

การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดย
ใช้ต้นทุนต่ำ

DESIGN AND DEVELOPMENT OF LOW-COST ELECTRICAL IMPEDANCE
TOMOGRAPHY EQUIPMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดย
ใช้ต้นทุนต่ำ

DESIGN AND DEVELOPMENT OF LOW-COST ELECTRICAL IMPEDANCE
TOMOGRAPHY EQUIPMENT



นายจิรพันธ์ บำเพ็ญ

61010159

นางสาวนงนภัส วานกระ

61010529

อาจารย์ที่ปรึกษา
รศ.ดร. พิพัฒน์ พรหมมี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ

DESIGN AND DEVELOPMENT OF LOW-COST ELECTRICAL IMPEDANCE

TOMOGRAPHY EQUIPMENT

ผู้จัดทำ

1. นายจิรพันธ์ บำเพ็ญ 61010159
2. นางสาวนนกัศ วานกระ 61010529


..... อาจารย์ที่ปรึกษา
(รศ.ดร. พิพัฒน์ พรหมมี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปริญญาานิพนธ์เรื่อง “การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ” จะไม่สามารถสำเร็จลุล่วงได้เลย หากไม่ได้รับความช่วยเหลือและความอนุเคราะห์จากอาจารย์ที่ปรึกษา คือ รศ.ดร. พิพัฒน์ พรหมมี ที่คอยให้คำปรึกษาและคำสั่งสอนตลอดมา อีกทั้งยังคอยแนะนำ คอยตรวจสอบ แก้ไขข้อบกพร่องทุกขั้นตอนการจัดทำปริญญาานิพนธ์เล่มนี้ รวมถึงสนับสนุนสถานที่ เครื่องมือและอุปกรณ์ต่าง ๆ ที่ใช้ในระหว่างการจัดทำปริญญาานิพนธ์ขอขอบคุณ ดร. ณรงค์ บริจินดากุล อาจารย์ Nanyang Technological University ที่ให้ความรู้ด้าน EIT อีกทั้งพี่ ๆ นักศึกษาปริญญาเอกที่คอยให้ความช่วยเหลือและคำแนะนำเป็นอย่างดี และสุดท้ายเพื่อน ๆ นักศึกษาปริญญาตรีชั้นปีที่ 4 ภาควิชาวิศวกรรมโทรคมนาคมห้อง T108-A ที่เป็นกำลังใจให้กันตลอดมา ทำให้ปริญญาานิพนธ์ชิ้นนี้เป็นรูปเป็นร่างขึ้นมา

ขอกราบพระคุณ บิดา มารดา ครอบครัวและเพื่อน ๆ ทุกคน ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมา และที่สำคัญที่สุดคือการสนับสนุน ให้โอกาสทางด้านการศึกษานั้นมีค่าอย่างยิ่งแก่ผู้จัดทำ

นายจิรพันธ์ บำเพ็ญ
นางสาวนงนภัส วานกระ
ผู้จัดทำ

การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความ
ต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ
DESIGN AND DEVELOPMENT OF LOW-COST
ELECTRICAL IMPEDANCE TOMOGRAPHY EQUIPMENT

โดย นายจิรพันธ์ บำเพ็ญ 61010159

นางสาวนงนภัส ว่านกระ 61010529

อาจารย์ที่ปรึกษา รศ.ดร. พิพัฒน์ พรหมมี

บทคัดย่อ

บทความนี้นำเสนอการออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้า (Electrical Impedance Tomography : EIT) โดยใช้ต้นทุนต่ำ การออกแบบจะประกอบไปด้วยวงจรกำเนิดสัญญาณรูปไซน์ วงจรเปลี่ยนแรงดันเป็นกระแสซึ่งจะใช้เป็นวงจรขยายสัญญาณแบบป้อนกลับกระแส การใช้ไมโครคอนโทรลเลอร์ Arduino Uno R3 ในการควบคุมมัลติเพล็กซ์เซอร์สำหรับการทำงานในส่วนของการรับ-ส่งสัญญาณ รูปแบบการทำงานจะทำการจ่ายสัญญาณไซน์รูปแบบกระแสให้กับตัวโพรบที่ถูกติดตั้งไว้รอบตัว Vessel จากนั้นเมื่อจ่ายสัญญาณไซน์ วงจรเรียงกระแสแม่ข่ายแบบเต็มคลื่นและมัลติเพล็กซ์เซอร์จะทำหน้าที่รับสัญญาณและจ่ายสัญญาณในแต่ละจุด หลังจากที่ได้รับสัญญาณจะแปลงสัญญาณแรงดันไฟฟ้าจากอนาล็อกเป็นสัญญาณดิจิทัลเพื่อส่งไปประมวลผล แสดงตำแหน่งวัตถุที่อยู่ใน Vessel ซึ่งเครื่องมือที่ใช้ในการประมวลผลใช้ซอฟต์แวร์ที่มีชื่อว่า EIDORS ไฟไนต์เอลิเมนต์ บนโปรแกรม MATLAB โดยในบทความนี้มีการทดลองจริงเพื่อยืนยันหลักการสร้างของงานที่ออกแบบมา พบว่าเป็นไปตามทฤษฎีทุกประการ

ABSTRACT

This paper presents the design and development of a low-cost Electrical Impedance Tomography (EIT) device. The various subsystems, sinusoidal oscillator, voltage-to-current converter (V/I), multiplexer microcontroller (Arduino UNO R3) is incorporated to be transmitter and receiver. The current-feedback OPAMP (CFOA) is required as a V/I converter. The sinusoidal current injects into the probes located in circular vessel positions in the transmitter. The rectifier circuit and multiplexer provide multiple positions receiving signals. The receiving signals have been converted to digital signals and sent to reconstruct to detect the vessel's object. The MATLAB-based EIDORS finite element program is used to generate the image from the receiving data. The experimental results have confirmed the workability of the different conductivity objects tested in the vessel.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 หลักการและวิธีการทำงานของชุดอุปกรณ์ EIT	2
2.2 โปรแกรม Proteus	5
2.3 วงจรขยายสัญญาณแบบป้อนกลับกระแส	6
2.4 วงจรเรียงกระแส (Rectifier Circuits)	7
2.5 ไมโครคอนโทรลเลอร์	11
2.6 วงจร Multiplexer	14
2.7 โปรแกรมการสร้างภาพตัดขวางความต้านทานไฟฟ้า EIDORS	17
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	23
3.1 การออกแบบโครงสร้างโดยรวม	23
3.2 เครื่องมือที่ใช้ในการทดลอง	31
3.3 การจัดเก็บผลการทดลอง	32

สารบัญ (ต่อ)

	หน้า
บทที่ 4	
ผลการทดลอง	33
4.1 วงจรจ่ายกระแสสำหรับชุด EIT	33
4.2 วงจรรับกระแสสำหรับชุด EIT	41
4.3 ทดสอบการทำงานของวงจรทั้งหมด	49
4.4 การวิเคราะห์และประมวลผลสัญญาณดิจิทัลเป็นภาพ	57
บทที่ 5	
สรุปผลและข้อเสนอแนะ	61
5.1 สรุปผล	61
5.2 ข้อเสนอแนะ	61
บรรณานุกรม	62
ภาคผนวก Coding	64

สารบัญรูป

รูปที่	หน้า
2.1	3
การจำลองการวัด EIT ที่มีอิเล็กโทรดติดอยู่กับลำตัวผู้ป่วยในระยะห่างเท่ากัน และมีอุปกรณ์อิเล็กทรอนิกส์สำหรับการแสดงภาพ	
2.2	4
2 หลักการหมุนตำแหน่งของการจ่ายกระแสไฟฟ้า (I) และการวัดแรงดันตกคร่อม (V) ในการ กำหนดค่าอิเล็กโทรด	
2.3	5
ลำดับวิถัจกรการหายใจสี่รู้งโดยใช้ภาพ EIT สัมพัทธ์ตาม ΔZ_n	
2.4	6
ไอคอนโปรแกรม Proteus 8 Professional	
2.5	7
โครงสร้างพื้นฐานของวงจรขยายสัญญาณแบบป้อนกลับกระแส	
2.6	8
วงจรงเรียงกระแสแมนย่าแบบครึ่งคลื่น	
2.7	8
สัญญาณที่ออกมาจากวงจรงเรียงกระแสแมนย่าแบบครึ่งคลื่น	
2.8	9
การทำงานของวงจรงเรียงกระแสแมนย่าแบบครึ่งคลื่นของช่วงครึ่ง Cycle บวกและช่วง Cycle ลบ	
2.9	10
วงจรงเรียงกระแสแมนย่าแบบเต็มคลื่น	
2.10	11
วงจรงเรียงกระแสแมนย่าแบบเต็มคลื่นที่สร้างจากวงจรงขยายผลต่างของสัญญาณ	
2.11	11
บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	
2.12	15
โครงสร้างการทำงานของ MUX	
2.13	16
Pin Configuration	
2.14	16
Function Diagrams	
2.15	18
โครงสร้างของ fwd_model	
2.16	19
โครงสร้างของ Data	
2.17	20
โครงสร้างของ inv_model	
2.18	20
โครงสร้างของ Imaging	
2.19	22
ตัวอย่างการสร้างภาพแบบ 2 มิติด้วยวิธี Difference imaging	
3.1	23
บล็อกไดอะแกรมภาพรวมของโครงการงาน	

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.2 วงจรขยายสัญญาณแบบป้อนกลับกระแสโดยใช้ CFOA เบอร์ AD844	24
3.3 การออกแบบวงจรรับกระแส โดยใช้ Op-Amp เบอร์ LF347	24
3.4 โค้ดการทำงานของวงจรควบคุมการจ่ายกระแส	25
3.5 โค้ดการทำงานของวงจรควบคุมการรับกระแส	26
3.6 Schematic ของวงจรจ่ายกระแส	27
3.7 PCB Layout สำหรับวงจรจ่ายกระแสที่มีการจัดเรียงเรียบร้อย	28
3.8 แผ่น PCB ของวงจรจ่ายกระแส	28
3.9 Schematic ของวงจรรับกระแส	29
3.10 PCB Layout สำหรับวงจรรับกระแสที่มีการจัดเรียงเรียบร้อย	30
3.11 โค้ดการทำงานการแปลงสัญญาณดิจิตอลเป็นภาพ	31
4.1 สัญญาณอินพุตเทียบกับเอาต์พุตจากโปรแกรม Proteus	33
4.2 วงจรขยายสัญญาณแบบป้อนกลับกระแสโดยใช้ CFOA เบอร์ AD844	34
4.3 สัญญาณอินพุตเทียบกับเอาต์พุตจากวงจรจ่ายกระแส	34
4.4 วงจรทดสอบควบคุมการจ่ายกระแสที่ออกแบบในโปรแกรม Proteus	35
4.5 สัญญาณที่ออกมาจาก Channel 0 และ Channel 1	36
4.6 สัญญาณที่ออกมาจาก Channel 2 และ Channel 3	36
4.7 รูปแบบการจ่ายกระแสและการวัดแรงดันไฟฟ้า Vessel	37
4.8 การทดสอบวงจรจ่ายกระแส โดยใช้ Arduino Uno R3 ในการควบคุม	37
4.9 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 1 และตัวที่ 2	38
4.10 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 3 และตัวที่ 4	38
4.11 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 5 และตัวที่ 6	38
4.12 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 7 และตัวที่ 8	39
4.13 ทดสอบการทำงานของ PCB ของวงจรจ่ายกระแส	39
4.14 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 1 และตัวที่ 2	40

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.15 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 3 และตัวที่ 4	40
4.16 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 5 และตัวที่ 6	40
4.17 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 7 และตัวที่ 8	41
4.18 ผลลัพธ์ของสัญญาณที่ได้ออกมา	42
4.19 การต่อวงจรรับกระแสโดยใช้ LF347 ลงบนแผ่นไฟโต้บอร์ด	42
4.20 ผลลัพธ์ของสัญญาณที่ได้ออกมาจากวงจรรับกระแส	43
4.21 วงจรทดสอบควบคุมการรับกระแสในโปรแกรม Proteus	44
4.22 ผลลัพธ์ของสัญญาณที่ได้ออกมา	45
4.23 การทดสอบวงจรรับกระแส โดยใช้ Arduino Uno R3 ในการควบคุม	45
4.24 สัญญาณที่วัดได้จาก Electrode ตัวที่ 1 และตัวที่ 2	46
4.25 สัญญาณที่วัดได้จาก Electrode ตัวที่ 3 และตัวที่ 4	46
4.26 ทดสอบการทำงานของ PCB ของวงจรรับกระแส	47
4.27 สัญญาณที่วัดได้จาก Electrode ตัวที่ 1 และตัวที่ 2	47
4.28 สัญญาณที่วัดได้จาก Electrode ตัวที่ 3 และตัวที่ 4	48
4.29 สัญญาณที่วัดได้จาก Electrode ตัวที่ 5 และตัวที่ 6	48
4.30 สัญญาณที่วัดได้จาก Electrode ตัวที่ 7 และตัวที่ 8	48
4.31 โค้ดการทำงานของวงจรจ่ายกระแสและรับกระแส	49
4.32 วงจรการทำงานทั้งหมดบนโปรแกรม Proteus โดยใช้ Arduino ควบคุม	50
4.33 ผลลัพธ์ของสัญญาณที่ได้จากการทดสอบบนโปรแกรม Proteus	51
4.34 วงจรทั้งหมดบนแผ่นไฟโต้บอร์ดโดยใช้ Arduino ควบคุมการทำงาน	51
4.35 สัญญาณที่ได้จาก Electrode ตัวที่ 1 และตัวที่ 2 ในการทดสอบการทำงานวงจรทั้งหมด บนไฟโต้บอร์ดโดยใช้ Arduino ควบคุม	52
4.36 วงจรทั้งหมดบนแผ่น PCB โดยใช้ Arduino ควบคุมการทำงาน	52

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.37 สัญญาณที่ได้จาก Electrode ตัวที่ 1 และตัวที่ 2 ในการทดสอบการทำงานวงจรทั้งหมด บนแผ่น PCB โดยใช้ Arduino ควบคุม	53
4.38 การวัดสัญญาณจาก Multiplexer	53
4.39 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 1, Channel 2 และ Channel 3	54
4.40 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 4, Channel 5 และ Channel 6	54
4.41 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 7 และ Channel 8	55
4.42 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 1, Channel 2 และ Channel 3	55
4.43 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 4, Channel 5 และ Channel 6	56
4.44 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 7 และ Channel 8	56
4.45 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าสูง	58
4.46 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุ	59
4.47 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าสูงและความนำไฟฟ้าต่ำ	60

สารบัญตาราง

ตารางที่		หน้า
2.1	Pin Function ของ Multiplexer เบอร์ 4051	17
4.1	Truth Table ของ Multiplexer เบอร์ 4051	25
4.2	การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าสูง	57
4.3	การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุ	58
4.4	การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าสูงและความนำไฟฟ้าต่ำ	60

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ จัดทำขึ้นโดยมีจุดประสงค์ที่สำคัญ เพื่อให้สามารถเกิดเป็นชุดต้นแบบของอุปกรณ์การสร้างภาพจากความต้านทานที่มีต้นทุนต่ำได้สำเร็จ แม้ว่าจะตัวงานจะมีต้นทุนต่ำประสิทธิภาพของตัวชิ้นงานก็ยังสามารถมีประสิทธิภาพที่ดีใกล้เคียงกับตัวชิ้นงานที่มีอยู่ในท้องตลาดได้ ซึ่งหากสามารถทำให้สำเร็จขึ้นมาได้ ก็จะสามารถนำไปประยุกต์ใช้งานได้หลากหลายด้าน ไม่ว่าจะเป็นทางด้านการแพทย์ ด้านอุตสาหกรรม ด้านธรณีฟิสิกส์ และด้านอื่น ๆ ในอนาคตต่อไป

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาการออกแบบอุปกรณ์การสร้างภาพความต้านทานไฟฟ้า
- 2) เพื่อพัฒนาอุปกรณ์การสร้างภาพความต้านทานไฟฟ้า
- 3) เพื่อจัดทำต้นแบบชุดจำลอง EIT โดยใช้ต้นทุนต่ำ

1.3 ขอบเขตของปริญญาานิพนธ์

- 1) ทำวงจรจ่ายและรับกระแสได้ โดยใช้อุปกรณ์อิเล็กทรอนิกส์
- 2) ใช้ Multiplexer ในการแบ่งกระแสสัญญาณได้ โดยใช้ Arduino ในการควบคุม
- 3) ใช้คอมพิวเตอร์ส่วนบุคคลและซอฟต์แวร์ในการประมวลให้ได้ภาพที่เราต้องการ
- 4) ต่อยอดอุปกรณ์การสร้างภาพความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 หลักการและวิธีการทำงานของชุดอุปกรณ์ EIT

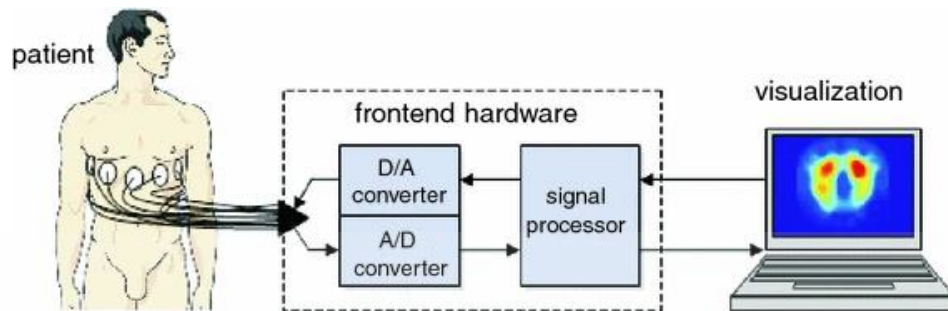
2.1.1 การสร้างภาพจากความต้านทานไฟฟ้า (EIT)

เอกซเรย์อิมพีแดนซ์ไฟฟ้า (EIT) ใช้กระแสไฟฟ้าความถี่ต่ำเพื่อตรวจสอบร่างกาย วิธีการนี้มีความไวต่อการเปลี่ยนแปลงของการนำไฟฟ้า โดยการจ่ายปริมาณกระแสไฟฟ้าที่กำหนดไว้ และทำการวัดสนามศักย์ไฟฟ้าที่เกิดขึ้นที่พื้นผิวของร่างกาย เป็นไปได้ที่จะคิดกลับข้อมูลดังกล่าวเพื่อกำหนดการนำไฟฟ้าหรือความต้านทานบริเวณพื้นผิวของร่างกายที่ถูกตรวจสอบโดยกระแส วิธีนี้ยังสามารถใช้ในหลักการเปลี่ยนแปลงภาพในค่าคงที่ไดอิเล็กตริกที่ความถี่สูง ซึ่งเป็นเหตุผลว่าทำไมวิธีการนี้จึงมักเรียกว่าเอกซเรย์อิมพีแดนซ์ แทนที่จะเป็นการนำไฟฟ้าหรือการต้านทานเอกซเรย์ อย่างไรก็ตามในแง่ของการพัฒนาอย่างเต็มที่จนถึงปัจจุบัน คือการถ่ายภาพของการนำไฟฟ้า/ความต้านทานไฟฟ้า แม้ว่าวิธีการของ EIT ยังไม่ได้รับการยอมรับในกลุ่มการถ่ายภาพทางการแพทย์ แต่ก็แสดงให้เห็นว่าทำงานได้ดีทั้งทางธรณีวิทยาและทางอุตสาหกรรม ดังนั้นจึงเป็นไปได้ที่การประยุกต์ใช้ภาพทางการแพทย์ในอนาคต อาจตามมาค่อนข้างเร็วจากความก้าวหน้าที่ทำขึ้นสำหรับการใช้งานด้านอื่น ๆ [1]

2.1.2 หลักการพื้นฐานของเอกซเรย์อิมพีแดนซ์ไฟฟ้า (EIT)

ในเนื้อเยื่อชีวภาพ การนำกระแสไฟฟ้าขึ้นอยู่กับปริมาณของของเหลว และความเข้มข้นของไอออนในเนื้อเยื่อ เมื่อพิจารณาปอดจะสังเกตได้ว่ากระแสไฟฟ้าไม่สามารถไหลผ่านถุงลมได้ แต่จะไหลผ่านผนังผนังกันถุงลมเท่านั้น ในขณะที่การเพิ่มปริมาตรของถุงลม เมื่อถุงลมขยายตัวจึงทำให้ความต้านทานไฟฟ้าของเนื้อเยื่อเพิ่มขึ้น และการลดปริมาตรของถุงลม จะทำให้ความต้านทานไฟฟ้าของเนื้อเยื่อลดลง

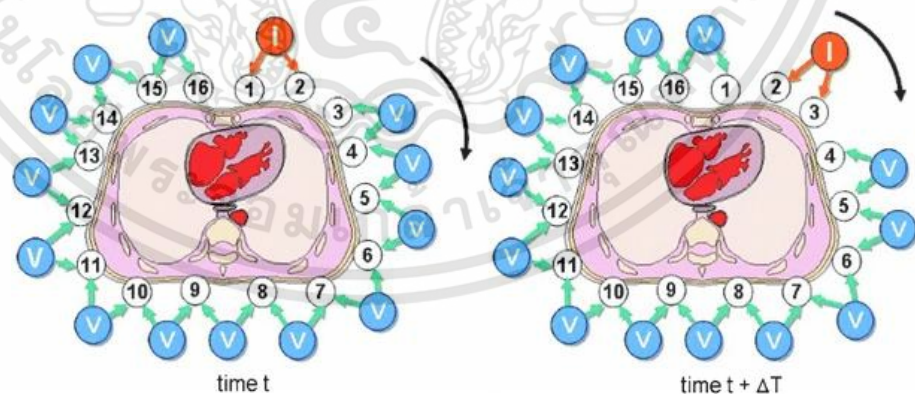
ใน EIT ความต้านทานไฟฟ้าที่กระจายตัวอยู่ทุกเซลล์ภายในส่วนตัดขวางของร่างกาย จะแสดงเป็นภาพเอกซเรย์ เนื่องจากอิมพีแดนซ์ Z ถูกกำหนดโดยแรงดันไฟฟ้า V และกระแส I อิมพีแดนซ์ที่ได้ อาจคำนวณได้จากการวัดกระแส และแรงดันไฟฟ้าที่สอดคล้องกันที่พื้นผิว ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 การจำลองการวัด EIT ที่มีอิเล็กโทรดติดอยู่กับลำตัวผู้ป่วยในระยะห่างเท่ากัน และมีอุปกรณ์อิเล็กทรอนิกส์สำหรับการแสดงภาพ

มีคู่อิเล็กโทรดสำหรับจ่ายกระแสและอิเล็กโทรดอีกคู่จะทำกรวัดแรงดันไฟฟ้า โดยทั่วไปจะมีการกำหนดค่ากระแสที่จ่ายเข้าไป เช่น 5 มิลลิแอมป์ที่ความถี่ 50 กิโลเฮิร์ตซ์ เพื่อให้มีการเว้นการทำงานของอิเล็กโทรดและผิวหนังที่บริเวณนั้น จึงทำให้ EIT ขยายการทำงานไปโอิมพีแดนซ์อิเล็กโทรด โดยการเปลี่ยนตัวจ่ายกระแสและวัดแรงดันไฟฟ้าทั่วร่างกายในระนาบตัดขวาง โดยทั่วไปสำหรับการตรวจปอดจะใช้พื้นที่ว่างระหว่างซี่โครงซี่ที่ 4 ถึง 6

ในการจัดเรียงอิเล็กโทรดที่เรียงตัวกันอยู่บนผิวหนังเป็นลักษณะเส้นรอบนั้น มีวิธีรวมอิเล็กโทรดเข้าด้วยกัน โดยโครงแบบที่นิยมใช้กันมากที่สุดเรียกว่า “การกำหนดค่าโวลท์ที่อยู่ติดกัน” และรวมอิเล็กโทรดที่อยู่ใกล้เคียงเข้าด้วยกันเป็นคู่ของอิเล็กโทรดที่อยู่ติดกัน ดังแสดงในรูปที่ 2.2 แสดงให้เห็นถึงหลักการหมุนตำแหน่งของการจ่ายกระแสไฟฟ้า (I) และการวัดแรงดันตกรวม (V) ในการกำหนดค่าอิเล็กโทรดนี้



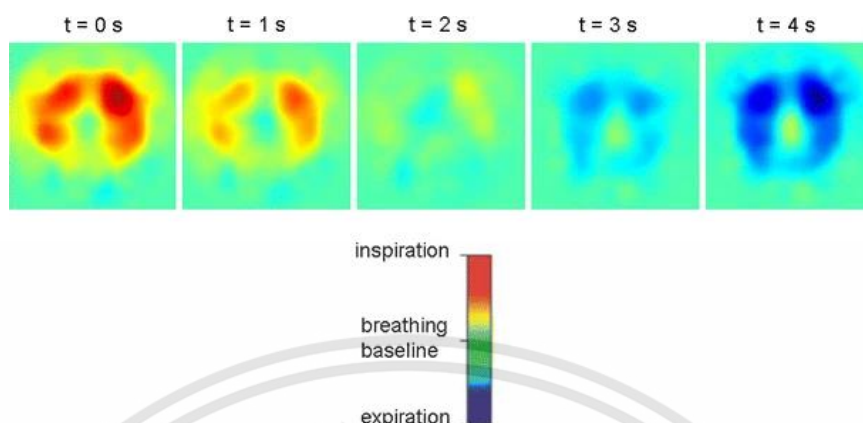
รูปที่ 2.2 หลักการหมุนตำแหน่งของการจ่ายกระแสไฟฟ้า (I) และการวัดแรงดันตกรวม (V) ในการกำหนดค่าอิเล็กโทรด

จากภาพจะเห็นได้ว่า $N = 16$ อิเล็กโทรดในโครงแบบไดรฟ์ที่อยู่ติดกันที่มีการเปลี่ยนตำแหน่งการจ่ายกระแสและวัดแรงดัน ดังแสดงในรูปที่ 2.2 ทางซ้ายมือ กระแสไฟฟ้า I จะถูกจ่ายที่เวลา t ที่อิเล็กโทรด 1 และ 2 ในขณะที่แรงดันไฟฟ้าที่สอดคล้องกันจะถูกวัดตามลำดับที่คู่อิเล็กโทรดที่เหลืออยู่ หลังจากนั้นในเวลา $t + \Delta T$ ตำแหน่งการจ่ายกระแสจะหมุนไปยังอิเล็กโทรดคู่ถัดไป (อิเล็กโทรดตัวที่ 2 และ 3 ดังแสดงในรูปที่ 2.2 ทางขวามือ) และแรงดันไฟฟ้าที่สอดคล้องกันจะถูกวัดตามลำดับที่อิเล็กโทรดอีก 13 คู่อิเล็กโทรดที่เหลืออยู่ สามารถประมวลผลเป็นหนึ่งภาพได้ทันทีเมื่อคู่อิเล็กโทรดเวียนจ่ายกระแสครบทุกคู่

ความละเอียดชั่วขณะของ EIT ค่อนข้างสูง (13-50 เฟรม/วินาที) ซึ่งอาจมีศักยภาพครอบคลุมทั้งระบบการหายใจแบบเรียลไทม์ ในทางตรงกันข้ามความละเอียดเชิงพื้นที่ของภาพจะต่ำอย่างมาก จึงต้องจำกัดระยะห่างระหว่างอิเล็กโทรดให้อยู่ในช่วงที่ดีที่สุด (ประมาณ 2-3 เซนติเมตร)

การได้ภาพ EIT ที่สมบูรณ์ที่แม่นยำเป็นเรื่องยาก เนื่องจากต้องใช้ความรู้โดยละเอียดเกี่ยวกับรูปทรงและการวางตำแหน่งอิเล็กโทรดสมบูรณ์ เพื่อหลีกเลี่ยงปัญหาและปรับปรุงคุณภาพของภาพ EIT Barber และต่อมา Hahn นำเสนอแนวคิดของการสร้างภาพ EIT “สัมพัทธ์” ในที่นี้ อิมพีแดนซ์เฉลี่ยในช่วงเวลาหนึ่งจะถูกลบออกจากค่าอิมพีแดนซ์จริง ซึ่งกำหนดการขยายคอนทราสต์และยกเลิกองค์ประกอบทางเรขาคณิตส่วนใหญ่ที่พบได้ทั่วไปในภาพ EIT สมบูรณ์ อย่างไรก็ตามราคาที่ย่ำแย่คือการสูญเสียข้อมูลอิมพีแดนซ์สัมบูรณ์ กล่าวคือมีเพียงการเปลี่ยนแปลงอิมพีแดนซ์ ΔZ_n เท่านั้นที่ยังคงมองเห็นได้ ในขณะที่ข้อมูลเกี่ยวกับการช่วยหายใจในระดับพื้นฐานจะถูกเก็บรักษาไว้ การวินิจฉัยแยกโรคของบริเวณอิมพีแดนซ์คงที่ (ไม่มีการหายใจ) จะไม่สามารถทำได้โดยตรงอีกต่อไป (เช่น การแยกความแตกต่างระหว่าง , ปอดบวม, ภาวะลมรั่วที่ปอด, ภาวะถุงลมในปอดขยายตัวได้ไม่เต็มที่)

วิธีทั่วไปในการแสดงภาพ EIT คือการใช้รูปแบบสีรุ้ง โดยให้สีแดงหมายถึงการหายใจเข้า (ค่าอิมพีแดนซ์สัมพัทธ์สูงสุด) สีเขียวสำหรับการหายใจที่อยู่ในเกณฑ์มาตรฐาน และสีน้ำเงินหมายถึงการหายใจออก (ค่าอิมพีแดนซ์สัมพัทธ์ต่ำสุด) ดังแสดงในรูปที่ 2.3 ที่แสดงตัวอย่างการแสดงผลภาพที่ได้



รูปที่ 2.3 ลำดับวีดิทัศน์การหายใจสร้างโดยใช้ภาพ EIT สัมพันธ์ตาม ΔZ_{L}

ในเอกสารสรุปล่าสุดได้มีการแนะนำมาตราส่วนสีที่แตกต่างกัน โดยที่สีแดงหมายถึงอิมพีแดนซ์ต่ำ และสีน้ำเงินหมายถึงอิมพีแดนซ์สูงโดยมีการเพิ่มสีดำเข้าไป อุปกรณ์เชิงพาณิชย์บางตัวอาจมีการใช้โทนสีที่คล้ายกันตั้งแต่สีดำไปจนถึงสีน้ำเงินเข้มสำหรับการหายใจออก[2]

2.2 โปรแกรม Proteus

โปรแกรม Proteus หรือ Proteus VSM (Virtual System Modeling) เป็นโปรแกรมที่พัฒนาขึ้น โดยบริษัท แล็บเซ็นเตอร์อิเล็กทรอนิกส์ จำกัด (Lab center Electronics Ltd.) ที่ประเทศอังกฤษ โปรแกรม Proteus มีชื่อเต็มว่า Lab center Electronics Proteus ซึ่งภายในโปรแกรมจะประกอบด้วยส่วนประกอบหลัก 2 ส่วน คือ ISIS และ ARES โปรแกรม Proteus จะมีอยู่หลายเวอร์ชันให้เลือกใช้งาน ซึ่งเวอร์ชันในปัจจุบัน คือเวอร์ชัน 8 ซึ่งมีไอคอนตัวโปรแกรมดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ไอคอนโปรแกรม Proteus 8 Professional

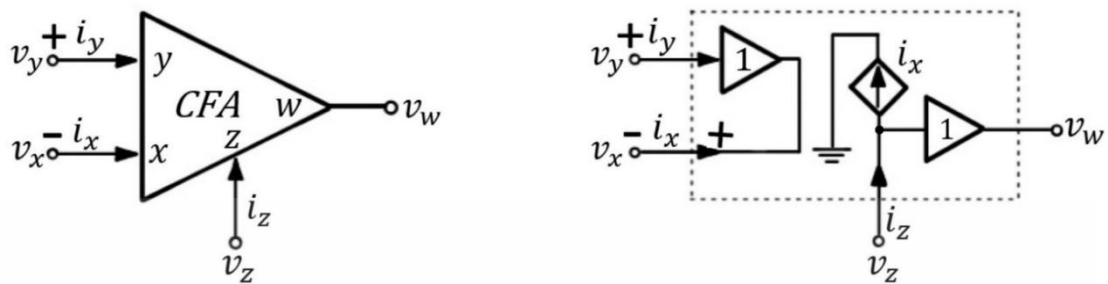
โปรแกรม Proteus เป็นโปรแกรมที่มีความสามารถในการจำลองการทำงานของวงจรอิเล็กทรอนิกส์ได้หลากหลายรูปแบบ ไม่ว่าจะเป็นแบบอนาล็อกและแบบดิจิทัล หรือทั้งแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนาล็อกและดิจิทัลผสมกัน นอกจากนี้ Proteus ยังสามารถออกแบบลายวงจรพิมพ์ (PCB) ได้อีกด้วย จุดเด่นของโปรแกรม Proteus ที่เป็นที่ยอมรับและชื่นชอบก็คือ การจำลองการทำงานของวงจรอิเล็กทรอนิกส์ ที่ใช้ไมโครคอนโทรลเลอร์ตระกูลต่าง ๆ ไม่ว่าจะเป็น PIC, MCS-51, AVR และ ARM เป็นต้น ทำให้นักเขียนโปรแกรมหรือโปรแกรมเมอร์สามารถตรวจสอบได้ว่าโปรแกรม หรือซอร์สโค้ด (Source Code) ที่เขียนขึ้นมานั้น สามารถสนับสนุนกับวงจรฮาร์ดแวร์ที่ต่อได้หรือไม่ ถ้าโปรแกรม (Source Code) ที่เขียนขึ้น ไม่สนับสนุนกับวงจรฮาร์ดแวร์ที่ต่อโปรแกรมเมอร์ก็จะทำการพัฒนาโปรแกรม (Source Code) ที่เขียนขึ้นใหม่ หรือปรับปรุงวงจรฮาร์ดแวร์ใน Proteus จนกว่าโปรแกรมที่เขียนขึ้นและฮาร์ดแวร์ที่ต่อ สามารถสนับสนุนซึ่งกันและกัน ทำให้การสร้างโครงงานต่าง ๆ สามารถประหยัดเวลาและค่าใช้จ่ายเป็นอย่างมาก เพราะในอดีตการเขียนโปรแกรมขึ้นมานั้น จะต้องต่อวงจรจริงเพื่อทดสอบ ทำให้เสียเวลาและค่าใช้จ่ายมาก ในกรณีที่วงจรฮาร์ดแวร์และโปรแกรมที่เขียนขึ้นไม่สนับสนุนซึ่งกันและกัน

2.3 วงจรขยายสัญญาณแบบป้อนกลับกระแส

วงจรขยายสัญญาณแบบป้อนกลับกระแส (Current Feedback Operational Amplifier) จะประกอบไปด้วยองค์ประกอบหลัก 3 ส่วน คือ ส่วนที่หนึ่งวงจรตามแรงดันฝั่งอินพุต ส่วนที่สองวงจรสะท้อนกระแส และส่วนที่สามวงจรตามแรงดันฝั่งเอาต์พุต วงจรตามแรงดันฝั่งอินพุตจะต่ออยู่ระหว่างขั้วอินพุตทั้งสองของวงจรขยาย ขั้วบวกของวงจรขยายคือขั้วอินพุตของวงจรตามแรงดันและขั้วลบของวงจรขยายคือขั้วเอาต์พุตของวงจรตามแรงดัน ทำให้ขั้วบวกของวงจรขยายมีค่าอิมพีแดนซ์ที่สูง และขั้วลบของวงจรขยายมีค่าอิมพีแดนซ์เท่ากับศูนย์ วงจรสะท้อนกระแสซึ่งต่ออยู่ในภาคที่สองของวงจรจะคัดลอกกระแสจากขั้วลบของวงจรขยายและสะท้อนสู่พอร์ต Z ซึ่งมีค่าอิมพีแดนซ์สูงเพื่อแปลงกระแสเป็นแรงดัน ค่าอิมพีแดนซ์ที่พอร์ต X จะเป็นตัวกำหนดกระแสที่พอร์ต Z วงจรตามแรงดันฝั่งเอาต์พุตซึ่งต่ออยู่ในส่วนสุดท้ายของวงจรจะทำหน้าที่ส่งผ่านแรงดันจากพอร์ต Z ไปสู่เอาต์พุตของวงจร และทำหน้าที่ในการขับโหลด ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 โครงสร้างพื้นฐานของวงจรรขยายสัญญาณแบบบ่อนกลับกระแส

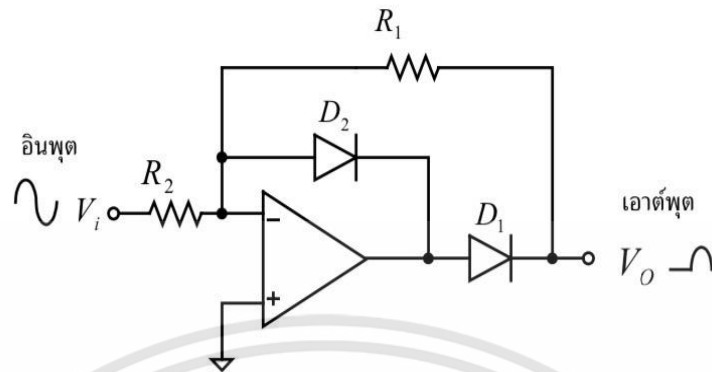
2.4 วงจรเรียงกระแส (Rectifier Circuits)

วงจรรเรียงกระแส (Rectifier Circuit) โดยทั่วไปแล้วไม่ว่าจะเป็นแบบครึ่งคลื่นหรือแบบเต็มคลื่น เราจะพบว่าเมื่อไดโอดนำกระแสแล้วจะต้องมีแรงดันตกคร่อมที่ค่าหนึ่งเสมอถ้าเป็นไดโอดชนิดซิลิกอนจะมีแรงดันตกคร่อมที่ค่าๆหนึ่งเสมอ ถ้าเป็นไดโอดชนิดซิลิกอนจะมีแรงดันตกคร่อมขณะนำกระแสประมาณ 0.7 V ในขณะที่ไดโอดชนิดเจอร์เมเนียมมีค่าประมาณ 0.3 V ถ้าต้องการจะใช้วงจรรเรียงกระแสสัญญาณไฟสลับที่มีแอมพลิจูดต่ำ เช่น 200 mV (0.2 V) เราจะพบว่า ไดโอดทั้งสองชนิดไม่สามารถจะนำกระแสได้เลย หมายความว่า จะไม่มีสัญญาณออกมาที่เอาต์พุตของวงจรรเรียงกระแสเลย

ดังนั้นในกรณีที่เรากำลังต้องการเรียงกระแสหรือเปลี่ยนสัญญาณไฟสลับที่มีแอมพลิจูดต่ำให้เป็นไฟตรงเราจะใช้วงจรรเรียงกระแสอีกแบบหนึ่งที่เรียกว่า “วงจรรเรียงกระแสแบบแม่นยำ” (Precision Rectifier) โดยวงจรรนี้จะใช้ออปแอมป์มาใช้ในการทำงานด้วย ซึ่งจะแบ่งออกเป็น 2 ชนิดเช่นเดียวกับวงจรรเรียงกระแสธรรมดา ได้แก่ วงจรรเรียงกระแสแม่นยำแบบครึ่งคลื่น (Halfwave precision rectifier) และวงจรรเรียงกระแสแม่นยำแบบเต็มคลื่น (Fullwave precision rectifier)

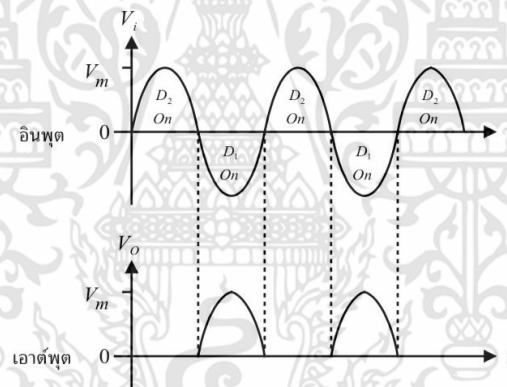
2.4.1 วงจรรเรียงกระแสแม่นยำแบบครึ่งคลื่น

วงจรรเรียงกระแสแม่นยำแบบครึ่งคลื่นดังแสดงดังรูปที่ 2.6 จะมีการใช้ไดโอดเหมือนกับในวงจรรเรียงกระแสแบบธรรมดาพร้อมกับออปแอมป์เพื่อช่วยในการกำจัดผลของแรงดันที่ตกคร่อมในตัวไดโอด



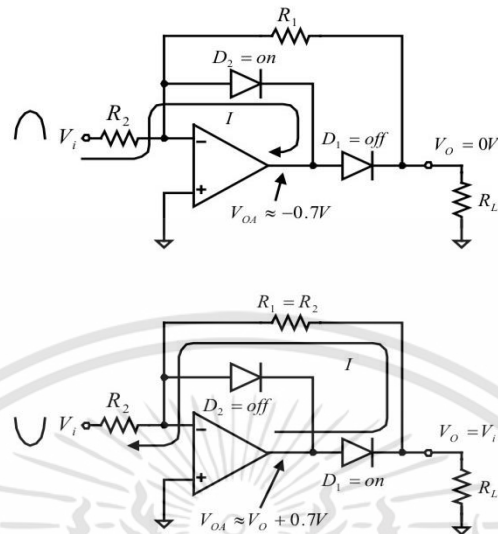
รูปที่ 2.6 วงจรเรียงกระแสแม่นยำแบบครึ่งคลื่น

สำหรับการทำงาน เมื่อสัญญาณอินพุต V_i มีค่าเป็นบวก ไดโอด D_2 ได้รับไบอัสตรง เนื่องจากมีกระแสผ่านตัวมัน แรงดันเอาต์พุต V_o ในช่วงนี้จะมีค่าเป็นศูนย์ เพราะไดโอด D_1 ได้รับไบอัสกลับ พิจารณาดังแสดงในรูปที่ 2.7 ช่วงนี้เอาต์พุตของออปแอมป์มีแรงดันประมาณ -0.7 V



รูปที่ 2.7 สัญญาณที่ออกมาจากวงจรเรียงกระแสแม่นยำแบบครึ่งคลื่น

เมื่ออินพุต V_i มีค่าเป็นลบ จะสังเกตว่าไดโอด D_2 ได้รับไบอัสกลับและในช่วงนี้ ไดโอด D_1 จะได้รับไบอัสตรง เนื่องจากออปแอมป์สร้างแรงดันบวกออกมาที่เอาต์พุต จึงมีกระแสไหลผ่านทั้ง D_1 ไปที่ R_1 และ R_2 แล้วกลับไป V_i เมื่อเรากำหนดให้ $R_1 = R_2$ ก็จะได้ว่า $V_o = -V_i$ (ในช่วง V_i เป็นลบ) นั่นคือจะได้ V_o เป็นสัญญาณบวกนั่นเอง และสามารถสรุปการทำงานของวงจรในช่วง V_i เป็น Cycle บวก และครึ่ง Cycle ลบ ดังแสดงในรูปที่ 2.8

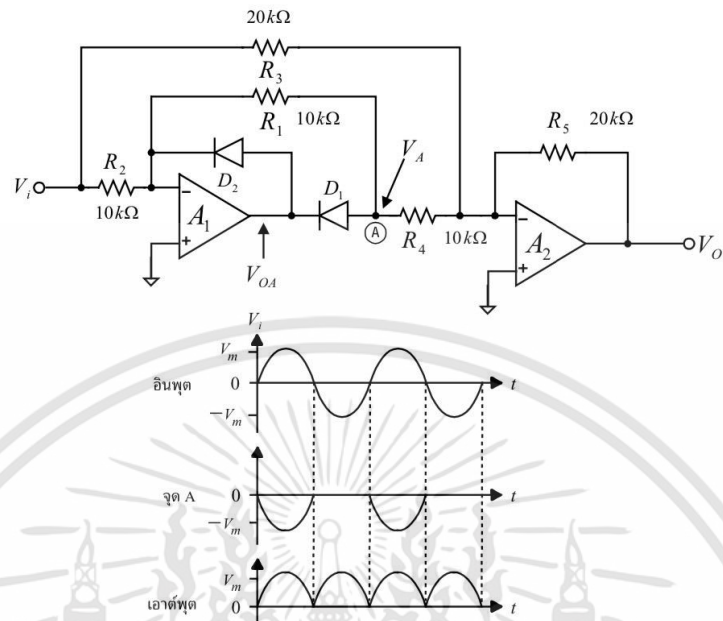


รูปที่ 2.8 การทำงานของวงจรเรียงกระแสแม่นยำแบบครึ่งคลื่นของช่วงครึ่ง Cycle บวกและช่วง Cycle ลบ

จากรูปที่ 2.8 พบว่า แรงดันตกคร่อมไดโอดจะไม่ส่งผลกับแรงดันเอาต์พุต ดังนั้นวงจรนี้จึงเรียกว่า “วงจรเรียงกระแสแม่นยำแบบแม่นยำ” ทำให้สามารถใช้ไดโอดที่มีแอมพลิจูดต่ำ ๆ ได้และวงจรนี้ถ้าต้องการ V_0 ที่มีทิศทางตรงข้ามกับสัญญาณเดิม ก็เพียงแต่สลับขั้วของไดโอดทั้งสองเท่านั้น

2.4.2 วงจรเรียงกระแสแม่นยำแบบเต็มคลื่น

จากวงจรเรียงกระแสแม่นยำแบบครึ่งคลื่นในรูปที่ 2.6 เราสามารถปรับปรุงวงจรเดิมโดยการกลับขั้วไดโอดเพื่อให้ได้สัญญาณ V_0 ออกมาเป็นลบ และใช้ออปแอมป์อีกหนึ่งตัว ต่อเป็นวงจรบวกสัญญาณก็จะได้วงจรเรียงกระแสแม่นยำแบบเต็มคลื่นและสัญญาณที่จุดต่าง ๆ ในวงจร ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 วงจรเรียงกระแสแม่นยำแบบเต็มคลื่น

สำหรับการทำงานสามารถวิเคราะห์ได้ดังนี้ ในช่วงที่ V_i เป็น Cycle บวกจะพบว่า แรงดันที่จุด A (V_A) จะมีค่าเป็นลบ ตามรูปที่ 2.9 เนื่องจากมีกระแสไหลจาก V_i ผ่าน R_1 , R_2 , D_1 และไหลเข้าออปแอมป์ A_1 ทำให้มีค่าเป็น $V_{OA} \approx -0.7V$ ดังนั้นจาก เรื่องวงจรรขยายผลบวก สัญญาณจะได้ดังสมการที่ 2.1

$$V_0 = -R_5 \left[\frac{V_A}{R_4} + \frac{V_i}{R_3} \right] \quad (2.1)$$

$$V_0 = -20k\Omega \left[\frac{-V_i}{10k\Omega} + \frac{V_i}{20k\Omega} \right]$$

$$V_0 = 2V_i - V_i$$

$$V_0 = V_i$$

ดังนั้นในช่วงที่ V_i เป็น Cycle บวก เราจึงได้ V_0 เป็นครึ่ง Cycle บวกด้วยดังแสดง
ในรูปที่ 2.9

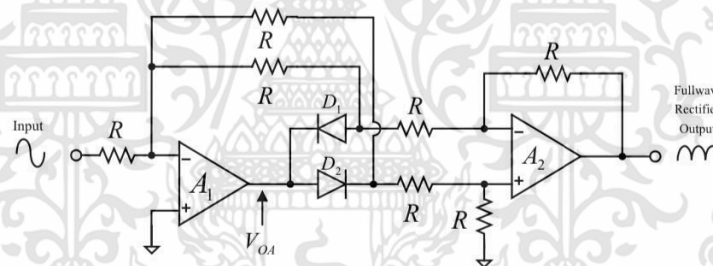
สำหรับในครึ่ง Cycle ลบ จะมีกระแสไหลออกจากออปแอมป์ A_1 ผ่านไดโอด D_2 ไดโอด D_1 จะไม่นำกระแส V_A ที่ช่วงนี้จึงเป็นศูนย์ ดังนั้น

$$V_0 = -R_5 \left[\frac{V_A}{R_4} + \frac{V_i}{R_3} \right] \quad (2.2)$$

$$V_0 = -20k\Omega \left[\frac{0}{10k\Omega} + \frac{V_i}{20k\Omega} \right]$$

$$V_0 = -V_i$$

ดังนั้นในช่วงที่ V_i เป็น Cycle ลบ เราจึงได้ V_0 เป็นครึ่ง Cycle บวกตามรูปที่ 2.9 วงจรในรูปที่ 2.10 ก็เป็นวงจรเรียงกระแสแม่นยำแบบเต็มคลื่นอีกแบบหนึ่ง โดยใช้ วงจรขยายผลต่างสัญญาณแทนวงจรบวกสัญญาณ



รูปที่ 2.10 วงจรเรียงกระแสแม่นยำแบบเต็มคลื่นที่สร้างจากวงจรขยายผลต่างของสัญญาณ [4]

2.5 ไมโครคอนโทรลเลอร์



รูปที่ 2.11 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ดังรูปที่ 2.11 ได้รับความนิยมอีกบอร์ดหนึ่งเนื่องจากมีราคาไม่แพง ซึ่งส่วนใหญ่โปรเจกต์และไลบรารีต่างๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็น ขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ Arduino ซึ่งบอร์ด Arduino Uno ได้ถูกพัฒนาขึ้นมา ตั้งแต่ R2,R3 และมีรุ่นชิพไอซีเป็นแบบ SMD ในการศึกษาบอร์ดไมโครคอนโทรลเลอร์ Arduino ของเล่มนี้ใช้เป็น บอร์ดรุ่น Arduino Uno R3 มีคุณสมบัติของบอร์ดดังนี้

- 1) ใช้ไอซีไมโครคอนโทรลเลอร์ ATmega328
- 2) ใช้แรงดันไฟฟ้าเลี้ยงไอซีไมโครคอนโทรลเลอร์ ATmega328 มีค่า 5 โวลต์
- 3) แรงดันไฟฟ้าป้อนที่บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 อยู่ในช่วง 7 – 12 โวลต์
- 4) มีพอร์ตดิจิตอลอินพุต/เอาต์พุต (Digital VO) จำนวน 14 พอร์ต (มี PWM output จำนวน 6 พอร์ต)
- 5) มีพอร์ตอนาล็อกอินพุต (Analog Input) จำนวน 6 พอร์ต
- 6) สามารถจ่ายกระแสไฟฟ้า แต่ละพอร์ตได้ 40 มิลลิแอมป์ (mA)
- 7) สามารถจ่ายกระแสไฟฟ้าในพอร์ต 3.3 V จ่ายได้ 50 มิลลิแอมป์ (mA)
- 8) มีพื้นที่หน่วยความจำโปรแกรม 32 กิโลไบต์ (KB)
- 9) มีพื้นที่หน่วยความจำชั่วคราวแบบ SRAM 2 กิโลไบต์ (KB)
- 10) มีพื้นที่หน่วยความจำถาวรแบบ EEPROM 1 กิโลไบต์ (KB)
- 11) ใช้ความถี่สัญญาณนาฬิกา 16 เมกะเฮิร์ตซ์ (MHz)

2.5.1 ภาคจ่ายไฟ

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 สามารถใช้พลังงานไฟฟ้าจากพอร์ต USB ของคอมพิวเตอร์ หรือแหล่งพลังงานไฟฟ้าจากภายนอก โดยบอร์ดสามารถเลือกแหล่งจ่ายไฟฟ้าโดยอัตโนมัติ ในส่วนของแหล่งพลังงานไฟฟ้าจากภายนอก สามารถใช้ได้ทั้งแบบไฟฟ้ากระแสสลับและไฟฟ้ากระแสตรงจากอะแดปเตอร์ หรือจากแบตเตอรี่โดยมีขั้วไฟฟ้าของอะแดปเตอร์ หรือจากแบตเตอรี่โดยมีขั้วไฟฟ้าของอะแดปเตอร์สามารถเชื่อมต่อ ด้วยการเสียบปลั๊กขนาด 2.1 มม. เข้ากับแจ็คพาวเวอร์ของบอร์ด ช่วงแรงดันของแหล่งจ่ายไฟฟ้าที่แนะนำ ควรมีค่าอยู่ในช่วง 7-12 โวลต์ แต่ถ้าใช้แรงดันมากกว่า 12 โวลต์ ส่งผลให้ไอซีควบคุมมีแรงดันไฟฟ้ามากเกินไปและเกิดความเสียหายต่อบอร์ดได้ ขาพาวเวอร์ชัฟฟลาย มีดังนี้

- V_{in} เป็นขารับแรงดันไฟฟ้าเลี้ยงบอร์ด Arduino จากแหล่งจ่ายไฟฟ้าภายนอก
- 5 V เป็นขาจ่ายแรงดันไฟฟ้า 5 โวลต์ ที่ได้จากแรงดัน V_{in} ผ่านวงจรรีเกิลูเลเตอร์ภายในบอร์ดหรือจากแรงดันไฟฟ้าที่พอร์ต USB
- 3.3 V เป็นขาจ่ายแรงดันไฟฟ้า 3.3 โวลต์ ที่สร้างขึ้นโดยวงจรรีเกิลูเลเตอร์ภายในบอร์ดจ่ายกระแสสูงสุดคือ 50 มิลลิแอมป์
- GND เป็นขาราวนด์

2.5.2 หน่วยความจำ

ไมโครคอนโทรลเลอร์ ATmega328 มีหน่วยความจำแบบแฟลชสำหรับการจัดเก็บโปรแกรม ขนาด 32 กิโลไบต์ (มีหน่วยความจำใช้สำหรับการบูต ขนาด 0.5 กิโลไบต์) มีหน่วยความจำชั่วคราวแบบสแตติกแรม (SRAM) ขนาด 2 กิโลไบต์ และมีหน่วยความจำถาวรแบบอีพรอม (EEPROM) ขนาด 1 กิโลไบต์

2.5.3 พอร์ตอินพุต - เอาต์พุต

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตดิจิทัลทั้งหมด 14 ขา สามารถเป็นได้ทั้งพอร์ตอินพุตหรือเอาต์พุต โดยใช้ฟังก์ชัน `pinMode()` , `digitalWrite()` และ `digitalRead()` แต่ละขาทำงานที่แรงดัน 5 โวลต์ ตัวต้านทานต่อแบบพูลอัพอยู่ภายในมีค่าความต้านทาน 20-50 กิโลโอห์ม

สามารถจ่ายหรือรับกระแสไฟฟ้าได้สูงสุด 40 มิลลิแอมป์ และนอกจากนี้แล้วบางพอร์ตยังมีหน้าที่พิเศษ ดังนี้

- พอร์ต 0 เป็นขา RX ใช้เป็นพอร์ตรับสัญญาณสื่อสารแบบอนุกรม
- พอร์ต 1 เป็นขา TX ใช้เป็นพอร์ตส่งสัญญาณสื่อสารแบบอนุกรม
- พอร์ต 2 และ 3 เป็นพอร์ตรับสัญญาณอินเตอร์รัปต์ภายนอก (Interrupts) พอร์ตเหล่านี้สามารถกำหนดค่าให้รับสัญญาณขัดจังหวะได้ทั้งแบบลอจิกสูง ลอจิกต่ำ หรือแบบอื่นๆ
- พอร์ต 10,11,12 และ 13 เป็นพอร์ตสื่อสารแบบ SPI
- พอร์ต 13 เป็นพอร์ตควบคุมแอลอีดีที่ติดตั้งบนบอร์ด เมื่อขา 13 จ่ายเอาต์พุตลอจิก “1” ทำให้แอลอีดีติดสว่าง และเมื่อจ่ายลอจิก “0” ทำให้แอลอีดีดับ

2.5.4 อนาล็อกอินพุต

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตอนาล็อกอินพุต 6 ขา คือ ขา A0 ถึง ขา A5 ซึ่งแต่ละช่องมีความละเอียดขนาด 10 บิต แบ่งระดับความแตกต่างได้ 1024 ค่า โดยเริ่มต้นจากระดับแรงดัน 0 โวลต์ จนถึงระดับแรงดัน 5 โวลต์ และสามารถเปลี่ยนระดับแรงดันอ้างอิงได้โดยใช้ แรงดันอ้างอิงจากภายนอกที่ขา AREF ร่วมกับฟังก์ชัน `analogReference()` นอกจากนี้ยังมีที่บางขาที่มีหน้าที่พิเศษ ดังนี้

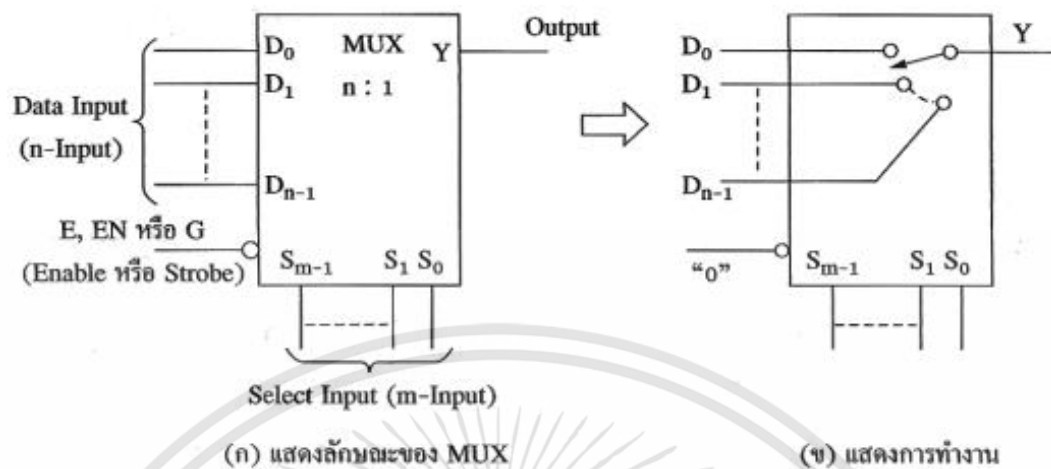
- พอร์ต A4 (SDA) และพอร์ต A5 (SCL) เป็นพอร์ตสื่อสารแบบ IRC
- พอร์ต Aref แรงดันอ้างอิงสำหรับอินพุตอนาล็อก ใช้งานร่วมกับฟังก์ชัน `analogReference()`

2.5.5 การสื่อสาร

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตสื่อสารเชื่อมต่อกับคอมพิวเตอร์ หรือบอร์ด Arduino อื่นๆ หรือไมโครคอนโทรลเลอร์เบอร์อื่นๆ หลายรูปแบบ ตามความสามารถของไมโครคอนโทรลเลอร์ ATmega328 ที่มีพอร์ตสื่อสารอนุกรมแบบ UART ที่พอร์ตดิจิทัล ขา 0 (RX) และพอร์ตดิจิทัล ขา 1 (TX) ช่องทางการสื่อสารแบบอนุกรมยังเชื่อมโยงผ่านพอร์ต USB และยังปรากฏเป็นพอร์ต COM เสมือนซอฟต์แวร์บนเครื่องคอมพิวเตอร์ ด้วยเฟิร์มแวร์ 8U2 คอมพิวเตอร์ สามารถเชื่อมต่อได้โดยใช้ไดรเวอร์ USB มาตรฐาน และไม่ต้องใช้ไดรเวอร์ภายนอก [5]

2.6 วงจร Multiplexer

วงจรมัลติเพล็กซ์ (Multiplexer) หรือมักซ์ (MUX) หรือวงจรถูกเลือกข้อมูล (Data Selector) เป็นวงจรที่มีเอาต์พุตเพียง 1 เอาต์พุตแต่มีอินพุตหลายอินพุต ทำหน้าที่เลือกข้อมูลจากอินพุตมาใช้เพียงครั้งละข้อมูลเดียวลักษณะของ MUX แสดงดังรูปที่ 2.12 (ก) ส่วนรูปที่ 2.12 (ข) แสดงการทำงานของ MUX ซึ่งเปรียบเสมือนสวิตช์แบบหมุน (Rotary switch)



รูปที่ 2.12 โครงสร้างการทำงานของ MUX

เอาต์พุตของ MUX มี 1 เอาต์พุต (Y) อาจจะเป็นแบบ Action High หรือ Active Low ก็ได้แต่บางตัวอาจจะมีเอาต์พุตทั้ง 2 แบบ ซึ่งเรียกว่าเป็นแบบ Complement Output อินพุตของ MUX แบ่งเป็น 3 กลุ่ม คือ

1. ขา Enable หรือ Strobe (E, EN หรือ G) ทำหน้าที่ควบคุมให้ MUX ทำงานหรือไม่ทำงานโดย Enable เป็นอินพุตแบบ Active Low

- ถ้าขา Enable ได้รับลอจิก 0 MUX จะทำงานตามปกติ
- ถ้าขา Enable ได้รับลอจิก 1 MUX จะไม่ทำงาน (เอาต์พุตไม่ Active)

2. Data Input เป็นอินพุตที่รับสัญญาณข้อมูล (Data) เข้ามา

3. Select Input เป็นขาที่ทำหน้าที่เลือกข้อมูล (Data) จากขา Data Input ไปใช้

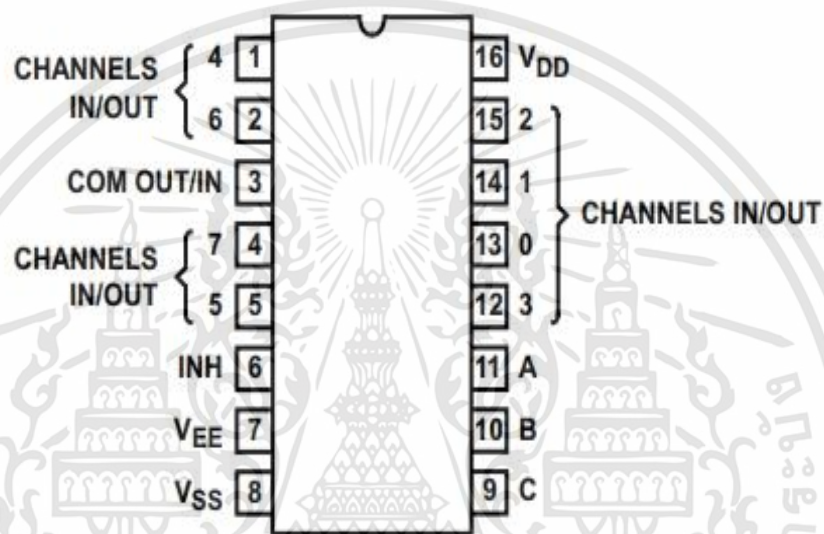
งาน

ถ้า Data Input = n, Select Input = m จะได้ความสัมพันธ์ระหว่างจำนวน Data Input กับจำนวน Select Input เป็น $n = 2^m$ การกำหนดขนาดของ MUX กำหนดในรูป “จำนวน Data Input : 1” หรือ “จำนวน Data Input-to-1”

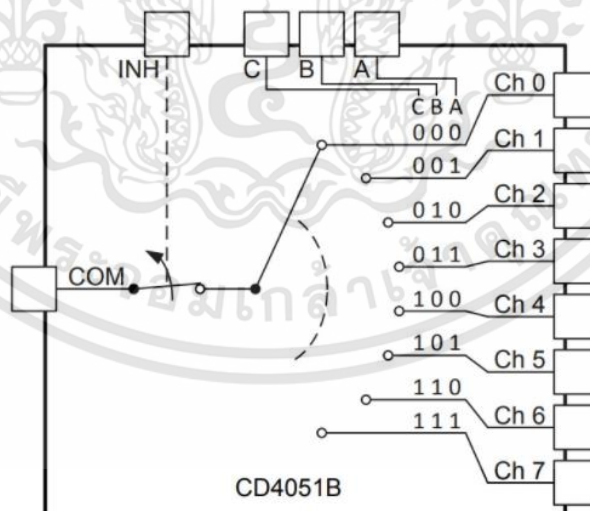
ซึ่งจะมุ่งเน้นไปที่ MUX ขนาด 8 : 1 มีสมการเอาต์พุตเป็น

$$Y = m_0D_0 + m_1D_1 + \dots + m_7D_7 = \sum_{i=0}^7 m_iD_i \quad (2.3)$$

วงจรมัลติเพล็กซ์ขนาดเข้า 8 ออก 1 หรือเรียกอีกอย่างว่า 8-line-to-1-line Data Selector / Multiplexers ซึ่งมีการเลือกใช้ต่างกันออกไปตามแต่ละเบอร์ ถ้าเลือกใช้เบอร์ 4051 จะมีจำนวนอินพุตรับข้อมูลเข้าจำนวน 8 อินพุต (X_0 ถึง X_7) มีขาซีล็คที่คอยทำการควบคุมจำนวน 3 ขา (A, B, C) และมีขาเอาต์พุต Y ดังวงจรที่แสดงในรูปที่ 2.13 รูปที่ 2.14 และตารางที่ 2.1 เมื่อต้องการให้มัลติเพล็กซ์เซอร์ ทำงานได้ต้องควบคุมผ่านขา A, B, C [6]



รูปที่ 2.13 Pin Configuration



รูปที่ 2.14 Function Diagrams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 Pin Function ของ Multiplexer เบอร์ 4051 [7]

Pin		I/O	Description
No.	Name		
1	CH 4 IN/OUT	I/O	Channel 4 in/out
2	CH 6 IN/OUT	I/O	Channel 6 in/out
3	COM OUT/IN	I/O	Common out/in
4	CH 7 IN/OUT	I/O	Channel 7 in/out
5	CH 5 IN/OUT	I/O	Channel 5 in/out
6	INH	I	Disables all channels
7	V _{EE}	-	Negative power input
8	V _{SS}	-	Ground
9	C	I	Channel select C
10	B	I	Channel select B
11	A	I	Channel select A
12	CH 3 IN/OUT	I/O	Channel 3 in/out
13	CH 0 IN/OUT	I/O	Channel 0 in/out
14	CH 1 IN/OUT	I/O	Channel 1 in/out
15	CH 2 IN/OUT	I/O	Channel 2 in/out
16	V _{DD}	-	Positive power input

2.7 โปรแกรมการสร้างภาพตัดขวางความต้านทานไฟฟ้า EIDORS

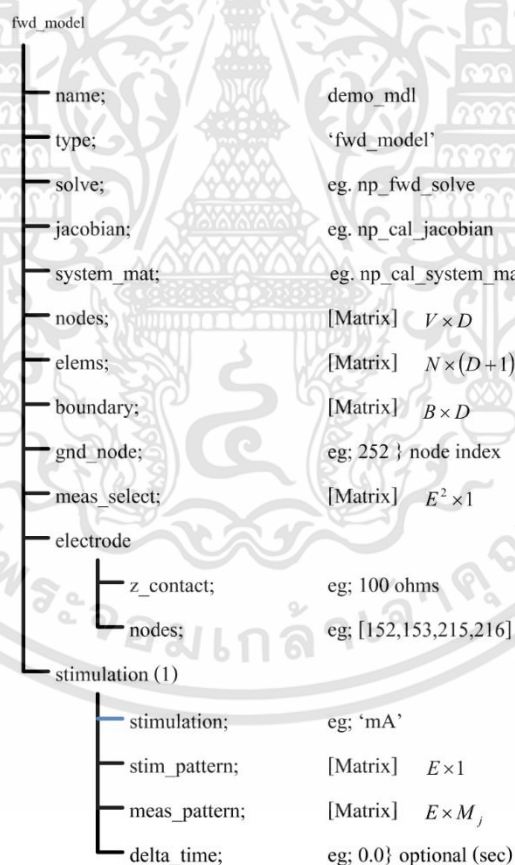
EIDORS เป็นซอฟต์แวร์อิสระ (free software) ที่เหมาะสำหรับการสร้างภาพตัดขวางความต้านทานไฟฟ้า (electrical impedance tomography : EIT) และภาพตัดขวางการกระจายตัวทางแสง (diffuse optical tomography : DOT) EIDORS รุ่นแรกมาจากวิทยานิพนธ์ของ Vauhkonen (1997) ที่ถูกพัฒนาบน MATLAB สำหรับเมชแบบสองมิติ โดยแก้ปัญหาแบบฟอร์เวิร์ด จากนั้นสร้างภาพและแสดงผล เพื่อที่จะรองรับการแก้ปัญหการสร้างภาพของโมเดลแบบสามมิติ จึงได้มีโครงการใหม่ขึ้นมาเรียกว่า EIDORS3D (รุ่นที่ 2) โดยวิทยานิพนธ์ของ Polydorides (2002) แพคเกจของซอฟต์แวร์ EIDORS มีการแบ่งปัน (share) ในด้านพื้นฐานเชิง

ตัวเลขและอัลกอริทึมที่เหมือนกัน แต่มีการแบ่งปันรหัส (code) ของซอฟต์แวร์น้อยมาก ซอฟต์แวร์แต่ละตัวจะจำลองตัวกลางโดยใช้ไฟไนต์เอลิเมนต์ และสร้างภาพด้วยเทคนิคการแปลงกลับของเรกูลาไรเซชัน (regularized inverse)

2.7.1 สถาปัตยกรรมของซอฟต์แวร์ (software architecture)

ซอฟต์แวร์ EIDORS ประกอบด้วยข้อมูล (data), ภาพ (image), fwd_model (forward model) และ inv_model (inverse model) โดยทั้งหมดมีคุณสมบัติตามชื่อและชนิด

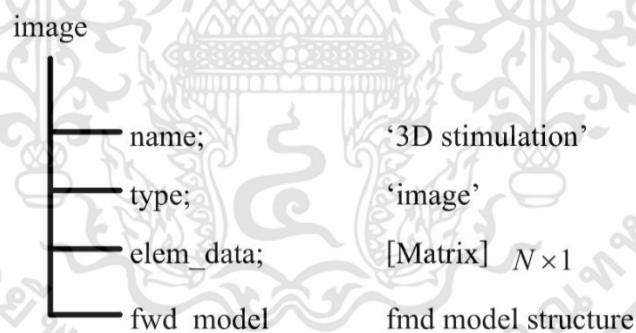
2.7.1.1 fwd_model เป็นส่วนที่ซับซ้อนที่สุดของซอฟต์แวร์ EIDORS เนื่องจากถูกออกแบบให้แทนที่โมเดลไฟไนต์เอลิเมนต์, ตำแหน่งและคุณสมบัติของอิเล็กโทรด รวมทั้งรูปแบบการกระตุ้น นอกจากนี้ยังมีฟังก์ชันที่ใช้แก้ปัญหาแบบฟอร์เวิร์ด ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 โครงสร้างของ fwd_model [8]

นอกจากนี้ EIDORS ยังมีฟังก์ชันใช้งานทั่วไป เช่น `mk_circ_tank.m`, `mk_stim_patterns.m` และ `mk_common_model.m` ฟังก์ชันเหล่านี้จะทำให้การสร้างโมเดลของไฟไนต์เอลิเมนต์แบบวงกว้าง และทรงกระบอกง่ายขึ้น และสามารถกำหนดโพรโทคอลของการกระตุ้นแบบชั่วคราวคิง (ทั้งจ่ายกระแสและวัดแรงดัน) ใน `fwd_model` ยังประกอบด้วยฟังก์ชันที่ใช้แก้ปัญหาแบบฟอร์เวิร์ด คือ `solve`, `jacobian` และ `system_mat` โดยรูปแบบของฟังก์ชันคือ `fwd_solve()`, `calc_jacobian()` และ `calc_system_mat()`

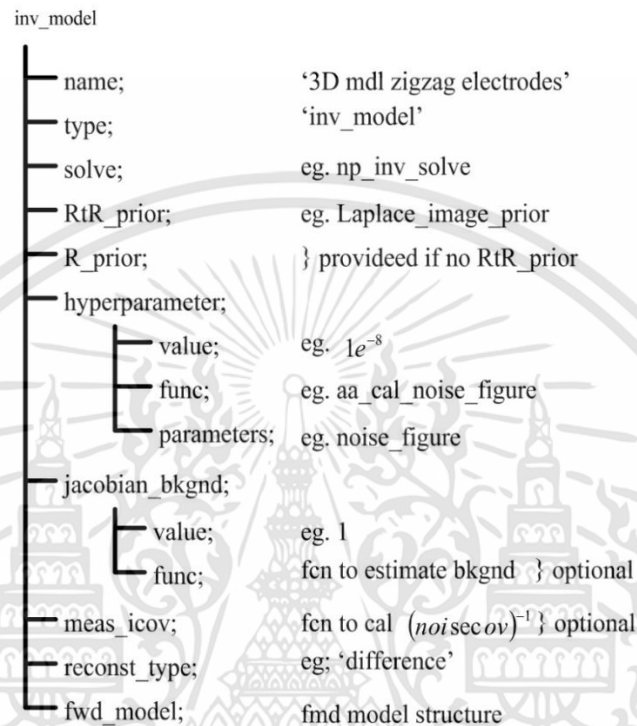
2.7.1.2 ข้อมูล (Data) แทนกรอบของข้อมูลที่วัดหรือจำลองได้ โครงสร้างของข้อมูลแสดงดังภาพที่ 4 ประกอบด้วย `frame_data`, `meas` และ `time` (acquisition time) ซึ่งมีหน่วยเป็นวินาที ในการประยุกต์ใช้งาน `time` ถูกกำหนดให้ขึ้นกับจุดเริ่มต้นอื่น ๆ เช่น จุดเริ่มต้นของการทดลอง โดยอาจจะมีค่าเท่ากับ 0 หรือ -1 สำหรับเวลาที่ไม่รู้ หรือการจำลองข้อมูล `meas` มีขนาดเมตริกซ์เท่ากับ $M \times 1$ โดย M คือจำนวนของการวัดในแต่ละเฟรม ซึ่งในบางครั้งข้อมูล (data object) อาจประกอบไปด้วยทางเลือก 2 ทางคือรูปแบบ (configuration) คือตัวอักษรที่ผู้ใช้ระบุขึ้นร่วมกับฟังก์ชันที่คล้ายกัน และ `fwd model` เป็นการรับรองว่าข้อมูลที่เสนอมีความถูกต้องและสามารถนำไปใช้สร้างโมเดลที่ต้องการ [8]



รูปที่ 2.16 โครงสร้างของ Data [8]

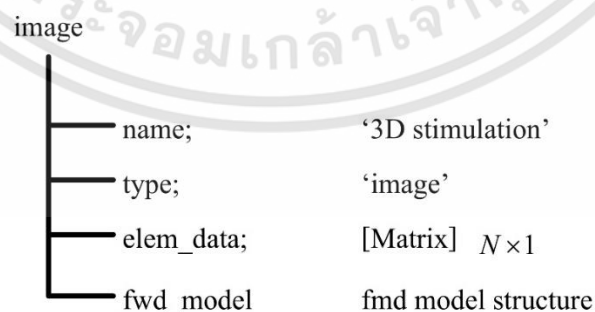
2.7.1.3 `inv_model` แสดงดังภาพที่ 2.17 จะเห็นได้ว่ากลุ่มของข้อมูลที่ใช้จำเป็นสำหรับการสร้างภาพ ชนิดของการสร้างภาพแบบพื้นฐานแบ่งออกเป็น 2 แบบคือ 'difference' สร้างภาพจากข้อมูลที่แตกต่างกัน 2 ข้อมูล กับ 'static' สร้างภาพจากข้อมูลเดียว ซึ่งฟังก์ชันที่มีอยู่บนพื้นฐานของอัลกอริธึมการสร้างภาพร่วมกับวิธีเรกูลาไรเซชัน โดยมีข้อมูลที่จำเป็นคือ ข้อมูลก่อนหน้า (image prior) และการเลือกค่าไฮเปอร์พารามิเตอร์ (hyperparameter) ใน

กรณีทั่วไปค่าไฮเปอร์พารามิเตอร์อาจมีค่าคงที่ แต่ในกรณีที่ซับซ้อนมาก ๆ การเลือกค่าพารามิเตอร์อาจถูกระบุด้วยฟังก์ชัน เช่น 'aa_calc_noise_figure' [8]



รูปที่ 2.17 โครงสร้างของ inv_model [8]

2.7.1.4 ภาพ (Image) แทนค่าตัวนำไฟฟ้าที่สร้างหรือจำลองขึ้น จากภาพที่ 2.18 จะเห็นได้ว่าโครงสร้างของภาพ โดย elem_data เป็นเมตริกซ์ขนาด $N \times 1$ คือค่าของแต่ละเอลิเมนต์ของภาพในโมเดลไฟไนต์เอลิเมนต์ [8]

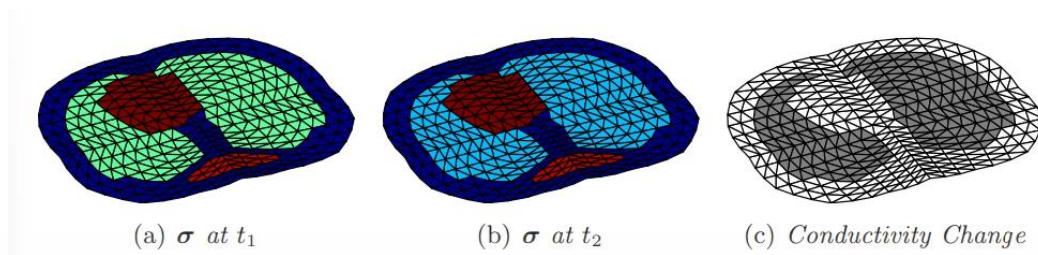


รูปที่ 2.18 โครงสร้างของ Image [8]

2.7.2 การสร้างภาพ (reconstruction)

การสร้างภาพตัดขวางความต้านทานไฟฟ้าเป็นเทคนิคการสร้างภาพซึ่งคำนวณการกระจายสภาพความนำไฟฟ้าภายในโมเดล จากการวัดแรงดันไฟฟ้าบริเวณอิเล็กโทรดที่สัมผัสกับสารละลายไอออนภายในโมเดล ระบบการสร้างภาพตัดขวางความต้านทานไฟฟ้าประกอบด้วย การเก็บข้อมูลและการสร้างภาพ หลังจากทำการเก็บข้อมูลที่ได้เข้ามายังคอมพิวเตอร์ จากนั้นนำค่าที่ได้ไปคำนวณหาค่าการกระจายแรงดันไฟฟ้าภายในโมเดล เรียกกระบวนการนี้ว่าการแก้ปัญหาแบบฟอร์เวิร์ด โดยนิยมทดลองภายใต้โมเดลอิเล็กโทรดแบบสมบูรณ์ (complete electrode model) ซึ่งหมายถึงมีความต้านทานสัมผัสเกิดขึ้นระหว่างสารละลายกับอิเล็กโทรด โมเดลที่นิยมใช้คือไฟไนต์เอลิเมนต์โมเดล เนื่องจากมีความยืดหยุ่นกับวัตถุที่มีความซับซ้อน ดังนั้นค่าผลเฉลยโดยประมาณที่คำนวณออกมาได้จะมีความแม่นยำมากขึ้น ต่อมานำค่าแรงดันไฟฟ้าภายในที่คำนวณได้มาแก้ปัญหาค่าการกระจายของอิมพีแดนซ์ภายในโมเดล เรียกขั้นตอนนี้ว่าการแก้ปัญหาแบบอินเวอร์ส เนื่องจากสมการที่ใช้ในการคำนวณค่าการกระจายของอิมพีแดนซ์ภายในตัวกลางเป็นสมการเชิงเส้นที่มีโมเดลทางคณิตศาสตร์คือ $Ax = b$ โดยปกติถ้าต้องการแก้หาค่า x ทำได้โดยการอินเวอร์สเมตริกซ์ A แต่เนื่องจาก A เป็นซิงกูลาร์เมตริกซ์ (singular matrix) จึงไม่สามารถหาค่าอินเวอร์สของเมตริกซ์ได้ ดังนั้นจึงใช้วิธีเรกูลาไรเซชัน (regularization) เพื่อให้ A เป็นนอน-ซิงกูลาร์เมตริกซ์ (non-singular matrix) และสามารถหาค่า x ได้ สุดท้ายประมาณค่าอิมพีแดนซ์ด้วยอัลกอริธึมของเกาส์-นิวตันแบบรอบเดียว

อัลกอริธึมของการสร้างภาพมีด้วยกันสองชนิดหลัก ๆ คือ การสร้างภาพแบบคงที่ (static imaging) เป็นการประมาณค่าสภาพความนำไฟฟ้าของตัวกลางจากข้อมูลที่ขอบของโมเดล และการสร้างภาพแบบต่างกัน (difference imaging) เป็นการประมาณค่าความเปลี่ยนแปลงของสภาพความนำไฟฟ้าในตัวกลางของข้อมูลที่วัดได้ในช่วงเวลา (t_1, t_2) ดังแสดงในภาพที่ 2.19 สำหรับอิมพีแดนซ์ที่มีการเปลี่ยนแปลงน้อยในช่วงเวลาหนึ่ง สามารถคำนวณโดยใช้อัลกอริธึมแบบเป็นเชิงเส้น (linearized algorithm) ในขั้นตอนเดียว แต่ถ้าอิมพีแดนซ์มีการเปลี่ยนแปลงมาก อาจจะต้องแก้ปัญหาด้วยอัลกอริธึมแบบไม่เป็นเชิงเส้นร่วมกับการทำซ้ำ



รูปที่ 2.19 ตัวอย่างการสร้างภาพแบบ 2 มิติด้วยวิธี difference imaging [9]

กำหนดให้สัญญาณ $z = v_2 - v_1$ โดย v_1 คือเวกเตอร์ของแรงดันที่วัดได้ที่เวลา t_1 และ v_2 คือเวกเตอร์ของแรงดันที่วัดได้ที่เวลา t_2 การประมาณค่าของภาพที่ต่างกัน สามารถคำนวณจากสมการ

$$\hat{x} = Bz \quad (1)$$

ซึ่ง $\hat{x} = \Delta\sigma$ คือการเปลี่ยนแปลงสภาพความนำไฟฟ้าในช่วงเวลา t_1 และ t_2 ส่วน B คือตัวดำเนินการของการสร้างภาพเชิงเส้นด้วยวิธีเรกูลาไรซ์ (regularized linearized reconstruction operator)

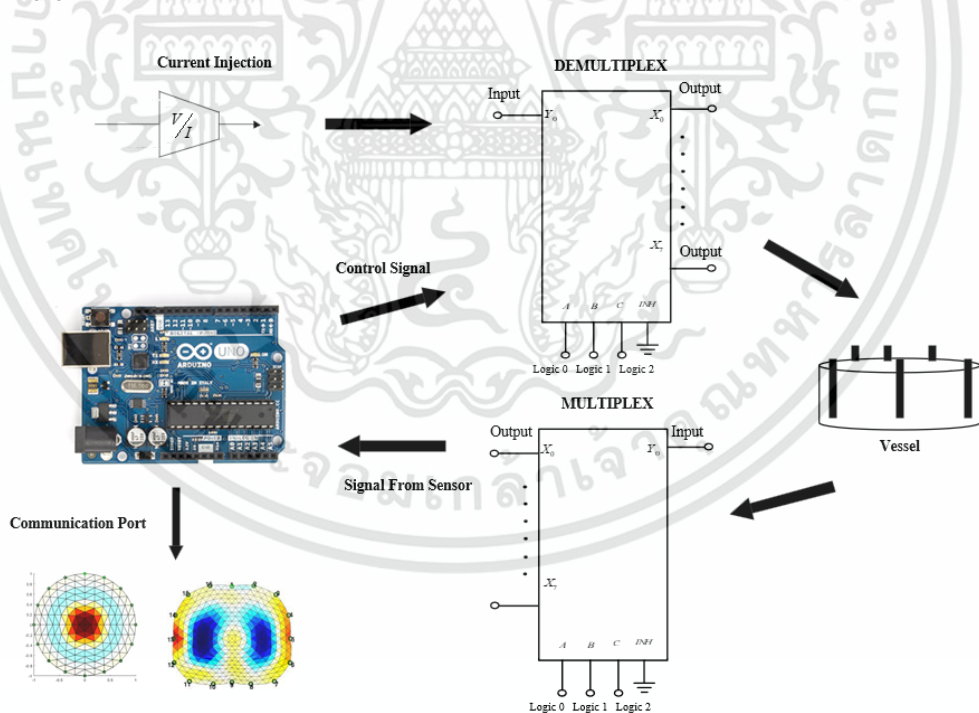
การคำนวณค่าอิมพีแดนซ์หรือการเปลี่ยนแปลงของอิมพีแดนซ์อยู่บนพื้นฐานของข้อมูลที่ได้จากการวัดแรงดันบริเวณขอบของโมเดล ซึ่งเป็นปัญหาแบบอินเวอร์สที่เป็น ill-conditioned นั้นหมายความว่าเมื่ออินพุตเปลี่ยนแปลงน้อย ส่งผลให้อเอาท์พุตเปลี่ยนแปลงมาก ทำให้ระบบไม่เสถียร ดังนั้นจึงต้องการวิธีที่จะพัฒนาเงื่อนไขนี้ ซึ่งส่วนใหญ่ใช้วิธีเรกูลาไรเซชัน (regularization) โดยเกี่ยวข้องกับการได้มาของข้อมูลที่มีลักษณะตรงกับความเป็นจริงแลกกับการสูญเสียเงื่อนไขก่อนหน้า (prior condition) ในการแก้ปัญหา [9]

บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

3.1 การออกแบบโครงสร้างโดยรวม

ในการจัดทำปฏิญานิพนธ์เล่มนี้ ผู้จัดทำได้ทำการออกแบบการออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ ซึ่งในบทนี้จะนำเสนอในส่วนของการออกแบบระบบทั้งหมดของปฏิญานิพนธ์ โดยภาพรวมการทำงานของระบบจะดังแสดงในรูปที่ 3.1 โดยการทำงานของระบบเมื่อวงจรจ่ายกระแสผ่าน Multiplexer โดยมีการควบคุมการจ่ายกระแสด้วยบอร์ด Arduino ไปยังโหนด ที่ติดตั้งไว้ในชุดจำลอง EIT กระแสที่ป้อนเข้าไปจะกระจายตัวอยู่ในน้ำที่เป็นสื่อกลางในการนำกระแส เมื่อมีวัตถุลงไปใต้น้ำค่ากระแสไฟฟ้าที่ตรวจจับได้จะมีค่าแตกต่างไปจากเดิม ค่ากระแสจะถูกตรวจจับโดยวงจรตรวจจับสัญญาณของชุดอุปกรณ์ส่งผ่าน Multiplexer กลับไปยังบอร์ด Arduino เพื่อทำการเปลี่ยนค่าแรงดันไฟฟ้าเป็นค่าดิจิทัลและส่งไปยังคอมพิวเตอร์ให้นำสัญญาณในแต่ละโนดไปประมวลผลสร้างเป็นภาพในลำดับต่อไป

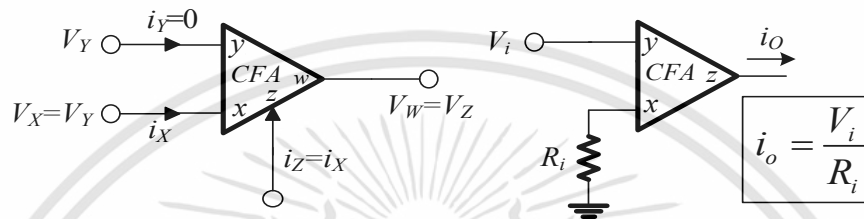


รูปที่ 3.1 บล็อกไดอะแกรมภาพรวมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การออกแบบวงจรจ่ายกระแสสำหรับชุด EIT โดยใช้โปรแกรม Proteus

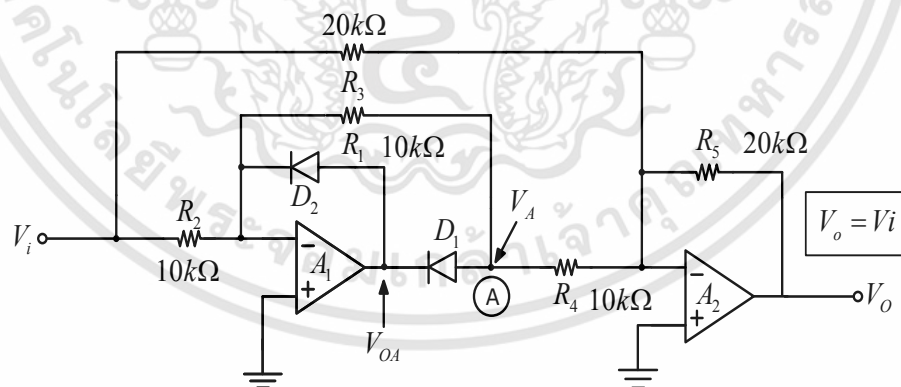
ในการออกแบบวงจรขยายสัญญาณแบบป้อนกลับกระแสสำหรับแบบจำลอง EIT นี้ จะใช้วงจร CFOA เบอร์ AD844 ทำหน้าที่เป็นวงจรแลกเปลี่ยนแรงดันเป็นกระแสโดยจะทำการออกแบบในโปรแกรม Proteus ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 วงจรขยายสัญญาณแบบป้อนกลับกระแสโดยใช้ CFOA เบอร์ AD844

3.1.2 ออกแบบวงจรรับกระแสสำหรับชุดจำลอง EIT โดยใช้โปรแกรม Proteus

ในการออกแบบวงจรรับกระแสโดยใช้วงจรเรียงกระแสแบบเต็มคลื่นสำหรับแบบจำลอง EIT นี้ จะใช้วงจรเรียงกระแสแบบเต็มคลื่น โดยใช้ Op-Amp เบอร์ LF347 และไดโอดเบอร์ 1N4148 ทำหน้าที่เป็นตัวรับกระแสและแปลงกระแส ดังรูปที่ 3.3



รูปที่ 3.3 การออกแบบวงจรรับกระแส โดยใช้ Op-Amp เบอร์ LF347

3.1.3 เขียนโปรแกรมควบคุมการจ่ายกระแสภายในชุดจำลอง EIT โดยใช้

Arduino

ในการควบคุมการจ่ายกระแสของวงจรจ่ายกระแส เราจะใช้ตัว Arduino ในการควบคุมตัวมัลติเพล็กซ์เซอร์เบอร์ HC4051 ในการควบคุมการจ่ายกระแสให้กับตัว Vessel ของ EIT system โดยการเขียนโปรแกรมจะเป็นการควบคุมตัว Logic ของตัวมัลติเพล็กซ์เซอร์ตามรูปที่ 3.4 เพื่อควบคุมให้เอาต์พุตของมัลติเพล็กซ์ออกมาตามใน Channel ที่เราต้องการ โดยตาราง Logic จะเป็นตามตาราง Truth Table ของ Multiplexer เบอร์ 4051

```
void loop()
{
  for(int i = 0; i < 8 ; i++){
    s3 = bitRead(i,0);
    s4 = bitRead(i,1);
    s5 = bitRead(i,2);
    digitalWrite(dPin,s3);
    digitalWrite(ePin,s4);
    digitalWrite(fPin,s5);
    delayMicroseconds(1);
  }
}
```

รูปที่ 3.4 โค้ดการทำงานของโปรแกรมควบคุมการจ่ายกระแส

จากรูป 3.4 เห็นได้ว่าคำสั่ง bitRead จะทำการอ่านค่า bit ของ Number เป็นการกำหนดค่าให้กับ S3, S4, S5 หลังจากนั้นคำสั่ง digitalWrite จะทำการกำหนดค่า High หรือ Low ให้กับ digital pin ที่กำหนด ซึ่ง pin ทั้ง 3 จะเป็นตัวกำหนด Chanel ในการจ่ายกระแส ตามตาราง logic

3.1.4 เขียนโปรแกรมควบคุมการรับกระแสภายในชุดจำลอง EIT โดยใช้

Arduino

ในการควบคุมการรับกระแสของวงจรรับกระแส เราจะใช้ตัว Arduino ในการควบคุมตัวมัลติเพล็กซ์เซอร์เบอร์ HC4051 ในการควบคุมการรับกระแสเพื่อนำไปประมวลผลในโปรแกรม MATLAB เพื่อทำการประมวลผลเป็นภาพในขั้นตอนต่อไป โดยจะทำการเขียนโปรแกรมการควบคุมตัว Logic ของตัวมัลติเพล็กซ์เซอร์ตามรูปที่ 3.5 เพื่อควบคุมการรับค่าเข้ามาจากแต่ละ Channel ที่เราต้องการ โดยมีค่าการควบคุมตามตาราง Logic เหมือนกับวงจรจ่ายกระแสดังตารางที่ 2.1

```

for(int i = 0 ; i < 8 ; i++){
  s0 = bitRead(i,0);
  s1 = bitRead(i,1);
  s2 = bitRead(i,2);
  digitalWrite(aPin,s0);
  digitalWrite(bPin,s1);
  digitalWrite(cPin,s2);
  delayMicroseconds(1);
}

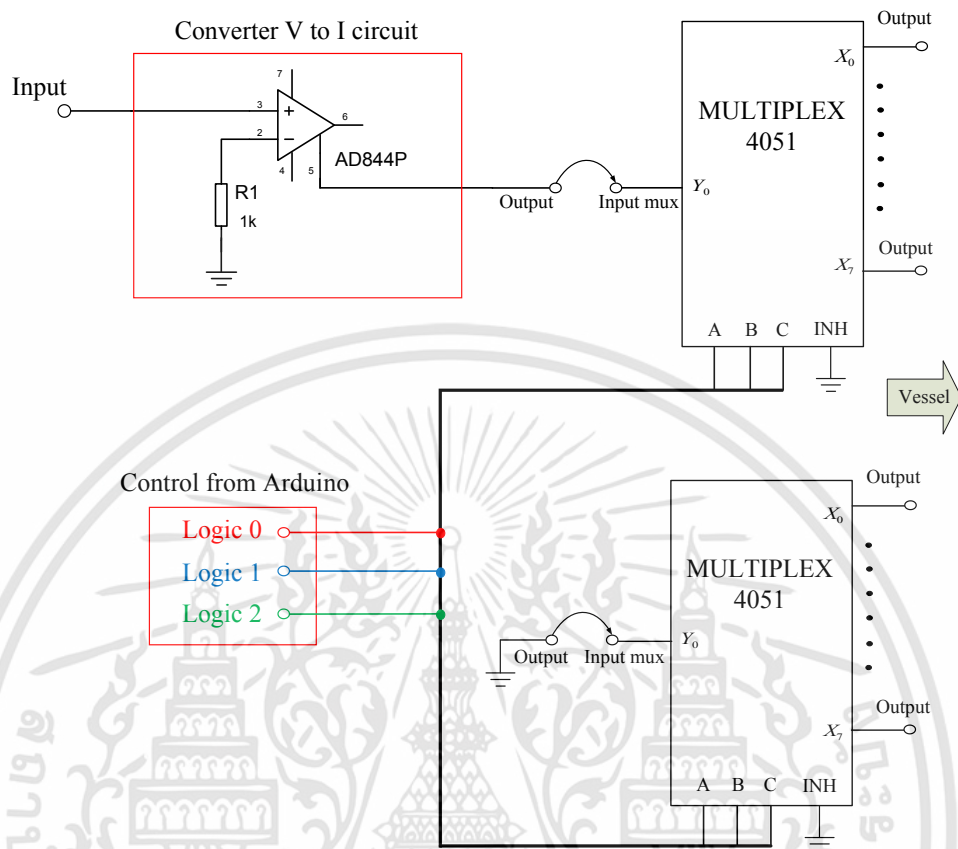
```

รูปที่ 3.5 โค้ดการทำงานของควมคุมการรับกระแส

จากรูป 3.5 เห็นได้ว่าคำสั่ง bitRead จะทำการอ่านค่า bit ของ Number เป็นการกำหนดค่าให้กับ S0, S1, S2 หลังจากนั้นคำสั่ง digitalWrite จะทำการกำหนดค่า High หรือ Low ให้กับ digital pin ที่กำหนด ซึ่ง pin ทั้ง 3 จะเป็นตัวกำหนด Chanel ในการรับกระแส ตามตาราง Logic

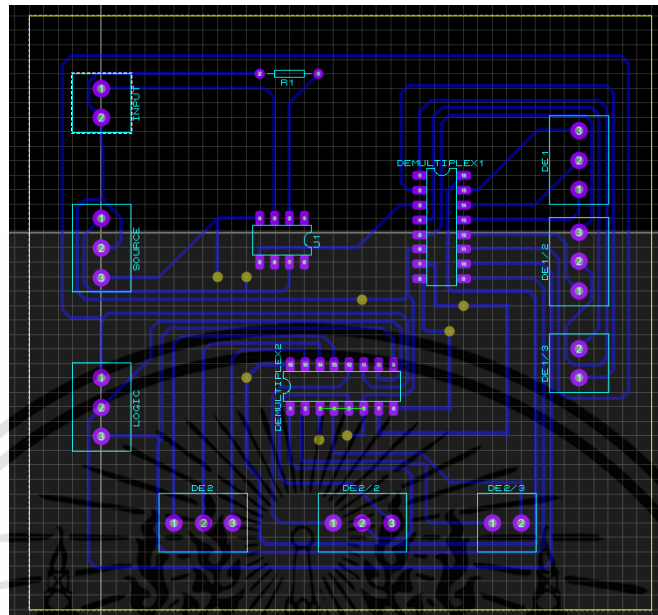
3.1.5 ออกแบบ PCB สำหรับวงจรจ่ายกระแส

ในการออกแบบ PCB สำหรับวงจรจ่ายกระแส โดยใช้เป็นวงจรขยายสัญญาณแบบป้อนกระแสกลับ (Current Feedback Operational Amplifier) โดยการออกแบบตัว PCB จะต้องออกแบบตัว Schematic ของวงจรจ่ายกระแสขึ้นมาก่อนได้ดังรูปที่ 3.6



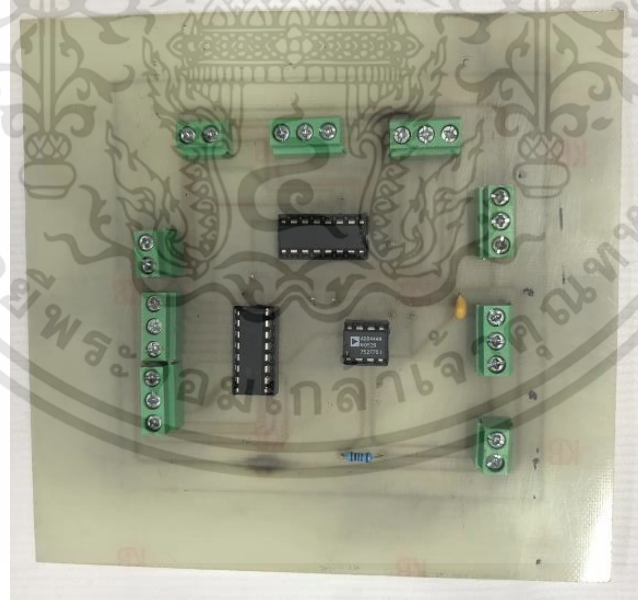
รูปที่ 3.6 Schematic ของวงจรจ่ายกระแส

จากนั้นเมื่อได้แบบ Schematic มาเรียบร้อยแล้วก็จะทำการออกแบบตัว PCB Layout ในโปรแกรม Proteus โดยหน้าตาของการออกแบบ PCB Layout ทำการจัดวางตำแหน่งของอุปกรณ์ให้เรียบร้อยและไม่ซับซ้อน เพื่อความสะดวกในการกัดปรินท์วงจรและบัดกรีวงจร แสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 PCB Layout สำหรับวงจรจ่ายกระแสที่มีการจัดเรียงเรียบร้อย

เมื่อได้ PCB Layout ของวงจรจ่ายกระแสมาเรียบร้อยแล้วก็จะทำการกัดปรินต์และ
บัดกรีวงจรให้เรียบร้อย โดยเมื่อเรียบร้อยแล้วก็จะวงจรจ่ายกระแสที่สมบูรณ์ได้ดังรูปที่ 3.8

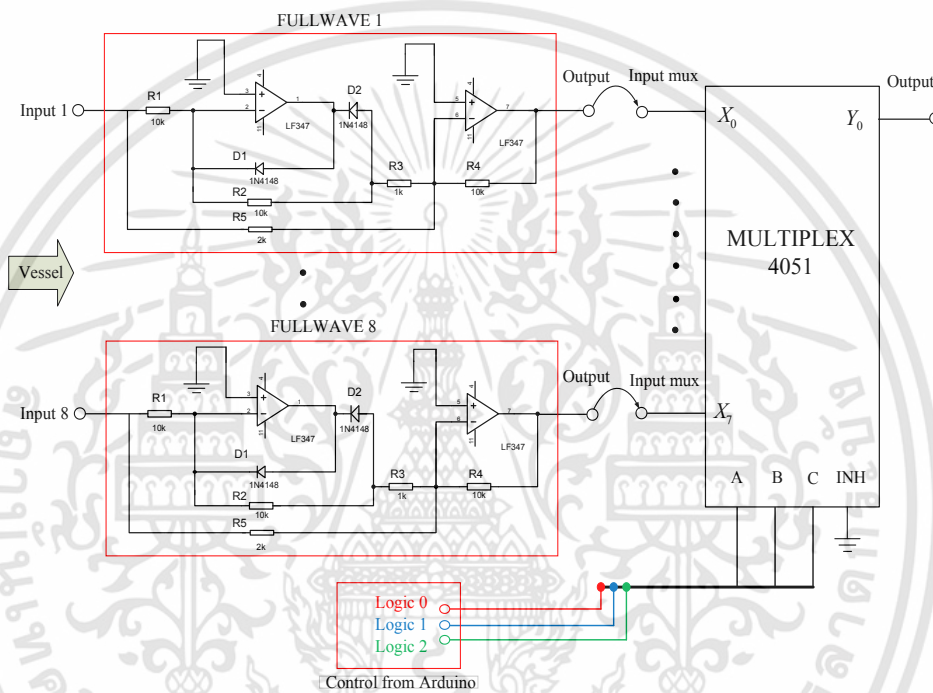


รูปที่ 3.8 แผ่น PCB ของวงจรจ่ายกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

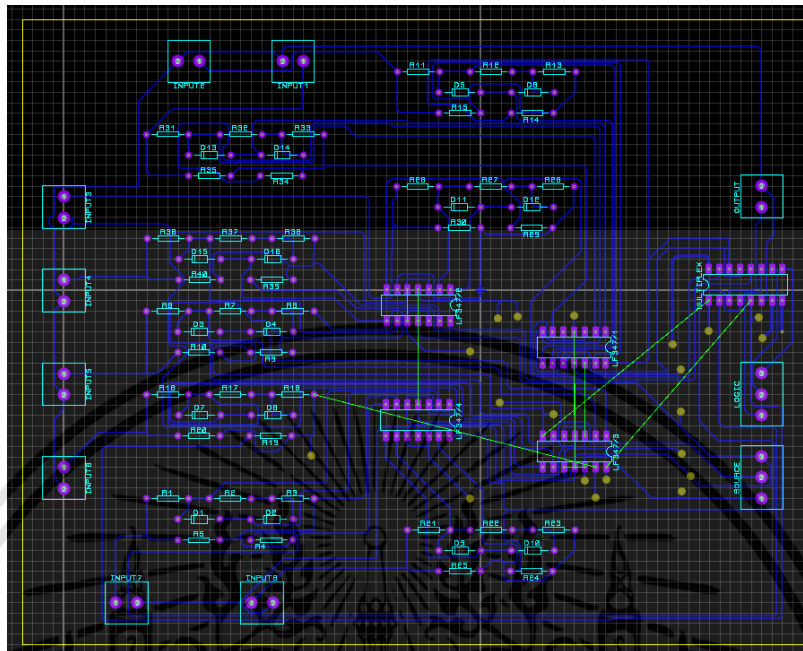
3.1.6 ออกแบบ PCB สำหรับวงจรรับกระแส

ในการออกแบบ PCB สำหรับวงจรรับกระแส โดยใช้เป็นวงจรเรียงกระแสแบบเต็มคลื่น (Full wave precision rectifier) โดยการที่จะออกแบบตัว PCB จะออกแบบตัว Schematic ของวงจรเรียงกระแสแบบเต็มคลื่นมาก่อน ดังรูปที่ 3.9



รูปที่ 3.9 Schematic ของวงจรรับกระแส

จากนั้นเมื่อได้แบบ Schematic มาเรียบร้อยแล้วก็จะทำการออกแบบตัว PCB Layout ในโปรแกรม Proteus โดยหน้าตาของการออกแบบ PCB Layout ทำการจัดวางตำแหน่งของอุปกรณ์ให้เรียบร้อยและไม่ซับซ้อน เพื่อความสะดวกในการกัดปรินต์วงจรและบัดกรีวงจร แสดงได้ดังรูปที่ 3.10



รูปที่ 3.10 PCB Layout สำหรับวงจรรับกระแสที่มีการจัดเรียงเรียบร้อย

3.1.7 เขียนโปรแกรมการแปลงสัญญาณดิจิทัลเป็นภาพโดยใช้ซอฟต์แวร์ที่ชื่อว่า EIDORS บนโปรแกรม MATLAB

ในการรับค่าสัญญาณดิจิทัลของ EIT system เข้าสู่โปรแกรม EIDORS ผ่านโปรแกรม Matlab โดยเขียนโปรแกรมการแปลงสัญญาณดิจิทัลเป็นภาพ ซึ่งมีการกำหนดค่าพารามิเตอร์ดังต่อไปนี้

- `n_elec` คือจำนวนของอิเล็กโทรดต่อระนาบ กำหนดให้มีค่าเท่ากับ 8
- `n_rings` คือจำนวนระนาบของอิเล็กโทรด กำหนดให้มีค่าเท่ากับ 1
- `options` คือทางเลือกของการวัดแรงดัน ซึ่งในโค้ดนี้ กำหนดให้เป็น `no_meas_current` หมายถึงไม่ทำการวัดบนอิเล็กโทรดที่มีกระแสไฟฟ้า และ `no_rotate_meas` หมายถึงไม่เปลี่ยนตำแหน่งการวัดกระแสไฟฟ้าไปตาม pattern
 - ฟังก์ชัน `mk_stim_pattern` คือฟังก์ชันที่ใช้กำหนดรูปแบบการจ่ายกระแสและวัดแรงดัน และกำหนดให้ เป็น `{ad}` ซึ่งหมายถึงทำการวัดค่าแรงดันตกคร่อมระหว่างอิเล็กโทรดที่อยู่ข้างกัน [10]
 - ฟังก์ชัน `mk_common_model` เป็นฟังก์ชันในการสร้างโมเดล EIT แบบทั่วไป เพื่อให้ผู้ใช้ไม่จำเป็นต้องเขียนโค้ดโมเดล EIT ขึ้นใหม่ [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโค้ดดังแสดงในรูปที่ 3.11 ได้กำหนดรูปแบบโมเดลเป็น 'c2C' ซึ่งหมายถึง โมเดล วงกลม 2 มิติ อิเล็กโทรดแบบจุด

```
n_elec= 8;
n_rings= 1;
options = {'no_meas_current','no_rotate_meas'};
[st, ms]= mk_stim_patterns(n_elec, n_rings, '{ad}','{ad}', options, 2);

inv2d= mk_common_model('c2C',n_elec);
inv2d.fwd_model.stimulation = st;
inv2d.fwd_model.meas_select = ms;

load DATA.mat
dl= background;
d2= data;

img= inv_solve( inv2d, dl, d2);
show_slices(img);
```

รูปที่ 3.11 โค้ดการทำงานการแปลงสัญญาณดิจิตอลเป็นภาพ

จากรูปที่ 3.11 จะเห็นได้ว่าคำสั่ง load จะทำหน้าที่โหลดไฟล์ DATA.mat ที่เป็นค่าสัญญาณดิจิตอลที่ได้จาก EIT system มาเก็บไว้ ในส่วนของฟังก์ชันคำสั่ง inv_solve จะทำหน้าที่หาค่าสภาพความนำไฟฟ้าที่เปลี่ยนไป และแสดงผลเป็นภาพด้วยฟังก์ชัน show_slice

3.2 เครื่องมือที่ใช้ในการทดลอง

ในโครงงานนี้มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 AD844	1	ตัว
3.2.2 LF347	4	ตัว
3.2.3 Arduino Uno R3	1	บอร์ด
3.2.4 HC4051	3	ตัว
3.2.5 ตัวต้านทาน 1 กิโลโอห์ม	3	ตัว
3.2.6 ตัวต้านทาน 10 กิโลโอห์ม	3	ตัว
3.2.7 ตัวต้านทาน 2 กิโลโอห์ม	2	ตัว

3.2.8 1N4148	2	ตัว
3.2.9 แผ่นทองแดง ขนาด 10.7×10.2 ซม.	1	แผ่น
3.2.10 แผ่นทองแดง ขนาด 18×15 ซม.	1	แผ่น

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบวงจรจ่ายและรับกระแสโดยใช้โปรแกรม Proteus

ทำการทดสอบการทำงานของวงจรจ่ายและรับกระแส ตรวจสอบค่ากระแสที่ได้ ให้สอดคล้องกับค่ากระแสตั้งต้นที่ทำการป้อนเข้าไป และเป็นไปตามทฤษฎีที่ได้ศึกษามา ทดสอบอุปกรณ์ให้มีความเหมาะสมต่อวงจรที่ทำ

3.3.2 ทดสอบการทำงานร่วมกันระหว่างวงจรจ่ายและรับกระแสภายในชุดจำลอง EIT

ทำการทดสอบการทำงานของวงจรจ่ายและรับกระแสจากการทดสอบต่อจริงบนโพรโตบอร์ด ตรวจสอบค่ากระแสที่ได้ให้สอดคล้องกับค่าที่ทำการทดสอบโดยใช้โปรแกรม Proteus

3.3.3 การทดสอบการใช้ Arduino Uno R3 ในการควบคุมการจ่ายและรับกระแส

ทำการทดสอบการใช้ Arduino Uno R3 ในการควบคุมการจ่ายและรับกระแส ตรวจสอบโค้ดควบคุมการทำงานของวงจร ให้ตัววงจรที่สร้างขึ้นมาทำงานตามที่ต้องการ ตรวจสอบการเก็บค่าข้อมูลจากการใช้โค้ดให้สอดคล้องกับการทดสอบโดยใช้โปรแกรม Proteus

3.3.4 การทดสอบวงจรจ่ายและรับกระแส

ทำการทดสอบการทำงานของวงจรจ่ายและรับกระแส ตรวจสอบค่ากระแสที่ได้ ให้สอดคล้องกับค่ากระแสตั้งต้นที่ทำการป้อนเข้าไป และเป็นไปตามทฤษฎีที่ได้ศึกษามา ทดสอบอุปกรณ์ให้มีความเหมาะสมต่อวงจรที่ทำ

3.3.5 การทดสอบวิเคราะห์และประมวลผลสัญญาณดิจิทัลเป็นภาพ

ทำการทดสอบวิเคราะห์และประมวลผลสัญญาณดิจิทัลเป็นภาพ ให้สอดคล้องกับภาพตั้งต้นที่ได้จากการใส่วัตถุลงใน Vessel

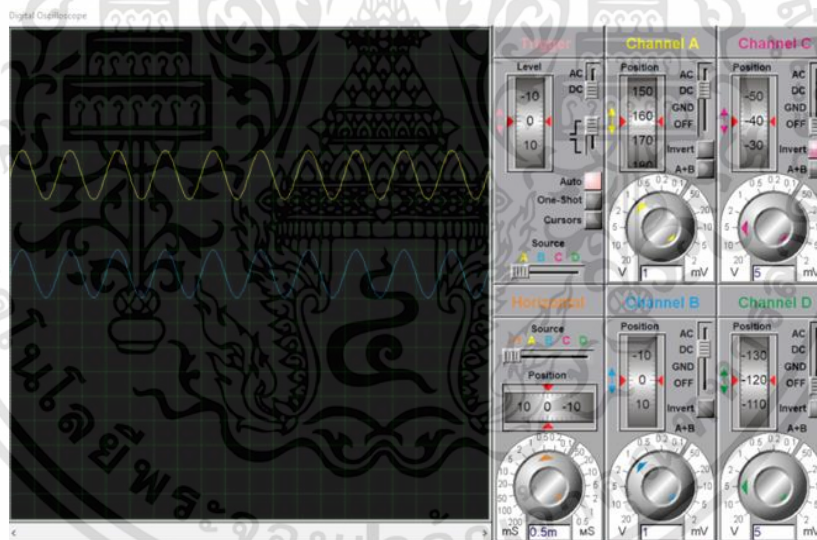
บทที่ 4

ผลการทดลอง

4.1 วงจรจ่ายกระแสสำหรับชุด EIT

4.1.1 การทดสอบวงจรจ่ายกระแสโดยใช้โปรแกรม Proteus

ในการทดสอบวงจรจ่ายกระแสในโปรแกรม Proteus ที่ได้ออกแบบไว้ดังรูปที่ 3.2 จากบทก่อนหน้านี้ โดยจะป้อนสัญญาณ Sine เข้าที่ขาบวกอินพุตของ AD844 ขาลบต่อกับตัวต้านทาน 1k โอห์มแล้วต่อลงกราวด์ ขา 7 ต่อเข้ากับไฟบวก V_{CC} ส่วนขา 4 ต่อเข้ากับไฟลบ V_{EE} และขา 5 เป็นเอาต์พุตโดยต่อเข้ากับตัวต้านทาน 1k โอห์ม จากนั้นทำการวัดสัญญาณที่ได้ออกมา ซึ่งสามารถแสดงได้ในรูปที่ 4.1 โดยที่สีเหลืองคือสัญญาณอินพุตและสีน้ำเงินคือสัญญาณเอาต์พุต โดยสัญญาณก่อนเข้า AD844 และหลังเข้า AD844 จะมีขนาดสัญญาณเท่ากัน

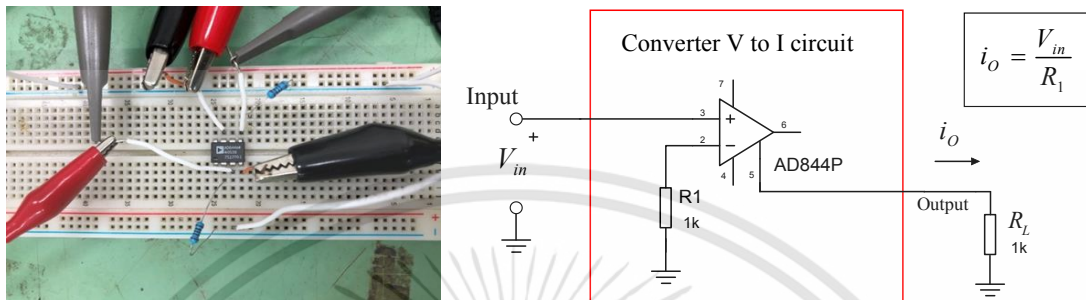


รูปที่ 4.1 สัญญาณอินพุตเทียบกับเอาต์พุตจากโปรแกรม Proteus

4.1.2 ทดสอบวงจรจ่ายกระแสลงบนแผ่นไฟโต้บอร์ด

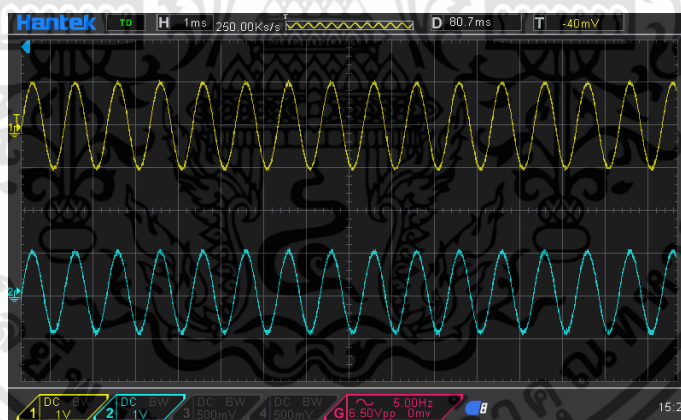
ในขั้นตอนนี้จะเป็นการสร้างวงจรจ่ายกระแสเพื่อทดสอบการทำงาน โดยทำการต่อวงจรลงแผ่นไฟโต้บอร์ด เพื่อทดสอบการทำงานของ CFOA เบอร์ AD844 โดยขาที่ 3 จะเป็นอินพุต

ป้อนสัญญาณ ขาที่ 4 เป็นโฟลบ ขาที่ 7 เป็นไฟบวก และขาที่ 5 เป็นเอาต์พุตของ
กระแส แสดงได้ดังรูปที่ 4.2



รูปที่ 4.2 วงจรขยายสัญญาณแบบป้อนกลับกระแสโดยใช้ CFOA เบอร์ AD844

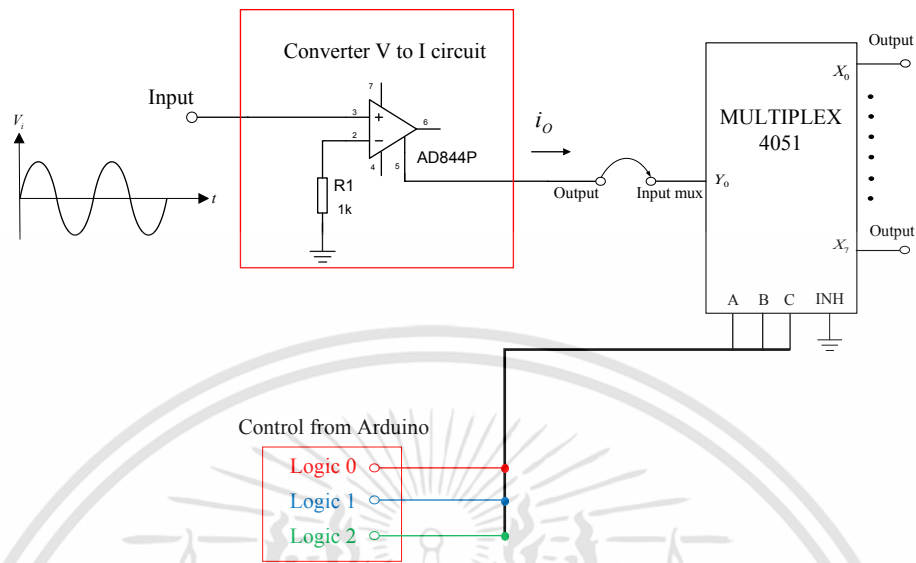
หลังจากทำการต่อวงจรจ่ายกระแสไฟโต้บอร์ดแล้ว จะทำการป้อนสัญญาณ Sine ความถี่ 1 kHz ขนาด $2 V_{pp}$ ลงไปที่ขาอินพุตของ AD844 ป้อนไฟบวก V_{CC} ไปที่ขาที่ 7 มีค่าเป็น +7 Volt ป้อนโฟลบ V_{EE} ไปที่ขาที่ 4 มีค่าเป็น -7 Volt จากนั้นจะวัดสัญญาณอินพุตและเอาต์พุต ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 สัญญาณอินพุตเทียบกับเอาต์พุตจากวงจรจ่ายกระแส

4.1.3 ทดสอบการควบคุมการจ่ายกระแส โดยใช้ Arduino Uno R3 ใน Proteus

การทดสอบการควบคุมการจ่ายกระแสภายในโปรแกรม Proteus เป็นการทดสอบการทำงาน โดยการจำลองระบบการจ่ายกระแส ก่อนที่จะทำการทดสอบอุปกรณ์จริงบนไฟโต้บอร์ด ซึ่งตัววงจรที่ทำการจำลองระบบจ่ายกระแสนั้นเป็นดังแสดงในรูปที่ 4.4



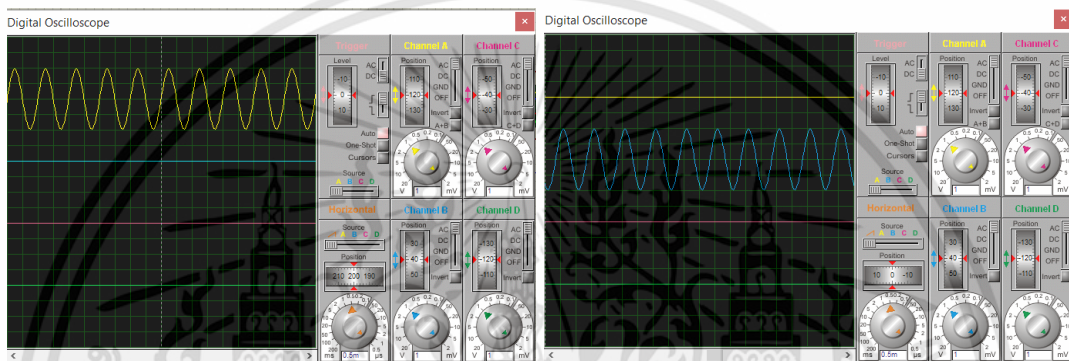
รูปที่ 4.4 วงจรทดสอบควบคุมการจ่ายกระแสที่ออกแบบในโปรแกรม Proteus

การทดสอบควบคุมการจ่ายกระแสให้กับตัว Vessel ใน EIT system ในโปรแกรม Proteus โดยเราจะใช้ Arduino Uno R3 ในการควบคุมการทำงานของมัลติเพล็กซ์โดยเป็นการควบคุม Logic ของมัลติเพล็กซ์ เพื่อทดสอบว่า Output ออกมาอย่างไร สามารถแสดงผลตาม Channel ในตาราง Logic ที่แสดงดังตารางที่ 4.1

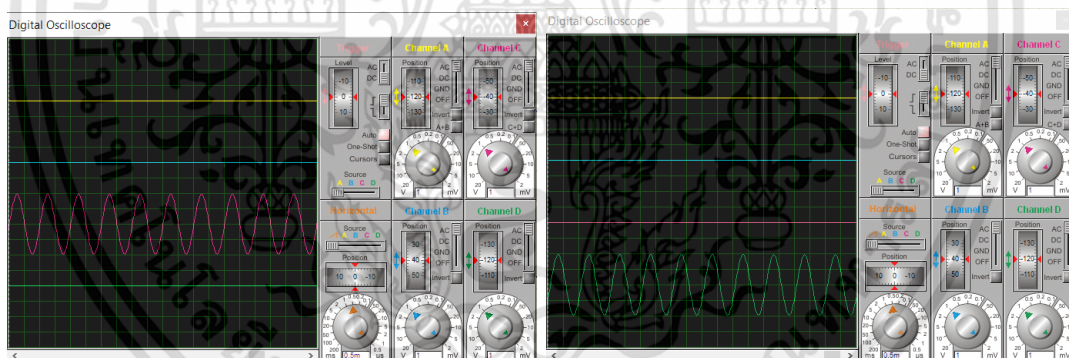
ตารางที่ 4.1 Truth Table ของ Multiplexer เบอร์ 4051

Input				Channel ON
E	C	B	A	
0	0	0	0	Y ₀
0	0	0	1	Y ₁
0	0	1	0	Y ₂
0	0	1	1	Y ₃
0	1	0	0	Y ₄
0	1	0	1	Y ₅
0	1	1	0	Y ₆
0	1	1	1	Y ₇
1	X	X	X	switches off

ทดสอบโดยทำการป้อนสัญญาณไซน์ ความถี่ 1 กิโลเฮิร์ตซ์มีขนาดเป็น $2 V_{pp}$ ให้กับ วงจรจ่ายกระแส จากนั้นวงจรจ่ายกระแสจะทำการป้อนกระแสผ่านตัวมัลติเพล็กซ์เซอร์เบอร์ HC4051 เพื่อเป็นอินพุตให้กับขั้ว Electrode โดยการทำงานของมัลติเพล็กซ์เซอร์จะมี Arduino Uno R3 คอยควบคุมการทำงานของมัลติเพล็กซ์เซอร์ไว้ จะเห็นได้ว่าเมื่อ Arduino Uno R3 ป้อน Logic 000 เข้าไปยังมัลติเพล็กซ์เซอร์ โดยเอาต์พุตที่ได้ออกมาจะแสดงดังรูป 4.5 และทำการป้อน Logic 001 เข้าไปยังมัลติเพล็กซ์เซอร์ โดยเอาต์พุตที่ได้ออกมาจะแสดงดังรูป 4.6



รูปที่ 4.5 สัญญาณที่ออกมาจาก Channel 0 และ Channel 1



รูปที่ 4.6 สัญญาณที่ออกมาจาก Channel 2 และ Channel 3

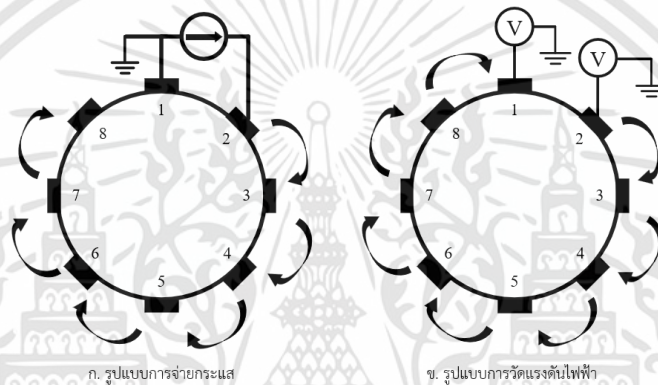
จากรูปที่ 4.5 และ 4.6 จะเห็นได้ว่าสัญญาณเอาต์พุตจะออกมาตาม Channel 0 Channel 1, Channel 2 และ Channel 3 ดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ทดสอบการควบคุมการจ่ายกระแสโดยใช้ Arduino Uno R3 บนแผ่นโฟโต้

บอร์ด

การทดสอบการควบคุมการจ่ายกระแสบนแผ่นโฟโต้บอร์ด จะทำการต่อวงจรจ่ายกระแสเข้ากับตัวมัลติเพล็กซ์เซอร์ 2 ตัว โดยตัวหนึ่งจะเป็นตัวจ่ายกระแสให้กับขั้ว Electrode รอบ Vessel และอีกตัวหนึ่งจะเป็นกราวด์ให้กับ Electrode โดยมี Arduino Uno R3 ในการควบคุมการทำงานของมัลติเพล็กซ์เซอร์ โดยรูปแบบของการจ่ายกระแสเราจะใช้รูปแบบดังรูปที่ ดังแสดงในรูปที่ 4.7(ก)



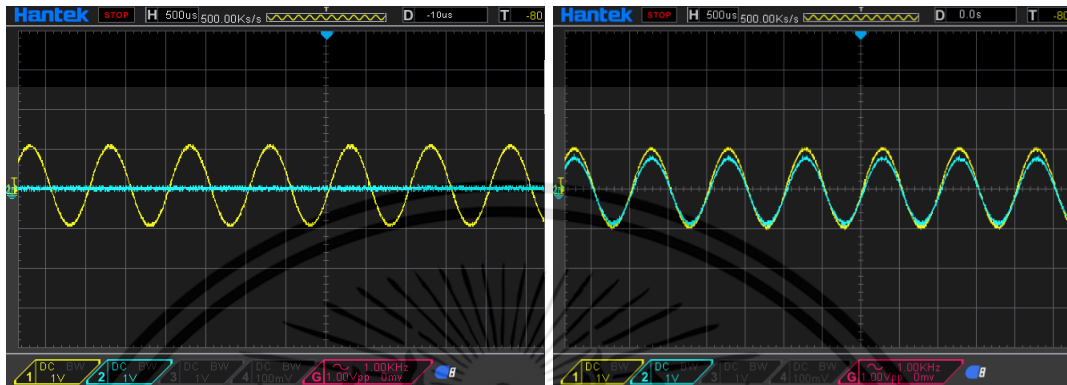
รูปที่ 4.7 รูปแบบการจ่ายกระแสและการวัดแรงดันไฟฟ้า Vessel



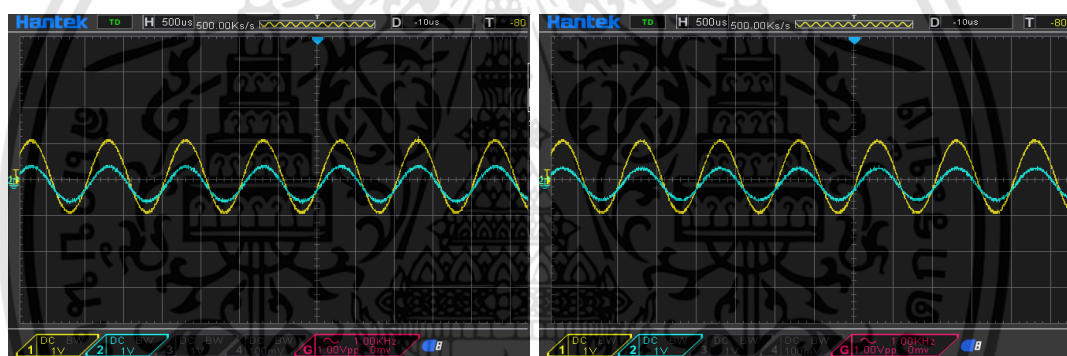
รูปที่ 4.8 การทดสอบวงจรจ่ายกระแส โดยใช้ Arduino Uno R3 ในการควบคุม

จากนั้นเราจะวัดสัญญาณของแต่ละจุดว่าสัญญาณที่ออกมาเป็นอย่างไร โดยกำหนดให้รูปที่ 4.8 มีตำแหน่งการจ่ายกระแสเช่นเดียวกับรูปที่ 4.7(ก) สำหรับทำการทดสอบรูปแบบการจ่ายแรงดันแรก โดยการทดสอบนี้จะวัดแค่แบบแรกโดยให้จุดที่ 1 เป็น Ground และจุดที่ 2 เป็น I_{in} จากนั้นทำการวัดค่าสัญญาณจาก Electrode ตัวที่ 1, 2, 3, 4, 5, 6, 7 และ 8 โดยมีรูปแบบการวัดแรงดันไฟฟ้าดังแสดงในรูปที่ 4.7(ข) เทียบกับสัญญาณอินพุตเป็นดังที่ได้แสดงในรูปที่

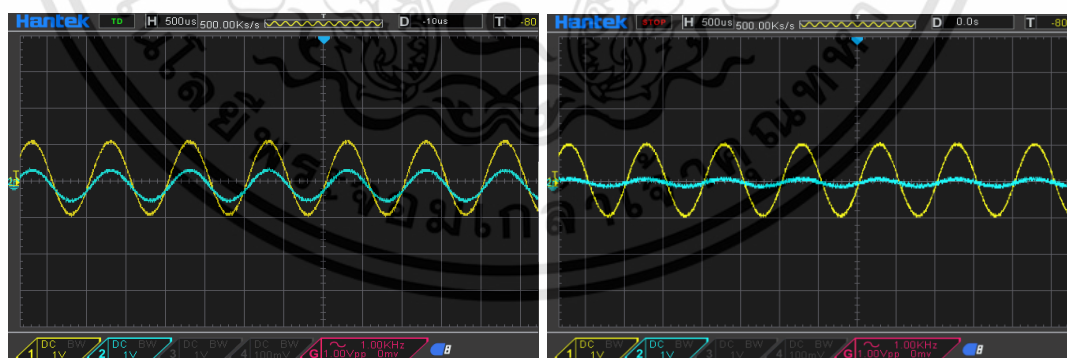
4.9, 4.10, 4.11 และ 4.12 โดยที่สัญญาณสีเหลืองคือสัญญาณอินพุตที่จ่ายเข้าวงจร และสัญญาณสีฟ้าคือสัญญาณเอาต์พุตของแต่ละ Electrode



รูปที่ 4.9 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 1 และตัวที่ 2

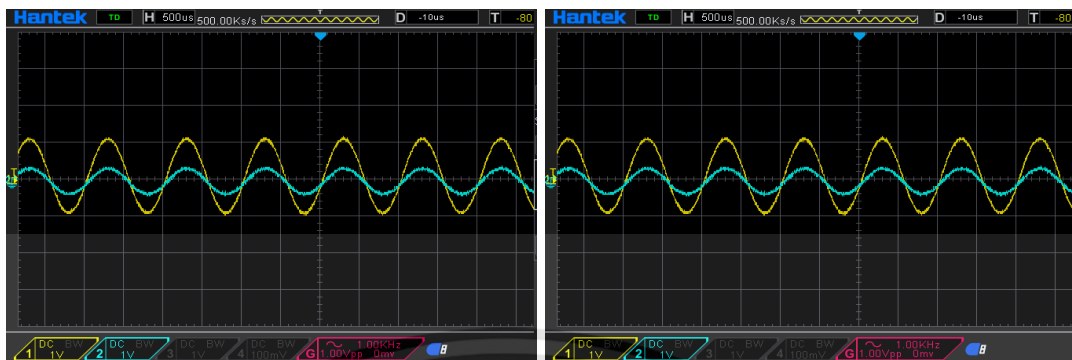


รูปที่ 4.10 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 3 และตัวที่ 4



รูปที่ 4.11 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 5 และตัวที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

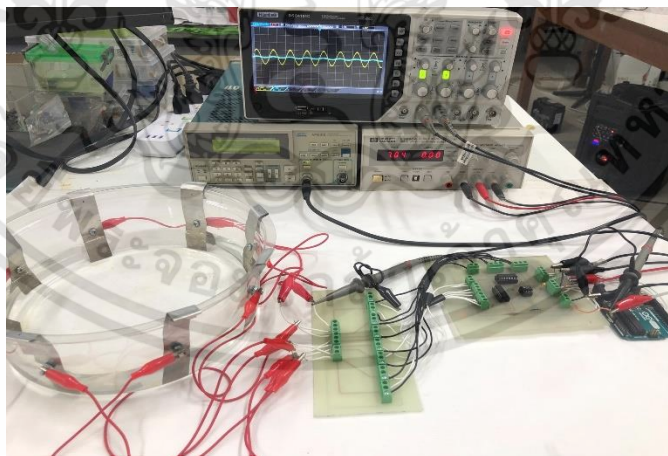


รูปที่ 4.12 สัญญาณอินพุตเทียบกับสัญญาณจาก Electrode ตัวที่ 7 และตัวที่ 8

หลังจากที่วัดสัญญาณของ Electrode แต่ละตัวรอบ Vessel ของการทำงานวงจรจ่ายกระแสแล้ว พบว่าในตัวที่ 1 สัญญาณเอาต์พุตจะหายไปเพราะที่จุดนี้จะจ่ายกราวด์เข้าไป ในตัวที่ 2 สัญญาณอินพุตและเอาต์พุตจะมีขนาดเท่ากันเพราะเป็น Electrode ตัวที่จ่ายกระแสเข้าไป ให้ชุดจำลอง EIT ใน ส่วนตัวที่ 3, 4, 5, 6, 7, และ 8 สัญญาณเอาต์พุตจะมีขนาดที่ลดลงมาจากสัญญาณอินพุตเดิมเนื่องจากอยู่ไกลจากจุดที่เป็นจุดป้อนสัญญาณอินพุต

4.1.5 ทดสอบการทำงานวงจรจ่ายกระแส

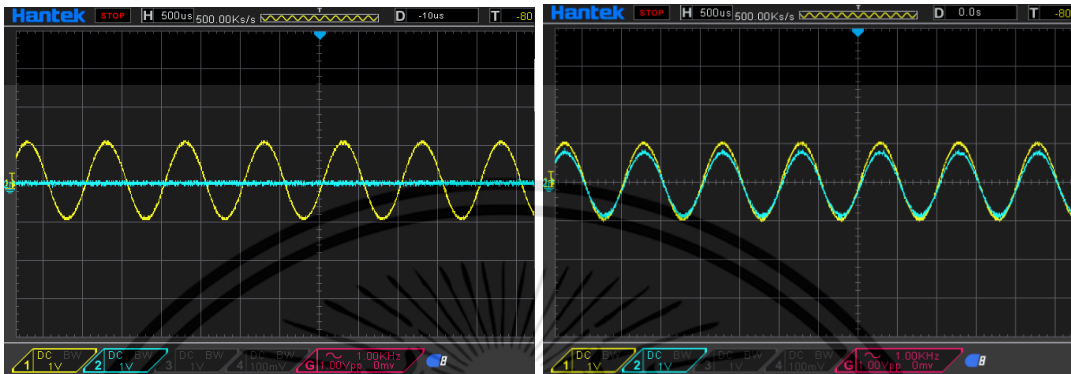
การทดสอบการทำงานวงจรจ่ายกระแสที่ได้จากการทำแผ่น PCB ออกมานั้น จะสามารถทดสอบการทำงานกับ Vessel เพื่อตรวจสอบว่าวงจรทำงานได้ ดังแสดงในรูปที่ 4.13



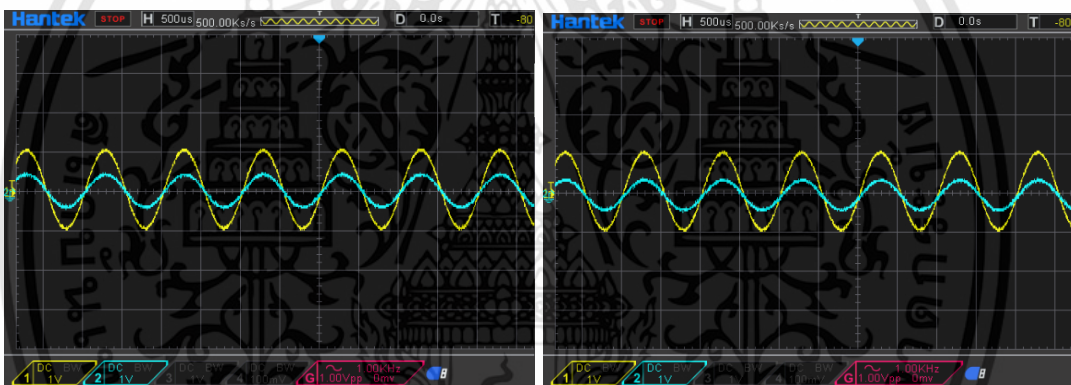
รูปที่ 4.13 ทดสอบการทำงานของ PCB ของวงจรจ่ายกระแส

เมื่อทำการต่อวงจรจ่ายกระแสเข้ากับตัว Vessel แล้ว ก็จะมีการป้อนสัญญาณและเชื่อมต่อกับ Arduino Uno R3 เพื่อควบคุมการทำงานของวงจร แล้วจากนั้นก็วัดสัญญาณที่ขั้ว

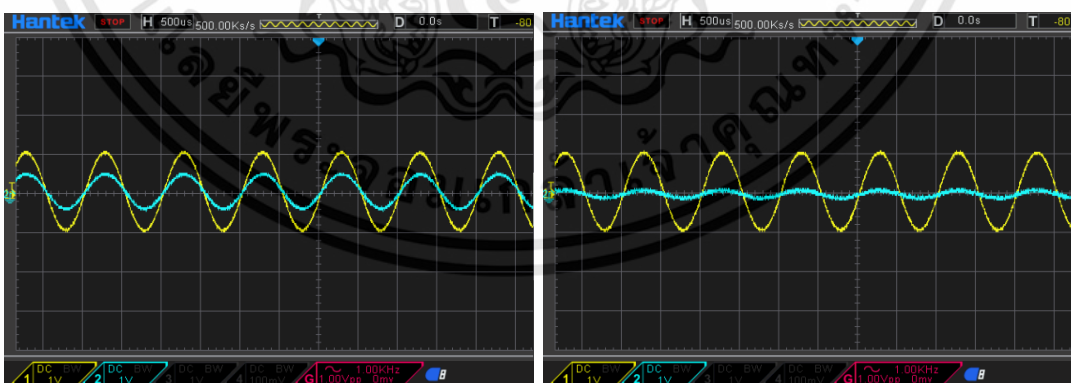
Electrode 8 จุดโดยดูแต่ละจุดว่าสัญญาณได้ตามที่ควรเป็นไหมดังรูปที่ 4.14, 4.15, 4.16 และ 4.17



รูปที่ 4.14 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 1 และตัวที่ 2

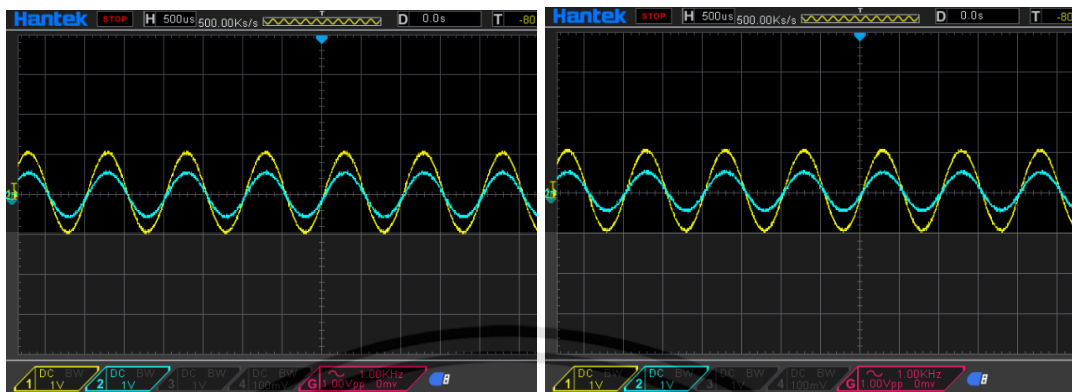


รูปที่ 4.15 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 3 และตัวที่ 4



รูปที่ 4.16 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 5 และตัวที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

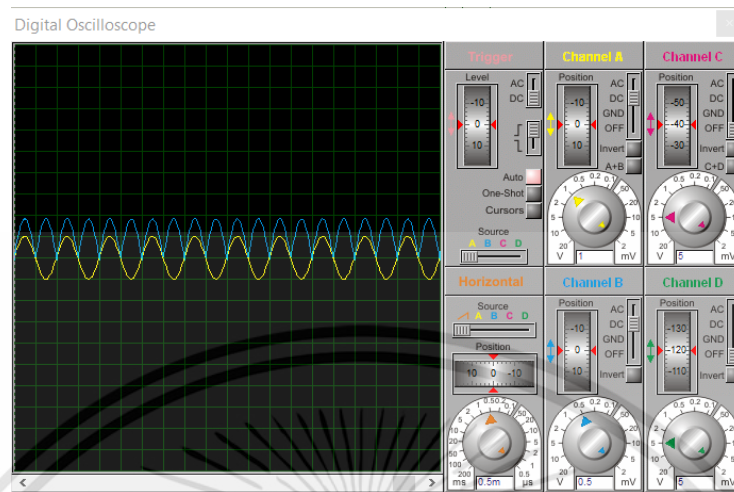


รูปที่ 4.17 สัญญาณอินพุตกับสัญญาณจาก Electrode ตัวที่ 7 และตัวที่ 8

4.2 วงจรรับกระแสสำหรับชุด EIT

4.2.1 การทดสอบวงจรรับกระแสโดยใช้โปรแกรม Proteus

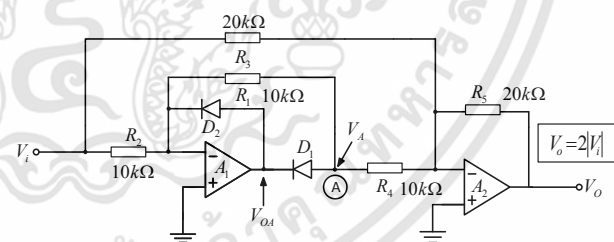
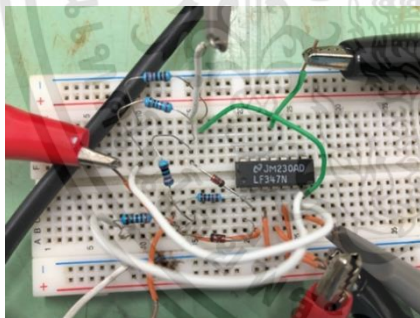
ในการทดสอบวงจรรับกระแสในโปรแกรม Proteus ที่ได้ออกแบบไว้ โดยจะป้อนสัญญาณ Sine เข้าที่ขาอินพุตของ LF347 โดยจะต่อเข้ากับตัวต้าน 10k โอห์มก่อนที่จะเข้าขาอินพุตขาลบ อินพุตบวกขา 3 และ 5 ต่อลงกราวด์ ขา 4 ต่อเข้ากับไฟบวก ขา 11 ต่อเข้ากับไฟลบ โดยขาเอาต์พุตขาที่ต่อกับไดโอดเบอร์ 1N4148 สองตัว เพื่อให้กระแสไหลไปทางเดียว ส่วนกระแสที่ไหลตรงข้ามจะถูกกั้นไว้โดยจะใช้ขาแคโทดเป็นขาอินพุตและแอนโนดเป็นเอาต์พุต ทำให้สัญญาณที่ได้ออกมาเป็นสัญญาณในลักษณะกลับเฟส โดยที่ไม่มีส่วนประกอบของไฟ DC จากนั้นจะนำสัญญาณนี้ไปรวมกับสัญญาณอินพุต ก็จะทำให้ได้สัญญาณที่มีแต่ฝั่งของไฟบวก โดยที่ไม่มีส่วนประกอบของไฟ DC เลย โดยเมื่อทำการวัดสัญญาณที่ได้ออกมาจะได้ดังนี้ โดยสีเหลืองคือสัญญาณอินพุต สีน้ำเงินคือสัญญาณเอาต์พุตที่มีแต่สัญญาณฝั่งบวก และมีขนาดเป็น 5 เท่าของสัญญาณเดิม ดังแสดงในรูปที่ 4.18



รูปที่ 4.18 ผลลัพธ์ของสัญญาณที่ได้ออกมา

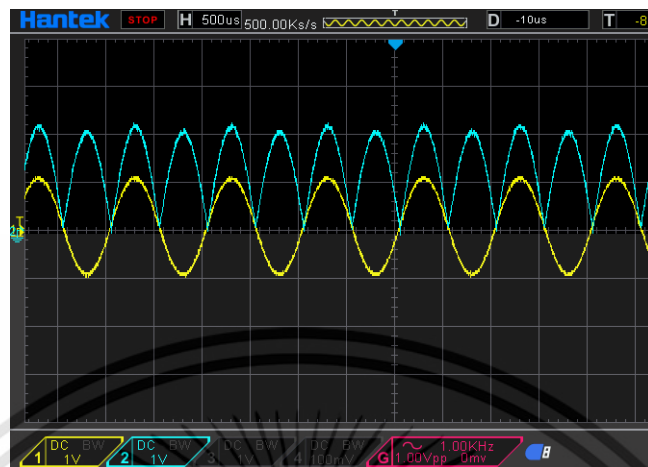
4.2.2 ทดสอบวงจรรับกระแสลงบนแผ่นไฟโต้บอร์ด

ในขั้นตอนนี้จะเป็นการสร้างวงจรรับกระแส โดยทำการต่อวงจรลงบนแผ่นไฟโต้บอร์ด เพื่อทดสอบการทำงานของวงจรรับกระแสโดยใช้วงจรเรียงกระแสแบบเต็มคลื่น โดยใช้ Op-Amp เบอร์ LF347 โดยขาที่ 2 จะเป็นอินพุตขาลบป้อนสัญญาณ Sine ขาที่ 4 เป็นไฟบวก ขาที่ 11 เป็นไฟลบ ขาที่ 7 เป็นเอาต์พุตของกระแสแสดงได้ดังรูปที่ 4.19



รูปที่ 4.19 การต่อวงจรรับกระแสโดยใช้ LF347 ลงบนแผ่นไฟโต้บอร์ด

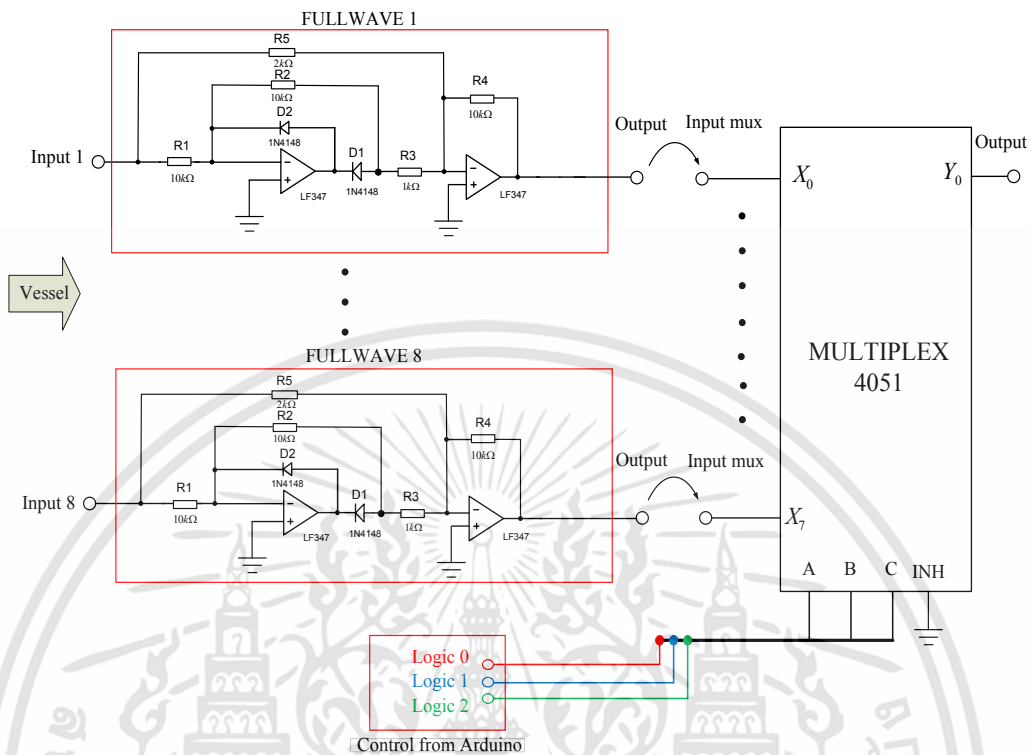
หลังจากทำการต่อวงจรรับกระแสลงบนไฟโต้บอร์ดแล้ว จะทำการป้อนสัญญาณ Sine ความถี่ 1 kHz ขนาด $2 V_{pp}$ ลงไปที่ขาอินพุตขาลบของ LF347 ป้อนไฟบวก V_{CC} ไปที่ขาที่ 4 มีค่าเป็น +7 Volt ป้อนไฟลบ V_{EE} ไปที่ขาที่ 11 มีค่าเป็น -7 Volt แสดงดังรูปที่ 4.20



รูปที่ 4.20 ผลลัพธ์ของสัญญาณที่ได้ออกมาจากวงจรรับกระแส

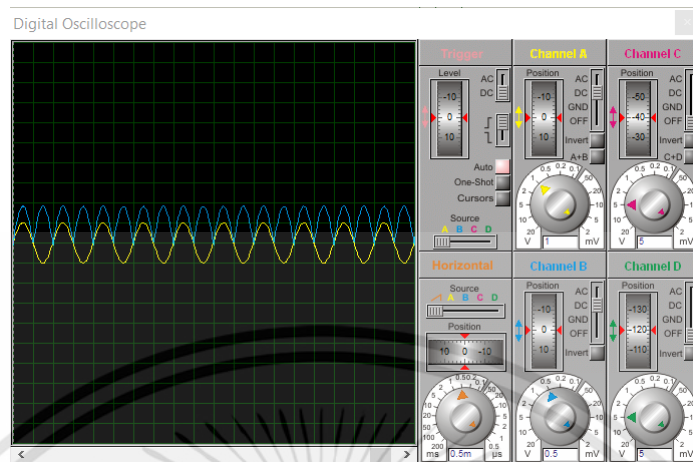
4.2.3 ทดสอบการควบคุมการรับกระแสโดยใช้ Arduino Uno R3 ใน Proteus

ในการทดสอบควบคุมการรับกระแสจากตัว Vessel ใน EIT system ในโปรแกรม Proteus จะเห็นได้ว่าตัวกระแสผ่านวงจร Full wave ก่อนที่จะเข้ามัลติเพล็กซ์เซอร์ โดยในโปรแกรมนั้น จะจำลองกระแสที่ได้จากวงจร Full wave เพียงวงจรเดียว และใช้ตัวต้านทานปรับค่าได้จำนวน 8 ตัวจำลองแทนความต้านทานที่อ่านได้จาก Vessel ของ EIT system โดยการทดสอบจะเริ่มจากป้อนสัญญาณไซน์ ความถี่ 1 กิโลเฮิร์ตซ์ มีเป็นขนาด $2 V_{pp}$ ให้กับวงจร Full wave หลังจากผ่านวงจร Full wave กระแสจะไหลผ่านตัวต้านทานปรับค่าได้ ก่อนที่จะเข้ามัลติเพล็กซ์เซอร์เบอร์ HC4051 เป็นอินพุต 8 ตัว โดยการทำงานของมัลติเพล็กซ์เซอร์ให้รับค่ากระแสจาก channel ที่ต้องการนั้น จะมี Arduino Uno R3 คอยควบคุมการทำงานของมัลติเพล็กซ์เซอร์ไว้ ดังแสดงในรูปที่ 4.21



รูปที่ 4.21 วงจรทดสอบควบคุมการรับกระแสในโปรแกรม Proteus

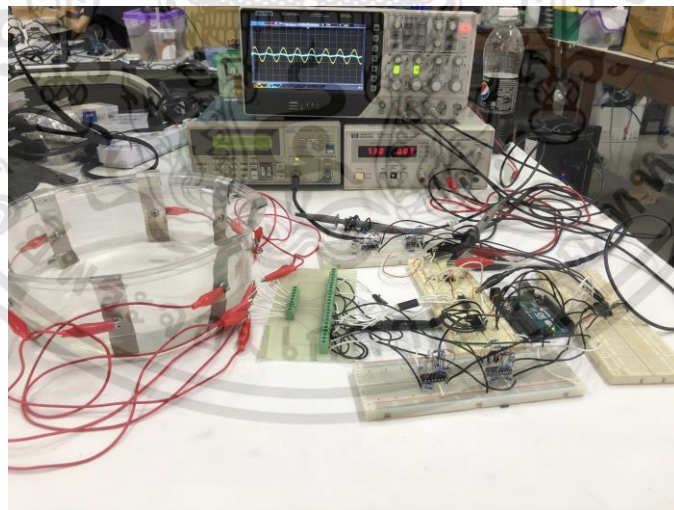
ในการทดสอบวงจรรับกระแสในโปรแกรม Proteus ที่ได้ออกแบบไว้ โดยจะป้อนสัญญาณ Sine เข้าที่ขาอินพุตของ LF347 โดยจะต่อเข้ากับตัวต้าน 10k โอห์มก่อนที่จะเข้าขาอินพุตขาลบ อินพุตบวกขา 3 และ 5 ต่อลงกราวด์ ขา 4 ต่อเข้ากับไฟบวก ขา 11 ต่อเข้ากับไฟลบ โดยขาเอาต์พุตขาที่จะต่อกับไดโอดเบอร์ 1N4148 สองตัว เพื่อให้กระแสไหลไปทางเดียว ส่วนกระแสที่ไหลตรงข้ามจะถูกกั้นไว้โดยจะใช้ขาแคโทดเป็นขาอินพุตและแอนโนดเป็นเอาต์พุต ทำให้สัญญาณที่ได้ออกมาเป็นสัญญาณในลักษณะกลับเฟส โดยที่ไม่มีส่วนประกอบของไฟ DC จากนั้นจะนำสัญญาณนี้ไปรวมกับสัญญาณอินพุต ก็จะทำให้ได้สัญญาณที่มีแต่ฝั่งของไฟบวก โดยที่ไม่มีส่วนประกอบของไฟ DC เลย โดยเมื่อทำการวัดสัญญาณที่ได้ออกมาจะได้ดังนี้ โดยสี่เหลี่ยมคือสัญญาณอินพุต สีน้ำเงินคือสัญญาณอินพุตที่กลับเฟส และไม่มีส่วนประกอบของไฟ DC และสี่เหลี่ยมคือสัญญาณเอาต์พุตที่มีแต่สัญญาณฝั่งบวก และมีขนาดเป็น 5 เท่าของสัญญาณเดิม ดังแสดงในรูปที่ 4.22



รูปที่ 4.22 ผลลัพธ์ของสัญญาณที่ได้ออกมา

4.2.4 ผลทดสอบการควบคุมการรับกระแสโดยใช้ Arduino Uno R3 บนแผ่นไฟ โต้บอร์ด

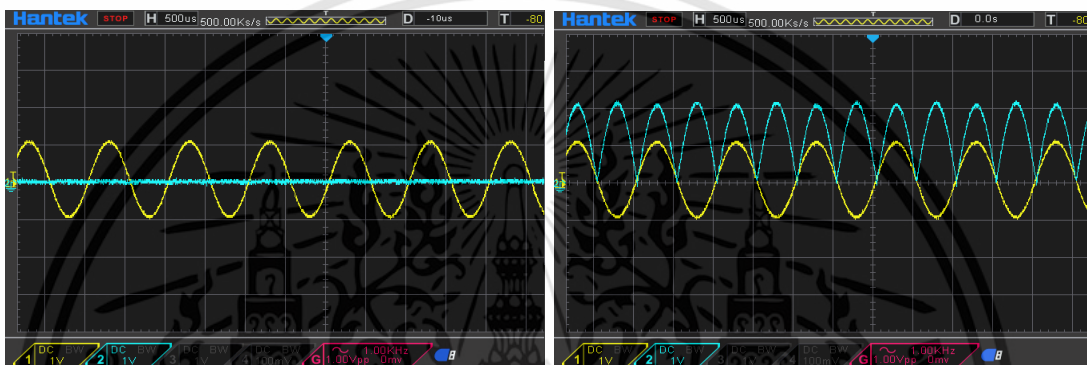
ทำการทดสอบของจริงโดยการต่อลงแผ่นไฟโต้บอร์ด ซึ่งในส่วนของวงจรรับกระแสใช้ตัวมัลติเพล็กซ์เซอร์ในการรับสัญญาณจากจุดใน Vessel ทั้ง 8 ตัว ต่อ 1 รูปแบบการจ่ายสัญญาณ รูปแบบการวัดสามารถแสดงได้ดังรูปที่ 4.7(ข) โดยตัวมัลติเพล็กซ์เซอร์ที่รับสัญญาณจะมี Arduino Uno R3 ควบคุมการทำงานอยู่ โดยสามารถแสดงได้ดังรูปที่ 4.23



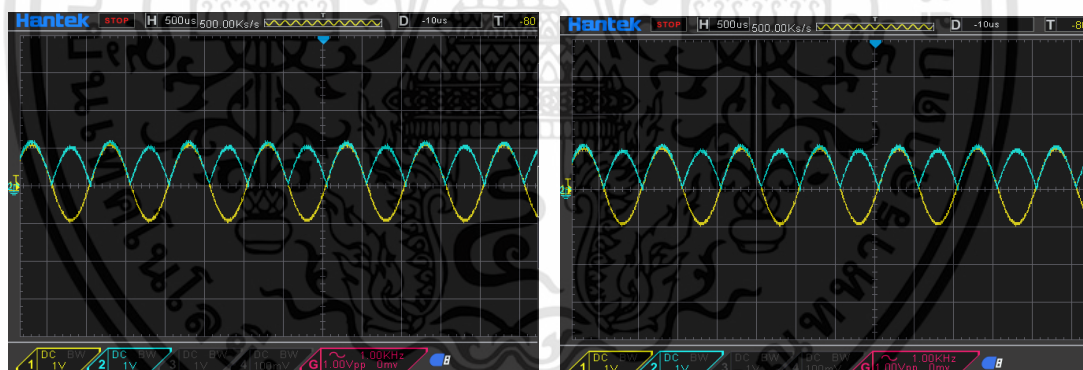
รูปที่ 4.23 การทดสอบวงจรรับกระแส โดยใช้ Arduino Uno R3 ในการควบคุม

เมื่อทำการต่อวงจรรับกระแสเข้าตัว Vessel ที่ได้มีการต่อวงจรจ่ายกระแสเข้าไปเรียบร้อยแล้ว ก็จะทำกรวัดสัญญาณที่ได้จากขั้ว Electrode รอบ Vessel โดยผ่านวงจรรับกระแส

และผ่านตัวมัลติเพล็กซ์เซอร์เป็นตัวสุดท้าย จากนั้นก็จะวัดสัญญาณที่ออกมาจากมัลติเพล็กซ์เซอร์ โดยสัญญาณที่ออกมาจะเป็นสัญญาณที่ได้จากขั้ว Electrode โดยการวัดในที่นี้จะใช้รูปแบบตามรูปที่ 4.7(ข) ซึ่งสัญญาณที่วัดออกมาจะเรียงลำดับกันออกมาตั้งแต่ Electrode ตัวที่ 1 ถึง 8 โดยทดสอบนี้จะทำการวัดด้วยกันสี่จุดแสดงในรูปที่ 4.24 และ 4.25 ซึ่งทั้งสองรูปนี้เป็นสัญญาณที่วัดได้จาก Electrode ตัวที่ 1, 2, 3, และ 4 โดยสัญญาณสี่เหลี่ยมเป็นอินพุต สัญญาณสี่ฟ่าเป็นเอาต์พุตที่วัดจากวงจรรับกระแส



รูปที่ 4.24 สัญญาณที่วัดได้จาก Electrode ตัวที่ 1 และตัวที่ 2

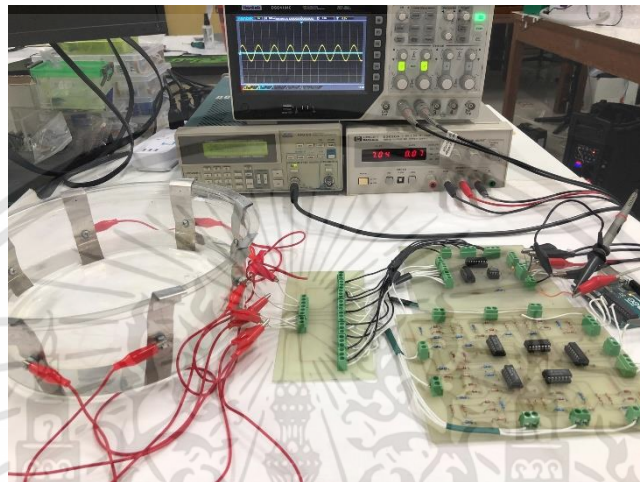


รูปที่ 4.25 สัญญาณที่วัดได้จาก Electrode ตัวที่ 3 และ ตัวที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

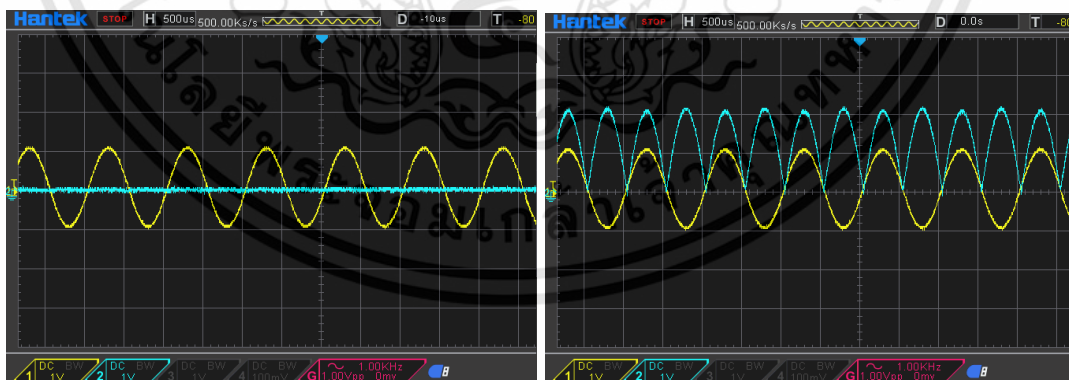
4.2.5 ทดสอบการทำงานของวงจรรีเซ็ต

การทดสอบการทำงานของวงจรรีเซ็ตที่ได้จากการทำแผ่น PCB ออกมานั้น จะสามารถทดสอบการทำงานกับ Vessel เพื่อตรวจสอบว่าวงจรทำงานได้ ดังแสดงในรูปที่ 4.26



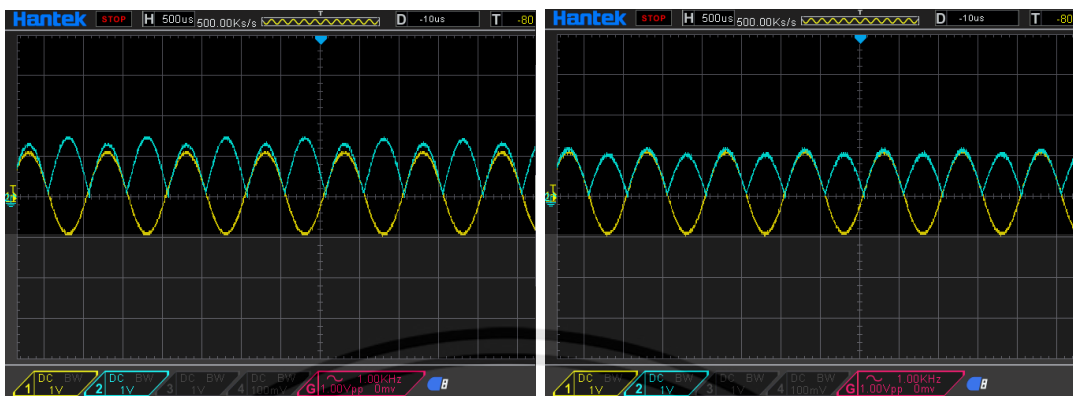
รูปที่ 4.26 ทดสอบการทำงานของ PCB ของวงจรรีเซ็ต

เมื่อทำการต่อวงจรจ่ายกระแสเข้ากับตัว Vessel แล้ว ก็จะทำให้การป้อนสัญญาณและเชื่อมต่อกับ Arduino Uno R3 เพื่อควบคุมการทำงานของวงจร แล้วจากนั้นก็วัดสัญญาณที่ขั้ว Electrode 8 จุด โดยดูแต่ละจุดว่าสัญญาณได้ตามที่ควรเป็นไหมดังรูปที่ 4.27, 4.28, 4.29 และ 4.30

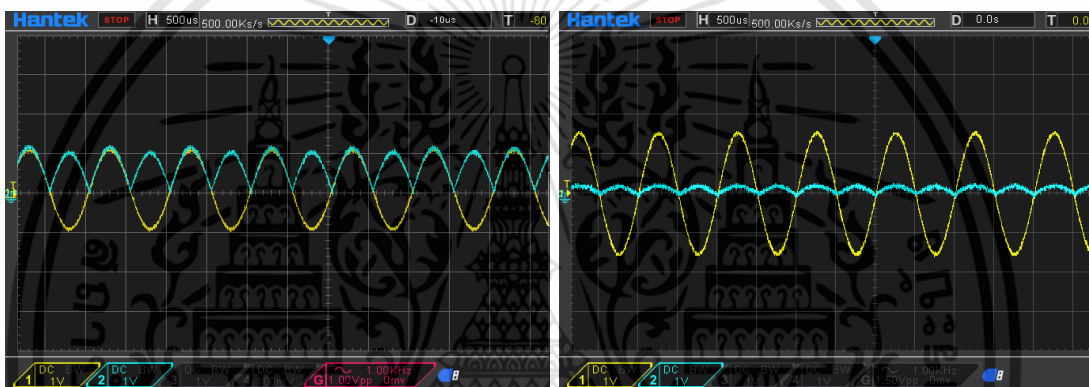


รูปที่ 4.27 สัญญาณที่วัดได้จาก Electrode ตัวที่ 1 และตัวที่ 2

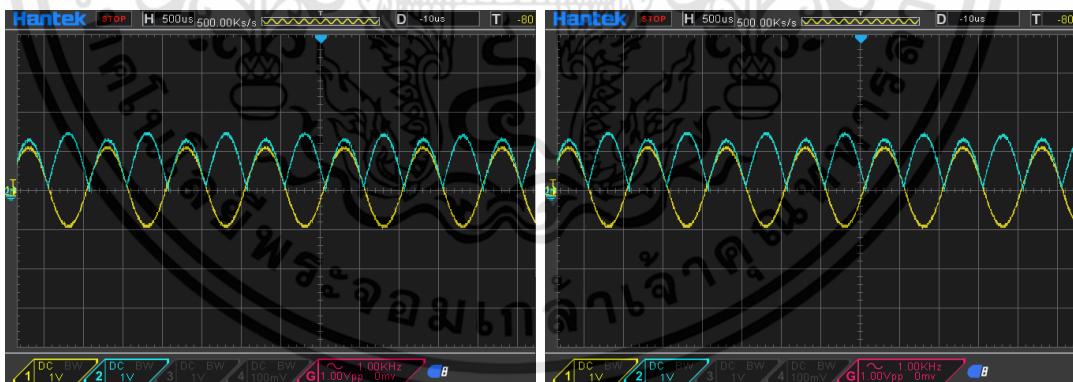
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 สัญญาณที่วัดได้จาก Electrode ตัวที่ 3 และตัวที่ 4



รูปที่ 4.29 สัญญาณที่วัดได้จาก Electrode ตัวที่ 5 และตัวที่ 6



รูปที่ 4.30 สัญญาณที่วัดได้จาก Electrode ตัวที่ 7 และตัวที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบการทำงานของวงจรทั้งหมด

4.3.1 ทดสอบการทำงานของวงจรทั้งหมดโดยใช้ Arduino ควบคุมบน Proteus

การทดสอบการทำงานของวงจรทั้งหมดโดยใช้ Arduino จะเขียนโค้ดควบคุมการทำงานของตัวมัลติเพล็กซ์เซอร์ ของทั้งวงจรจ่ายกระแสและวงจรรับกระแส โดยจะทำการรวมโค้ดของทั้งฝั่งส่งและรับเข้าด้วยกัน ดังแสดงในรูปที่ 4.31

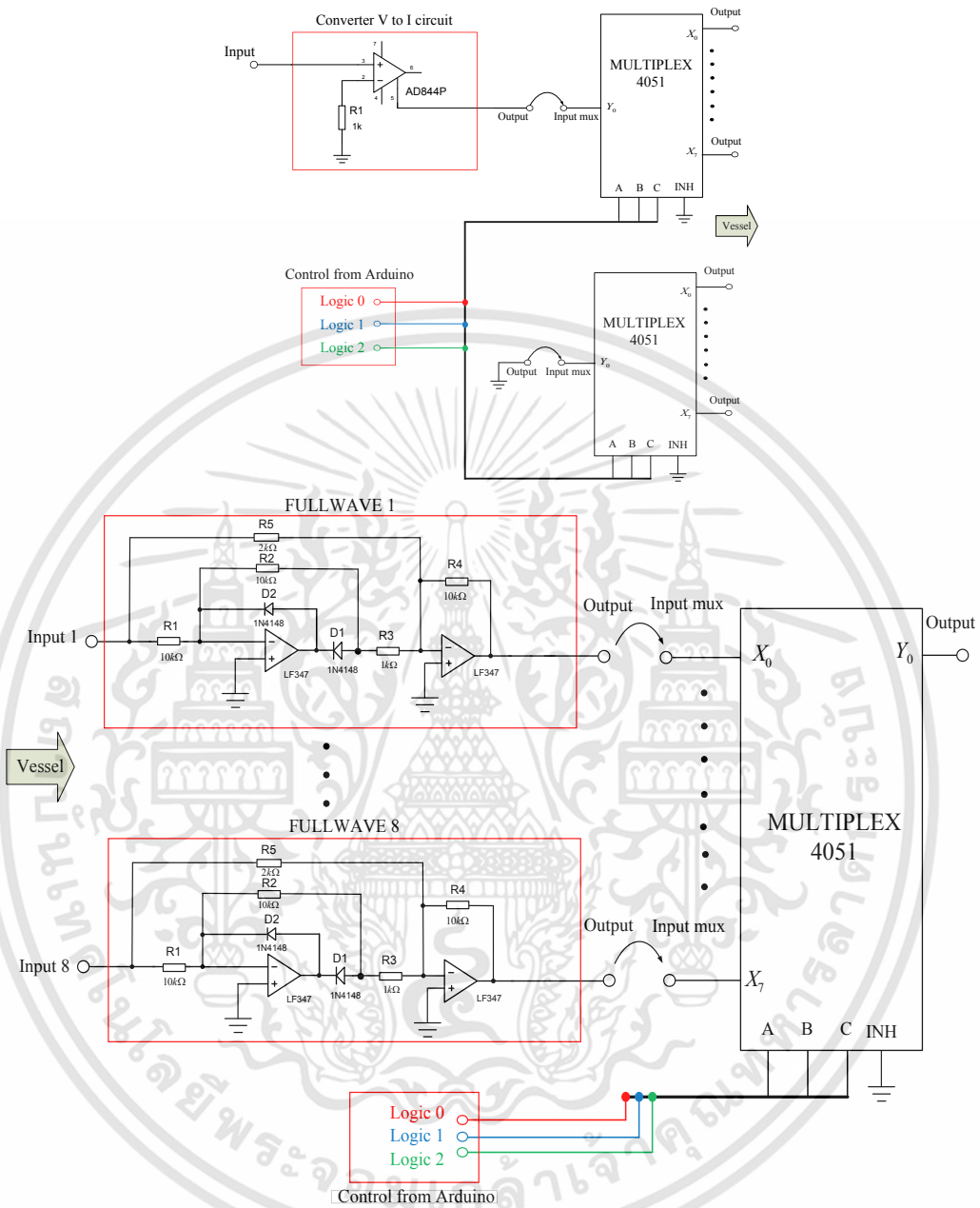
```

for(int i = 0; i < 8 ; i++){
  s3 = bitRead(i,0);
  s4 = bitRead(i,1);
  s5 = bitRead(i,2);
  digitalWrite(dPin,s3);
  digitalWrite(ePin,s4);
  digitalWrite(fPin,s5);
  Serial.print("round ");
  Serial.println(i);
  delay(1);
for(int i = 0 ; i < 8 ; i++){
  s0 = bitRead(i,0);
  s1 = bitRead(i,1);
  s2 = bitRead(i,2);
  digitalWrite(aPin,s0);
  digitalWrite(bPin,s1);
  digitalWrite(cPin,s2);
  delayMicroseconds(1);

```

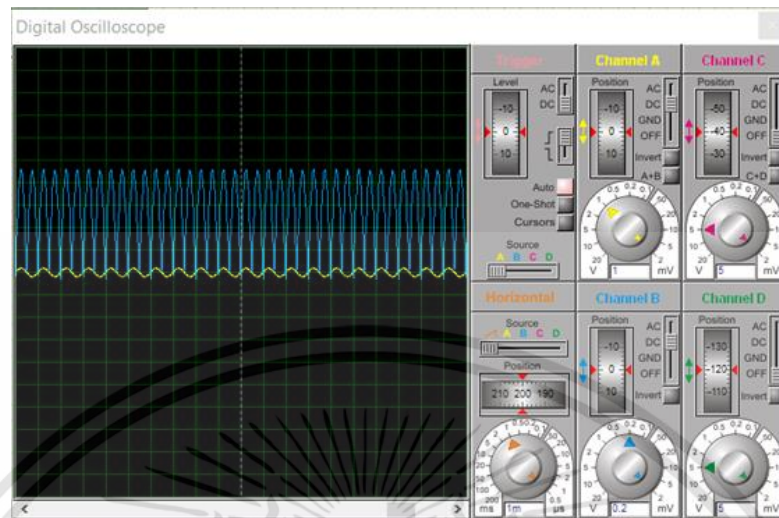
รูปที่ 4.31 โค้ดการทำงานของวงจรจ่ายกระแสและรับกระแส

ทำการทดสอบในโปรแกรม Proteus ก่อน โดยจะทำการต่อวงจรจ่ายกระแสและรับกระแสรวมกันทั้งหมด โดยมี Arduino Uno R3 ในการควบคุมทั้งหมด ดังรูปที่ 4.32 จากนั้นจะทำการดูผลของสัญญาณที่ได้ออกมา โดยจะดูสัญญาณที่ออกมาจากวงจรจ่ายกระแส และวงจรรับกระแส โดยจะดูหลังจากที่สัญญาณผ่านตัวมัลติเพล็กซ์เซอร์มาแล้ว แสดงได้ดังรูปที่ 4.33 ซึ่งสัญญาณที่ได้มา สัญญาณสีเหลืองจะเป็นสัญญาณอินพุต และสีน้ำเงินจะเป็นสัญญาณเอาต์พุตที่มีแต่ฝั่งบวก



รูปที่ 4.32 วงจรการทำงานทั้งหมดบนโปรแกรม Proteus โดยใช้ Arduino ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

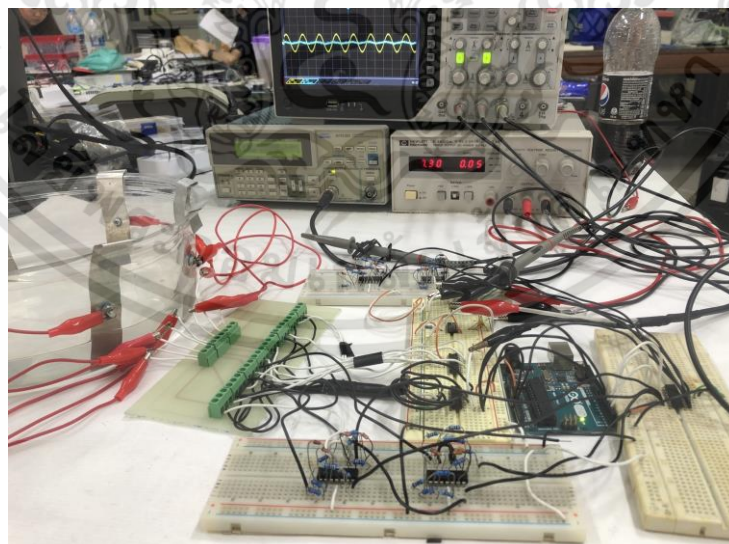


รูปที่ 4.33 ผลลัพธ์ของสัญญาณที่ได้จากการทดสอบบนโปรแกรม Proteus

4.3.2 ทดสอบการทำงานของวงจรทั้งหมดบนแผ่นไฟโต้บอร์ดโดยใช้ Arduino

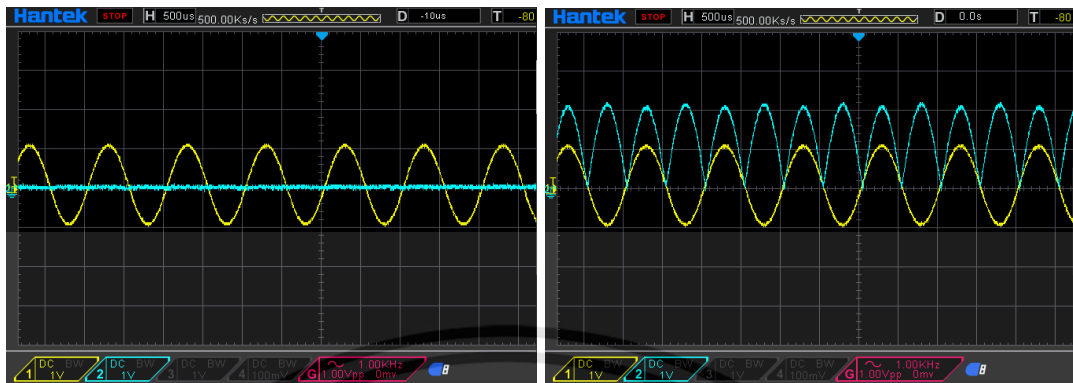
ควบคุม

ทำการทดสอบจริงโดยทำการต่อวงจรจ่ายและรับกระแสบนแผ่นไฟโต้บอร์ดเข้ากับขั้ว Electrode รอบ Vessel โดยใช้ Arduino Uno R3 ควบคุมการทำงานทั้งหมด ดังแสดงในรูปที่ 4.34 จากนั้นทำการวัดสัญญาณที่รับได้ ดังแสดงในรูปที่ 4.35 โดยสัญญาณที่ออกมาสี่เหลี่ยมคือ สัญญาณอินพุตและสี่ฟาคือสัญญาณเอาต์พุต



รูปที่ 4.34 วงจรทั้งหมดบนแผ่นไฟโต้บอร์ดโดยใช้ Arduino ควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

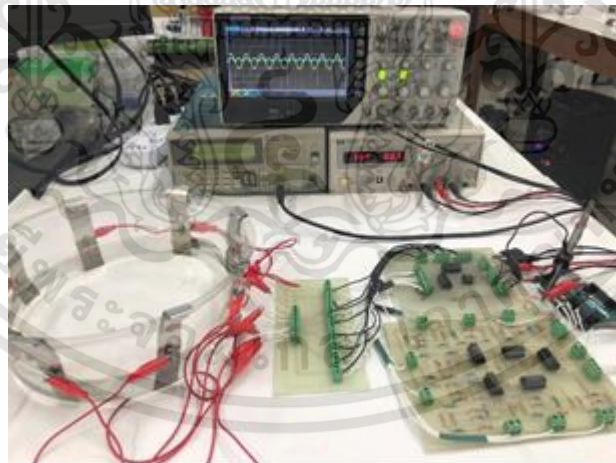


รูปที่ 4.35 สัญญาณที่ได้จาก Electrode ตัวที่ 1 และตัวที่ 2 ในการทดสอบการทำงานวงจรทั้งหมดบนโฟโต้บอร์ดโดยใช้ Arduino ควบคุม

4.3.3 ทดสอบการทำงานของวงจรบนแผ่น PCB ทั้งหมดโดยใช้ Arduino

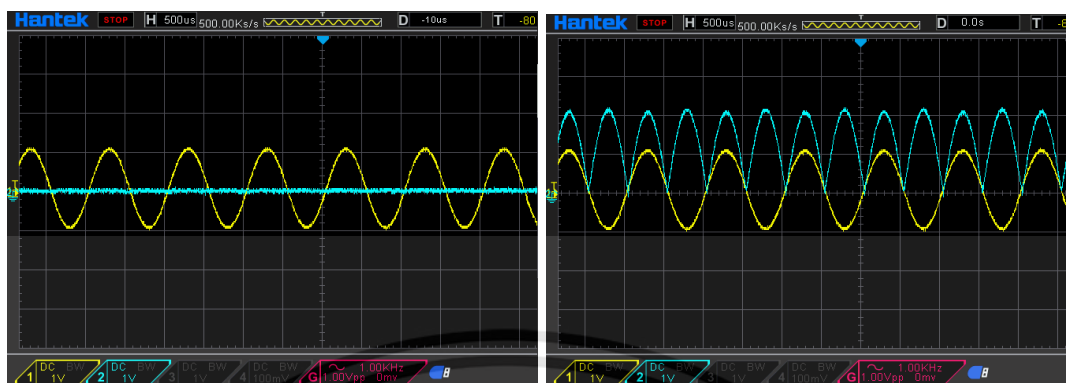
ควบคุม

ทำการทดสอบจริงโดยใช้วงจรจ่ายและรับกระแสบนแผ่น PCB ทั้งสองวงจรต่อเข้ากับขั้ว Electrode รอบ Vessel โดยใช้ Arduino Uno R3 ควบคุมการทำงานทั้งหมด ดังแสดงในรูปที่ 4.36 จากนั้นทำการวัดสัญญาณที่ได้รับได้ ดังแสดงในรูปที่ 4.37 โดยสัญญาณที่ออกมาสี่เหลี่ยมคือสัญญาณอินพุตและสี่ฟากคือสัญญาณเอาต์พุต



รูปที่ 4.36 วงจรทั้งหมดบนแผ่น PCB โดยใช้ Arduino ควบคุมการทำงาน

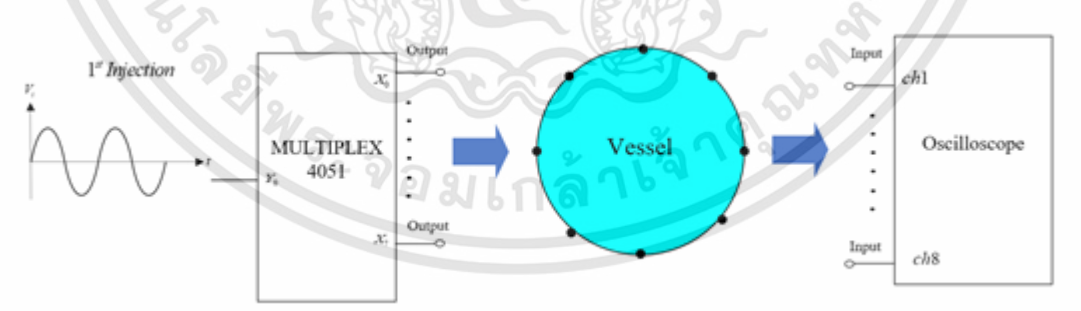
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



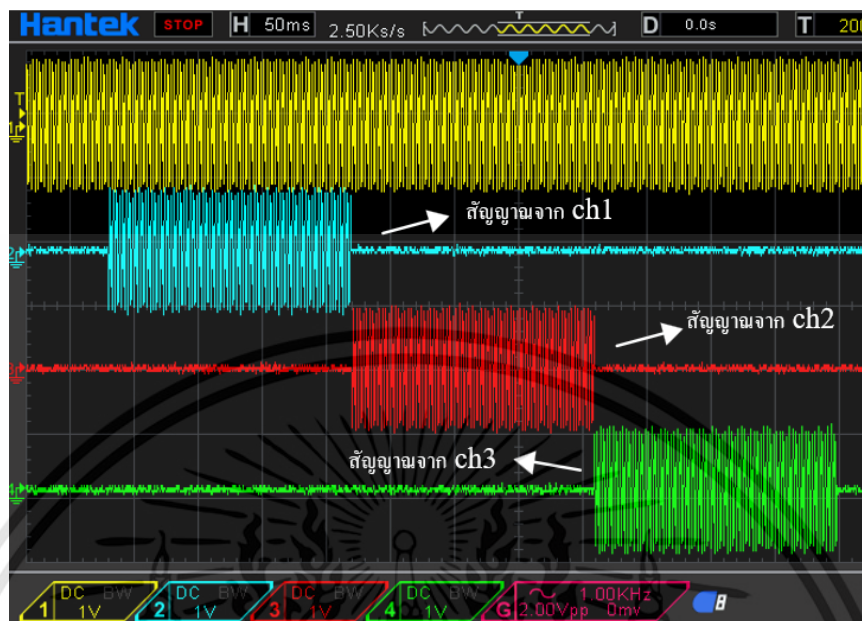
รูปที่ 4.37 สัญญาณที่ได้จาก Electrode ตัวที่ 1 และตัวที่ 2 ในการทดสอบการทำงานวงจรทั้งหมดบนแผ่น PCB โดยใช้ Arduino ควบคุม

4.3.4 ทดสอบการทำงานของวงจรมัลติเพล็กซ์เซอร์บนแผ่น PCB ทั้งหมดโดยใช้ Arduino ควบคุม

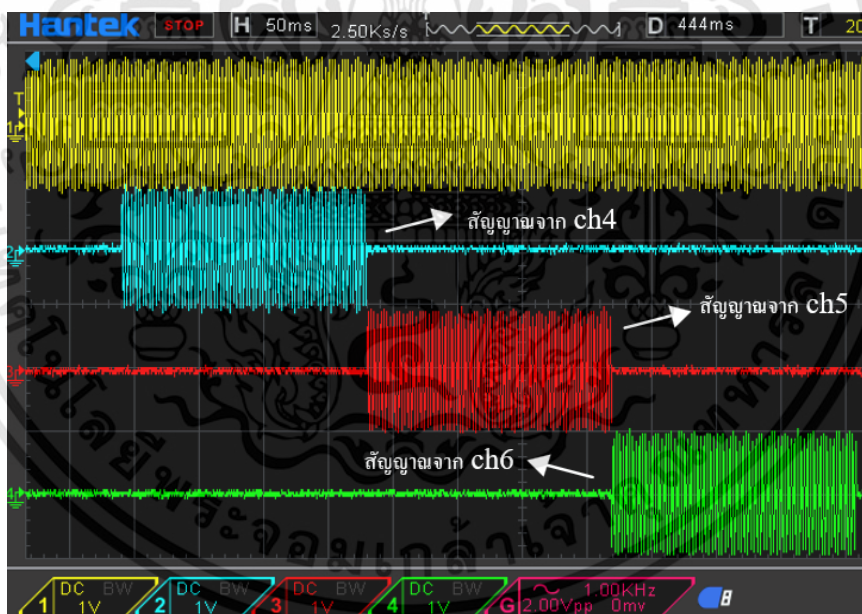
ทำการทดสอบการทำงานของวงจรมัลติเพล็กซ์เซอร์โดยใช้วงจรจ่ายกระแสและรับกระแสบนแผ่น PCB โดยทั้งสองวงจรต่อเข้ากับขั้ว Electrode รอบ Vessel โดยใช้ Arduino Uno R3 ควบคุมการทำงานทั้งหมด โดยตัวอย่างของการวัดสัญญาณจะวัดที่มัลติเพล็กซ์เซอร์ที่ต่อเข้ากับขั้ว Electrode จากนั้นวัดสัญญาณจาก Electrode 8 จุดดังรูปที่ 4.38 เพื่อดูว่าสัญญาณเอาต์พุตออกมาตามตารางที่ 4.1 หรือไม่ โดยวัดจากทั้งด้านฝั่งส่ง สามารถแสดงในรูปที่ 4.39, 4.40 และ 4.41 จากฝั่งรับแสดงดังรูปที่ 4.42 , 4.43 และ 4.44 ซึ่งเอาต์พุตของทั้งฝั่งส่งและรับ ที่ออกมาได้จะเรียงตามตารางที่ 4.1



รูป 4.38 การวัดสัญญาณจาก Multiplexer

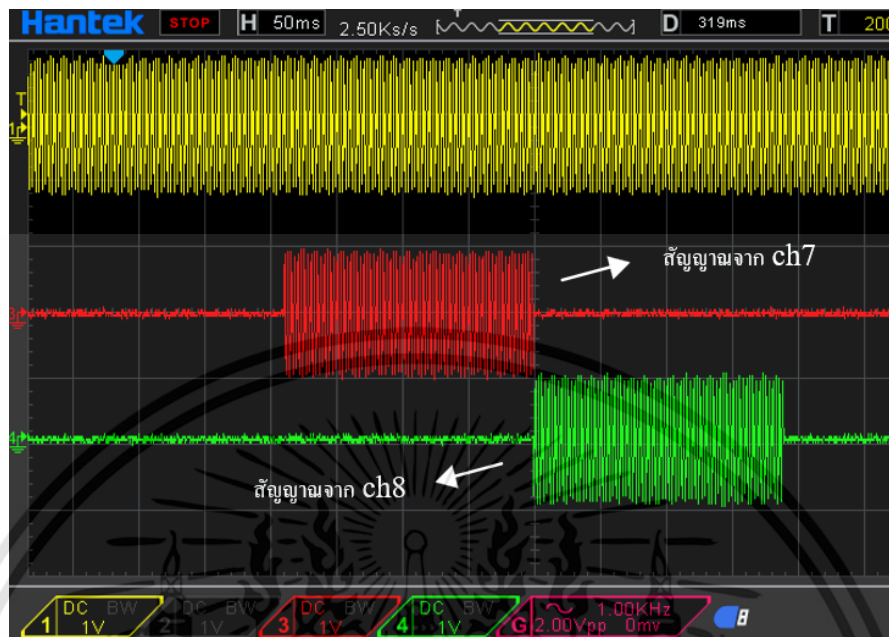


รูปที่ 4.39 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 1 , Channel 2 และ Channel 3

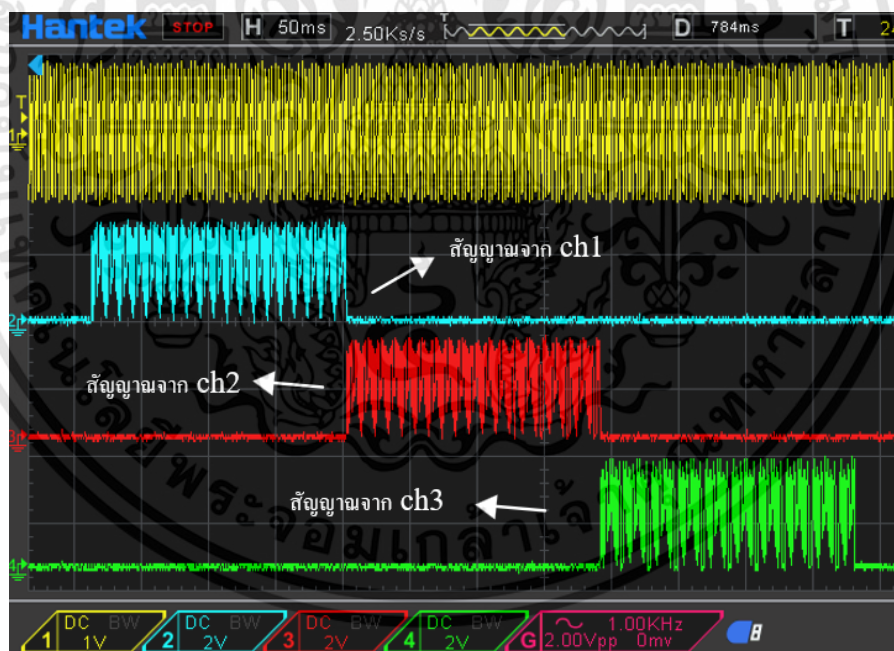


รูปที่ 4.40 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 4 , Channel 5 และ Channel 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

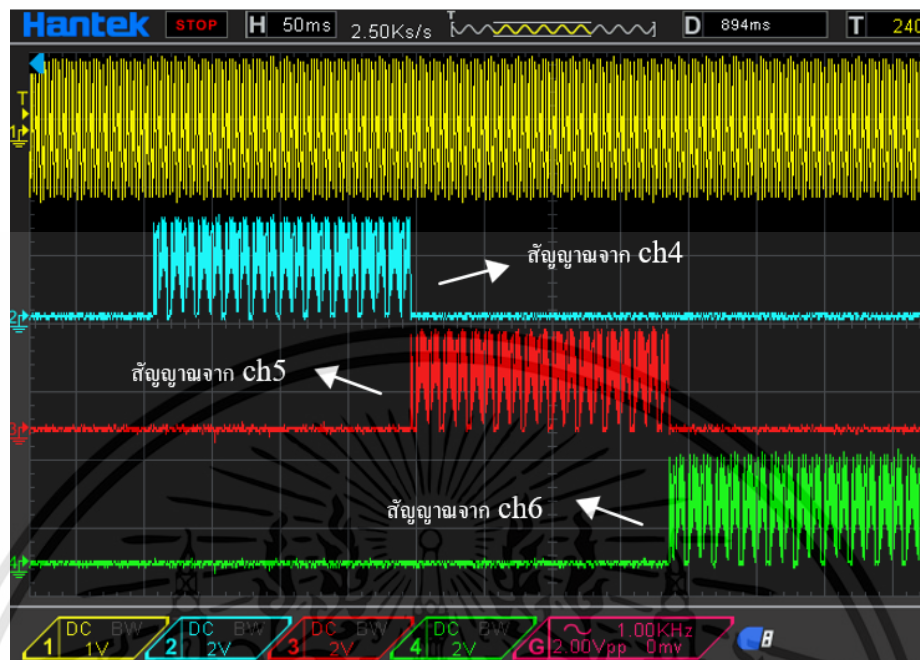


รูปที่ 4.41 สัญญาณที่วัดได้จากฝั่งส่ง Multiplexer Channel 7 และ Channel 8

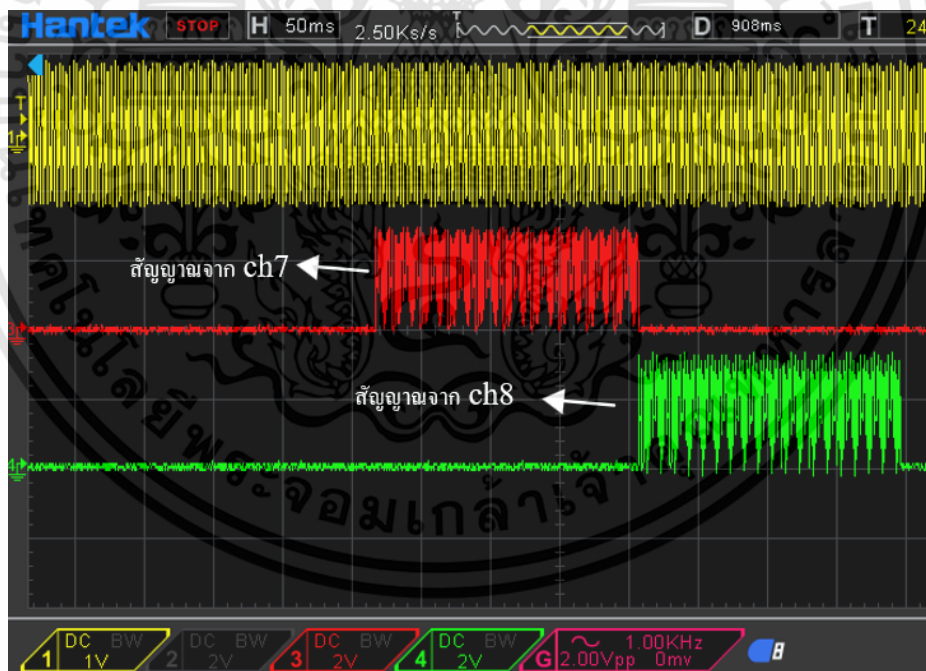


รูปที่ 4.42 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 1 , Channel 2 และ Channel 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.43 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 4 , Channel 5 และ Channel 6



รูปที่ 4.44 สัญญาณที่วัดได้จากฝั่งรับ Multiplexer Channel 7 และ Channel 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การวิเคราะห์และประมวลผลสัญญาณดิจิทัลเป็นภาพ

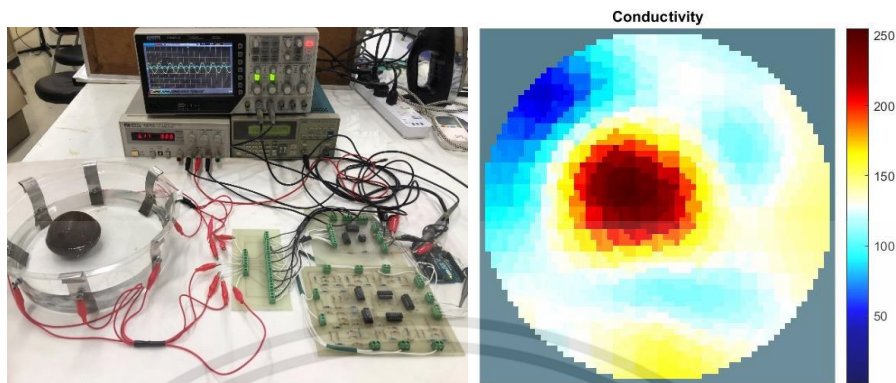
เป็นการวิเคราะห์และประมวลผลภาพจากสัญญาณดิจิทัลที่ได้จากชุด EIT เข้าโปรแกรม EIDORS ผ่านทางโปรแกรม MATLAB โดยใช้โค้ดวิเคราะห์ภาพดังหัวข้อที่ 3.1.7

4.4.1 ทดสอบประมวลผลเป็นภาพ กรณีใส่วัตถุที่มีความนำไฟฟ้าสูง

ในการทดสอบการวิเคราะห์สัญญาณดิจิทัลเป็นภาพ จำเป็นที่จะต้องมียุทธศาสตร์ 2 ส่วนด้วยกัน คือค่าข้อมูลก่อนหน้า (image prior) และค่าไฮเปอร์พารามิเตอร์ (hyperparameter) ซึ่งโดยทั่วไปค่าไฮเปอร์พารามิเตอร์จะมีค่าคงที่ จากการต่อชุดอุปกรณ์ EIT ดังแสดงในภาพที่ 4.45 โดยใส่ลูกเหล็กที่มีความนำไฟฟ้าสูงลงไปในอุปกรณ์ และทำการเก็บค่าสัญญาณ 2 ชุด ซึ่งค่าที่ทำการวัดแรงดันตกคร่อมระหว่าง Electrode ข้างเคียง ได้ผลดังตารางที่ 4.2

ตารางที่ 4.2 การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าสูง

Electrode Round	(2,1)	(3,2)	(4,3)	(4,5)	(5,6)	(6,7)	(7,8)	(8,1)
1	0.15	-1.85	0.26	-0.03	0.01	-0.03	0.17	-0.22
2	-0.17	0.18	-1.68	-0.18	0.02	-0.05	-0.07	-0.10
3	-0.27	-0.18	0.14	-1.85	0.26	-0.07	0.04	0.06
4	0.08	0.25	-0.51	0.11	-2.05	0.11	0.01	0.03
5	0.04	0.14	0.32	-0.32	0.15	-1.85	0.17	0.03
6	0.01	0.10	0.15	0.29	-0.28	0.17	-2.00	0.17
7	-0.05	0.05	0.02	0.10	0.22	-0.36	0.19	-1.89
8	-1.86	0.18	0.04	-0.01	0.40	0.35	-0.22	0.18



รูปที่ 4.45 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าสูง

หลังจากที่เก็บค่าสัญญาณดิจิทัลจาก Vessel ที่ยังไม่ม้วัดลงไป เป็นค่าข้อมูลชุดที่ 1 และค่าสัญญาณดิจิทัลจาก Vessel ที่มีวัดความนำไฟฟ้าสูงลงไป เป็นค่าข้อมูลชุดที่ 2 แล้ว ทำการประมวลผลผ่านโปรแกรม EIDORS แล้วจะได้ภาพดังที่แสดงในรูปที่ 4.45

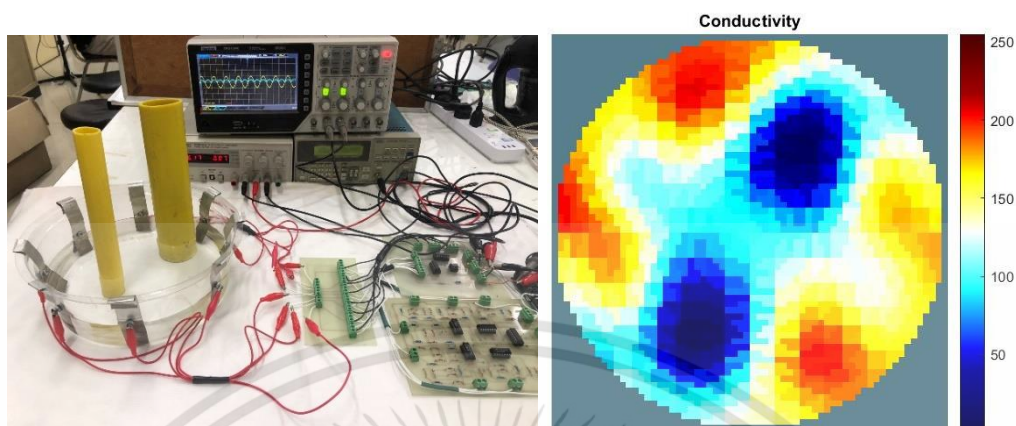
4.4.2 ทดสอบประมวลผลเป็นภาพ กรณีใส่วัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุ

จากการต่อชุดอุปกรณ์ EIT ดังภาพที่ 4.46 โดยวัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุลงในอุปกรณ์ และทำการเก็บค่าสัญญาณ 2 ชุด ซึ่งค่าที่ทำการวัดแรงดันตกคร่อมระหว่าง Electrode ข้างเคียง ได้ผลดังตารางที่ 4.3

ตารางที่ 4.3 การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุ

Electrode Round	(2,1)	(3,2)	(4,3)	(4,5)	(5,6)	(6,7)	(7,8)	(8,1)
1	0.25	-1.91	-0.02	0.04	-0.01	0.10	0.23	-0.32
2	-0.23	0.20	-1.73	0.01	-0.04	0.05	0.08	0.23
3	0.28	-0.24	0.14	-1.91	0.27	-0.04	0.01	0.05
4	0.09	0.25	-0.66	0.15	-2.2	0.15	0.01	-0.01
5	0.03	0.12	0.34	-0.5	0.16	-1.85	0.22	-0.02
6	-0.01	0.09	0.12	0.24	-1.29	0.24	-1.94	0.21
7	0.02	0.05	0.04	0.10	0.17	-1.33	0.20	-1.87
8	-1.89	0.06	0.01	-0.08	0.03	0.30	-1.25	0.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.46 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าต่ำสองวัตถุ

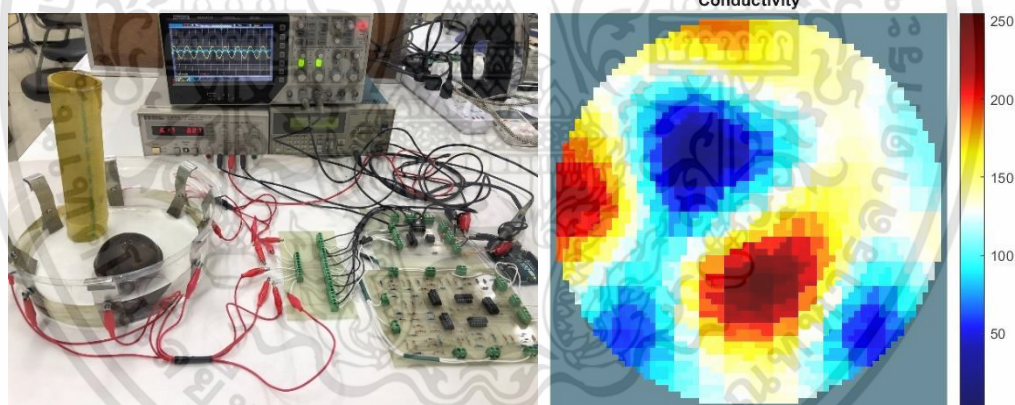
หลังจากที่เก็บค่าสัญญาณดิจิทัลจาก Vessel ที่ยังไม่มีวัตถุลงไป เป็นค่าข้อมูลชุดที่ 1 และค่าสัญญาณดิจิทัลจาก Vessel ที่มีวัตถุความนำไฟฟ้าต่ำสองวัตถุลงไป เป็นค่าข้อมูลชุดที่ 2 แล้ว ทำการประมวลผลผ่านโปรแกรม EIDORS แล้วจะได้ภาพดังที่แสดงในรูปที่ 4.46

4.4.3 ทดสอบประมวลผลเป็นภาพ กรณีใส่วัตถุที่มีความนำไฟฟ้าสูงและความนำไฟฟ้าต่ำ

จากการต่อชุดอุปกรณ์ EIT ดังภาพที่ 4.47 โดยใส่วัตถุที่มีความนำไฟฟ้าสูงและวัตถุที่มีความนำไฟฟ้าต่ำ ลงไปในอุปกรณ์ และทำการเก็บค่าสัญญาณ 2 ชุด ซึ่งค่าที่ทำกรวัดแรงดันตกคร่อมระหว่าง Electrode ข้างเคียง ได้ผลดังตารางที่ 4.4

ตารางที่ 4.4 การวัดค่าแรงดันไฟฟ้าสำหรับ Electrode จากจุดที่ 1-8 กรณีใส่วัตถุที่มีความนำไฟฟ้าสูงและความนำไฟฟ้าต่ำ

Electrode Round	(2,1)	(3,2)	(4,3)	(4,5)	(5,6)	(6,7)	(7,8)	(8,1)
1	0.25	-1.90	0.25	0.01	-0.01	0.11	0.24	-1.30
2	-1.23	0.19	-1.74	-0.01	-0.04	0.05	0.11	0.21
3	0.28	-1.23	0.15	-1.89	0.26	-0.05	0.02	0.05
4	0.08	0.27	-0.62	0.22	-2.13	0.13	-0.01	0.03
5	0.02	0.12	0.31	-1.08	0.17	-1.71	0.26	0.01
6	-0.04	0.06	0.06	0.26	-1.17	0.17	-1.76	0.26
7	-0.06	0.03	-0.03	0.12	0.11	-1.01	0.27	-1.27
8	-1.59	0.05	-0.03	0.08	0.04	0.17	-1.05	0.23



รูปที่ 4.47 ภาพที่ได้จากวัตถุที่มีความนำไฟฟ้าสูงและความนำไฟฟ้าต่ำ

หลังจากที่เก็บค่าสัญญาณดิจิทัลจาก Vessel ที่ยังไม่มีวัตถุลงไป เป็นค่าข้อมูลชุดที่ 1 และค่าสัญญาณดิจิทัลจาก Vessel ที่มีวัตถุที่มีความนำไฟฟ้าสูงและวัตถุที่มีความนำไฟฟ้าต่ำลงไป เป็นค่าข้อมูลชุดที่ 2 แล้ว ทำการประมวลผลผ่านโปรแกรม EIDORS แล้วจะได้ภาพดังที่แสดงในรูปที่ 4.47

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีเป้าหมายเพื่อศึกษา ออกแบบ และสร้างระบบสร้างภาพจากความต้านทานด้วยโครงสร้างไมซ์บ์ซ็อนและใช้ต้นทุนต่ำ โดยผู้จัดทำได้ทำศึกษา การออกแบบและสร้างระบบสร้างภาพจากความต้านทาน ด้วยวงจรจ่ายกระแสจะใช้เพียงวงจรขยายสัญญาณแบบป้อนกระแสกลับ (CFOA) เพียงตัวเดียว ทำการต่อยอดกับวงจรมัลติเพล็กซ์เซอร์ ส่วนด้านตรวจจับแรงดันจะใช้วงจรเรียงกระแสแม่นยำแบบเต็มคลื่น เพื่อแปลงสัญญาณไซน์ให้เป็นแรงดันไฟตรง โดยมีไมโครคอนโทรลเลอร์ Arduino Uno R3 ในการควบคุมมัลติเพล็กซ์เซอร์ และเก็บข้อมูลค่าแรงดันไฟฟ้าเพื่อไปประมวลผลโดยใช้ EIDORS ที่ทำงานบนแพลตฟอร์ม MATLAB ซึ่งจากผลการทดลองแสดงให้เห็นว่าสามารถสร้างภาพจากความต้านทานไฟฟ้าได้อย่างถูกต้อง และยืนยันผลได้จากภาพที่ใส่วัตถุลงใน Vessel มีการแสดงถึงความนำไฟฟ้าที่เปลี่ยนไปในภาพ

5.2 ข้อเสนอแนะ

การออกแบบและพัฒนาอุปกรณ์การสร้างภาพจากความต้านทานไฟฟ้าโดยใช้ต้นทุนต่ำ ในส่วนของการออกแบบวงจรขยายสัญญาณแบบป้อนกระแสกลับและวงจรเรียงกระแสแม่นยำแบบเต็มคลื่น พบปัญหาในส่วนของ การออกแบบและสร้างวงจร เนื่องจากสัญญาณที่ได้มีความคลาดเคลื่อนจากทฤษฎีที่ได้มีการศึกษาไว้ ซึ่งมีสาเหตุมาจากสายสัญญาณที่มีจำนวนมาก ทำให้เกิดสัญญาณรบกวนขึ้น มีผลทำให้เอาต์พุตที่ออกมา มีความคลาดเคลื่อนจากที่ได้ศึกษาไว้ ซึ่งสามารถแก้ไขได้โดยลดจำนวนสายสัญญาณลงหรือทำการใช้สายสัญญาณร่วมกัน อีกทั้งยังพบว่าเมื่อใช้ Electrode จำนวน 8 ตัว ภาพความต้านทานไฟฟ้าที่ได้มีความละเอียดค่อนข้างน้อย สามารถแก้ไขโดยการเพิ่มจำนวน Electrode ในชุดอุปกรณ์ EIT และเมื่อนำชุดอุปกรณ์ EIT ที่ออกแบบและสร้างตามปริญญานิพนธ์นี้ จะพบว่ามีความต่ำกว่าที่มีตามท้องตลาดเป็นอย่างมาก

บรรณานุกรม

- [1] Beatriz Lobo, Cecilia Hermosa, Ana Abella and Federico Gordo. “Electrical impedance tomography.”
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5799136/>.
- [2] S. Leonhardt, B. Lachmann. “*Electrical impedance tomography: the holy grail of ventilation and perfusion monitoring?*.” Springer and ESICM. (2012) : 1918-1920.
- [3] อินทอร วงษาเวียง. “ความรู้พื้นฐานเกี่ยวกับโปรแกรม PROTEUS 8 PROFESSIONAL.”
<https://sites.google.com/site/intuonwsw/khea-su-th-reiyn/bth-thi-1-khwam-ru-phun-than-keiyw-kab-porkaerm-proteus-8-professional>.
- [4] ภมร ศิลาพันธ์, วินัย ไจกล้ำ, มนตรี ศิริปรัชญานันท์. *วงจรเรียงกระแสเต็มคลื่นโหมดกระแสแบบแม่นยำและเป็นอิสระจากอุณหภูมิโดยใช้อุปกรณ์แอคทีฟเพียงตัวเดียว*. กรุงเทพฯ : การประชุมวิชาการระดับชาติด้านครุศาสตร์อุตสาหกรรม ครั้งที่ 1, 2551.
- [5] สุรียา ศรีวิเศษ. “บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3.”
<https://sites.google.com/site/mikhorkhxnthorllexr1/bxrd-mikhorkhxnthorllexr-arduino-uno-r3>.
- [6] สมหวัง ศุภพล. “วงจร Multiplexer (MUX) และ วงจร Demultiplexer (DMUX).” หนังสือดิจิทัลประยุกต์. บทที่ 12. หน้า 234-264. เดือนตุลาคม พศ.2561.
- [7] Texas Instruments. “CD405xB CMOS Single 8-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion.”
https://www.ti.com/lit/ds/symlink/cd4051b.pdf?ts=1635385723173&ref_url=https%253A%25
- [8] นิตา ดุก่อง. “ต้นแบบจำลองภาพตัดขวางความต้านทานไฟฟ้าแบบ 3 มิติสำหรับเต้านม” วิทยานิพนธ์, สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, มหาวิทยาลัยสงขลานครินทร์, 2553
- [9] B.M. Graham, “*Enhancements in Electrical Impedance Tomograph (EIT) Image Reconstruction for 3D Lung Imaging.*” PhD thesis, University of Ottawa, November, 2006.

[10] m2html © 2005. “mk_stim_patterns.”

http://eidors3d.sourceforge.net/doc/index.html?eidors/models/mk_stim_patterns.html.

[11] m2html © 2005. “mk_common_model.”

http://eidors3d.sourceforge.net/doc/index.html?eidors/models/mk_common_model.html.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Arduino ควบคุมการทำงานของ Multiplexer

```

#include <Mux.h>

// ReadAnalogAverage
// Reads input A0
// Global Variables
int Min;
int Max;
int Analog;
int numReading = 20;
unsigned long Sum;
unsigned long Average; //Should be equal to the bias.
float AnalogVoltage[8];
float Voltage1[64];

//กำหนด
#define aPin 2
#define bPin 3
#define cPin 4
#define dPin 5
#define ePin 6
#define fPin 7

int s0 = 0;
int s1 = 0;
int s2 = 0;
int s3 = 0;
int s4 = 0;
int s5 = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int num1 = 0;
int num2 = 1;
int num3 = 2;
int num4 = 3;
int num5 = 4;
int num6 = 5;
int num7 = 6;
int num8 = 7;
int num9 = 8;

void setup()
{
  pinMode(aPin,OUTPUT);
  pinMode(bPin,OUTPUT);
  pinMode(cPin,OUTPUT);
  pinMode(dPin,OUTPUT);
  pinMode(ePin,OUTPUT);
  pinMode(fPin,OUTPUT);
  Serial.begin(9600);
  delay(200);/"Stabilization time". Probably not necessary
}

//Main Loop
void loop()
{
  for(int i = 0; i < 8 ; i++){
    s3 = bitRead(i,0);
    s4 = bitRead(i,1);
    s5 = bitRead(i,2);
    digitalWrite(dPin,s3);
    digitalWrite(ePin,s4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(fPin,s5);
delayMicroseconds(1);
for(int i = 0 ; i < 8 ; i++){
s0 = bitRead(i,0);
s1 = bitRead(i,1);
s2 = bitRead(i,2);
digitalWrite(aPin,s0);
digitalWrite(bPin,s1);
digitalWrite(cPin,s2);
delayMicroseconds(1);

Min = 1023; //initilize/reset to limit
Max = 0; //initilize/reset to limit
Sum = 0; //Initialize/reset
//Take 1 round for 20 readings value , find Voltage for each point. This loop takes
about 100ms.
int mynum[20];
for (int i = 0; i <20; i++)
{
Analog = analogRead(A0);
mynum[i] = Analog;
delayMicroseconds(100);
}
bubbleSort(mynum,20);
//Serial.print("Sorted Array: ");
for(int i=0; i<20; i++) {
//Serial.print(mynum[i]);
//Serial.print(",");
}
Max = (mynum[17]+mynum[18]+mynum[19])/3;
AnalogVoltage[i] = (Max* 5.00)/1023.00;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay(1);
}
if(num1 == i){
Voltage1[0] = AnalogVoltage[0];
Voltage1[1] = AnalogVoltage[1];
Voltage1[2] = AnalogVoltage[2];
Voltage1[3] = AnalogVoltage[3];
Voltage1[4] = AnalogVoltage[4];
Voltage1[5] = AnalogVoltage[5];
Voltage1[6] = AnalogVoltage[6];
Voltage1[7] = AnalogVoltage[7];
}
else if(num2 == i){
Voltage1[8] = AnalogVoltage[0];
Voltage1[9] = AnalogVoltage[1];
Voltage1[10] = AnalogVoltage[2];
Voltage1[11] = AnalogVoltage[3];
Voltage1[12] = AnalogVoltage[4];
Voltage1[13] = AnalogVoltage[5];
Voltage1[14] = AnalogVoltage[6];
Voltage1[15] = AnalogVoltage[7];
}
else if(num3 == i){
Voltage1[16] = AnalogVoltage[0];
Voltage1[17] = AnalogVoltage[1];
Voltage1[18] = AnalogVoltage[2];
Voltage1[19] = AnalogVoltage[3];
Voltage1[20] = AnalogVoltage[4];
Voltage1[21] = AnalogVoltage[5];
Voltage1[22] = AnalogVoltage[6];
Voltage1[23] = AnalogVoltage[7];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if(num4 == i){
Voltage1[24] = AnalogVoltage[0];
Voltage1[25] = AnalogVoltage[1];
Voltage1[26] = AnalogVoltage[2];
Voltage1[27] = AnalogVoltage[3];
Voltage1[28] = AnalogVoltage[4];
Voltage1[29] = AnalogVoltage[5];
Voltage1[30] = AnalogVoltage[6];
Voltage1[31] = AnalogVoltage[7];
}
else if(num5 == i){
Voltage1[32] = AnalogVoltage[0];
Voltage1[33] = AnalogVoltage[1];
Voltage1[34] = AnalogVoltage[2];
Voltage1[35] = AnalogVoltage[3];
Voltage1[36] = AnalogVoltage[4];
Voltage1[37] = AnalogVoltage[5];
Voltage1[38] = AnalogVoltage[6];
Voltage1[39] = AnalogVoltage[7];
}
else if(num6 == i){
Voltage1[40] = AnalogVoltage[0];
Voltage1[41] = AnalogVoltage[1];
Voltage1[42] = AnalogVoltage[2];
Voltage1[43] = AnalogVoltage[3];
Voltage1[44] = AnalogVoltage[4];
Voltage1[45] = AnalogVoltage[5];
Voltage1[46] = AnalogVoltage[6];
Voltage1[47] = AnalogVoltage[7];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(num7 == i){
Voltage1[48] = AnalogVoltage[0];
Voltage1[49] = AnalogVoltage[1];
Voltage1[50] = AnalogVoltage[2];
Voltage1[51] = AnalogVoltage[3];
Voltage1[52] = AnalogVoltage[4];
Voltage1[53] = AnalogVoltage[5];
Voltage1[54] = AnalogVoltage[6];
Voltage1[55] = AnalogVoltage[7];
}
else if(num8 == i){
Voltage1[56] = AnalogVoltage[0];
Voltage1[57] = AnalogVoltage[1];
Voltage1[58] = AnalogVoltage[2];
Voltage1[59] = AnalogVoltage[3];
Voltage1[60] = AnalogVoltage[4];
Voltage1[61] = AnalogVoltage[5];
Voltage1[62] = AnalogVoltage[6];
Voltage1[63] = AnalogVoltage[7];
}
}
for(int i=0; i<64; i++){
Serial.print("Voltage value = ");
Serial.println(Voltage1[i]);
delay(1);
}
}
void bubbleSort(int a[], int size) {
for(int i=0; i<(size-1); i++) {
for(int o=0; o<(size-(i+1)); o++) {
if(a[o] > a[o+1]) {

```

```
int t = a[0];  
a[0] = a[0+1];  
a[0+1] = t;  
}  
}  
}  
}  
//End
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code MATLAB ควบคุมการทำงานการแปลงสัญญาณดิจิทัลเป็นภาพ

```
clear
clc

n_elec= 8;
n_rings= 1;
options = {'no_meas_current','no_rotate_meas'};
[st, ms]= mk_stim_patterns(n_elec, n_rings, 'ad}','ad}', options, 2);

inv2d= mk_common_model('c2c',n_elec);
inv2d.fwd_model.stimulation = st;
inv2d.fwd_model.meas_select = ms;

load background.mat
d1= BG1;
d2= data1;

img= inv_solve( inv2d,d1,d2);
show_fem(img);
```