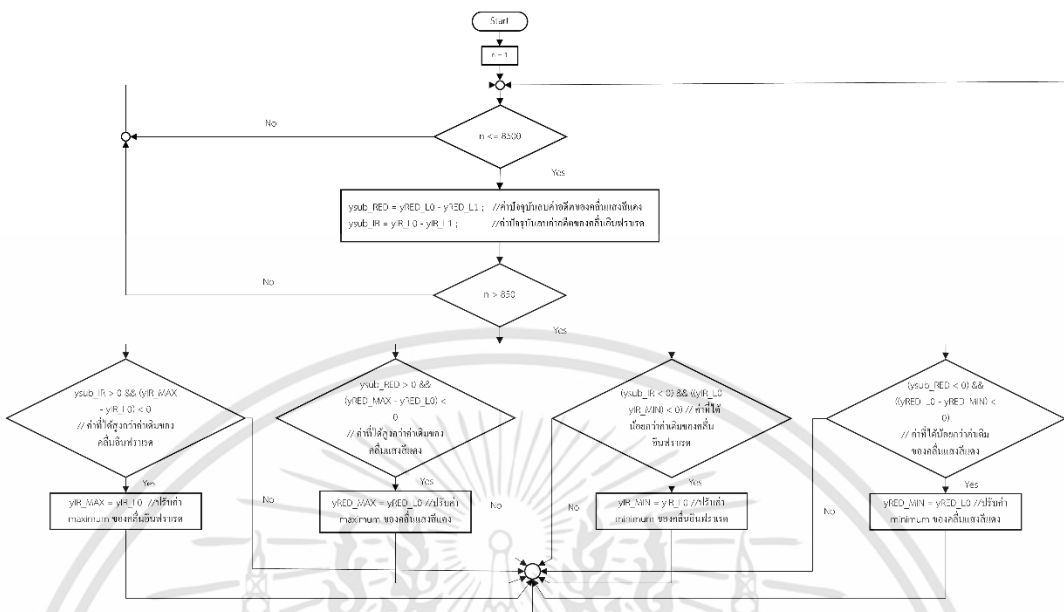
$$\text{LPF}(n) = 710.6115x(n) + 1.9779y(n-1) - 0.9781y(n-2) \quad (3.25)$$

จะได้สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดที่ผ่านตัวกรองความถี่ต่ำผ่านดังสมการที่ 3.25 โดยมีแผนภาพการทำงานของการดีมอดูเลตสัญญาณ แสดงได้ดังรูปที่ 3.14



รูปที่ 3.14 แผนภาพการทำงานของขั้นตอนการหาค่าสูงสุดและต่ำสุดของสัญญาณ

หลังจากนั้นทำการหาจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดโดย นำค่าในแต่ละช่วงไปทำการหาจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีโดยการกำหนดเงื่อนไขว่าถ้าค่าปัจจุบันที่รับเข้ามามีค่ามากกว่าค่าอดีต นำค่าปัจจุบันไปเทียบกับค่าสูงสุดที่เก็บเอาไว้ก่อนหน้านี้ ถ้าค่าปัจจุบันสูงกว่าค่าสูงสุดที่เก็บเอาไว้ให้ปรับค่าปัจจุบันเป็นค่าสูงสุดตัวใหม่ ซึ่งจะทำการหาค่าของสัญญาณพีพีจีของคลื่นแสงสีแดงและคลื่นอินฟราเรด ($yRED_{max}$, yIR_{max}) และในทำนองเดียวกัน ถ้าทำการเปลี่ยนเงื่อนไขให้ตรงกันข้ามจะได้ค่าต่ำสุดทั้งของสัญญาณพีพีจีคลื่นแสงสีแดงและคลื่นอินฟราเรด ($yRED_{min}$, yIR_{min}) ออกมา โดยมีแผนภาพการทำงานของการทำงานหาค่าสูงสุดและต่ำสุด แสดงได้ดังรูปที่ 3.15



รูปที่ 3.15 แผนภาพการหาค่าจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีทั้งสอง

หลังจากนั้นทำการหาอัตราส่วนการดูดซับแสง (R) ได้จากสมการที่ 3.26

$$R = \frac{\frac{AC_{RED}}{DC_{RED}}}{\frac{AC_{IR}}{DC_{IR}}} = \frac{(yRED_{max} - yRED_{min}) / ((yRED_{max} + yRED_{min}) / 2)}{(yIR_{max} - yIR_{min}) / ((yIR_{max} + yIR_{min}) / 2)} \tag{3.26}$$

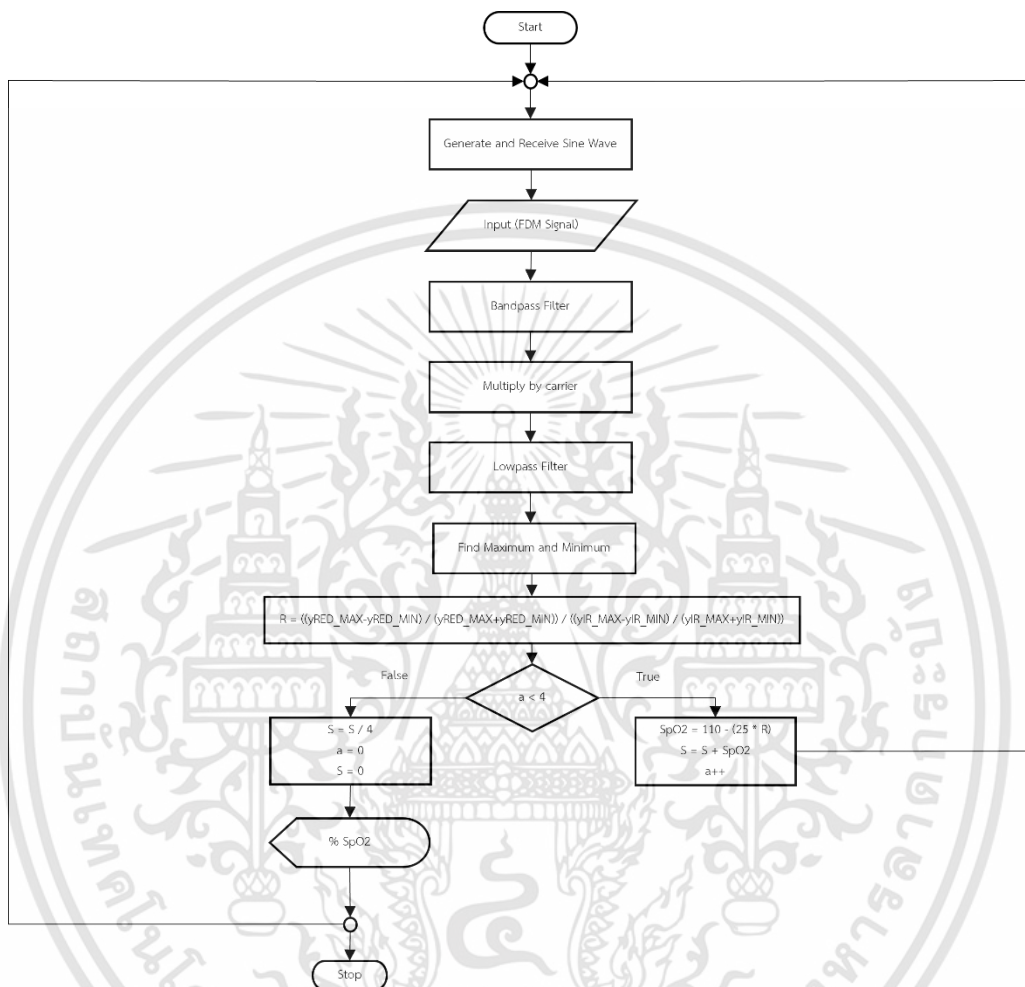
- โดยที่
- yRED_{max} คือ ค่าสูงสุดของสัญญาณพีพีจีของคลื่นแสงสีแดง
 - yRED_{min} คือ ค่าต่ำสุดของสัญญาณพีพีจีของคลื่นแสงสีแดง
 - yIR_{max} คือ ค่าสูงสุดของสัญญาณพีพีจีของคลื่นอินฟราเรด
 - yIR_{min} คือ ค่าต่ำสุดของสัญญาณพีพีจีของคลื่นอินฟราเรด

เมื่อทำการหาอัตราส่วนการดูดซับแสงเรียบร้อยแล้ว นำค่าอัตราส่วนการดูดซับแสงที่ได้มาคำนวณหาค่าความอิ่มตัวของออกซิเจนในเลือด (%SpO₂) สามารถหาค่าความอิ่มตัวของออกซิเจนในเลือดได้จากสมการที่ 3.27

$$\%SpO_2 = 110 - (25 \times R) \tag{3.27}$$

โดยจะกำหนดให้โปรแกรมทำการวนซ้ำตั้งแต่ต้นเป็นจำนวนทั้งหมด 4 รอบ เพื่อนำมาหาค่าเฉลี่ยของค่าความอิ่มตัวของออกซิเจนในเลือด จากนั้นทำการแสดงผลค่าความอิ่มตัวของ

ออกซิเจนในเลือดเป็นการสิ้นสุดการประมวลผลข้อมูลบนไมโครคอนโทรลเลอร์ ESP32 โดยสามารถแสดงแผนภาพการหาค่าความอิ่มตัวของออกซิเจนในเลือดได้ดังรูปที่ 3.16

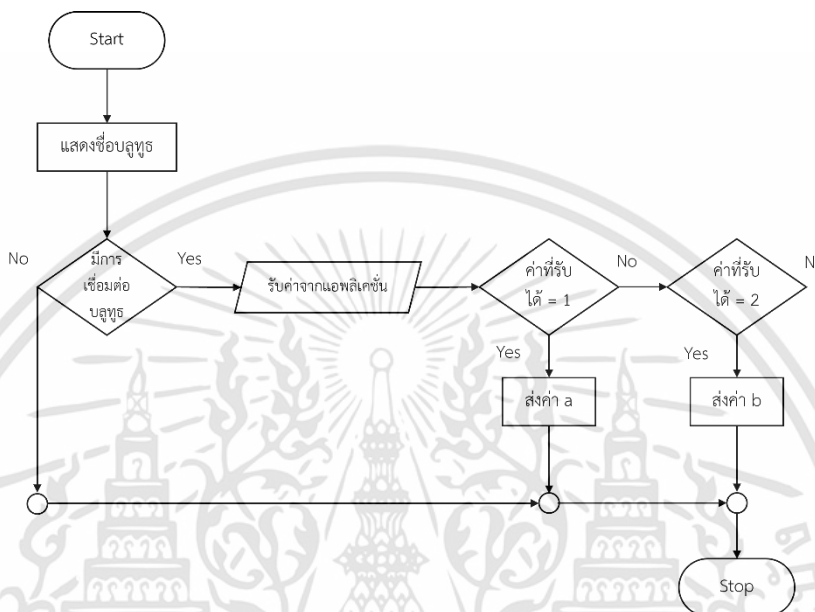


รูปที่ 3.16 แผนภาพการหาค่าความอิ่มตัวของออกซิเจนในเลือด

3.1.5 ออกแบบและเขียนโปรแกรมรับและส่งข้อมูลผ่านบลูทูธ

เขียนโปรแกรม Arduino IDE รับส่งข้อมูลผ่านบลูทูธ สามารถอธิบายขั้นตอนการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32 ได้ดังรูปที่ 3.17 โดยโปรแกรมเริ่มจากการแสดงชื่อบลูทูธบนหน้าแอปพลิเคชัน สำหรับการรอเชื่อมต่อ แล้วจะทำการตรวจสอบว่ามีการเชื่อมต่อบลูทูธกับอุปกรณ์เคลื่อนที่แล้วหรือไม่ ถ้ามีการใช้งานบลูทูธแล้วจะสามารถรับค่าจากแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ที่ได้ โดยจะกำหนดไว้ว่าเมื่อมีการรับค่า 1 เข้ามาจะให้ส่งค่า a

กลับไปแสดงบนหน้าแอปพลิเคชัน (ค่า a คือค่าที่ได้จากการสุ่ม) และเมื่อมีการรับค่า 2 เข้ามาจะให้ส่งค่า b กลับไปแสดงบนหน้าแอปพลิเคชัน (ค่า b คือค่าของ a ที่ได้จากการสุ่มล่าสุด) สามารถเขียนคำสั่งในการรับและส่งข้อมูลผ่านบลูทูธได้ดังรูปที่ 3.18



รูปที่ 3.17 แผนภาพการรับและส่งข้อมูลผ่านบลูทูธบนไมโครคอนโทรลเลอร์ ESP32

```

#include "BluetoothSerial.h"
BluetoothSerial ESP_BT;

int incoming, a, b;

void setup() {
  Serial.begin(19200);
  ESP_BT.begin("ESP32 Control"); //Name of your Bluetooth interface
}

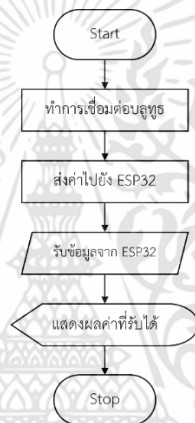
void loop() {
  //Receive Bluetooth signal
  delay(1000);
  a = random(60, 100);
  //ESP_BT.write(a);
  if (ESP_BT.available())
  {
    incoming = ESP_BT.read(); //Read what we receive
  }
  if (incoming == 1) {
    ESP_BT.write(a); //Send DATA A
    b = a;
  }
  else if (incoming == 2) {
    ESP_BT.write(b); //Send DATA B
  }
}

```

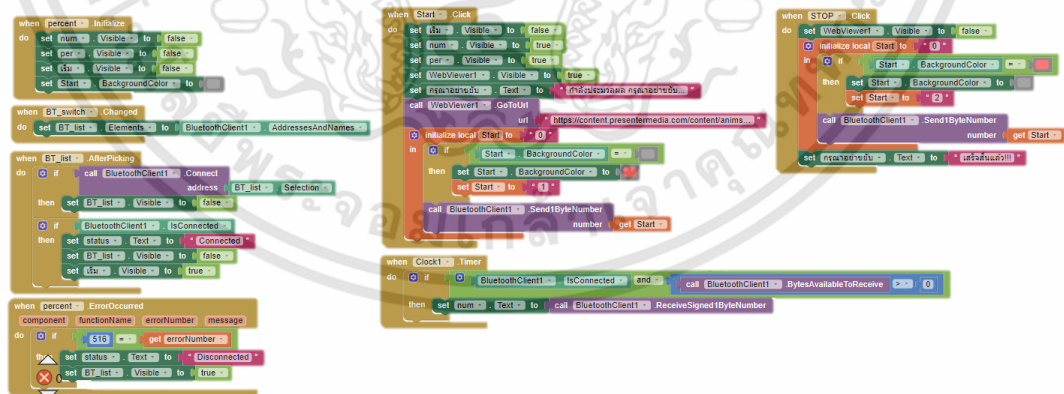
รูปที่ 3.18 คำสั่งในการรับและส่งข้อมูลผ่านบลูทูธ

3.1.6 ออกแบบและเขียนแอปพลิเคชันบนอุปกรณ์เคลื่อนที่

เขียนแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ด้วย MIT App Inventor สามารถอธิบายขั้นตอนการทำงานของแอปพลิเคชันได้ดังรูปที่ 3.19 โดยโปรแกรมเริ่มจากการทำการเชื่อมต่อกับบลูทูธ จากนั้นเมื่อผู้ใช้งานกดปุ่มเริ่มต้นการใช้งาน (Start) จะมีการส่งค่า 1 ไปยัง ESP32 แล้วจะรอรับข้อมูล (ค่า a) จากไมโครคอนโทรลเลอร์ ESP32 ส่งกลับมาเพื่อแสดงค่าที่ได้บนหน้าจอแอปพลิเคชัน และเมื่อกดปุ่มหยุดการใช้งาน (Stop) จะมีการส่งค่า 2 ไปยังไมโครคอนโทรลเลอร์ ESP32 แล้วจะรอรับข้อมูล (ค่า b) จากไมโครคอนโทรลเลอร์ ESP32 ส่งกลับมาเพื่อแสดงค่าที่ได้บนหน้าจอแอปพลิเคชัน สามารถเขียนบล็อกการทำงานของแอปพลิเคชันได้ดังรูปที่ 3.20



รูปที่ 3.19 แผนภาพการทำงานของแอปพลิเคชัน

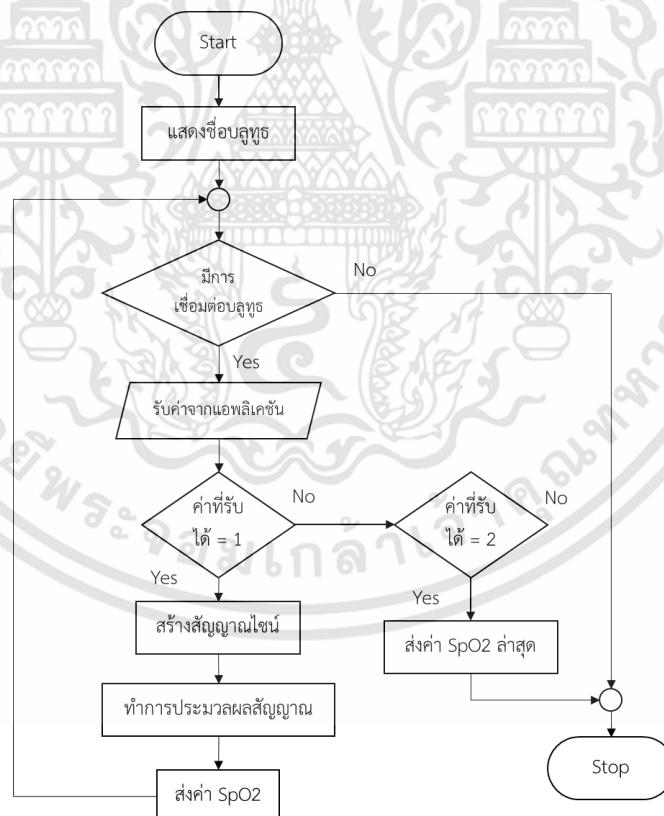


รูปที่ 3.20 บล็อกการทำงานของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7 ออกแบบและเขียนโปรแกรมรวมระบบ

เขียนโปรแกรม Arduino IDE รวมระบบทั้งส่วนของการสร้างสัญญาณไซน์ ซึ่งเป็นสัญญาณอินพุตให้กับวงจรตัวกำเนิดสีแดงและแสงอินฟราเรด ส่วนของการประมวลผลสัญญาณ และส่วนการส่งข้อมูลที่ได้ไปแสดงผลที่หน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ สามารถอธิบายขั้นตอนการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32 ได้ดังรูปที่ 3.21 โดยโปรแกรมเริ่มจากการแสดงชื่อบลูทูธบนหน้าแอปพลิเคชัน สำหรับการรอเชื่อมต่อ แล้วจะทำการตรวจสอบว่ามี การเชื่อมต่อบลูทูธกับอุปกรณ์เคลื่อนที่แล้วหรือไม่ ถ้ามีการใช้งานบลูทูธแล้วจะสามารถรับค่าจาก แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ได้ โดยจะกำหนดไว้ว่าเมื่อมีการรับค่า 1 เข้ามาจะให้เริ่มทำการ สร้างสัญญาณไซน์ที่ได้ออกแบบไว้ดังหัวข้อ 3.1.2 ต่อมาทำการประมวลผลสัญญาณดังที่ออกแบบไว้ ในหัวข้อ 3.1.4 แล้วทำการส่งค่าความอิมพัลส์ของออกซิเจนในเลือดไปแสดงผลบนหน้าแอปพลิเคชัน จากนั้นทำการวนซ้ำตั้งแต่ต้นไปเรื่อย ๆ และถ้ามีการรับค่า 2 เข้ามาจะให้ส่งค่าความอิมพัลส์ของ ออกซิเจนในเลือดที่วัดได้ล่าสุดกลับไปแสดงผลบนหน้าแอปพลิเคชัน และหยุดการทำงานของโปรแกรม



รูปที่ 3.21 แผนภาพการทำงานของระบบบนไมโครคอนโทรลเลอร์ ESP32

3.2 เครื่องมือที่ใช้ในการทดลอง

ในปฏิญานพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 เครื่องออสซิลโลสโคป (Oscilloscope)

เครื่องออสซิลโลสโคปเป็นเครื่องมือวัดสัญญาณต่าง ๆ โดยจะแสดงผลออกมาเป็นกราฟผ่านหลอดภาพที่ฉาบด้วยฟอสเฟอร์ดังรูปที่ 3.22 โดยผู้จัดทำได้นำมาใช้สำหรับแสดงสัญญาณไซน์ความถี่ 80 และ 190 เฮิร์ต ซึ่งเป็นสัญญาณที่สร้างจากไมโครคอนโทรลเลอร์ ESP32 และใช้สำหรับตรวจสอบแรงดันไฟฟ้าที่พอร์ตต่าง ๆ ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมมิตัวออกซิเจนในเลือด



รูปที่ 3.22 เครื่องออสซิลโลสโคป [16]

3.2.2 เครื่องจ่ายไฟ (Power Supply)

เครื่องจ่ายไฟเป็นอุปกรณ์ที่จ่ายพลังงานไฟฟ้าให้กับโหลดไฟฟ้าต่าง ๆ เป็นการแปลงพลังงานไฟฟ้าจากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่ง เครื่องจ่ายไฟดังรูปที่ 3.23 สามารถควบคุมแรงดันไฟฟ้าหรือกระแสไฟฟ้าเอาต์พุตให้มีค่าคงที่แน่นอน โดยผู้จัดทำได้นำมาใช้สำหรับจ่ายไฟให้กับวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมมิตัวออกซิเจนในเลือด



รูปที่ 3.23 เครื่องจ่ายไฟ [17]

3.2.3 เครื่องสร้างสัญญาณ (Signal Generator)

เครื่องสร้างสัญญาณดังรูปที่ 3.24 เป็นเครื่องมือวัดทางไฟฟ้าอิเล็กทรอนิกส์ ทำหน้าที่ให้สัญญาณรูปร่างต่าง ๆ ที่เป็นสัญญาณมาตรฐาน สามารถปรับได้ทั้งแอมพลิจูดและความถี่ ใช้วัดเปรียบเทียบหรืออ้างอิงสัญญาณ



รูปที่ 3.24 เครื่องสร้างสัญญาณ [18]

3.2.4 โปรแกรม Arduino IDE

ใช้เป็นเครื่องมือสำหรับเขียนโปรแกรมและอัปโหลดโปรแกรมเข้าสู่บอร์ด เพื่อควบคุมการทำงานของบอร์ดไมโครคอนโทรลเลอร์ ESP32 ให้สร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรตซ์ ซึ่งเป็นสัญญาณอินพุตให้กับวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด ทำการประมวลผลสัญญาณที่ได้รับจากเซนเซอร์ตัวรับแสงของหัววัดความเข้มตัวออกซิเจนในเลือด และทำการส่งข้อมูลผ่านบลูทูธไปยังหน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ โดยมีโลโก้ของโปรแกรมดังรูปที่ 3.25



รูปที่ 3.25 โลโก้โปรแกรม Arduino IDE [19]

3.2.5 โปรแกรม Proteus

ใช้เป็นเครื่องมือสำหรับออกแบบลายวงจรพิมพ์ของวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความเข้มตัวออกซิเจนในเลือด และนำลายวงจรที่ได้ไปทำการกัดปรินบนแผ่นทองแดง สร้างเป็นวงจรสำหรับการทดสอบ โดยมีโลโก้ของโปรแกรมดังรูปที่ 3.26



PROTEUS

รูปที่ 3.26 โลโก้โปรแกรม Proteus [20]

3.2.6 โปรแกรม MATLAB

ใช้เป็นเครื่องมือสำหรับเขียนโปรแกรมรับสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความเข้มตัวออกซิเจนในเลือด และนำสัญญาณนั้นมาทำการประมวลผลเพื่อให้ได้สัญญาณ PPG ของคลื่นแสงสีแดงและคลื่นแสงอินฟราเรดที่สามารถนำไปคำนวณหาอัตราส่วนการดูดซับแสง และค่าความเข้มตัวของออกซิเจนในเลือดต่อไปได้ โดยมีโลโก้ของโปรแกรมดังรูปที่ 3.27

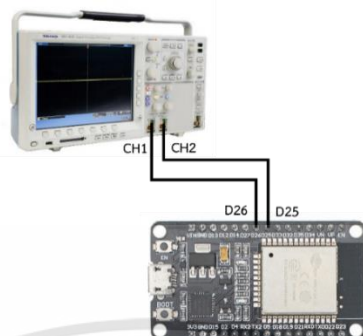


รูปที่ 3.27 โลโก้โปรแกรม MATLAB [21]

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบการทำงานของโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต์

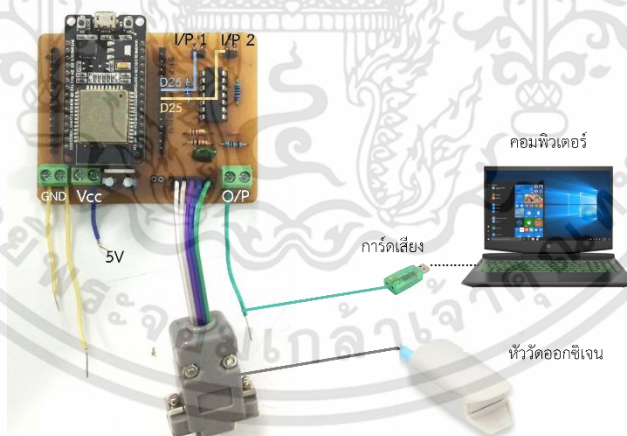
ทำการอัปโหลดโปรแกรมลงไมโครคอนโทรลเลอร์ ESP32 จากนั้นทำการติดตั้งไมโครคอนโทรลเลอร์ ESP32 กับเครื่องออสซิลโลสโคป โดยให้ช่องสัญญาณที่หนึ่ง (CH1) ต่อเข้ากับขาที่ 26 ของไมโครคอนโทรลเลอร์ ESP32 และช่องสัญญาณที่สอง (CH2) ต่อเข้ากับขาที่ 25 ของไมโครคอนโทรลเลอร์ ESP32 และทำการดูสัญญาณไซน์บนหน้าจอเครื่องออสซิลโลสโคป แสดงการติดตั้งการทดสอบได้ดังรูปที่ 3.28



รูปที่ 3.28 การติดตั้งการทดสอบการทำงานของโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิร์ต

3.3.2 การทดสอบวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และ วงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือด

ทำการติดตั้งวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือดที่สร้างเสร็จสิ้นแล้วเข้ากับเครื่องจ่ายไฟสำหรับจ่ายไฟ 5 โวลต์ ที่พอร์ต Vcc แล้วเชื่อมต่อไมโครคอนโทรลเลอร์ ESP32 ขา 25 และ ขา 26 เข้ากับพอร์ตอินพุต 1 และพอร์ตอินพุต 2 ตามลำดับ จากนั้นนำหัววัดความอืดตัวออกซิเจนในเลือดมาเชื่อมต่อที่หัวเชื่อมต่อ DB-9 แล้วนำการ์ดเสียงเชื่อมต่อที่เอาต์พุตของวงจร เพื่อที่จะนำสัญญาณที่ได้มาเข้าคอมพิวเตอร์ได้ดังรูปที่ 3.29

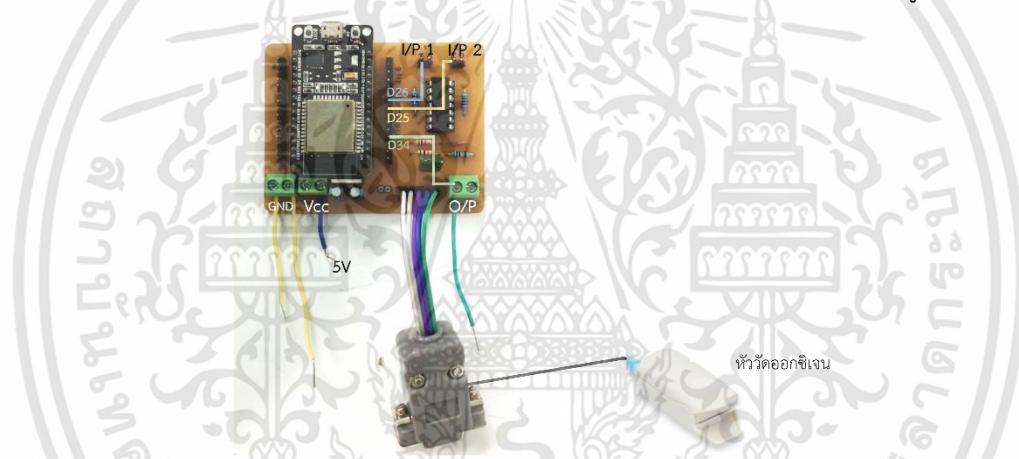


รูปที่ 3.29 การติดตั้งวงจรสำหรับการทดสอบวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอืดตัวออกซิเจนในเลือด

ทำการอัปโหลดโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต ดังหัวข้อที่ 3.3.1 ลงบนไมโครคอนโทรลเลอร์ ESP32 แล้วทำการเก็บสัญญาณที่ได้เมื่อนำหัววัดความอิมิต์วออกซิเจนในเลือดมาหนีบที่ปลายนิ้วมือ จากนั้นทำการประมวลผลข้อมูลที่ได้ผ่านโปรแกรม MATLAB

3.3.3 การประมวลผลสัญญาณที่ได้รับบนไมโครคอนโทรลเลอร์ ESP32

ทำการติดตั้งวงจรขับตัวกำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรมหาสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมิต์วออกซิเจนในเลือดที่สร้างเสร็จสิ้นแล้วเข้ากับเครื่องจ่ายไฟสำหรับจ่ายไฟ 5 โวลต์ ที่พอร์ต Vcc แล้วเชื่อมต่อไมโครคอนโทรลเลอร์ ESP32 ขา 25, ขา 26 และขา 34 เข้ากับพอร์ตอินพุต 1, พอร์ตอินพุต 2 และพอร์ตเอาต์พุตของวงจร ตามลำดับ จากนั้นนำหัววัดความอิมิต์วออกซิเจนในเลือดมาเชื่อมต่อที่หัวเชื่อมต่อ DB-9 ได้ดังรูปที่ 3.30



รูปที่ 3.30 การติดตั้งสำหรับการประมวลผลสัญญาณที่ได้รับบนไมโครคอนโทรลเลอร์ ESP32

ทำการอัปโหลดโปรแกรมที่ได้ตั้งหัวข้อที่ 3.1.4 ลงบนไมโครคอนโทรลเลอร์ ESP32 แล้วทำการเก็บสัญญาณที่ได้เมื่อนำหัววัดความอิมิต์วออกซิเจนในเลือดมาหนีบที่ปลายนิ้วมือ จากนั้นทำการบันทึกผลที่ได้บนซีเรียลพล็อตเตอร์

3.3.4 การทดสอบการรับและส่งข้อมูลผ่านบลูทูธ

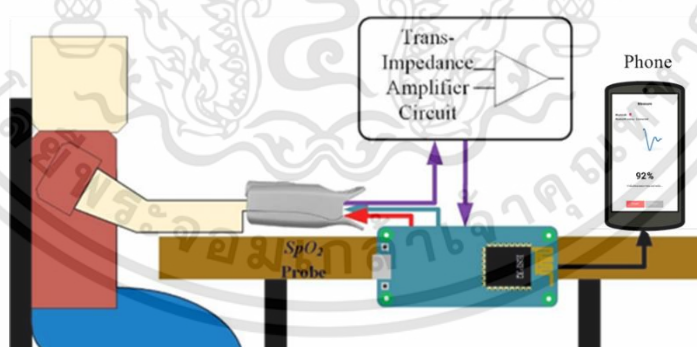
ทำการอัปโหลดโปรแกรมการรับส่งข้อมูลผ่านบลูทูธดังหัวข้อที่ 3.1.5 ลงบนไมโครคอนโทรลเลอร์ ESP32 แล้วทำการเชื่อมต่อบลูทูธระหว่างอุปกรณ์เคลื่อนที่กับไมโครคอนโทรลเลอร์ ESP32 สังเกตผลที่ได้บนซีเรียลมอนิเตอร์และหน้าแอปพลิเคชัน เมื่อมีการรับและส่งข้อมูล

3.3.5 การทดสอบระบบรวมทั้งหมด

ทำการอัปโหลดโปรแกรมตั้งหัวข้อที่ 3.1.7 ลงบนไมโครคอนโทรลเลอร์ ESP32 แล้วทำการเชื่อมต่อบลูทูธระหว่างอุปกรณ์เคลื่อนที่กับไมโครคอนโทรลเลอร์ ESP32 จากนั้นนำหัววัดความอิ่มตัวออกซิเจนในเลือดมาหนีบที่ปลายนิ้วมือ และทำการกดปุ่มเริ่มต้นการใช้งานที่หน้าแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ สังเกตผลที่ได้บนซีเรียลมอนิเตอร์และหน้าแอปพลิเคชัน

3.3.6 การทดลองวัดค่าเปอร์เซ็นต์ความอิ่มตัวออกซิเจนในเลือดเทียบกับเครื่องมาตรฐาน

ทำการทดลองวัดค่าเปอร์เซ็นต์ความอิ่มตัวออกซิเจนในเลือดโดยใช้เครื่องวัดความอิ่มตัวออกซิเจนในเลือดที่สร้างเสร็จสิ้นแล้ววัดเทียบกับเครื่องวัดความอิ่มตัวออกซิเจนในเลือดปลายนิ้วยี่ห้อ Unigo ซึ่งใช้สำหรับเป็นเครื่องวัดมาตรฐานในการทดลอง ทำการทดลองกับนักศึกษาภาควิชาโทรคมนาคมจำนวน 50 คน (หญิง 25 คน และชาย 25 คน) โดยแต่ละคนจะทำการทดลองวัดค่าออกซิเจนในเลือดด้วยเครื่องที่เราสร้างขึ้นมา และเครื่องมาตรฐานในขณะเวลาเดียวกัน โดยหนึ่งคนจะทำการวัดติดต่อกันทั้งหมด 5 ครั้ง ครั้งละ 5 วินาที และสำหรับสภาพแวดล้อมในการทดลองจะทำการทดลองในห้องที่มีการควบคุมแสง และให้ผู้ทดลองนั้นนั่งนิ่งเพื่อที่จะลดสัญญาณรบกวนที่เกิดจากการเคลื่อนไหว นำผลที่ได้ไปคำนวณหาประสิทธิภาพการทำงานของเครื่องวัดความอิ่มตัวออกซิเจนในเลือด แสดงการติดตั้ง และสภาพแวดล้อมสำหรับการทดลองได้ดังรูปที่ 3.31



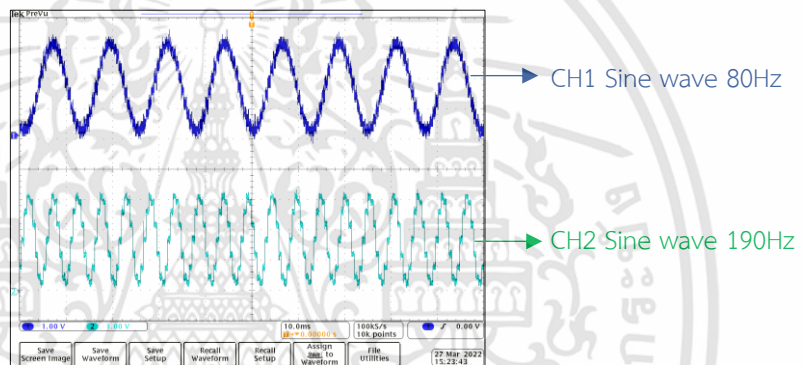
รูปที่ 3.31 การติดตั้ง และสภาพแวดล้อมสำหรับการทดลอง

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการทำงานของโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต์

เมื่อทำการอัปโหลดโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต์ ลงบนไมโครคอนโทรลเลอร์ ESP32 และทำการติดตั้งดังหัวข้อที่ 3.3.1 เรียบร้อยแล้ว จะได้สัญญาณไซน์ความถี่ 80 และ 190 เฮิรต์ ที่แสดงบนหน้าจอเครื่องออสซิลโลสโคปดังรูปที่ 4.1

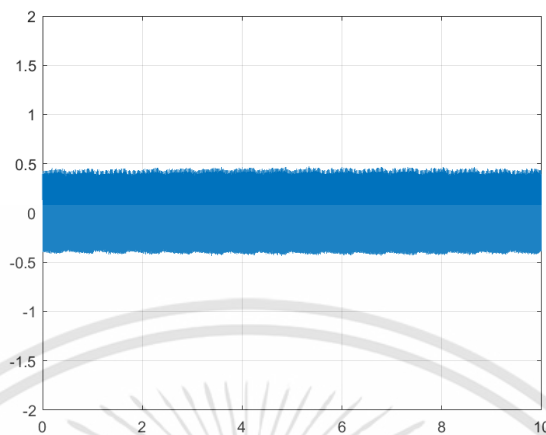


รูปที่ 4.1 สัญญาณไซน์ที่ได้บนหน้าจอเครื่องออสซิลโลสโคป

จะเห็นว่าที่ช่องสัญญาณหนึ่งได้สัญญาณไซน์ความถี่ 80 เฮิรต์ และช่องสัญญาณสองได้สัญญาณไซน์ความถี่ 190 เฮิรต์ และทั้งสองสัญญาณนี้มีค่าแรงดันไฟฟ้าเท่ากับ 3 โวลต์

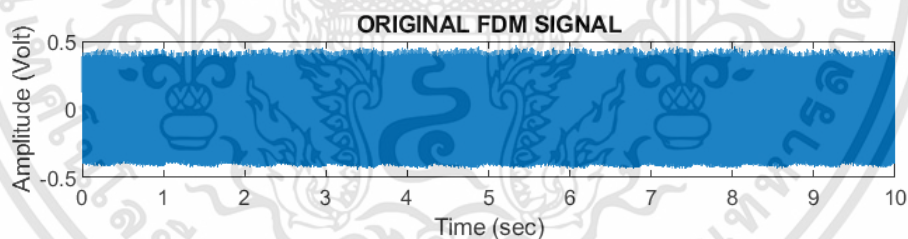
4.2 ผลการทดสอบวงจรขับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความเข้มตัวออกซิเจนในเลือด

หลังจากทำการติดตั้งอุปกรณ์ดังหัวข้อที่ 3.3.2 และอัปโหลดโปรแกรมสร้างสัญญาณไซน์ความถี่ 80 และ 190 เฮิรต์ เรียบร้อยแล้ว เมื่อนำหัววัดความเข้มตัวออกซิเจนในเลือดมาหนีบที่ปลายนิ้วมือ และรันโปรแกรม MATLAB จะได้สัญญาณที่โปรแกรม MATLAB รับได้ หรือสัญญาณเอฟดีเอ็มผ่านการ์ดเสียงเข้าเครื่องคอมพิวเตอร์ ดังรูปที่ 4.2



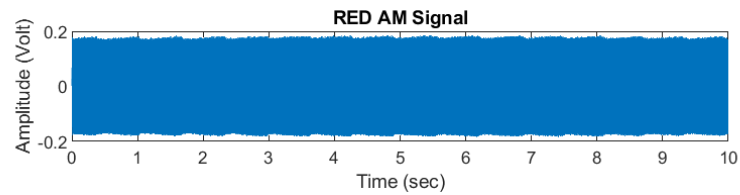
รูปที่ 4.2 สัญญาณเอฟดีเอ็มที่โปรแกรม MATLAB รับเข้ามา

โดยสัญญาณเอฟดีเอ็มที่รับเข้ามาคือ การรวมกันของสัญญาณเอเอ็มของคลื่นแสงสีแดง (สัญญาณพีพีจีที่มอดูเลตกับคลื่นแสงสีแดง), สัญญาณเอเอ็มของคลื่นอินฟราเรด (สัญญาณพีพีจีที่มอดูเลตกับคลื่นอินฟราเรด) และไฟตรง ดังนั้นจึงนำสัญญาณเอฟดีเอ็มที่รับเข้ามาทำการลบด้วยค่าเฉลี่ยของสัญญาณเอฟดีเอ็ม เพื่อกำจัดค่าไฟตรง จะได้สัญญาณเอฟดีเอ็มที่ผ่านการลบด้วยค่าเฉลี่ยของสัญญาณเอฟดีเอ็มดังรูปที่ 4.3

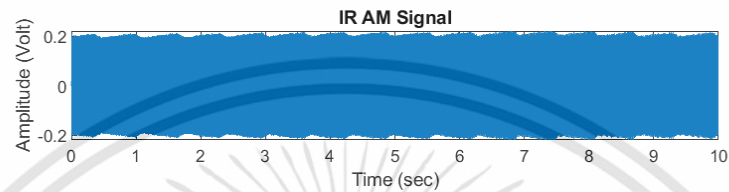


รูปที่ 4.3 สัญญาณเอฟดีเอ็มที่ผ่านการลบด้วยค่าเฉลี่ย

ซึ่งสัญญาณเอฟดีเอ็มที่ผ่านการลบด้วยค่าเฉลี่ยคือ สัญญาณเอเอ็มของคลื่นแสงสีแดง รวมกันกับสัญญาณเอเอ็มของคลื่นอินฟราเรด ดังนั้นทำการแยกสัญญาณเอฟดีเอ็มที่ผ่านการลบด้วยค่าเฉลี่ยเป็นสัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรด โดยนำสัญญาณเอฟดีเอ็มนั้นมาผ่านตัวกรองแถบความถี่ผ่าน จะได้สัญญาณเอเอ็มของคลื่นแสงสีแดง ดังรูปที่ 4.4 และสัญญาณเอเอ็มของคลื่นอินฟราเรดดังรูปที่ 4.5

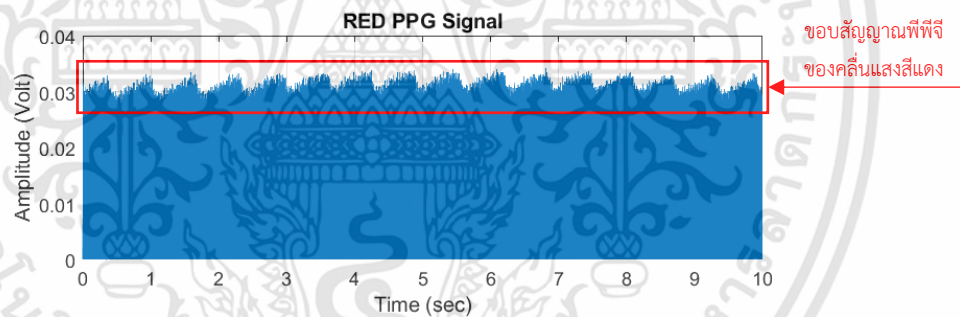


รูปที่ 4.4 สัญญาณเอเอ็มของคลื่นแสงสีแดง

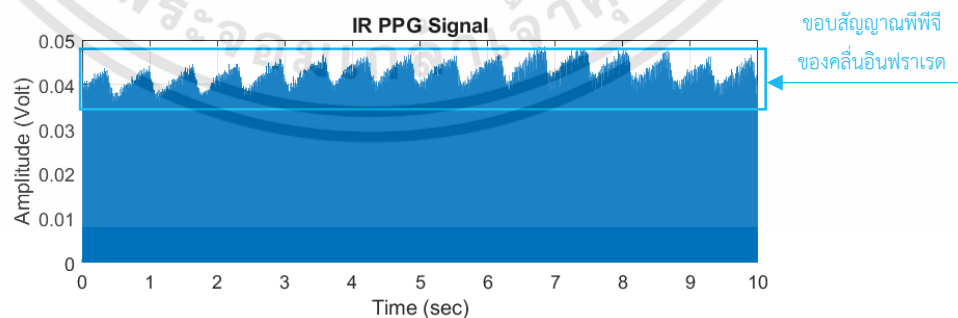


รูปที่ 4.5 สัญญาณเอเอ็มของคลื่นอินฟราเรด

เมื่อได้สัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรดแล้ว จึงทำการตัดมอดูเลตด้วยวิธีตรวจจับขอบ โดยนำสัญญาณเอเอ็มทั้งสองมายกกำลังสอง จะได้สัญญาณเอเอ็มยกกำลังสองของคลื่นแสงสีแดงดังรูปที่ 4.6 และจะได้สัญญาณเอเอ็มยกกำลังสองของคลื่นอินฟราเรดดังรูปที่ 4.7



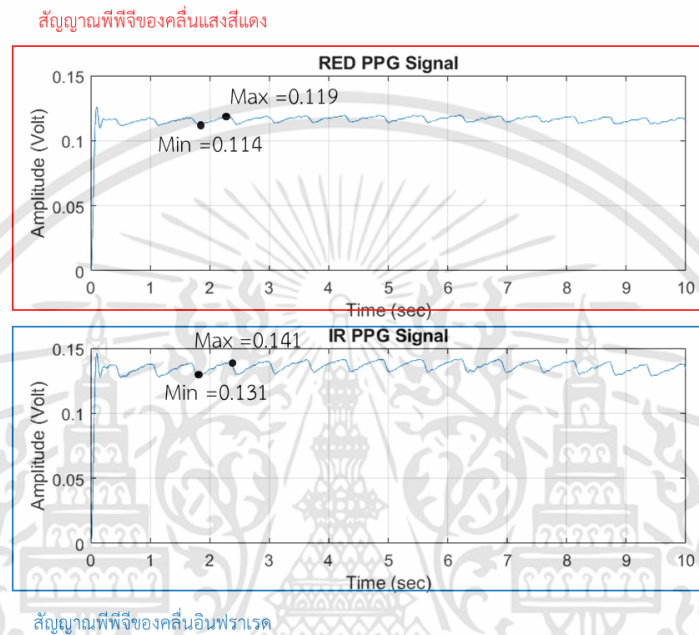
รูปที่ 4.6 สัญญาณเอเอ็มยกกำลังสองของคลื่นแสงสีแดง



รูปที่ 4.7 สัญญาณเอเอ็มยกกำลังสองของคลื่นอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.6 และ 4.7 จะเห็นว่าสัญญาณเริ่มเห็นเป็นขอบสัญญาณพีพีจีแล้ว จากนั้นนำสัญญาณเอเอ็มยกกำลังสองของคลื่นแสงสีแดง (รูปที่ 4.6) และสัญญาณเอเอ็มยกกำลังสองของคลื่นอินฟราเรด (รูปที่ 4.7) มาผ่านตัวกรองความถี่ต่ำผ่าน จะได้สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดดังรูปที่ 4.8



รูปที่ 4.8 สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรด

เมื่อทำการสุ่มสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรด จากรูปที่ 4.8 มาหนึ่งลูกคลื่นในช่วงเวลาเดียวกัน เพื่ออ่านค่าจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีทั้งสองแล้วนำมาคำนวณหาอัตราการดูดซับแสงดังสมการที่ 4.1

$$R = \frac{\frac{AC_{RED}}{DC_{RED}}}{\frac{AC_{IR}}{DC_{IR}}} = \frac{\left(\frac{Max-Min}{Max+Min}\right)_{RED}}{\left(\frac{Max-Min}{Max+Min}\right)_{IR}} \quad (4.1)$$

โดยจะอ่านค่าสัญญาณพีพีจีของคลื่นแสงสีแดงที่จุดสูงสุดได้ 0.119 และจุดต่ำสุดได้ 0.114 ส่วนสัญญาณพีพีจีของคลื่นอินฟราเรดจะอ่านค่าจุดสูงสุดได้ 0.141 และจุดต่ำสุดได้ 0.131 จากนั้นนำค่าจุดสูงสุดและจุดต่ำสุดของสัญญาณพีพีจีทั้งสองมาคำนวณหาอัตราการดูดซับแสงดังสมการที่ 4.2

$$R = \frac{(0.119 - 0.114 / \frac{0.119 + 0.114}{2})_{RED}}{(0.141 - 0.131 / \frac{0.141 + 0.131}{2})_{IR}} = 0.5837 \quad (4.2)$$

จะได้ค่าอัตราการดูดซับแสง เท่ากับ 0.5837 สุดท้ายเมื่อได้ค่าอัตราการดูดซับแสงแล้ว นำค่าดังกล่าวมาคำนวณหาค่าความอิ่มตัวของออกซิเจนในเลือดได้จากสมการที่ 4.3

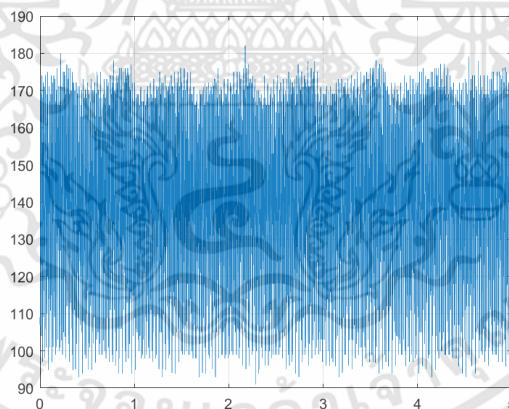
$$\%SpO_2 = 110 - (25 \times R)$$

$$\%SpO_2 = 110 - (25 \times 0.5837) = 95.41\% \quad (4.3)$$

จะได้ค่าความอิ่มตัวของออกซิเจนในเลือด เท่ากับ 95.41%

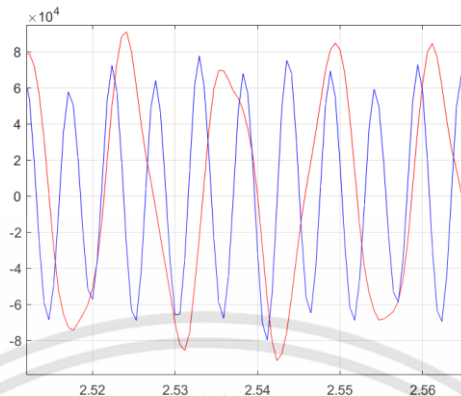
4.3 การประมวลผลสัญญาณที่ได้รับบนไมโครคอนโทรลเลอร์ ESP32

เมื่อติดตั้งวงจรจับตัวกำเนิดแสง และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของ หัววัดความอิ่มตัวของออกซิเจนในเลือดเรียบร้อยแล้ว นำหัววัดความอิ่มตัวของออกซิเจนในเลือดหนีบเข้าที่ปลายนิ้ว จะได้สัญญาณเอฟดีเอ็มที่ไมโครคอนโทรลเลอร์ ESP32 รับเข้ามา สามารถนำสัญญาณดังกล่าวมาพล็อตแสดงผลบนโปรแกรม MATLAB ได้ดังรูปที่ 4.9



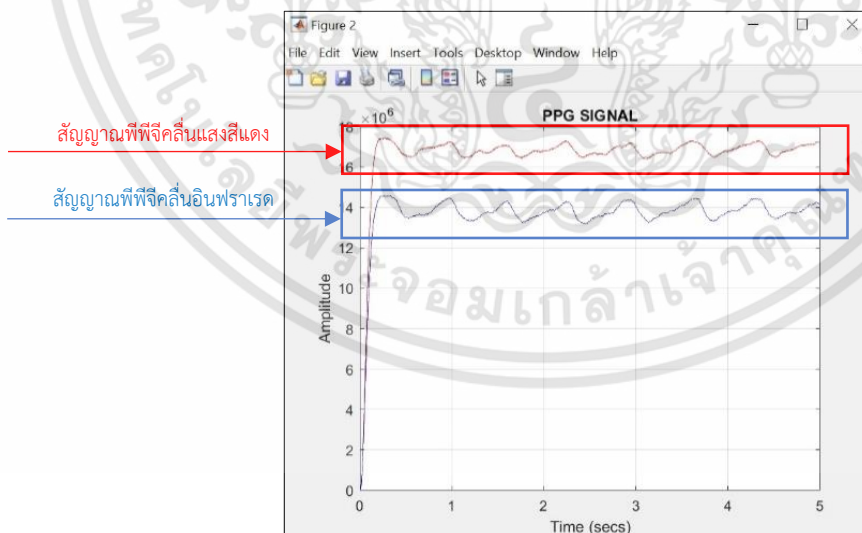
รูปที่ 4.9 สัญญาณเอฟดีเอ็มที่รับเข้ามาพล็อตแสดงผลบนโปรแกรม MATLAB

เมื่อนำสัญญาณเอฟดีเอ็มที่รับเข้ามาผ่านตัวกรองแถบความถี่ผ่าน แล้วนำสัญญาณที่ผ่านตัวกรองแถบความถี่ผ่านมาพล็อตแสดงผลบนโปรแกรม MATLAB ดังรูปที่ 4.10 จะเห็นว่าตัวกรองแถบความถี่ผ่านมีการแยกสัญญาณเอฟดีเอ็มเป็นสัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรด โดยกราฟเส้นสีแดงคือสัญญาณเอเอ็มของคลื่นแสงสีแดง และกราฟเส้นสีน้ำเงินคือสัญญาณเอเอ็มของคลื่นอินฟราเรด



รูปที่ 4.10 สัญญาณเอเอ็มของคลื่นแสง และคลื่นอินฟราเรดบนโปรแกรม MATLAB

จากนั้นในขั้นตอนการติมอดูเลตสัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรดจะทำด้วยวิธีการติมอดูเลตสัญญาณแบบโคฮีเรนต์ดีเทคชัน โดยการนำสัญญาณเอเอ็มคูณกับสัญญาณที่มีความถี่และเฟสตรงกันกับสัญญาณคลื่นพาห์ กล่าวคือนำสัญญาณเอเอ็มของคลื่นแสงสีแดงคูณกับสัญญาณไซน์ความถี่ 80 เฮิร์ต และนำสัญญาณเอเอ็มของคลื่นอินฟราเรดคูณกับสัญญาณไซน์ความถี่ 190 เฮิร์ต จากนั้นก็นำสัญญาณผลคูณของคลื่นแสงสีแดง และสัญญาณผลคูณของคลื่นอินฟราเรดมาทำการกรองด้วยตัวกรองความถี่ต่ำผ่านจะได้สัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดมาพล็อตแสดงผลบนโปรแกรม MATLAB ได้ดังรูปที่ 4.11



รูปที่ 4.11 กราฟสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดบนโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนการดูค่าอัตราส่วนการดูดซับแสง และค่าความอิ่มตัวของออกซิเจนในเลือดจากการประมวลสัญญาณแต่ละรอบ สามารถดูได้โดยการรันโปรแกรมบนซีเรียลมอนิเตอร์ จะเห็นว่าได้ค่าอัตราส่วนการดูดซับแสงเท่ากับ 0.53 และค่าความอิ่มตัวของออกซิเจนในเลือดเท่ากับ 96.79% ดังรูปที่ 4.12

```

COM3
[
137.00 -47954.92 43545.82 17214368.00 14136888.00;
132.00 -56986.65 9526.39 17207820.00 14131212.00;
119.00 -65292.34 -30088.76 17203264.00 14122813.00;
103.00 -73397.28 -58099.41 17201574.00 14114952.00;
95.00 -77641.22 -60162.16 17202826.00 14109485.00;
102.00 -72426.26 -32422.74 17205756.00 14103689.00;
118.00 -55790.81 12958.04 17208232.00 14094822.00;
]
FDM = data(:, 1)
AM_RED = data(:, 2)
AM_IR = data(:, 3)
PPG_RED = data(:, 4)
PPG_IR = data(:, 5)
figure
plot(t, FDM)
figure
plot(t, AM_RED, 'r', t, AM_IR, 'b')
figure
plot(t, PPG_RED, 'r', t, PPG_IR, 'b')
filename = 'test1.mat'
save(filename)
R = 0.53
SpO2 = 96.79

```

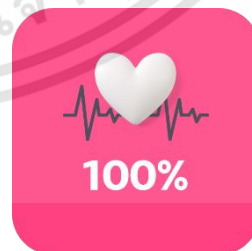
รูปที่ 4.12 อัตราส่วนการดูดซับแสง และความอิ่มตัวของออกซิเจนในเลือดบนซีเรียลมอนิเตอร์

4.4 การทดสอบการรับและส่งข้อมูลผ่านบลูทูธ

เริ่มต้นการทดสอบการรับและส่งข้อมูลผ่านบลูทูธจากการติดตั้งแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ โดยการแสกนบาร์โค้ด ดังรูปที่ 4.13 (ก) จะได้ไฟล์ APK (Android Package Kit) ที่เป็นไฟล์ที่สามารถติดตั้งแอปพลิเคชันลงบนอุปกรณ์เคลื่อนที่ได้ กดยอมรับการติดตั้ง จากนั้นเมื่อการติดตั้งเสร็จสิ้นแล้วจะได้ไอคอนของแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ ดังรูปที่ 4.13 (ข)



(ก)



(ข)

รูปที่ 4.13 บาร์โค้ดสำหรับการติดตั้งแอปพลิเคชัน และไอคอนของแอปพลิเคชัน

หลังจากนั้นทำการอัปโหลดโปรแกรมการรับส่งข้อมูลผ่านบลูทูธลงบนไมโครคอนโทรลเลอร์ ESP32 และทำการเชื่อมต่อบลูทูธระหว่างอุปกรณ์เคลื่อนที่กับไมโครคอนโทรลเลอร์ ESP32 จะได้ผลการทดสอบการรับและส่งข้อมูลผ่านบลูทูธบนซีเรียลมอนิเตอร์ได้ดังรูปที่ 4.14

```

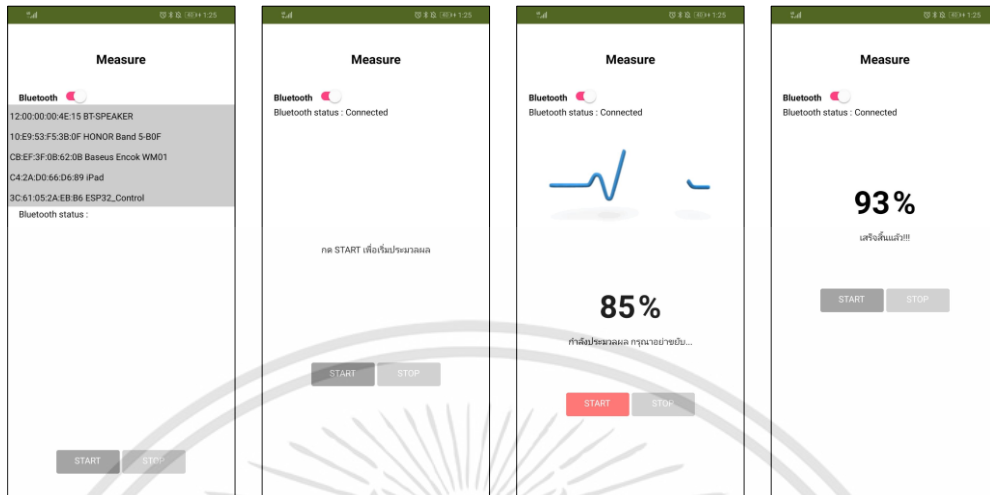
COM3
ESP_BT.available
incoming = 1
a = 64
a = 66
a = 92
a = 90
a = 90
ESP_BT.available
incoming = 2
b = 90
b = 90
b = 90
b = 90
Autoscroll Show timestamp Newline 19200 baud Clear output

```

รูปที่ 4.14 ผลทดสอบการรับและส่งข้อมูลผ่านบลูทูธบนซีเรียลมอนิเตอร์

จะเห็นว่ามี การเชื่อมต่อบลูทูธระหว่างอุปกรณ์เคลื่อนที่กับไมโครคอนโทรลเลอร์ ESP32 และเมื่อมีการรับค่า 1 จากแอปพลิเคชันบนอุปกรณ์เคลื่อนที่เข้ามา ไมโครคอนโทรลเลอร์ ESP32 จะทำการส่งค่า a ที่ได้จากการสุ่มกลับไปยังแอปพลิเคชันเรื่อย ๆ จนกระทั่งมีการรับค่า 2 เข้ามา ไมโครคอนโทรลเลอร์ ESP32 จะทำการส่งค่า b กลับไปยังแอปพลิเคชัน

และในส่วนของหน้าแอปพลิเคชันจะได้ผลการทดสอบดังรูปที่ 4.15 เมื่อเปิดหน้าแอปพลิเคชันจะมีการเข้าใช้งานดังรูป (ก) ซึ่งเป็นขั้นตอนการเชื่อมต่อบลูทูธ โดยจะมีการแสดงรายชื่อบลูทูธให้เลือกในการเชื่อมต่อ และเมื่อเลือกเชื่อมต่อบลูทูธชื่อ ESP32_Control จะได้ผลลัพธ์ดังรูป (ข) ที่พร้อมสำหรับการเริ่มใช้งาน จากนั้นเมื่อผู้ใช้งานกดปุ่มเริ่มการใช้งาน ดังรูป (ค) ที่หน้าแอปพลิเคชันจะมีการแสดงค่า a ที่ได้รับจากไมโครคอนโทรลเลอร์ ESP32 ไปเรื่อย ๆ จนกว่าผู้ใช้งานจะกดปุ่มหยุดการใช้งาน เมื่อกดหยุดการใช้งานแล้วที่หน้าแอปพลิเคชันจะแสดงค่า b ที่รับมาจากไมโครคอนโทรลเลอร์ ESP32 ได้ดังรูป (ง)



(ก) การเชื่อมต่อบลูทูธ (ข) พร้อมใช้งาน (ค) เมื่อกด Start (ง) เมื่อกด Stop

รูปที่ 4.15 ผลทดสอบการรับและส่งข้อมูลผ่านบลูทูธบนหน้าจอแอปพลิเคชัน

4.5 การทดสอบระบบรวมทั้งหมด

เมื่อติดตั้งวงจร และอัปโหลดโปรแกรมดังหัวข้อที่ 3.1.7 ลงบนไมโครคอนโทรลเลอร์ ESP32 ทำเชื่อมต่อบลูทูธกับแอปพลิเคชัน จากนั้นทำการวัดค่าความอิ่มตัวของออกซิเจนในเลือด จะได้ผลการทดสอบระบบรวมบนซีเรียลมอนิเตอร์ดังรูปที่ 4.16

```

CCME3
ESP_BT_available
incoming = 1
m = 0.93
spO2 = 92.84
n = 1.02
spO2 = 91.65
m = 0.96
spO2 = 92.52
n = 1.05
m = 0.93
spO2 = 92.72
n = 0.99
spO2 = 92.17
n = 0.90
spO2 = 94.51
n = 1.09
n = 1.01
spO2 = 92.94
ESP_BT_available
incoming = 2

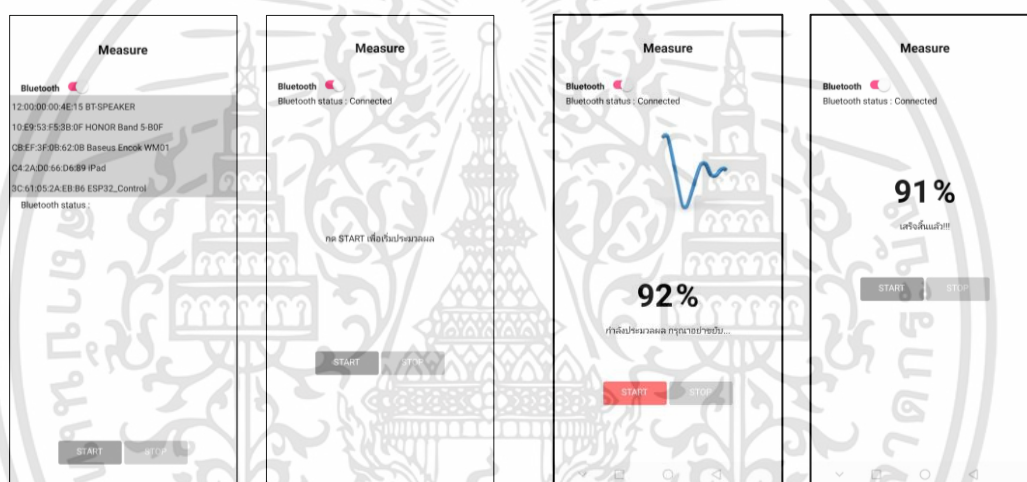
```

รูปที่ 4.16 ผลทดสอบระบบรวมบนซีเรียลมอนิเตอร์

จะเห็นว่าเมื่อมีการเชื่อมต่อบลูทูธระหว่างอุปกรณ์เคลื่อนที่กับไมโครคอนโทรลเลอร์ ESP32 และมีการรับค่า 1 จากแอปพลิเคชันบนอุปกรณ์เคลื่อนที่เข้ามา ไมโครคอนโทรลเลอร์ ESP32 จะทำการประมวลผลสัญญาณ แล้วทำการส่งค่าความอิ่มตัวของออกซิเจนในเลือดไปแสดงผลบนหน้าจอแอปพลิเคชัน โปรแกรมจะทำงานไปเรื่อย ๆ จนกระทั่งมีการรับค่า 2 เข้ามา ไมโครคอนโทรลเลอร์ ESP32 จะทำการส่งค่าความอิ่มตัวของออกซิเจนในเลือดที่ได้จากการคำนวณล่าสุดกลับไปยังแอปพลิเคชัน และหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในส่วนของหน้าแอปพลิเคชันจะได้ผลการทดสอบดังรูปที่ 4.17 เมื่อเปิดหน้าแอปพลิเคชันจะมีการเข้าใช้งานดังรูป (ก) ซึ่งเป็นขั้นตอนการเชื่อมต่อบลูทูธ โดยจะมีการแสดงรายชื่อบลูทูธให้เลือกในการเชื่อมต่อ และเมื่อเลือกเชื่อมต่อบลูทูธชื่อ ESP32_Controll จะได้ผลลัพธ์ดังรูป (ข) ที่พร้อมสำหรับการเริ่มใช้งาน จากนั้นเมื่อผู้ใช้งานกดปุ่มเริ่มการใช้งาน ดังรูป (ค) ที่หน้าแอปพลิเคชันจะมีการแสดงค่าความอิ่มตัวของออกซิเจนในเลือดที่ได้รับจากไมโครคอนโทรลเลอร์ ESP32 ไปเรื่อย ๆ จนกว่าผู้ใช้งานจะกดปุ่มหยุดการใช้งาน เมื่อกดหยุดการใช้งานแล้ว ที่หน้าแอปพลิเคชันจะแสดงค่าความอิ่มตัวของออกซิเจนในเลือดที่ได้จากการคำนวณล่าสุดจากไมโครคอนโทรลเลอร์ ESP32 และสิ้นสุดการประมวลผลได้ดังรูป (ง)



(ก) การเชื่อมต่อบลูทูธ (ข) พร้อมใช้งาน (ค) เมื่อกด Start (ง) เมื่อกด Stop

รูปที่ 4.17 ผลทดสอบระบบรวมบนหน้าจอแอปพลิเคชัน

4.6 ผลการทดลองเทียบกับเครื่องมาตรฐาน

4.6.1 ผลของเปอร์เซ็นต์ความอิ่มตัวของออกซิเจนในเลือดจากการทดลองเทียบกับเครื่องมาตรฐาน

จากหัวข้อที่ 3.3.6 จะได้ค่าเปอร์เซ็นต์ความอิ่มตัวของออกซิเจนในเลือดจากเครื่องวัดความอิ่มตัวของออกซิเจนในเลือด และจากเครื่องมาตรฐาน แสดงได้ดังตารางที่ 4.1 และ 4.2

ตารางที่ 4.1 ค่าเปอร์เซ็นต์ความอิ่มตัวออกซิเจนในเลือดที่ได้จากการทดลอง (หญิง)

ลำดับที่	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)
1	97	97	98	98	96	97	94	96	97	97
2	96	98	96	98	96	97	96	97	96	97
3	94	96	94	96	94	96	95	95	95	95
4	95	98	98	98	96	98	98	98	95	98
5	97	98	97	96	95	95	95	98	97	98
6	95	97	96	98	93	98	94	98	96	97
7	97	97	97	98	99	99	98	99	97	99
8	97	97	96	98	98	99	96	99	97	98
9	96	98	97	98	97	98	99	98	97	98
10	99	98	95	98	99	98	98	98	97	98
11	97	98	97	98	98	98	98	98	99	98
12	99	98	99	99	93	99	97	98	97	98
13	98	98	96	98	94	95	94	94	93	94
14	97	98	98	98	98	98	98	98	98	98
15	97	97	97	97	97	98	97	98	97	97
16	98	97	98	98	96	97	97	98	96	98
17	95	96	95	97	96	98	96	98	96	98
18	96	96	95	96	97	97	96	98	95	98
19	96	98	98	98	97	98	97	98	97	98
20	95	97	94	97	96	97	95	98	96	98
21	96	98	97	98	98	98	96	98	97	98
22	98	97	97	98	97	96	97	97	97	96
23	95	97	97	98	95	98	96	98	96	98
24	96	96	96	96	96	96	96	97	96	97
25	96	98	96	98	95	98	95	97	96	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าเปอร์เซ็นต์ความอิ่มตัวออกซิเจนในเลือดที่ได้จากการทดลอง (ชาย)

ลำดับที่	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)	SpO2 (ทดลอง)	SpO2 (มาตรฐาน)
1	96	97	96	97	96	97	96	96	97	97
2	97	97	97	98	96	97	97	97	96	97
3	97	98	97	98	98	98	95	99	96	99
4	94	96	97	97	96	97	96	97	97	96
5	96	97	96	96	96	97	96	97	95	97
6	95	97	94	97	95	98	94	97	95	97
7	98	98	96	98	97	97	96	97	98	98
8	96	97	96	98	97	97	97	98	98	98
9	99	99	99	99	99	99	97	99	95	99
10	96	96	96	97	96	97	96	97	97	97
11	95	95	96	96	96	96	96	96	95	96
12	97	96	97	95	97	97	96	95	95	96
13	96	96	94	97	97	97	95	98	95	98
14	95	95	94	96	96	97	95	97	96	96
15	98	96	98	98	99	98	99	98	98	98
16	95	97	97	97	95	97	95	97	95	97
17	96	97	95	96	96	97	95	96	93	96
18	96	97	96	97	97	97	97	98	95	97
19	96	98	96	98	96	98	96	98	95	97
20	95	95	95	96	94	96	95	96	95	96
21	97	98	98	98	97	97	98	98	96	98
22	96	96	96	97	95	97	97	96	97	97
23	96	98	96	98	97	98	97	97	97	98
24	96	97	97	96	97	98	96	98	97	98
25	98	98	97	98	97	98	96	97	98	97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 คำนวนหาประสิทธิภาพการทำงานของเครื่องวัดความอิ่มตัวออกซิเจนในเลือด

ทำการคำนวณหาประสิทธิภาพการทำงานของเครื่องวัดความอิ่มตัวออกซิเจนในเลือด โดยการนำค่าเปอร์เซ็นต์ความอิ่มตัวออกซิเจนในเลือดจากการทดลองเทียบกับเครื่องมาตรฐานที่ได้มาทำการคำนวณหาค่าความเที่ยงตรง (%Accuracy) และค่าความแม่นยำ (Precision) โดยสามารถคำนวณหาค่าความเที่ยงตรงได้จากสมการที่ 4.4

$$\%Error = \left| \frac{\text{Measure SpO}_2 - \text{Standard SpO}_2}{\text{Standard SpO}_2} \right| \times 100$$

$$\%Accuracy = 100 - \%Error \quad (4.4)$$

และสามารถหาค่าความแม่นยำได้จากสมการที่ 4.5 ดังนี้

$$\text{Precision} = \mu \pm \frac{\sum |x - \mu|}{n} \quad (4.5)$$

โดยที่ x คือ ค่า SpO₂ ที่ได้จากการทดลองในแต่ละครั้ง

μ คือ ค่าเฉลี่ยของ SpO₂ ที่ได้จากการทดลอง

n คือ จำนวนครั้งที่ทำการทดลอง (สำหรับการทดลองนี้ n = 5)

จากสมการที่ 4.4 และสมการที่ 4.5 ในการทดลองจะได้ค่าความเที่ยงตรง และค่าความแม่นยำดังตารางที่ 4.3 และตารางที่ 4.4 นำค่าที่ได้หาค่าเฉลี่ยของค่าความเที่ยงตรงได้ 98.34 % และค่าเฉลี่ยของค่าความแม่นยำคือ $96.33 \pm 0.91\%$

ตารางที่ 4.3 ค่าความแม่นยำ และความเที่ยงตรงที่ได้จากการคำนวณ (หญิง)

อันดับ	ค่าความเที่ยงตรง	ค่าความแม่นยำ
1	99.377	1.12
2	98.565	0.00
3	98.750	0.48
4	98.367	1.28
5	98.771	0.96
6	97.137	1.04
7	99.189	0.72
8	98.778	0.64
9	98.776	0.72
10	98.776	1.28
11	99.387	0.64
12	99.387	0.96
13	99.169	1.60
14	99.796	0.32
15	99.592	0.00
16	98.975	0.8
17	98.979	0.48
18	98.771	0.64
19	98.979	0.40
20	97.742	0.64
21	98.776	0.64
22	99.173	0.32
23	97.955	0.64
24	99.588	0.00
25	97.751	0.48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ค่าความแม่นยำ และความเที่ยงตรงที่ได้จากการคำนวณ (ชาย)

อันดับ	ค่าความเที่ยงตรง	ค่าความแม่นยำ
1	99.381	0.32
2	99.384	0.48
3	98.178	0.88
4	99.171	0.72
5	98.969	0.32
6	97.326	0.48
7	99.386	0.80
8	99.182	0.64
9	98.788	1.44
10	99.381	0.32
11	99.791	0.48
12	98.952	0.72
13	98.160	0.88
14	98.965	0.64
15	99.175	0.48
16	98.351	0.64
17	98.546	0.80
18	98.971	0.64
19	97.955	0.32
20	98.958	0.32
21	99.388	0.64
22	99.173	0.64
23	98.776	0.48
24	98.769	0.48
25	99.179	0.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์เรื่อง เครื่องวัดความอิมตัวออกซิเจนในเลือดปลายนิ้วที่มีการส่งสัญญาณด้วยบลูทูธ จะแบ่งการทำงานออกเป็น 3 ส่วนคือ ส่วนการออกแบบและสร้างวงจรจับตัวให้กำเนิดแสงสีแดงและแสงอินฟราเรด และวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมตัวออกซิเจนในเลือด ส่วนการประมวลผลสัญญาณเพื่อนำไปคำนวณหาค่าความอิมตัวของออกซิเจนในเลือดปลายนิ้ว และส่วนการเขียนแอปพลิเคชันแสดงค่าความอิมตัวของออกซิเจนในเลือดปลายนิ้วบนอุปกรณ์เคลื่อนที่ โดยได้ทำการศึกษาลักษณะต่าง ๆ เพื่อนำมาประยุกต์ใช้ในการทำปริญญานิพนธ์ และสามารถสรุปผลการดำเนินงานได้ดังนี้

- 1) สามารถออกแบบและสร้างวงจรขยายสัญญาณจากเซนเซอร์ตัวรับแสงของหัววัดความอิมตัวออกซิเจนในเลือด
- 2) สามารถเขียนโปรแกรม MATLAB เพื่อดูภาพรวมการประมวลผลสัญญาณ และคำนวณหาค่าความอิมตัวของออกซิเจนในเลือดปลายนิ้ว ก่อนจะนำขั้นตอนดังกล่าวไปเขียนลงบนไมโครคอนโทรลเลอร์ ESP32
- 3) สามารถออกแบบและสร้างตัวกรองแถบความถี่ผ่าน เพื่อนำไปใช้ในการแยกสัญญาณเอพดีเอ็ม ให้เป็นสัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรดได้
- 4) สามารถออกแบบและสร้างตัวกรองความถี่ต่ำผ่าน เพื่อนำไปใช้ในการตีמודูเลตสัญญาณเอเอ็มของคลื่นแสงสีแดง และสัญญาณเอเอ็มของคลื่นอินฟราเรด ให้เป็นสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดได้ตามลำดับ
- 5) สามารถนำสัญญาณพีพีจีของคลื่นแสงสีแดง และสัญญาณพีพีจีของคลื่นอินฟราเรดมาคำนวณหาค่าความอิมตัวของออกซิเจนในเลือดได้
- 6) สามารถเขียนโปรแกรม Arduino IDE ในการสร้างสัญญาณคลื่นพาร์ การรับสัญญาณเอพดีเอ็มเข้าพอร์ต ADC การประมวลผลสัญญาณ การคำนวณหาค่าความอิมตัวของ

ออกซิเจนในเลือด และการส่งข้อมูลค่าความอิ่มตัวของออกซิเจนในเลือดผ่านบลูทูธของ ไมโครคอนโทรลเลอร์ ESP32

7) สามารถเขียนแอปพลิเคชันแสดงค่าความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วบน อุปกรณ์เคลื่อนที่ได้

8) สามารถนำผลการทดลองมาคำนวณหาประสิทธิภาพการทำงานของเครื่องวัดความ อิ่มตัวของออกซิเจนในเลือดปลายนิ้ว

5.2 ข้อเสนอแนะ

ปริญญาานิพนธ์เรื่อง เครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่มีการส่ง สัญญาณด้วยบลูทูธ สามารถนำมาใช้ในการวัดค่าความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วได้จริง แต่ก็ยังมีข้อจำกัดอยู่ คือ ขนาดของเครื่องวัดความอิ่มตัวของออกซิเจนในเลือดปลายนิ้วที่ค่อนข้างใหญ่ ยังไม่สะดวกในการพกพา

บรรณานุกรม

- [1] บดินทร์ ขวัญนิมิตร. “การวัดความอิมิตัวออกซิเจนของฮีโมโกลบินจากชีพจรในผู้ใหญ่.” สงขลา นครินทร์เวชสาร. (24 เมษายน 2549) : 246-251.
- [2] ยูวณารถ มลทองศิริ. “การศึกษาการวัดปริมาณความอิมิตัวออกซิเจนของฮีโมโกลบินและ อัตราการเต้นของหัวใจด้วยไมโครคอนโทรลเลอร์ ARDUINO NANO.” วิทยานิพนธ์ ปริญญาโทบริหารบัณฑิต, สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์, สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2558.
- [3] LM324. (2021). <https://thai.alibaba.com/product-detail/lm324n-324-lm324-linear-ic-quad-op-amp-dip14-60559255206.html>.
- [4] Ken Shelby. “TL084 Quad Op-Amp.” <https://protosupplies.com/product/tl084-quad-op-amp/>.
- [5] Sound Card. (2021). <https://www.joom.com/en/products/1482405857192154714-190-1-582-1431322592>.
- [6] ESP32 Pin I/O. (2021). <https://www.artronshop.co.th/article/51/esp32>.
- [7] ESP32. (2021). [https://shopee.co.th/%E0%B9%81%E0%B8%9C%E0%B8%87%E0%B8%A7%E0%B8%87%E0%B8%88%E0%B8%A3-WIFI-Bluetooth-\(ESP-WROOM-32\)-i.231496482.6420178399](https://shopee.co.th/%E0%B9%81%E0%B8%9C%E0%B8%87%E0%B8%A7%E0%B8%87%E0%B8%88%E0%B8%A3-WIFI-Bluetooth-(ESP-WROOM-32)-i.231496482.6420178399)
- [8] Gustavo Murta. “ESP-WROOM-32 DEVKIT V4 pinout.” <https://www.pinterest.com/pin/482940760037656205/>.
- [9] Amplitude Modulation. (2016). <http://www.dspace.spu.ac.th/bitstream/123456789/4669/4/6.pdf>.
- [10] Virasit Imtawil. “การมอดูเลต.” <http://eestaff.kku.ac.th/~virasit/192231/commu-book-virasit/Ch4-1.pdf>.
- [11] Tutorialspoint. “Amplitude Modulation.” https://www.tutorialspoint.com/analog_communication/analog_communication_amplitude_modulation.htm.

- [12] พัฒนะ คัยนันท์. “การลดสัญญาณปนเปื้อนจากการเคลื่อนไหวในการวัดความอิมิตัวออกซิเจนในเลือดโดยใช้ตัวกรองแบบปรับตัวได้ร่วมกับเซนเซอร์วัดความเข้มข้น.” วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ, สาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2559.
- [13] พีระพล ยูวณิชิตานนท์. “DSP9 IIR Filter Design การออกแบบตัวกรองดิจิทัลแบบ IIR.” https://embedsigproc.files.wordpress.com/2008/09/dsp_2552_9.pdf.
- [14] วงจรกรองความถี่ Filter Circuit. (2015). <https://navapadol.files.wordpress.com/2015/09/e0b89ae0b897e0b897e0b8b5e0b988-6.pdf>.
- [15] ไหวพจน์ ศรีธัญ. อุปกรณ์อิเล็กทรอนิกส์และวงจร. กรุงเทพฯ : สำนักพิมพ์วังอักษร, 2013.
- [16] Oscilloscope. (2019). <https://www.bellnw.com/manufacture/Tektronix/DPO4032.htm>.
- [17] Power Supply. (2019). <https://www.distrelec.biz/en/triple-output-dc-power-supply-32v-3a-96w-adjustable-precision-bk1672/p/30115295>.
- [18] Kasetony Tekstylne. “Signal Generator.” <http://kasetony-tekstylne.pl/775225/NEW-and-Calibrateed-GW/>.
- [19] Arduino IDE. (2020). <http://xstreamlc.com/product/arduino-programmingmonday-230-400pm-for-age-7/>.
- [20] Proteus. (2021). <https://crackswebsite.com/proteus-crack/>.
- [21] MATLAB. (2021). <https://1000logos.net/matlab-logo/>.
- [22] พิมพ์เพ็ญ พรเฉลิมพงศ์. “Accuracy.” <https://www.foodnetworksolution.com/wiki/word/4289/accuracy/>
- [23] Natalie Harrington. “How to Calculate Precision.” <https://www.wikihow.com/Calculate-Precision/>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งการประมวลผลสัญญาณ

```
clear all; close all; clc;
```

```
% ***** Audio Recorder *****
Fs=44000;
time = 10;
t = 1/Fs: 1/Fs: time;
bits = 16;
sig = audiorecorder(Fs,bits,1);           % 1 channel
recordblocking(sig, time);               % 10 seconds
signal = getaudiodata(sig);
figure(1);                               % แสดงรูปสัญญาณ FDM
plot(t, signal); ylim([-2 2]); grid on

% ***** Signal Processing *****
FDM_SIGNAL = (signal - mean(signal)) * 1;

figure(2);
subplot(311);
plot(t, FDM_SIGNAL); xlabel('Time (sec)'); % แสดงรูปสัญญาณ FDM ที่กำจัด DC
ylabel('Amplitude (Volt)'); title('ORIGINAL FDM SIGNAL');

% Filter with 2 BPFs
F1pass = 395;                             % F1 Carrier
F2pass = 757;                             % F2 Carrier
guardband = 100;

% Band Pass Filter Design
[B1, A1] = butter(3, [(F1pass-guardband) (F1pass+guardband)]/(Fs/2),
'bandpass'); % 1070 Hz
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[B2, A2] = butter(5, [(F2pass-guardband) (F2pass+guardband)]/(Fs/2),
'bandpass');
```

```
AM_RED = filter(B1, A1, FDM_SIGNAL); % BPF RED
```

```
AM_IR = filter(B2, A2, FDM_SIGNAL); % BPF IR
```

```
subplot(312);
```

```
plot(t, AM_RED); xlabel('Time (sec)'); % แสดงรูปสัญญาณ AM ของ RED
ylabel('Amplitude (Volt)'); title('RED AM Signal');
```

```
subplot(313);
```

```
plot(t, AM_IR); xlabel('Time (sec)'); % แสดงรูปสัญญาณ AM ของ IR
ylabel('Amplitude (Volt)'); title('IR AM Signal');
```

```
% Envelope Detector Demodulation
```

```
fullwave_RED = power(AM_RED, 2); % ยกกำลังสองสัญญาณ AM ของ RED
```

```
fullwave_IR = power(AM_IR, 2); % ยกกำลังสองสัญญาณ AM ของ IR
```

```
figure(3);
```

```
subplot(211);
```

```
plot(t, fullwave_RED); % แสดงรูปสัญญาณ AM ยกกำลังสองของ RED
xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('RED PPG Signal'); grid on;
```

```
subplot(212);
```

```
plot(t, fullwave_IR); % แสดงรูปสัญญาณ AM ยกกำลังสองของ IR
xlabel('Time (sec)'); ylabel('Amplitude (Volt)'); title('IR PPG Signal'); grid on;
```

```
% Low Pass Filter Design
```

```
[B3, A3] = butter(5, 10/(Fs/2), 'low');
```

```
RED = filter(B3, A3, fullwave_RED); % LPF RED
```

```
IR = filter(B3, A3, fullwave_IR); % LPF IR
```

```
RED = sqrt(RED);  
IR = sqrt(IR);  
  
figure(4);  
subplot(211);  
plot(t, RED); % แสดงรูปสัญญาณ PPG RED  
xlabel('Time (sec)');ylabel('Amplitude (Volt)'); title('RED PPG Signal'); grid on;  
subplot(212);  
plot(t, IR); % แสดงรูปสัญญาณ PPG IR  
xlabel('Time (sec)');ylabel('Amplitude (Volt)'); title('IR PPG Signal'); grid
```

คำสั่งการออกแบบตัวกรองความถี่แถบผ่าน

```
clear all; close all; clc;
```

```
Ts = 1/1700 ;
```

```
H0 = 2 ;
```

```
f0 = 80; % f0 ของตัวกรองสัญญาณของคลื่นแสงสีแดง
```

```
f0 = 190; % f0 ของตัวกรองสัญญาณของคลื่นคลื่นอินฟราเรด
```

```
BW = 30 ;
```

```
Q = f0 / BW
```

```
W0 = 2 * pi * f0 ;
```

```
B = ( H0 * W0 ) / Q
```

```
SQ = sqrt ( 4 - ( (1/Q)^2 ) );
```

```
A1 = 2 * ( exp ( - ( W0 * Ts ) / ( 2 * Q ) ) ) * cos ( ( SQ * ( W0 * Ts ) / 2 ) );
```

```
A2 = exp ( - ( W0 * Ts ) / Q );
```

คำสั่งการออกแบบตัวกรองความถี่ต่ำผ่าน

```
clear all; close all; clc;
```

```
Ts = 1/1500 ;
```

```
Q = 0.5 ;
```

```
H0 = 2 ;
```

```
f0 = 2 ;
```

```
W0 = 2 * pi * f0 ;
```

```
B = H0 * W0 * W0 ;
```

```
SQ = sqrt ( 4 - ( (1/Q)^2 ) );
```

```
A1 = 2 * ( exp ( - ( W0 * Ts ) / ( 2 * Q ) ) ) * cos ( ( SQ * ( W0 * Ts ) / 2 ) );
```

```
A2 = exp ( - ( W0 * Ts ) / Q );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

คำสั่งการทำงานในโปรแกรม ARDUINO IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งการทำงานบนไมโครคอนโทรลเลอร์ ESP32

```
#include <Ethernet.h>

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH1106.h>
#define OLED_SDA 21
#define OLED_SCL 22
Adafruit_SH1106 display(21, 22);

#include "BluetoothSerial.h"
BluetoothSerial ESP_BT;

#include "Arduino.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <driver/dac.h>
#define l 7600

int countpin = 22 ;
float freq1 = 40; // D25
float freq2 = 17; // D26
float array_1[l] = {} ;
float array_f1_sine[l] = {} ;
float array_f2_sine[l] = {} ;

float yRED_1 = 0 ;
float yRED_2 = 0 ;
float yRED_0 = 0 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float yRED_IL = 0 ;
float yRED_L0 = 0 ;
float yRED_L1 = 0 ;
float yRED_L2 = 0 ;
```

```
float yIR_0 = 0 ;
float yIR_1 = 0 ;
float yIR_2 = 0 ;
float yIR_IL = 0 ;
float yIR_L0 = 0 ;
float yIR_L1 = 0 ;
float yIR_L2 = 0 ;
```

```
float car_RED = 0 ;
float car_RED_0 = 0 ;
float car_RED_sine = 0 ;
float car_IR = 0 ;
float car_IR_sine = 0 ;
```

```
float y_1 = 0;
float y_2 = 0;
float y_3 = 0;
float y_4 = 0;
float PPG_RED = 0;
float PPG_IR = 0;
float ysub_RED = 0;
float ysub_IR = 0;
```

```
float R = 0;
float SpO2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float yRED_MAX = 0;
float yRED_MIN = 0 ;
float yIR_MAX = 0;
float yIR_MIN = 0 ;

int loopnum = l ;
int looptime;
int n = 0;
int i = 0;
int a = 0 ;
float S = 0;
float Sav = 0;

int filename = 0;
const int sensor = 34;

int incoming, c, b;

void setup()
{
  Serial.begin(115200);
  ESP_BT.begin("ESP32_Control"); //Name of your Bluetooth interface
  pinMode(countpin, OUTPUT);
  digitalWrite(countpin, LOW);

  delay(1000);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void loop()
{
  if (ESP_BT.available())
  { Serial.println("ESP_BT.available");
    incoming = ESP_BT.read();
    Serial.print("incoming = "); Serial.println(incoming);
  }

  if (incoming == 1)
  {
    /***** GEN AND RECEIVE *****/
    for (n = 1; n <= loopnum; n++)
    {
      // GEN SINE WAVE
      car_IR = int(128 + (120 * (sin((freq1 * n * PI) / 180)))); // freq1 = 40
      car_RED = int(128 + (120 * (sin((freq2 * n * PI) / 180)))); // freq2 = 17

      dacWrite(25, car_IR); // Sine wave1 190Hz
      dacWrite(26, car_RED); // Sine wave2 80Hz

      // RECEIVE SINE WAVE
      array_f1_sine[n] = car_IR ;
      array_f2_sine[n] = car_RED ;

      array_1[n] = analogRead(sensor) / 5 ; // FDM Signal

      // กำหนดค่าเริ่มต้นของจุดสูงสุดและจุดต่ำสุด
      yRED_MIN = 999999999 ; yRED_MAX = 0 ;
      yIR_MIN = 999999999 ; yIR_MAX = 0 ;
    }
  }
}

```

```

digitalWrite(countpin, !digitalRead(countpin));
delayMicroseconds(490); // fs = 1.7kHz
}
/***** END GEN AND RECESS *****/

/***** PROCESSING *****/
n = 0;
for (n = 2; n <= loopnum; n++)
{
  y_1 = 0.001 * (array_f2_sine[n] - 128); // Carrier RED
  y_3 = 0.001 * (array_f1_sine[n] - 128); // Carrier IR

  // BandPass Filter Design RED
  yRED_0 = (376.9911 * (array_1[n] - array_1[n - 1])) + (1.8129 * yRED_1)
    - (0.8950 * yRED_2);
  yRED_2 = yRED_1;
  yRED_1 = yRED_0;

  // BandPass Filter Design IR
  yIR_0 = (376.9911 * (array_1[n] - array_1[n - 1])) + (1.4471 * yIR_1)
    - (0.8950 * yIR_2);
  yIR_2 = yIR_1;
  yIR_1 = yIR_0;

  // Demodulation
  PPG_RED = yRED_0 * y_1 * (-0.001);
  PPG_IR = yIR_0 * y_3 * (-0.001);

  // Low Pass Filter Design
  yRED_L0 = 710.6115 * (PPG_RED) + 1.9779 * yRED_L1 - 0.9781 * yRED_L2;
  yIR_L0 = 710.6115 * (PPG_IR) + 1.9779 * yIR_L1 - 0.9781 * yIR_L2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ysub_RED = yRED_L0 - yRED_L1 ; // ค่าปัจจุบันลบค่าอดีตของ RED
ysub_IR = yIR_L0 - yIR_L1 ; // ค่าปัจจุบันลบค่าอดีตของ IR

if (n > 850)
{
if (ysub_IR > 0 && (yIR_MAX - yIR_L0) < 0) // ค่าของ IR ที่ได้สูงกว่าค่าเดิม
{
yIR_MAX = yIR_L0 ; // ปรับค่า Maximum IR
}
if (ysub_RED > 0 && (yRED_MAX - yRED_L0) < 0)
// ค่าของ RED ที่ได้สูงกว่าค่าเดิม
{
yRED_MAX = yRED_L0 ; // ปรับค่า Maximum RED
}
if ((ysub_IR < 0) && ((yIR_L0 - yIR_MIN) < 0) // ค่าของ IR ที่ได้น้อยกว่าค่าเดิม
{
yIR_MIN = yIR_L0 ; // ปรับค่า Minimum IR
}
if ((ysub_RED < 0) && ((yRED_L0 - yRED_MIN) < 0))
// ค่าของ RED ที่ได้น้อยกว่าค่าเดิม
{
yRED_MIN = yRED_L0 ; // ปรับค่า Minimum RED
}
}
}

yRED_L2 = yRED_L1 ;
yRED_L1 = yRED_L0 ;
yIR_L2 = yIR_L1 ;
yIR_L1 = yIR_L0 ;
}

/***** END PROCESSING *****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** CAL R & SpO2 *****/
R = ((yRED_MAX - yRED_MIN) / ((yRED_MAX + yRED_MIN) / 2)) /
    ((yIR_MAX - yIR_MIN) / ((yIR_MAX + yIR_MIN) / 2)) ;
if (a < 3)
{
    SpO2 = 110 - (25 * R);
    S = S + SpO2 ;
    a++ ;
    /*ESP_BT.write(SpO2);
    Serial.print("SpO2 = ");
    Serial.println(SpO2);*/
}
else
{
    Sav = S / 3 ; // เฉลี่ยค่า SpO2
    a = 0 ;
    S = 0;
    if (Sav <= 100)
    {
        ESP_BT.write(round(Sav));
    }
    else
    {
        ESP_BT.write(99);
    }
}
}
}
/***** END CAL R & SpO2 *****/
delay(1000);
}
/*****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้