

การออกแบบเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดเพื่อเป็นอุปกรณ์

การสอนเกี่ยวกับการวัดค่าทางไฟฟ้า

DESIGN OF AN ELECTRICALLY VERBAL MEASUREMENT FOR
TEACHING ELECTRICITY MEASUREMENT

นิเทศ ประเดชบุญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาการสอนวิทยาศาสตร์

บัณฑิตวิทยาลัย มหาวิทยาลัยราชภัฏเชียงใหม่

พ.ศ. 2562

หัวข้อวิทยานิพนธ์ : การออกแบบเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดเพื่อเป็นอุปกรณ์การสอนเกี่ยวกับการวัดค่าทางไฟฟ้า

ผู้วิจัย : นิเทศ ประเดชนบุญ

สาขาวิชา : การสอนวิทยาศาสตร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์

: อาจารย์ ดร.ชนัญ วิชาศิลป์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

: ผู้ช่วยศาสตราจารย์ ดร.เสรี ปานซาง

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดเพื่อเป็นสื่อการเรียนการสอนสำหรับผู้พิการทางสายตา โดยทำการออกแบบและพัฒนาเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดจาก มัลติมิเตอร์ระบบตัวเลขตราสินค้า UNI-T รุ่น UT61E ร่วมกับบอร์ด Raspberry Pi Model B+ แล้วทำการทดสอบการวัดค่าทางไฟฟ้าได้แก่ ค่าแรงดันไฟฟ้ากระแสตรง ค่ากระแสไฟฟ้าของไฟฟ้ากระแสตรง ค่าความต้านทานไฟฟ้าของตัวต้านทาน ค่าแรงดันไฟฟ้ากระแสสลับ ค่ากระแสไฟฟ้าของไฟฟ้ากระแสสลับ ค่าความถี่ไฟฟ้าและค่าตัวเก็บประจุ โดยทำการเปรียบเทียบค่าทางไฟฟ้าที่วัดได้กับค่าที่ได้รับจากบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0 พบว่าเครื่องมือวัดนี้สามารถแสดงผลเป็นเสียงพูดได้ถูกต้องตามข้อมูลที่ได้รับจากดิจิทัลมัลติมิเตอร์ และได้จัดการเรียนการสอน เพื่อประเมินความพึงพอใจในการใช้สื่อนวัตกรรมที่มีเครื่องมือวัดทางไฟฟ้าแสดงผลเป็นเสียงพูดเข้าไปร่วม โดยนำไปทดลองเรื่อง กฎของโอห์ม สำหรับนักเรียนระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป จำนวน 8 คน พบว่านักเรียนส่วนใหญ่มีความพึงพอใจต่อการทดลอง ในการทดลองเรื่อง กฎของโอห์มในทุก ๆ ด้านจากการประเมินโดยภาพรวม ค่าเฉลี่ยที่ได้เท่ากับ 4.74 อยู่ในระดับความพึงพอใจมากที่สุด

คำสำคัญ : เครื่องมือวัดแสดงผลด้วยเสียงพูด, มัลติมิเตอร์พูดได้

The Title : Design of an Electrically Verbal Measurement for Teaching Electricity Measurement

Author : Nited Pradedboon

Program : Master of Science Program in Science Teaching

Thesis Advisors : Dr. Chanade Wichasilp Chairman
: Assistant Professor Dr. Seri Pangsang Member

ABSTRACT

The objective of this research was to develop an electrically verbal measurement as an instructional medium for visually impaired students. The design and development of the electrical display instrument were based on the UNI-T brand digital multimeter system, UT61E model, in conjunction with the Raspberry Pi Model B+ board. The instrument was electrically tested for DC voltage, DC current resistance value of resistors, AC voltage, and AC current frequency and capacitor. The electrical values were then compared with the values derived from the Raspberry Pi board via USB 2.0 ports. It was found that this measuring instrument is able to display accurate speech results based on the data derived from the digital multimeter. An instructional session was organized to assess the satisfaction with this innovative medium that incorporated the electrically verbal measurement. The pedagogical implementation was conducted on the Ohm's Law with eight fourth-year physics and general science major students. It was revealed that the students were satisfied with all aspects of the experimental module. The overall mean of the assessment was 4.74, which was at the highest level of satisfaction

Keywords: Electrically Verbal Measurement, Talking Multimeter

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจาก อาจารย์ ดร.ชนนัญ วิชาศิลป์ อาจารย์ที่ปรึกษาหลัก ผู้ช่วยศาสตราจารย์ ดร.เสรี ปานซาง อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้กรุณาให้คำปรึกษา แนะนำและแก้ไขข้อบกพร่องต่าง ๆ มาด้วยตลอดจนสำเร็จเรียบร้อย ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณศาสตราจารย์ ดร.สมชาย ทองเต็ม ในการเป็นประธานสอบ แนะนำให้คำปรึกษา ข้อคิดเห็น ตรวจสอบและแก้ไขเค้าโครงวิทยานิพนธ์

ขอขอบพระคุณสาขาวิชาฟิสิกส์ มหาวิทยาลัยราชภัฏเชียงใหม่ ที่ให้ข้าพเจ้าได้รับการศึกษาในคณะวิทยาศาสตร์และเทคโนโลยีของมหาวิทยาลัยในระดับบัณฑิตศึกษาเพื่อเพิ่มพูนความรู้ในการปฏิบัติงาน และยังอนุเคราะห์สนับสนุนการวิจัยในเรื่องสถานที่ที่เอื้ออำนวยต่อการทำวิจัย และอุปกรณ์ที่ใช้ในการวิจัย และขอขอบพระคุณผู้ที่เกี่ยวข้องในด้านอื่น ๆ ที่ไม่ได้กล่าวนามที่มีส่วนช่วยเหลือให้งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี

ประโยชน์อันพึงได้จากการศึกษาในครั้งนี้ ขอให้เป็นกตเวทิตาแด่บิดา มารดาและครอบครัว ตลอดจนผู้เขียนหนังสือหรือบทความต่าง ๆ ที่ให้ความรู้แก่ผู้วิจัยจนสามารถทำให้งานวิจัยสำเร็จได้

นิเทศ ประเดชญญ

สารบัญ

	หน้า
บทคัดย่อ.....	ข
ABSTRACT.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญภาพ.....	ซ
บทที่	
1 บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	2
ประโยชน์ที่คาดว่าจะได้รับการจากการวิจัย.....	2
ขอบเขตของการวิจัย.....	3
ข้อตกลงเบื้องต้น.....	3
2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	4
พื้นฐานการใช้งานมัลติมีเตอร์เบื้องต้น.....	4
การวัดความต่างศักย์ไฟฟ้ากระแสตรง.....	6
การวัดความต่างศักย์ไฟฟ้ากระแสสลับ.....	7
หลักการและทฤษฎีเกี่ยวกับการสื่อสารแบบอนุกรม.....	8
ดิจิทัลมัลติมีเตอร์ระบบตัวเลข ตราสินค้า UNI-T.....	11
บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก ราสเบอร์รี่พาย (RASPBERRY PI).....	13
บอร์ด ET-USB/RS232 MINI.....	15
งานวิจัยที่เกี่ยวข้อง.....	18
3 วิธีดำเนินการวิจัย.....	21
รูปแบบการวิจัย.....	21
ขั้นตอนการดำเนินงาน.....	21
การใช้งานบอร์ดคอมพิวเตอร์ Raspberry Pi.....	27

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการวิเคราะห์ข้อมูล.....	57
ประชากรและกลุ่มตัวอย่าง.....	57
สถานที่ทดลอง.....	57
ความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้ สื่อนวัตกรรม.....	58
5 สรุป อภิปรายผล และข้อเสนอแนะ.....	62
สรุปผลการวิจัย.....	62
อภิปรายผล.....	62
ข้อเสนอแนะ.....	63
บรรณานุกรม.....	65
ประวัติผู้วิจัย.....	67
ภาคผนวก.....	68
ภาคผนวก ก คุณสมบัติและข้อจำกัดของมัลติมีเตอร์ ตราสินค้า UNI-T รุ่น UT61E.....	69
ภาคผนวก ข การติดตั้งระบบปฏิบัติการ Linux และการตั้งค่าพื้นฐาน.....	72
ภาคผนวก ค โปรแกรมภาษา Python สำหรับอ่านค่าจาก มัลติมีเตอร์ ตราสินค้า UNI-T รุ่น UT61E.....	80
ภาคผนวก ง แบบทดลอง เรื่อง กฎของโอห์ม.....	102

สารบัญตาราง

ตารางที่		หน้า
2.1	รายการชื่อสายสัญญาณของพอร์ตอนุกรม RS-232.....	10
3.1	การตั้งค่าการสื่อสารผ่านพอร์ตอนุกรม.....	23
3.2	โหมดของการวัด.....	24
3.3	ชุดข้อมูล Range ต่าง ๆ.....	25
3.4	รหัสแทนตัวเลข 0-9.....	25
3.5	ชุดข้อมูล STATUS	26
3.6	ชุดข้อมูล OPTION 1.....	26
3.7	ชุดข้อมูล OPTION 2.....	14
3.8	ชุดข้อมูล OPTION 3.....	27
3.9	ชุดข้อมูล OPTION 4.....	27
3.10	ความสัมพันธ์ของไฟล์เสียงและตำแหน่งจัดเก็บ.....	33
3.11	ความสัมพันธ์ของผลการวัดแรงดันไฟฟ้ากระแสตรง.....	40
3.12	ความสัมพันธ์ของผลการวัดความต้านทานไฟฟ้า.....	42
3.13	ความสัมพันธ์ของผลการวัดตัวเก็บประจุ.....	45
3.14	ความสัมพันธ์ของผลการวัดความถี่ไฟฟ้า.....	30
3.15	ความสัมพันธ์ของผลการวัดกระแสไฟฟ้ากระแสตรง.....	50

สารบัญภาพ

ภาพที่	หน้า
2.1 การใช้มัลติมิเตอร์วัดกระแสไฟฟ้า.....	5
2.2 การใช้มัลติมิเตอร์วัดความต่างศักย์ไฟฟ้ากระแสตรง.....	6
2.3 การใช้มัลติมิเตอร์วัดความต่างศักย์ไฟฟ้ากระแสสลับ.....	7
2.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส.....	8
2.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส.....	9
2.6 การจัดขาพอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25.....	10
2.7 มัลติมิเตอร์แบบตัวเลข.....	12
2.8 วงจรสายสัญญาณเชื่อมต่อมัลติมิเตอร์ผ่านพอร์ตอนุกรม.....	13
2.9 ตำแหน่งอุปกรณ์ของบอร์ด Raspberry Pi Model B+.....	14
2.10 บอร์ด ET-USB/RS232 MINI.....	15
2.11 ตำแหน่งขาสัญญาณของบอร์ด ET-USB/RS232 MINI.....	16
3.1 แนวการออกแบบมัลติมิเตอร์แสดงผลด้วยเสียง.....	22
3.2 ชุดข้อมูล 10 บิต.....	23
3.3 ชุดข้อมูล 140 บิต.....	23
3.4 ตำแหน่งอุปกรณ์บนบอร์ด Raspberry Pi.....	28
3.5 เว็บไซต์ GOOGLE TRANSLATE.....	30
3.6 เครื่องมือนักพัฒนา.....	31
3.7 เปิดหน้าต่างใหม่เพื่อบันทึกไฟล์ MP3.....	32
3.8 การบันทึกไฟล์ MP3.....	32
3.9 ขั้นตอนการทำงานของโปรแกรมส่วน A.....	36
3.10 ขั้นตอนการทำงานของโปรแกรมส่วน B.....	37
3.11 การต่ออุปกรณ์วัดแรงดันไฟฟ้ากระแสตรง.....	41
3.12 ผลค่าแรงดันไฟฟ้าและการเรียงลำดับไฟล์เสียงที่แรงดันทดสอบ 5V.....	41

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.13 กราฟเปรียบเทียบระหว่างค่าแรงดันไฟฟ้าจากหน้าจอมัลติมิเตอร์กับ ค่าแรงดันไฟฟ้าเฉลี่ยจากพอร์ต USB.....	42
3.14 การวัดความต้านทานขนาด $0.5M\Omega$	43
3.15 การแสดงผลค่าความต้านทานขนาด $0.5M\Omega$ และการเรียงลำดับไฟล์เสียง.....	44
3.16 กราฟเปรียบเทียบระหว่างค่าความต้านทานจากหน้าจอมัลติมิเตอร์กับค่าความ ต้านทานเฉลี่ยจากพอร์ต USB.....	44
3.17 การวัดตัวเก็บประจุขนาด 2200UF.....	46
3.18 การแสดงผลค่าตัวเก็บประจุขนาด 2200uF และการเรียงลำดับไฟล์เสียง.....	46
3.19 กราฟเปรียบเทียบระหว่างค่าตัวเก็บประจุจากหน้าจอมัลติมิเตอร์กับค่าตัวเก็บ ประจุเฉลี่ยจากพอร์ต USB.....	47
3.20 การทดสอบวัดความถี่ไฟฟ้า.....	48
3.21 การแสดงผลค่าวัดความถี่ 20MHz และการเรียงลำดับไฟล์เสียง.....	49
3.22 กราฟเปรียบเทียบระหว่างค่าความถี่ไฟฟ้าจากหน้าจอมัลติมิเตอร์กับค่าความถี่ ไฟฟ้าเฉลี่ยจากพอร์ต USB.....	49
3.23 การวัดกระแสไฟฟ้ากระแสตรง.....	51
3.24 การแสดงผลค่าการวัดกระแสไฟฟ้ากระแสตรง 30mA และการเรียงลำดับไฟล์ เสียง.....	52
3.25 กราฟเปรียบเทียบระหว่างค่ากระแสไฟฟ้ากระแสตรงจากหน้าจอมัลติมิเตอร์กับ ค่ากระแสไฟฟ้ากระแสตรงเฉลี่ยจากพอร์ต USB.....	52
3.26 เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด.....	53
3.27 เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด.....	54
3.28 ชุดทดลองกฎของโอห์ม.....	55
3.29 ใช้งานเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด.....	55
4.1 การจัดกิจกรรมการสอนโดยใช้สื่อนวัตกรรม.....	58

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

การศึกษาเป็นสิ่งสำคัญสำหรับคนในสังคม ทำให้คนในสังคมดำเนินชีวิตได้อย่างมีเป้าหมาย และสามารถอยู่ร่วมกันได้อย่างมีความสุข ดังนั้นทุกคนจำเป็นต้องได้รับการศึกษา ปัจจุบันสำคัญประการหนึ่งในการจัดการเรียนการสอนให้มีประสิทธิภาพคือ สื่อการเรียนการสอน สื่อการเรียนการสอนมีส่วนช่วยทำให้ประสิทธิภาพของการเรียนการสอนเพิ่มขึ้น โดยเฉพาะกลุ่มผู้เรียนอีกกลุ่มหนึ่งที่มีความบกพร่องทางร่างกายคือ กลุ่มผู้พิการทางสายตา ผู้เรียนกลุ่มนี้มีอุปสรรคอย่างมากในการเรียน เนื่องจากคำরাส่วนใหญ่ถูกพิมพ์อยู่ในรูปแบบสำหรับคนตาดีเท่านั้น ซึ่งความบกพร่องทางร่างกายไม่ได้หมายถึงการด้อยสติปัญญาหรือความสามารถ บุคคลที่มีความบกพร่องทางร่างกายนั้นก็นับว่าเป็นทรัพยากรบุคคลที่มีค่าของประเทศชาติไม่ยิ่งหย่อนไปกว่าคนสมบูรณ์

ในการสอนวิทยาศาสตร์มีเนื้อหาการเรียนการสอนเกี่ยวกับไฟฟ้าเช่น แรงดันไฟฟ้า กระแสไฟฟ้า ความต้านทาน ความเหนี่ยวนำ ความจุไฟฟ้า ซึ่งต้องมีการวัดแสดงผลเพื่อศึกษาความสัมพันธ์ของเนื้อหาดังกล่าว ซึ่งไม่ได้มีปัญหาสำหรับผู้เรียนที่มีความปกติ แต่กลับมีปัญหามากสำหรับผู้เรียนที่มีความบกพร่องทางสายตา ในระดับที่ไม่สามารถดำเนินการจัดการเรียนการสอน หรือการเรียนรู้ได้อย่างมีประสิทธิภาพ

ดังนั้นผู้วิจัยจึงได้สร้างนวัตกรรมที่เป็นเครื่องมือวัดทางไฟฟ้าเพื่อแก้ปัญหาเรื่องการวัดทางไฟฟ้าให้กับผู้พิการทางสายตา โดยดำเนินการออกแบบและสร้างอุปกรณ์แสดงค่าทางไฟฟ้าที่วัดได้จากมัลติมิเตอร์ด้วยเสียงพูด โดยใช้ข้อมูลการวัดที่ส่งออกจากดิจิตอลมัลติมิเตอร์มายังอุปกรณ์เพื่อแปลค่าข้อมูลทางไฟฟ้าเป็นเสียงพูด ซึ่งจะทำให้ผู้เรียนที่มีความบกพร่องทางสายตาทราบค่าทางไฟฟ้าและสามารถเชื่อมโยงความสัมพันธ์ระหว่างค่าทางไฟฟ้าได้

งานวิจัยนี้จึงได้ออกแบบสร้าง และพัฒนาสื่อการสอนให้เหมาะกับผู้เรียนที่มีความบกพร่องทางสายตาโดยเฉพาะ เพื่อสามารถนำมาใช้เป็นสื่อในการจัดการเรียนการสอนวิชาวิทยาศาสตร์ที่เกี่ยวกับการวัดค่าทางไฟฟ้าได้ และส่งเสริมให้เกิดความเข้าใจในเรื่องไฟฟ้าได้มากยิ่งขึ้น โดยงานวิจัยนี้จะเป็นการพัฒนามัลติมิเตอร์ที่แสดงผลแบบตัวเลขที่ใช้วัดค่าทางไฟฟ้าทั่วไป

ให้สามารถบอกค่าที่วัดได้แสดงผลเป็นเสียงพูดบอกปริมาณที่วัดได้อย่างถูกต้อง ซึ่งงานวิจัยนี้ถือได้ว่าเป็นนวัตกรรมทางการศึกษาชิ้นหนึ่ง สุคนธ์ สินธพานนท์ (2553) ให้ความหมายว่า นวัตกรรมทางการศึกษา หมายถึง สิ่งใหม่ ๆ ที่สร้างขึ้นมาเพื่อช่วยแก้ปัญหาเกี่ยวกับการจัดการเรียนการสอน หรือพัฒนาให้ผู้เรียนเกิดการเรียนรู้อย่างมีประสิทธิภาพ ได้แก่แนวคิด รูปแบบ วิธีการ กระบวนการ สื่อต่าง ๆ ที่เกี่ยวกับการศึกษา สำหรับงานวิจัยสำหรับคนพิการทางสายตา เช่น ภักธิมา เทียงตรง, (2553) หาประสิทธิผลในการใช้โปรแกรมคอมพิวเตอร์ฝึกพิมพ์ดีดสัมผัสสำหรับคนพิการ มีวัตถุประสงค์เพื่อเปรียบเทียบทักษะในการพิมพ์ดีดสัมผัสภาษาไทยของนักเรียนตาบอดที่ใช้วิธีการฝึกด้วยโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการทางตากับการใช้วิธีการฝึกด้วยวิธีปกติและศึกษาความพึงพอใจของนักเรียนตาบอดที่มีต่อโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการทางสายตาได้ใช้ประโยชน์ตาบอดที่ใช้วิธีการฝึกด้วยโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการทางตามีทักษะในการพิมพ์ดีดสัมผัสภาษาไทยสูงกว่านักเรียนตาบอดที่ใช้วิธีการฝึกด้วยวิธีปกติ และความพึงพอใจอยู่ในระดับมาก

วัตถุประสงค์ของการวิจัย

1. พัฒนาเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด
2. ทดสอบเปรียบเทียบค่าทางไฟฟ้าที่แสดงผลทางหน้าจอกับค่าที่ได้รับจากมัลติมิเตอร์ตราสินค้า UNI-T รุ่น UT61E ผ่านทางพอร์ต USB ของบอร์ดเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดที่สร้างขึ้น
3. เพื่อประเมินความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อนวัตกรรมในการทดลองเรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป

ประโยชน์ที่คาดว่าจะได้รับการวิจัย

1. ได้เครื่องมือวัดทางไฟฟ้าที่สามารถแสดงผลด้วยเสียงพูด
2. สามารถนำเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงไปใช้เป็นอุปกรณ์ในการประกอบการเรียนการสอนให้กับนักเรียนที่ปกติและนักเรียนที่มีความบกพร่องทางการเห็นสามารถนำมาใช้ในการสอนวิทยาศาสตร์ เรื่องไฟฟ้าเบื้องต้น

ขอบเขตของการวิจัย

1. ออกแบบและพัฒนาเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด โดยใช้มัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T รุ่น UT61E ร่วมกับบอร์ด Raspberry Pi Model B+
2. ทดสอบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด เพื่อนำไปใช้กับนักเรียนที่มีความบกพร่องทางการมองเห็นและนักเรียนปกติทั่วไป สามารถนำไปใช้ประกอบการเรียนการสอนด้านวิทยาศาสตร์ ที่มีเนื้อหาเกี่ยวข้องกับเรื่อง ไฟฟ้าเบื้องต้น

ข้อตกลงเบื้องต้น

การใช้งานเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดควรอยู่ภายใต้ความควบคุมดูแลของครูผู้สอนอย่างใกล้ชิด และควรทำความเข้าใจหลักการวัดค่าทางไฟฟ้าเบื้องต้นก่อน

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การออกแบบเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดเพื่อเป็นอุปกรณ์การสอนเกี่ยวกับการวัดค่าทางไฟฟ้าจำเป็นต้องศึกษาทฤษฎีดังต่อไปนี้

1. พื้นฐานการใช้งานมัลติมิเตอร์เบื้องต้น
2. การวัดกระแสไฟฟ้ากระแสตรง
3. การวัดความต่างศักย์ไฟฟ้ากระแสตรง
4. การวัดความต่างศักย์ไฟฟ้ากระแสสลับ
5. หลักการและทฤษฎีเกี่ยวกับการสื่อสารแบบอนุกรม
6. ดิจิตอลมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T
7. บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก ราสเบอร์รี่พาย (Raspberry Pi)
8. บอร์ด ET-USB/RS232 MINI
9. ภาษาไพทอน (Python)

พื้นฐานการใช้งานมัลติมิเตอร์เบื้องต้น

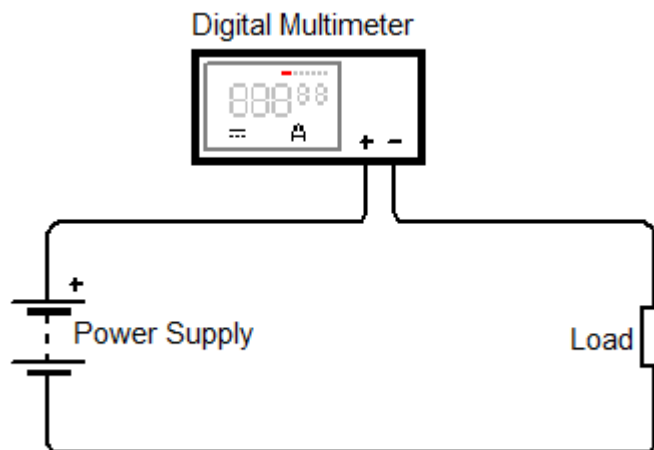
หลักการวัดปริมาณไฟฟ้าเบื้องต้นโดยใช้ดิจิตอลมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T รุ่น UT61E เป็นตัวอ้างอิงในการปรับตั้งการวัดปริมาณไฟฟ้า ในการใช้งานมัลติมิเตอร์แต่ละรุ่นต้องศึกษาคู่มือการใช้งานเบื้องต้นก่อนเสมอ ซึ่งคุณสมบัติของตัวดิจิตอลมัลติมิเตอร์แต่ละรุ่นจะมีช่วงของการวัดที่ต่างกันและมีความสามารถในการวัดปริมาณทางไฟฟ้าเพิ่มเติมอื่น ๆ เข้ามาแตกต่างกันออกไปด้วย เช่น วัดความถี่ไฟฟ้า วัดอุณหภูมิ เป็นต้น

การวัดกระแสไฟฟ้ากระแสตรง

ในการนำมัลติมิเตอร์ไปใช้ในการวัดกระแสไฟฟ้า ต้องปรับมัลติมิเตอร์ให้เป็นแอมมิเตอร์ก่อน โดยหมุนสวิตช์บนตัวมิเตอร์ ไปที่ตำแหน่งช่วงการวัดกระแสไฟฟ้ากระแสตรง (DCA) โดยอ้างอิงจากมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T รุ่น UT61E ซึ่งมี 4 ช่วงการวัดคือ

0-220 μ A , 0-2200 μ A , 0-22mA , 0-220mA ดังนั้นเราต้องใช้มัลติมิเตอร์วัดกระแสไฟฟ้าเหมือนกับหลักการวัดของแอมมิเตอร์ทุกขั้นตอน หลักการที่นำมัลติมิเตอร์ไปใช้ในการวัดกระแสไฟฟ้างี้

1. เลือกตำแหน่งที่ต้องการวัดกระแสไฟฟ้าและตรวจสอบทิศทางกระแสไฟฟ้า
2. เสียบสายวัดมิเตอร์สีดำที่ขั้วลบ (- COM) และสายวัดสีแดงที่ขั้วบวก (+) เข้ากับมัลติมิเตอร์
3. ตั้งช่วงการวัดที่เหมาะสม ในกรณีที่ทราบค่ากระแสในวงจรควรตั้งช่วงการวัดให้สูงกว่าค่ากระแสที่ทราบ แต่ในกรณีที่ไม่ทราบค่ากระแสในวงจร ควรตั้งช่วงการวัดที่สูง ๆ ไว้ก่อน แล้วค่อยปรับช่วงการวัดใหม่ ก่อนปรับช่วงการวัดใหม่ต้องเอาสายวัดออกจากวงจรทุกครั้งและต้องแน่ใจว่าค่าที่จะวัดได้นั้นมีค่าไม่เกินช่วงการวัดที่ปรับตั้งใหม่
4. นำสายโพรบของมัลติมิเตอร์ไปต่อแทรกหรือต่อแบบอนุกรม โดยใช้หัวโพรบวัดแต่ละบริเวณที่ต้องการวัด และต้องให้กระแสไฟฟ้าไหลเข้าทางขั้วบวกของมัลติมิเตอร์ หากมัลติมิเตอร์แสดงสัญลักษณ์ OV ให้รีบเอาสายวัดมิเตอร์ออกจากวงจรทันที แล้วเลือกช่วงการวัดที่สูงขึ้นจากนั้นทำการวัดค่าใหม่



ภาพที่ 2.1 การใช้มัลติมิเตอร์วัดกระแสไฟฟ้า

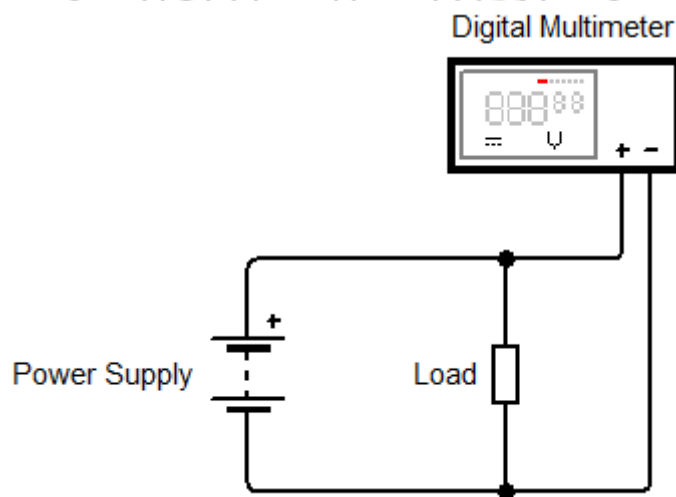
ที่มา: นิเทศ ประเดชนบุญ

การวัดค่ากระแสไฟฟ้าที่ไหลผ่านโหลด โดยการนำมัลติมิเตอร์ต่ออนุกรมกับโหลดที่ต้องการวัดค่ากระแสไฟฟ้า ดังแสดงในภาพที่ 2.1

การวัดความต่างศักย์ไฟฟ้ากระแสตรง

ในการนำมัลติมิเตอร์ไปใช้ในการวัดความต่างศักย์ ต้องปรับมัลติมิเตอร์ให้เป็นโวลท์มิเตอร์ก่อน โดยหมุนสวิตช์บนตัวมัลติมิเตอร์ ไปที่ตำแหน่งช่วงการวัดความต่างศักย์ไฟฟ้ากระแสตรง (DCV) โดยอ้างอิงจากมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T รุ่น UT61E ซึ่งมี 7 ช่วงการวัด คือ 0-220mV, 0-2.2V, 0-22V, 0-220V, 0-1000V หลักการนำมัลติมิเตอร์ไปใช้ในการวัดความต่างศักย์

1. เลือกตำแหน่งที่ต้องการวัดความต่างศักย์ และตรวจสอบทิศทางกระแสไหลของกระแสไฟฟ้า
2. เสียบสายวัดมิเตอร์สีดำที่ขั้วลบ (- COM) และสายวัดสีแดงที่ขั้วบวก (+) เข้ากับมัลติมิเตอร์
3. ตั้งช่วงการวัดให้สูงกว่าความต่างศักย์ของบริเวณนั้น โดยหมุนสวิตช์บนตัวมัลติมิเตอร์ ไปที่ตำแหน่งช่วงการวัดความต่างศักย์ไฟฟ้ากระแสตรง (DCV)
4. นำสายวัดมิเตอร์ไปต่อขนานหรือต่อคร่อมวงจร โดยใช้หัววัดแตะกับจุดที่ต้องการวัด และต้องให้กระแสไฟฟ้าไหลเข้าทางขั้วบวก (+) ของมัลติมิเตอร์เสมอ ถ้าวัดสลับขั้วตัวเลขของดิจิตอลมัลติมิเตอร์จะแสดงค่าติดลบ ให้เอาสายวัดมิเตอร์ออกจากวงจรทันที จากนั้นทำการสลับหัวโพรบวัดให้ถูกต้อง



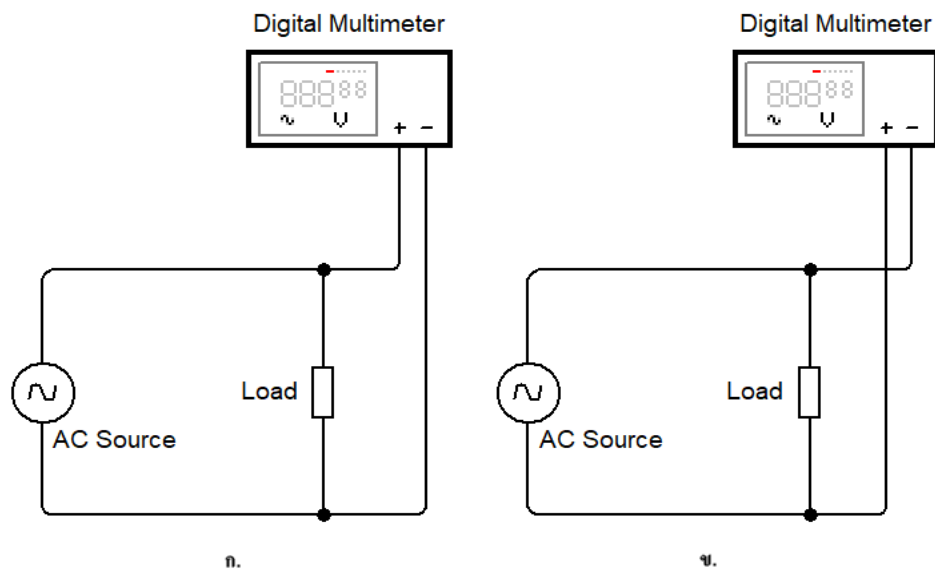
ภาพที่ 2.2 การใช้มัลติมิเตอร์วัดความต่างศักย์ไฟฟ้ากระแสตรง
ที่มา: นิเทศ ประเดษบุญ

5. การวัดค่าความต่างศักย์ไฟฟ้ากระแสตรงที่ตกคร่อมโหลด โดยการนำมัลติมิเตอร์ต่อคร่อมกับโหลดตัวที่ต้องการวัดค่า ดังแสดงในภาพที่ 2.2

การวัดความต่างศักย์ไฟฟ้ากระแสสลับ

การวัดความต่างศักย์ไฟฟ้ากระแสสลับ ไม่จำเป็นต้องให้กระแสไฟฟ้าไหลผ่านทางขั้วบวกเหมือนไฟฟ้ากระแสตรง เพราะไฟฟ้ากระแสสลับไม่มีขั้วตายตัว ขั้วแรงดันจะสลับไปสลับมาตลอดเวลา กล่าวคือสามารถต่อโดยให้สายวัดเส้นใดอยู่ข้างใดก็ได้ แต่วิธีวัดค่ายังใช้หลักการเดียวกันกับโวลต์มิเตอร์กระแสตรง ดังแสดงดังภาพที่ 2.3 ก. และ ข.

ก่อนที่จะนำมัลติมิเตอร์ไปวัดค่า ต้องทำการปรับมัลติมิเตอร์ให้เป็นโวลท์มิเตอร์กระแสสลับก่อน จากนั้นเลือกช่วงการวัดให้เหมาะสมโดยหมุนสวิตช์บนตัวมัลติมิเตอร์ไปที่ตำแหน่งช่วงการวัดความต่างศักย์ไฟฟ้ากระแสตรง (ACV) โดยอ้างอิงจากมัลติมิเตอร์ระบบตัวเลขตราสินค้า UNI-T รุ่น UT61E ซึ่งมี 4 ช่วงการวัด คือ 0-220mV, 0-2.2V, 0-22V, 0-220V, 0-750V

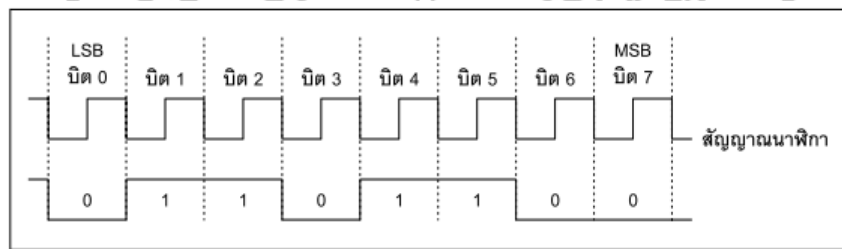


ภาพที่ 2.3 การใช้มัลติมิเตอร์วัดความต่างศักย์ไฟฟ้ากระแสสลับ
ที่มา: นิเทศ ประเดชนบุญ

การวัดความต่างศักย์ไฟฟ้ากระแสสลับ สามารถใช้สายวัดเส้นใดอยู่ข้างใดก็ได้ของตัวโหลด ดังแสดงในภาพที่ 2.3

หลักการและทฤษฎีเกี่ยวกับการสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับส่งสัญญาณด้วย ภาพที่ 9 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



ภาพที่ 2.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส
ที่มา: นิเทศ ประเดชญญ

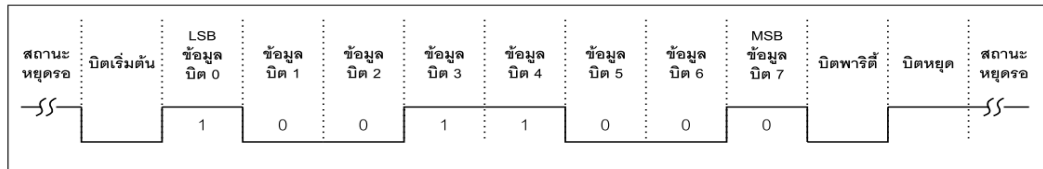
การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกา ร่วมด้วยเหมือนการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาที่ภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baud Rate) มีหน่วยเป็น บิตต่อวินาที (Bit Per Second : BPS) รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

ภาพที่ 2.5 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขาดาค้า (Data) จะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาดาค้า (Data) มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น (Start Bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนส่งข้อมูลในไบต์ที่ส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นตามด้วย บิตพาริตี (Parity Bit) ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตปิดท้าย ซึ่งจะ

ให้ขาด้ามีสถานะลอจิก “1” อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต , 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



ภาพที่ 2.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

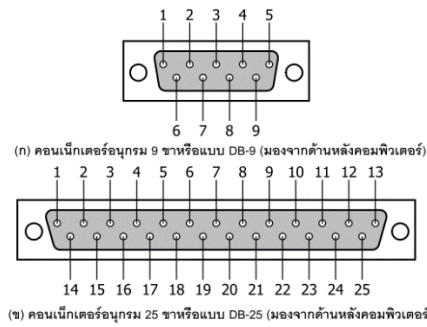
ที่มา: นิเทศ ประเดษบุญ

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส เรียกว่า (Universal Asynchronous Receiver/Transmitter : UART) มีอัตราความเร็วในการรับและส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์และอุปกรณ์ .

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (Odd) และแบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก “1” ภายในข้อมูลที่ส่งไป ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่ โดยต้องรวมบิตพาริตีเข้าไปด้วย บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART โดยภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตีให้ตรงกันว่าจะตรวจสอบพาริตีคี่หรือคู่ จากนั้นภาครับของ UART จะตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือคี่ เป็นการตรวจสอบความผิดพลาดของการถ่ายทอดข้อมูลที่ง่ายที่สุด

มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 มีระดับสัญญาณตั้งแต่ -3V ถึง -12V แสดงว่ามีข้อมูล (Mark) และ +3V ถึง +12V แสดงว่าเป็นช่องว่าง (Space) โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือแบบ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในภาพที่ 2.6



ภาพที่ 2.6 การจัดขาพอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25
ที่มา: นิเทศ ประเดษบุญ

ตารางที่ 2.1 รายการชื่อสายสัญญาณของพอร์ตอนุกรม RS-232

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

รายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังต่อไปนี้

1. Data Carrier Detect (DCD) หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม
2. Receive Data (Rx)D ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

3. Transmitted Data (TxD) ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

4. Data Terminal Ready (DTR) เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์

5. Signal Ground (GND) กราวด์ระบบทำหน้าที่เป็นแรงดันอ้างอิงของทุกๆ สัญญาณ

6. Data Set Ready (DSR) ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

7. Request To Send (RTS) เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณจาก RTS คือ CTS

8. Clear To Send (CTS) ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

9. Ring Indicator (RI) ใช้แสดงสถานะสัญญาณเรียกสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน

ดิจิตอลมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T

ดิจิตอลมัลติมิเตอร์สามารถวัดค่าปริมาณไฟฟ้าได้หลายประเภท เช่น วัดแรงดันไฟฟ้า กระแสตรง (DCV) แรงดันไฟกระแสสลับ (ACV) กระแส ไฟตรง (DCA) กระแสไฟสลับ (ACV) และความต้านทาน (Ω) เป็นต้น นอกจากนี้ในดิจิตอลมัลติมิเตอร์ บางรุ่นยังมีความสามารถเพิ่มมากขึ้นไปอีก สามารถวัดค่าปริมาณไฟฟ้าอื่น ๆ นอกเหนือจากค่าปกติได้ เช่น วัดอุณหภูมิ วัดความถี่ วัดค่าความจุของตัวเก็บประจุ วัดอัตราขยายของทรานซิสเตอร์ (hFE) และวัดขาทรานซิสเตอร์ได้ เป็นต้น ดิจิตอลมัลติมิเตอร์แต่ละรุ่นแต่ละแบบและแต่ละบริษัท มีส่วนประกอบ โครงสร้างและรายละเอียดต่าง ๆ ของเครื่องแตกต่างกันแต่การใช้งาน การวัดค่า การอ่านค่า มีหลักการที่คล้ายกัน โดยอ้างอิงรายละเอียดจากดิจิตอลมัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E ดังแสดงในภาพที่ 2.7 และอ้างอิงถึงของเขตคุณสมบัติ ข้อจำกัดของตัวเครื่องมัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E (ภาคผนวก ข)



ภาพที่ 2.7 มัลติมิเตอร์แบบตัวเลข

ที่มา: นิเทศ ประเดชนบุญ

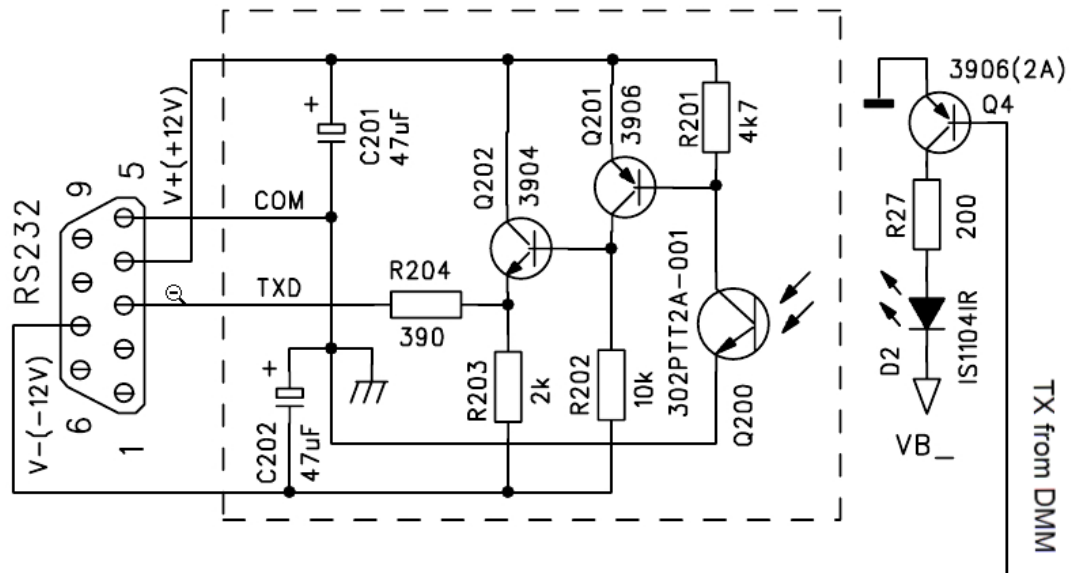
ลักษณะของเครื่องจะประกอบไปด้วยหน้าจอแสดงผล ที่แสดงด้วยตัวเลขแบบดิจิทัล โดยมีตัวเลขสูงสุด 4 หลัก ในการวัดปริมาณทางไฟฟ้าบางชนิด เช่น ความต่างศักย์ไฟฟ้ากระแสตรง ด้วยเครื่องวัดที่ใช้เข็มชี้เป็นตัวแสดงผล เมื่อต่อสายวัดผิดขั้ว เข็มของเครื่องวัดจะตีกลับในทิศตรงข้าม ในสภาวะเช่นนี้สำหรับดิจิทัลมัลติมิเตอร์แบบตัวเลขจะปรากฏเครื่องหมาย – บนจอแสดงผล ในการวัดปริมาณใด ๆ ที่ตั้งช่วงการวัดต่ำกว่าค่าที่จะวัด จอแสดงผลจะแสดงตัวอักษร OL และจะแสดงตัวอักษร UL เมื่อปริมาณที่วัดมีค่าต่ำกว่าค่าที่ตัวเครื่องจะสามารถอ่านค่าได้ นอกจากนี้เมื่อแหล่งจ่ายพลังงานให้เครื่องวัด คือ แบตเตอรี่ 9V อ่อนกำลัง ตัวอักษร LO BAT จะปรากฏบนจอแสดงถึงแบตเตอรี่พลังงานต่ำ เพื่อเตือนให้ผู้ใช้งานเปลี่ยนแบตเตอรี่ใหม่ ข้อควรระวังและการเตรียมสำหรับการวัดก่อนการวัดปริมาณใด ต้องแน่ใจว่า

1. บิดสวิตช์เลือกการวัดตรงกับปริมาณที่จะวัด

2. สวิตช์เลือกการวัดอยู่ในช่วงการวัดที่เหมาะสมไม่ต่ำกว่าปริมาณที่จะวัดในกรณีที่ไม่ทราบปริมาณที่จะวัดมีค่าอยู่ในช่วงการวัดใด ให้ตั้งช่วงการวัดที่มีค่าสูงสุดก่อนแล้วค่อยลดช่วงการวัดลงมาทีละช่วงและเนื่องจากช่องเสียบสายวัด (สีแดง) มีหลายช่อง คือ V, mA และ 10 A ต้องแน่ใจว่าเสียบสายวัดสีแดงในช่องเสียบตรงกับปริมาณที่จะวัด ในกรณีที่วัดความต่างศักย์ไฟฟ้าสูงตั้งแต่ 25 VAC หรือ 60 VDC ขึ้นไป ระวังอย่าให้ส่วนใดของร่างกายแตะวงจรที่กำลังวัดจะเป็นอันตรายได้ เมื่อใช้งานเสร็จแล้ว ให้เลื่อนสวิตช์ปิด-เปิด มาที่ OFF ถ้าไม่ได้ใช้เป็นเวลานาน ควรเอาแบตเตอรี่ออกด้วย สำหรับการวัดปริมาณกระแสสูง (~10A) ควรใช้เวลาวัดในช่วงสั้นไม่เกิน 30 วินาที

1. ยกสายวัดเส้นหนึ่งออกจากวงจรที่กำลังทดสอบ
2. ปรับช่วงการวัดหรือเลือกการวัดปริมาณอื่นตามต้องการ
3. ทำการวัด

การติดต่อสื่อสารกับมัลติมิเตอร์ UNI-T รุ่น UT61E



ภาพที่ 2.8 วงจรสายสัญญาณเชื่อมต่อมัลติมิเตอร์ผ่านพอร์ตอนุกรม
ที่ 1: นิเทศ ประเดชนบล

การติดต่อสื่อสารกับมัลติมิเตอร์ UNI-T รุ่น UT61E จะต้องทำการติดต่อผ่านทางพอร์ตอนุกรมโดยตัวมัลติมิเตอร์จะมีสายสัญญาณที่มีการแยกวงจรด้วยการใช้แสงจากเซนเซอร์ แบบ IR และมีการต่อร่วมกับอุปกรณ์อิเล็กทรอนิกส์ ดังแสดงในภาพที่ 2.8 ตัวบอร์ดควบคุมหลักของมัลติมิเตอร์จะทำการส่งสัญญาณข้อมูลออกมาทางขาส่งสัญญาณ (Tx from DMM) เพื่อส่งข้อมูลให้กับอุปกรณ์ภายนอกที่ได้ทำการเชื่อมต่อผ่านทางขั้วต่อคอนเน็คเตอร์ แบบ DB-9 ไร้ตลอดเวลา

บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก ราสเบอร์รี่พาย (Raspberry Pi)

บอร์ดราสเบอร์รี่พาย (Raspberry Pi) เป็นบอร์ดคอมพิวเตอร์ 32 บิต ขนาดเล็กขนาดเท่ากับบัตรเครดิต รองรับระบบปฏิบัติการลินุกซ์ (Linux) ที่บรรจุลงใน SD การ์ดถูกพัฒนาโดย

Raspberry Pi Foundation พร้อมจุดเชื่อมต่ออุปกรณ์ อินพุตเอาต์พุตทั้งผ่านพอร์ต USB, LAN, ช่องสัญญาณภาพ HDMI และ GPIO ตัวของบอร์ดราสเบอร์รี่พายถูกออกแบบมาเพื่อใช้เป็นสื่อการเรียนการสอน การเขียนโปรแกรมคอมพิวเตอร์สำหรับเด็ก ๆ และเป็นบอร์ดที่ราคาไม่แพงมาก ใช้งานง่ายเหมาะสำหรับผู้เริ่มต้น ทำให้เป็นที่นิยมแพร่หลายสู่บุคคลทั่วไปอย่างรวดเร็ว บอร์ด Raspberry Pi มีลักษณะและส่วนประกอบ ดังแสดงในภาพที่ 2.9



ภาพที่ 2.9 ตำแหน่งอุปกรณ์ของบอร์ด Raspberry Pi Model B+

ที่มา: นิเทศ ประเดชนูญ

คุณสมบัติของบอร์ด Raspberry Pi

1. ใช้ชิพ SoC Broadcom BCM2835 ซึ่งรวม CPU, GPU และ SDRAM ไว้ในตัวถึง
เดียวกัน
2. หน่วยประมวลผลกลาง (CPU) 700 MHz ARM11 ARM1176JZF-S core
3. หน่วยประมวลผลภาพ (GPU) Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG
1080p30 H.264
4. หน่วยความจำ SDRAM 1 GB
5. ขั้วต่อ USB 2.0 จำนวน 4 พอร์ต
6. ขั้วต่อสัญญาณภาพแบบ HDMI
7. ขั้วต่อสัญญาณเสียงโดยใช้แจ็ค 3.5 mm หรือ ผ่านทางขั้ว HDMI
8. คอนเน็คเตอร์สำหรับเชื่อมต่ออินพุตเอาต์พุต (GPIO) ,SPI , I²C , I²S และ UART
(GPIO ต่าง ๆ ไม่สามารถรับแรงดันอินพุต 5 VDC ได้สูงสุดแค่ 3.3 VDC เท่านั้น)
9. ระบบปฏิบัติการของบอร์ดจะทำงานผ่าน MICRO SD CARD

10. ขั้วต่อ LAN 10/100 Mbps
11. ใช้ไฟเลี้ยงบอร์ด 5 VDC กระแสอย่างน้อย 700 mA
12. ขนาดของบอร์ด 85.0 x 56.0 mm

บอร์ด ET-USB/RS232 MINI

เนื่องจากการติดต่อสื่อสารระหว่างดิจิทัลมัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E จะต้องใช้การติดต่อสื่อสารแบบอนุกรมและใช้สายสัญญาณแบบ DB-9 ร่วมกับบอร์ด ET-USB/RS232 MINI เพื่อทำการแปลงสัญญาณที่ได้รับให้อยู่ในรูปแบบของสัญญาณ USB ดังภาพที่ 2.10 ซึ่งจะถูกต้องใช้งานผ่านทางพอร์ต USB 2.0 ของตัวบอร์ด Raspberry Pi โดยคุณสมบัติของบอร์ด ET-USB/RS232 MINI มีดังนี้



ภาพที่ 2.10 บอร์ด ET-USB/RS232 MINI

ที่มา: นิเทศ ประเดชนูญ

คุณสมบัติของ ET-USB/RS232 MINI

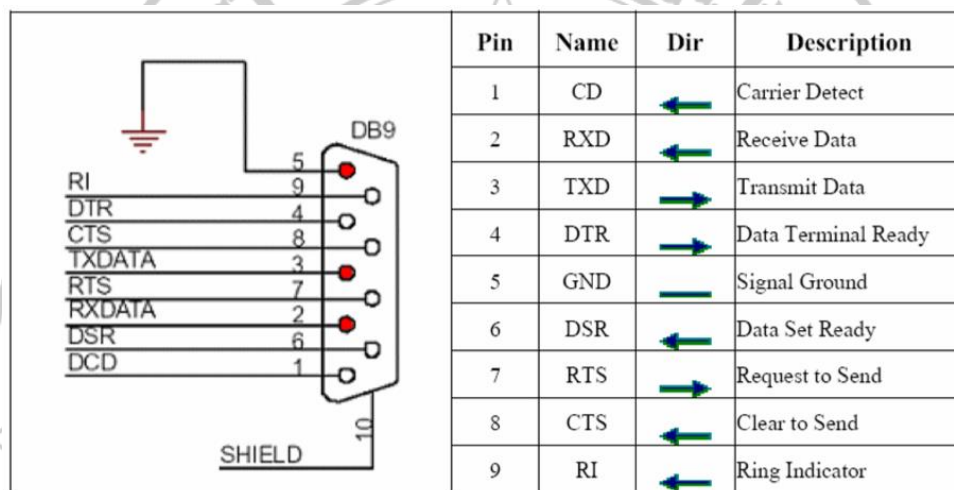
1. รองรับมาตรฐาน USB 1.1 และ USB 2.0
2. อัตราการรับส่งข้อมูล 300 bps ถึง 128 kbps
3. ใช้ไฟเลี้ยงจาก USB Port โดยตรงไม่ต้องต่อเพิ่มภายนอก
4. USB Connector แบบ TYPE A, RS232 Connector แบบ DP9 ตัวผู้
5. แสดงสถานะการทำงานด้วย LED 3 สี คือ การรับ (RX) สีเขียว, การส่ง (TX) สี

เหลือง และ Power (PWR) สีแดง

6. Driver รองรับ Windows / Linux
7. ขนาด กว้าง 3 cm x ยาว 6.8 cm

การต่อใช้งานโมดูล ET-USB/RS232 MINI

1. เสียบ ET-USB/RS232 MINI เข้ากับพอร์ต USB 2.0 ของบอร์ด Raspberry Pi จะสังเกตเห็นว่าไฟแสดงสถานะ PWR จะติดสว่างพร้อมที่จะใช้งาน
2. ต่อสายพอร์ตอนุกรม DB-9 ของมัลติมิเตอร์เข้ากับบอร์ด ET-USB/RS232 MINI โดยมีตำแหน่งขาใช้งานดังภาพที่ 2.11



ภาพที่ 2.11 ตำแหน่งขาสัญญาณของบอร์ด ET-USB/RS232 MINI

ที่มา: นิเทศ ประเดชนบุญ

ภาษาไพทอน (Python)

ภาษาไพทอน (Python) Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็น Open Source ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรี ๆ โดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้นและใช้งานได้กับงานหลายลักษณะ 5 คุณลักษณะเด่นของภาษา Python ดังนี้

1. สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือสามารถทำงานได้ทุก ๆ CPU หลาย ๆ ระบบปฏิบัติการ เพียงแต่ผู้เขียนโปรแกรมเขียนจากแพลตฟอร์มใด ๆ แล้วนำโปรแกรมที่ได้ไปให้ทำงานต่างแพลตฟอร์มกันได้

2. ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมต้นฉบับ โดยปกติแล้วโปรแกรม ภาษาทั่ว ๆ ไปจะต้องจัดซื้อโปรแกรมต้นฉบับเพื่อนำมาติดตั้งในราคาที่แพงมาก แต่โปรแกรมภาษาไพธอนสามารถดาวน์โหลดจาก www.python.org ได้โดยตรง แล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภท Open Source

3. ภาษาไพธอนได้นำเอาข้อดีของโปรแกรมในอดีตเข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java และ Perl เป็นต้น

4. มีความปลอดภัยสูง เนื่องจากภาษาไพธอนทำงานอยู่ด้าน Server เป็นหลัก เมื่อมีการร้องขอจากเครื่อง Client จะประมวลผลที่เครื่อง Server ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่อง Server ได้โดยตรงจึงมีความปลอดภัยสูง

5. ใช้ในการพัฒนา Web Service ซึ่งในปัจจุบันการพัฒนาซอฟต์แวร์ได้เน้นที่มีการแลกเปลี่ยนข้อมูลซึ่งกันและกันทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน ทำให้เกิดความสะดวกสบาย ไม่ต้องใช้ซอฟต์แวร์อื่น ๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไปเรียนรู้ได้เร็วกว่าโปรแกรมภาษาอื่น ๆ เพราะมีโครงสร้างภาษาที่ไม่ซับซ้อน ซึ่งโครงสร้างภาษาคลายคลึงกับภาษา C ถ้าโปรแกรมเมอร์ที่เคยใช้ภาษา C มาก่อนจะทำให้เรียนรู้ได้เร็วยิ่งขึ้น นอกจากนี้การเขียนโปรแกรมด้วยภาษาไพธอนจะมีความกระชับและสั้นกว่าภาษาซี

โครงสร้างของภาษาไพธอน (PYTHON)

1. Module จะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่าง ๆ และนอกจากยังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ใน package ซึ่งเป็นเหมือน directory ของ Module

2. Comment คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อโค้ดที่เราเขียนขึ้นและมันไม่มีผลต่อการทำงานของโปรแกรม

3. Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอนสำหรับการจบคำสั่งการทำงาน แต่อย่างไรก็ตามในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน (;)

4. Indentation and while space ในภาษา Python นั้นใช้ Whitespace และ Tab สำหรับกำหนดบล็อกของโปรแกรม เช่น คำสั่ง If Else For หรือ การประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านี้เป็นคำสั่งแบบบล็อก โดยจำนวนช่องว่างที่ใช้นั้นต้องเท่ากัน

5. Literals ในการเขียนโปรแกรม Literal คือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรมในภาษา Python นั้นมี Literal ของข้อมูลประเภท ต่าง ๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่ง ตัวอักษรและ boolean นี่เป็นตัวอย่างของการกำหนด Literal ให้กับตัวแปรในภาษา Python

6. Expressions คือการทำงานร่วมกันระหว่างค่าตั้งแต่หนึ่งไปจนถึงหลายค่า โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงานในภาษา Python นั้น Expression จะมีสองแบบ คือ Boolean expression เป็น การกระทำกันของตัวแปรและตัวดำเนินการและจะได้ผลลัพธ์เป็นค่า Boolean โดยทั่วไปแล้วมักจะเป็นตัวดำเนินการเปรียบเทียบค่าและตัวดำเนินการตรรกศาสตร์ และ Expression ทางคณิตศาสตร์ คือการกระทำกันกับตัวดำเนินการและได้ค่าใหม่ที่ไม่ใช่ Boolean

7. Keywords เป็นคำที่ถูกสงวนไว้ในการเขียนโปรแกรมภาษา Python เราไม่สามารถใช้คำสิ่งเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส หรือ identifier ใด ๆ ที่กำหนดขึ้นโดยโปรแกรมเมอร์ ซึ่งรายการของ Keyword ในภาษา Python มีดังนี้ False, None, True, and, as, assert, break, class, continue, def, del, elif, else, except, finally, for, From, global, if, import, in, is, lambda, nonlocal, not, or, pass, raise, return, try, while, with, yield

งานวิจัยที่เกี่ยวข้อง

อารีรัตน์ วงศ์จันทร์ชมพู (2558) ออกแบบระบบควบคุมคุณภาพกังชาโดยใช้บอร์ด Raspberry Pi เนื่องจากกังชามีมูลค่าสูงจึงจำเป็นต้องมีระบบรักษาความปลอดภัยเพื่อรายงานไปยังเกษตรกรผู้ควบคุมกังชาให้รับทราบว่ามีบุคคลใดที่เข้าออกกังชาและรวมถึงระบบแจ้งเตือนอุณหภูมิกังชาผ่านทางระบบแอปพลิเคชันไลน์ โดยผู้วิจัยได้ใช้บอร์ด raspberry pi ร่วมกับโมดูลวัดอุณหภูมิ กล้อง webcam และตัวตรวจจัดการเปิดปิดประตู สำหรับควบคุมคุณภาพกังชาและรักษาความปลอดภัยของกังชา ผลการวิจัยพบว่า หลังจากที่ได้ตรวจจบการเข้าออกประตูส่งออกสัญญาณลอจิกมีค่าเป็น “0” หรือ “1” ให้กับบอร์ด raspberry pi รับทราบสถานะ จากนั้นก็จะส่งงานให้กล้อง webcam บันทึกภาพบุคคลที่ผ่านเข้าออกไว้ ต่อจากนั้นก็ทำการตรวจวัดอุณหภูมิและความชื้นของกังชาโดยใช้เซนเซอร์ DHT22 โดยมีเงื่อนไขว่า อุณหภูมิ $\text{Temperature} \geq 28^{\circ}\text{C}$, ความชื้น $\text{Humidity} \geq 60\%$ ทำการเปรียบเทียบอุณหภูมิความชื้นแล้วส่งออกไปที่ระบบแอปพลิเคชันไลน์

ศุภฤกษ์ ศิวาลัย (2554) สร้างดิจิทัลมัลติมิเตอร์ส่งข้อมูลแบบไร้สาย เพื่อเป็นการแก้ไขปัญหการใช้เครื่องมือวัดในบริเวณที่มีพื้นที่แคบ สูง และจุดที่สามารถทำได้เพียงคนเดียวปัญหาเหล่านี้ อาจทำให้เกิดอุบัติเหตุได้รวมทั้งการสูญเสียชีวิตและทรัพย์สิน ซึ่งการส่งข้อมูลของค่าที่ได้

จากการวัดแบบไร้สายไปยังส่วนของการแสดงผลที่บริเวณอีกพื้นที่หนึ่ง จะสามารถช่วยเหลือการทำงานได้อย่างมีประสิทธิภาพ ดิจิตอลมัลติมิเตอร์ส่งข้อมูลแบบไร้สายสามารถวัดแรงดันไฟฟ้ากระแสตรง แรงดันไฟฟ้ากระแสสลับ กระแส และค่าความต้านทาน ในส่วนของชุดรับข้อมูลแบบไร้สายจะแสดงบนจอแอลซีดีและมีฟังก์ชันใช้งาน ให้เลือก 4 ฟังก์ชัน โดยทั้งสองส่วนนี้จะใช้ไมโครคอนโทรลเลอร์ ATmega128 ในการควบคุมการทำงานต่าง ๆ และใช้ Xbee รุ่น XB24-AWI-001 ในการส่งข้อมูลแบบไร้สาย ผลการดำเนินงานสามารถวัดแรงดันไฟฟ้ากระแสตรงได้ 0 - 250 โวลต์ มีความผิดพลาดตลอดย่านวัด ± 3.0 เปอร์เซ็นต์ แรงดันไฟฟ้ากระแสสลับได้ 0 - 250 โวลต์ มีความผิดพลาดตลอดย่านวัด ± 3.0 เปอร์เซ็นต์ กระแสไฟตรง 0 -1.5 แอมแปร์ มีค่าความผิดพลาดตลอดย่านวัด ± 5.0 เปอร์เซ็นต์ กระแสไฟสลับ 0 -1.5 แอมแปร์ มีค่าความผิดพลาดตลอดย่านวัด ± 5.0 เปอร์เซ็นต์ และค่าความต้านทาน 100 - 1M โอห์ม มีค่าความผิดพลาดตลอดย่านวัด ± 5.0 เปอร์เซ็นต์ การส่งข้อมูลไปยังชุดแสดงผลแบบไร้สายสามารถส่งได้ 30 เมตรและเลือกใช้ฟังก์ชันหยุดค่าขณะทำการวัดในค่านั้น ๆ และสามารถบันทึกค่าได้ 10 ค่าต่อย่านวัด

Runjam Alea (2556) มหาวิทยาลัย Turku จากคณะวิทยาศาสตร์ประยุกต์ วิจัยโดยการอ่านสัญญาณข้อมูลจากดิจิตอลมัลติมิเตอร์ด้วยบอร์ด Raspberry Pi เพื่อวิเคราะห์ว่าโปรโตคอลการสื่อสารที่เกิดขึ้นระหว่างสองอุปกรณ์ใช้สำหรับการพูดคุยกันที่ดำเนินการโดยใช้บอร์ด Raspberry Pi และมัลติมิเตอร์แบบดิจิตอล รุ่น Fluke 189 การเชื่อมต่อระหว่าง Raspberry Pi กับมัลติมิเตอร์โดยใช้สายเคเบิล USB Serial ที่มาพร้อมกับมัลติมิเตอร์และในการเข้าถึงข้อมูลจากมัลติมิเตอร์จะใช้ภาษาโปรแกรม Python ในการเข้าถึงข้อมูลโดยรูปแบบข้อมูลแบบอนุกรมและข้อมูลที่ได้รับการใช้โปรแกรม Python ในวิธีดิจิตอลนั้นได้ทำการทดสอบด้วยอุปกรณ์แบบอนาล็อก ดังนั้นคือ การวัดอุณหภูมิจะใช้สายเทอร์โมคัปเปิ้ล (Thermocouple) เสียบเข้าที่มัลติมิเตอร์รุ่น Fluke 189 เทียบกับอุณหภูมิที่อ่านได้จากปรอทแก้ว ผลการวิจัยพบว่า ค่าอุณหภูมิที่ได้รับจากมัลติมิเตอร์รุ่น Fluke 189 มีค่าใกล้เคียงกันมากกับอุณหภูมิที่อ่านได้จากปรอทแก้ว จึงสรุปผลได้ว่าค่าที่ได้รับจากโปรแกรมภาษา python มีความถูกต้องแม่นยำสูง

ขเจนศักดิ์ ไตรธิเลน (2553) พัฒนากิจกรรมการเรียนการสอนที่เน้นทักษะปฏิบัติเรื่องการใช้อุปกรณ์และการต่อวงจรไฟฟ้า โดยมีจุดมุ่งหมายคือ 1. เพื่อพัฒนาและหาประสิทธิภาพของกิจกรรมการเรียนการสอนให้มีประสิทธิภาพตามเกณฑ์ 80/80 2. เพื่อเปรียบเทียบทักษะปฏิบัติงานของนักเรียนก่อนเรียนและหลังเรียน 3. เพื่อเปรียบเทียบความรู้ความเข้าใจของนักเรียนก่อนเรียนและหลังเรียน 4. เพื่อศึกษาเจตคติของนักเรียนต่อการเรียนด้วยกิจกรรมการเรียนการสอน กลุ่มตัวอย่างที่ใช้ในการวิจัยเป็นนักเรียนชั้นมัธยมศึกษาปีที่ 1 โรงเรียนต่างอยพัฒนศึกษา ภาคเรียนที่ 1 ปีการศึกษา 2552 จำนวน 21 คน ผลการวิจัยพบว่า มีประสิทธิภาพของกิจกรรมการเรียน

การสอนสูงกว่าเกณฑ์ มีคะแนนทักษะปฏิบัติ และคะแนนความรู้ ความเข้าใจ หลังเรียนสูงกว่า ก่อนเรียนและนักเรียนมีเจตคติที่ดีต่อการเรียนอยู่ในระดับมาก

ทัศนัย ใจเย็น (2554) สร้างชุดการสอนเรื่องวงจรไฟฟ้า วิชางานไฟฟ้าและอิเล็กทรอนิกส์ เบื้องต้น หลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2546 สำนักงานคณะกรรมการการอาชีวศึกษา มีวัตถุประสงค์ เพื่อสร้างชุดการสอนเรื่อง วงจรไฟฟ้า วิชางานไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้น หลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2546 สำนักงานคณะกรรมการการอาชีวศึกษา เพื่อทดสอบประสิทธิภาพชุดการสอนที่สร้างขึ้น และเปรียบเทียบผลสัมฤทธิ์ทางการเรียนก่อน และหลังเรียนของชุดการสอนที่สร้างขึ้น กลุ่มตัวอย่างได้แก่ นักศึกษาแผนกช่างอิเล็กทรอนิกส์ ระดับประกาศนียบัตรวิชาชีพ ที่ลงทะเบียนเรียนวิชางานไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้นหลักสูตร ประกาศนียบัตรวิชาชีพ ภาคเรียนที่ 1 ปีการศึกษา 2554 วิทยาลัยสารพัดช่างบรรหาร-แจ่มใส จังหวัด สุพรรณบุรี จำนวน 20 คน ผลการวิจัยพบว่า ประสิทธิภาพชุดการสอนเรื่องวงจรไฟฟ้าสูงกว่า สมมติฐานที่ตั้งไว้และผลสัมฤทธิ์ทางการเรียนของนักศึกษาที่เรียนด้วยชุดการสอน เรื่องวงจรไฟฟ้า วิชางานไฟฟ้าและอิเล็กทรอนิกส์เบื้องต้นหลังเรียนสูงกว่าก่อนเรียนอย่างมีนัยสำคัญที่ระดับ 0.05

ภักธิมา เทียงตรง (2552) หาประสิทธิภาพในการใช้โปรแกรมคอมพิวเตอร์ฝึกพิมพ์ดีด สัมผัสสำหรับคนพิการ มีวัตถุประสงค์เพื่อเปรียบเทียบทักษะในการพิมพ์ดีดสัมผัสภาษาไทยของ นักเรียนตาบอดที่ใช้วิธีการฝึกด้วยโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการทางตากับการใช้วิธีการฝึก ด้วยวิธีปกติและศึกษาความพึงพอใจของนักเรียนตาบอดที่มีต่อโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการ ทางตา โดยกลุ่มตัวอย่างเป็นนักเรียนตาบอดที่กำลังศึกษาอยู่ในชั้นประถมศึกษาปีที่ 6 และมัธยมศึกษา ปีที่ 1-3 ภาคเรียนที่ 1 ปีการศึกษา 2552 โรงเรียนสอนคนตาบอดกรุงเทพ จำนวน 30 คน แบ่งเป็น กลุ่มควบคุม 15 คน และกลุ่มทดลอง 15 คน ผลการศึกษาพบว่า ทักษะในการพิมพ์ดีดสัมผัส ภาษาไทยของนักเรียนตาบอดที่ใช้วิธีการฝึกด้วยโปรแกรมฝึกพิมพ์ดีดสำหรับผู้พิการทางตามีทักษะ ในการพิมพ์ดีดสัมผัสภาษาไทยสูงกว่านักเรียนตาบอดที่ใช้วิธีการฝึกด้วยวิธีปกติ และความพึงพอใจ อยู่ในระดับมาก

บทที่ 3

วิธีดำเนินการวิจัย

รูปแบบการวิจัย

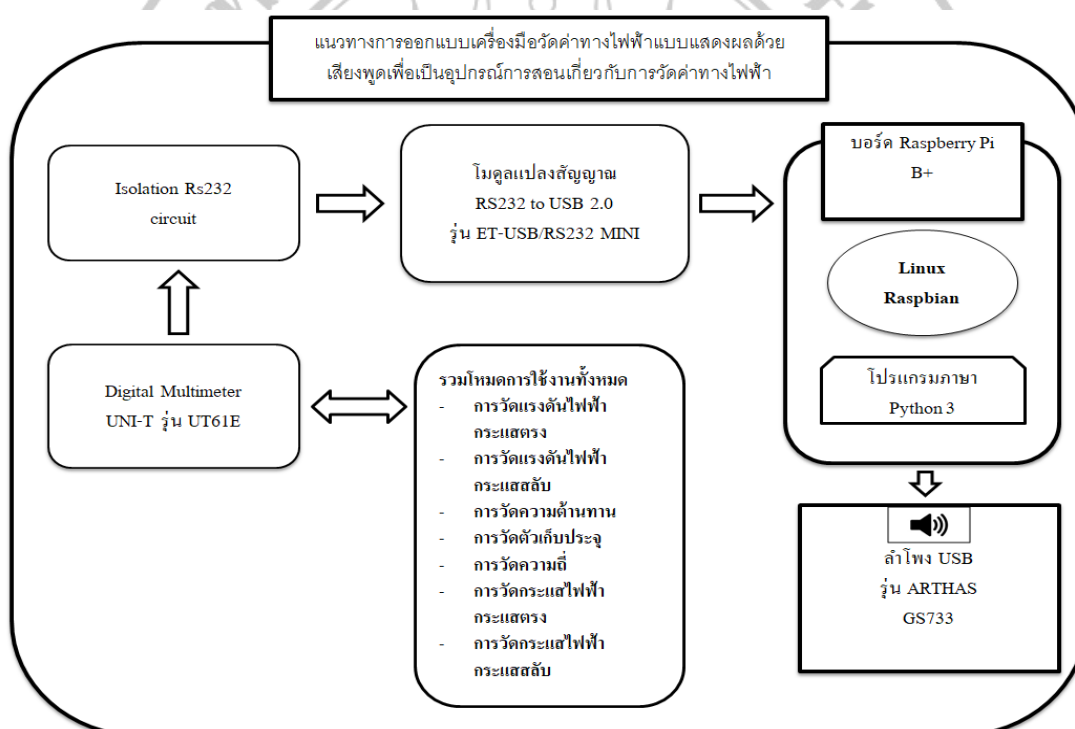
การออกแบบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด จำเป็นจะต้องมีการศึกษากระบวนการและวิธีการหลายด้าน เพื่อประกอบทุกส่วนให้เข้ากันและทำการแก้ไขปรับปรุงให้กระบวนการต่าง ๆ สอดคล้องไปในแนวทางเดียวกัน โดยมีการจำแนกออกเป็นการศึกษาด้านทฤษฎีด้านไฟฟ้าและอิเล็กทรอนิกส์ รวมกับการศึกษาด้านการเขียนโปรแกรมคอมพิวเตอร์ร่วมด้วย จึงจะสามารถทำให้ระบบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดทำงานได้อย่างถูกต้องเหมาะสมสำหรับใช้เป็นอุปกรณ์ในสื่อวัตกรรมการเรียนการสอนด้านวิทยาศาสตร์ โดยได้จำแนกขั้นตอนในการดำเนินงานดังต่อไปนี้

ขั้นตอนการดำเนินงาน

1. ศึกษาขั้นตอนการอ่านค่าจากมัลติมิเตอร์ไฟฟ้ารุ่น UT61E ผ่านการติดต่อสื่อสารแบบอนุกรม
2. ศึกษาการใช้งานบอร์ดคอมพิวเตอร์ Raspberry Pi
3. ศึกษาการสร้างไฟล์เสียง MP3 จากเว็บไซต์แปลภาษาเพื่อใช้ในการเรียกใช้งานจากโปรแกรมภาษา Python เพื่อแสดงผลด้วยเสียง
4. ศึกษาและเขียนโปรแกรมภาษา Python เพื่ออ่านค่าจาก มัลติมิเตอร์รุ่น UT61E และแสดงผลด้วยเสียง
5. ศึกษาทฤษฎีเกี่ยวกับการต่อวงจรไฟฟ้าเบื้องต้น
6. การทดสอบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดและทำการวัดค่าทางไฟฟ้าแต่ละโหมด ทำการวัดซ้ำจำนวน 3 ครั้งเพื่อหาค่าเฉลี่ยซึ่งแต่ละโหมดที่ทำการทดสอบมีดังนี้ คือ โหมดการวัดไฟฟ้ากระแสตรง โหมดวัดความต้านทาน โหมดวัดตัวเก็บประจุ โหมดวัดความถี่ไฟฟ้า และโหมดวัดกระแสไฟฟ้ากระแสตรง

7. สรุปและอภิปรายผลการทดลองและแสดงกราฟความสัมพันธ์ระหว่างค่าที่แสดงผลทางหน้าจอมัลติมิเตอร์และค่าเฉลี่ยของสัญญาณที่ถูกจำแนกผ่านทางพอร์ตยูเอสบีโดยแนวทางการออกแบบแสดงไว้ดังภาพที่ 3.1

8. ประเมินความพึงพอใจในสื่อการสอนและนวัตกรรม ที่มีเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด รวมในการทดลอง เรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป



ภาพที่ 3.1 แนวการออกแบบมัลติมิเตอร์แสดงผลด้วยเสียง

ที่มา: นิเทศ ประเดชญญ

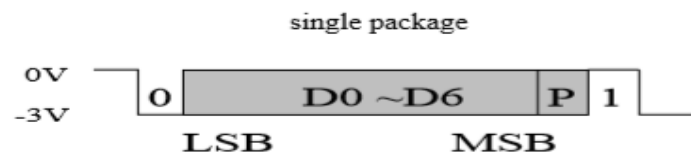
ขั้นตอนการอ่านค่าจากมัลติมิเตอร์ รุ่น UT61E ผ่านการติดต่อสื่อสารแบบอนุกรม

1. การอ่านค่าจากมัลติมิเตอร์ รุ่น UT61E ต้องตั้งค่าการติดต่อสื่อสารผ่านพอร์ตอนุกรมแบบ RS232 ดังแสดงในตารางที่ 3.1 ดังนี้

ตารางที่ 3.1 การตั้งค่าการสื่อสารผ่านพอร์ตอนุกรม

Baud Rate	19200
Start bit	1
Stop bit	1
Data bits	7
Parity	(odd)

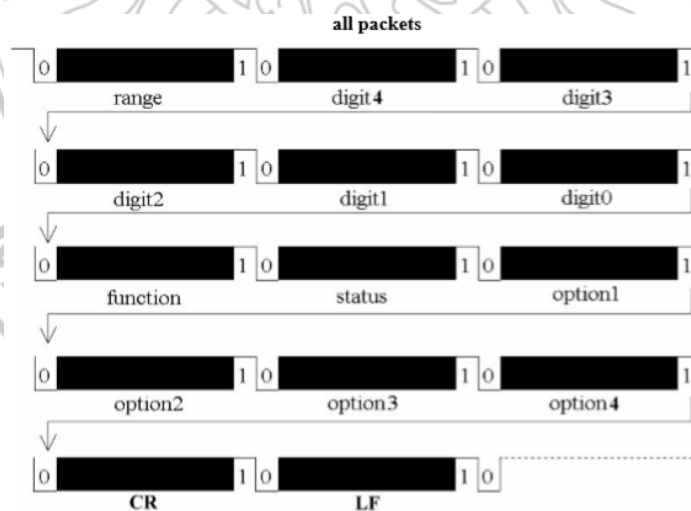
2. ข้อมูล 1 ชุด จะประกอบไปด้วยจำนวนบิตข้อมูล 10 บิต ดังแสดงในตารางที่ 3.1



ภาพที่ 3.2 ชุดข้อมูล 10 บิต

ที่มา: นิเทศ ประเดชญญ

3. ข้อมูลจะส่งออกจากพอร์ตอนุกรม โดยข้อมูลจะถูกส่งมาทีละ 14 ชุดข้อมูล รวมเป็น 140 บิต ดังภาพที่ 3.3 ประกอบด้วยรายละเอียดดังต่อไปนี้



ภาพที่ 3.3 ชุดข้อมูล 140 บิต

ที่มา: นิเทศ ประเดชญญ

4. Function ชุดข้อมูลนี้จะแสดงโหมดของการวัดในขณะนั้น ตามตารางที่ 3.2

ตารางที่ 3.2 โหมดของการวัด

รหัส	โหมด
0111011	Voltage
0111101	Auto μ A Current
0111111	Auto mA Current
0110000	22 A current
0111001	Manual A Current
0110011	Ω
0110101	Continuity
0110001	Diode
0110010	Frequency
0110110	Capacitance
0110100	Temperature
0111110	ADP

5. Range ชุดข้อมูลนี้จะแสดงถึงช่วงของการแสดงผลแบบเต็มสเกลของแต่ละโหมดการวัด ตัวอย่างเช่น ถ้า code ที่ได้รับเป็น '0110000' หมายความว่า ถ้าขณะวัดอยู่ในโหมดวัดแรงดันไฟฟ้า ค่าที่แสดงได้สูงสุดเป็น 2.2000V โดยจะแสดงผลแบบทศนิยม 4 ตำแหน่ง ตามตารางที่ 3.3

ตารางที่ 3.3 ชุดข้อมูล Range ต่าง ๆ

Code	V	*2-range auto A	22 A	Manual A	ADP	Ω	Frequency	Capacitor
0110000	2.2000V	Lower Range (IVSL)	22.000 A	2.2000A	ADP4	220.00 Ω	22.00Hz	22.000nF
0110001	22.000V	Higher Range (IVSH)		22.000A	ADP3	2.2000K Ω	220.0Hz	220.00nF
0110010	220.00V			220.00A	ADP2	22.000K Ω		2.2000 μ F
0110011	2200.0V			2200.0A	ADP1	220.00K Ω	22.000KHz	22.000 μ F
0110100	220.00mV			22000A	ADP0	2.2000M Ω	220.00KHz	220.00 μ F
0110101						22.000M Ω	2.2000MHz	2.2000mF
0110110						220.00M Ω	22.000MHz	22.000mF
0110111							220.00MHz	220.00mF

*It includes auto μ A, mA, 22.000A/220.00A, 220.00A/2200.0A.

6. DIGIT4 – DIGIT0 ชุดข้อมูลนี้แสดงถึงรหัสตัวเลขที่แสดงผลที่หน้าจอ LCD ของตัวมัลติมิเตอร์ โดยเป็นรหัสแสดงหมายเลข 0-9 และ DIGIT4 จะเป็นชุดข้อมูลแรกที่เข้ามาก่อน และ DIGIT0 จะเป็นตัวเลขชุดสุดท้ายตามลำดับ ซึ่งรหัสสามารถจำแนกตามตารางที่ 3.4

ตารางที่ 3.4 รหัสแทนตัวเลข 0-9

Digit	Code
0	0110000
1	0110001
2	0110010
3	0110011
4	0110100
5	0110101
6	0110110
7	0110111
8	0111000
9	0111001

7. STATUS ชุดข้อมูลนี้ใช้แสดงสถานะของค่าที่อ่านได้จากมัลติมิเตอร์ตามตารางที่ 3.5

8. บิต Judge ใช้แสดงสถานะอุณหภูมิ ถ้าเป็น '1' คือ องศาเซลเซียส และถ้าเป็น '0' คือ องศาฟาเรนไฮต์

9. บิต Sign ใช้แสดงเครื่องหมายสัญญาณ ถ้าค่าเป็น '1' คือสัญญาณเป็นลบ และถ้าเป็น '0' คือ สัญญาณเป็นบวก

10. บิต BATT ใช้แสดงสถานะแบตเตอรี่ ถ้าเป็น '1' คือแบตเตอรี่ต่ำ

11. บิต OL ถ้าเป็น '1' แสดงถึงค่าสัญญาณอินพุตที่ได้รับสูงเกินค่าที่จะวัดได้ (Overflow)

ตารางที่ 3.5 ชุดข้อมูล STATUS

0	1	1	Judge	Sign	BATT	OL
BIT6	BIT 5	BIT 4	BIT 3	BIT 2	BIT1	BIT 0

OPTION 1 ชุดข้อมูลนี้แสดงสถานะของการใช้งานมัลติมิเตอร์ในฟังก์ชัน ตามตารางที่ 3.6

1. บิต MAX / MIN จะแสดงถึงมัลติมิเตอร์อยู่ในการ โหมดการวัดค่าสูงและค่าต่ำสุดของสัญญาณ
2. บิต RMR จะเป็น '1' ถ้าขณะนั้นอยู่ในโหมด MAX หรือ MIN
3. บิต REL จะเป็น '1' ถ้าขณะนั้นอยู่ในโหมด REL/Zero mode

ตารางที่ 3.6 ชุดข้อมูล OPTION 1

0	1	1	MAX	MIN	REL	RMR
BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

12. OPTION 2 ชุดข้อมูลนี้แสดงสถานะของการใช้งานมัลติมิเตอร์ในฟังก์ชัน ตามตารางที่ 3.7

1. บิต UL จะเป็น '1' ตามเงื่อนไขดังต่อไปนี้ คือ ถ้าอยู่ในโหมดวัดความถี่ไฟฟ้า และค่าที่วัดได้น้อยกว่า 2 Hz ในย่านการวัด 22Hz หรือ ค่าที่วัดได้น้อยกว่า 20 Hz ในย่านการวัด 220Hz หรือวัดความกว้างของความถี่ (duty cycle) น้อยกว่า 10%
2. บิต P_{max} / P_{min} จะเป็น '1' จะแสดงการวัดค่าพีค(Peak) ของสัญญาณสูงสุดหรือพีคต่ำสุด

ตารางที่ 3.7 ชุดข้อมูล OPTION 2

0	1	1	UL	P_{MAX}	P_{MIN}	0
BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT1	BIT 0

13. OPTION 3 ชุดข้อมูลนี้แสดงสถานะของโหมดการวัด ตามตารางที่ 3.8

1. บิต DC ถ้าเป็น '1' แสดงถึงอยู่ในโหมดการวัดไฟฟ้ากระแสตรง
2. บิต AC ถ้าเป็น '1' แสดงถึงอยู่ในโหมดการวัดไฟฟ้ากระแสสลับ
3. บิต AUTO ถ้าเป็น '1' แสดงถึงอยู่ในโหมดการวัดแบบอัตโนมัติ ถ้าเป็น '0' คือ

โหมด Manual

ตารางที่ 3.8 ชุดข้อมูล OPTION 3

0	1	1	DC	AC	AUTO	VAHZ
BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

14. OPTION 4 ชุด ข้อมูลนี้แสดงสถานะของวงจรภายในซึ่งเราไม่นำชุดข้อมูลนี้มาใช้งาน ตามตารางที่ 3.9

1. บิต VBAR ถ้าเป็น '1' แสดงถึง ขา VBAR เชื่อมต่อถึงสัญญาณ V-
2. บิต Hold ถ้าเป็น '1' แสดงถึงเข้าสู่โหมด Hold
3. บิต LPF ถ้าเป็น '1' แสดงถึงวงจรกรองความถี่ต่ำกำลังทำงาน

ตารางที่ 3.9 ชุดข้อมูล OPTION 4

0	1	1	0	VBAR	Hold	LPF
BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0

15. CR ชุดข้อมูลนี้จะเป็นรหัสคงที่ คือ 0001101

16. LF ชุดข้อมูลนี้จะเป็นรหัสคงที่ คือ 0001010

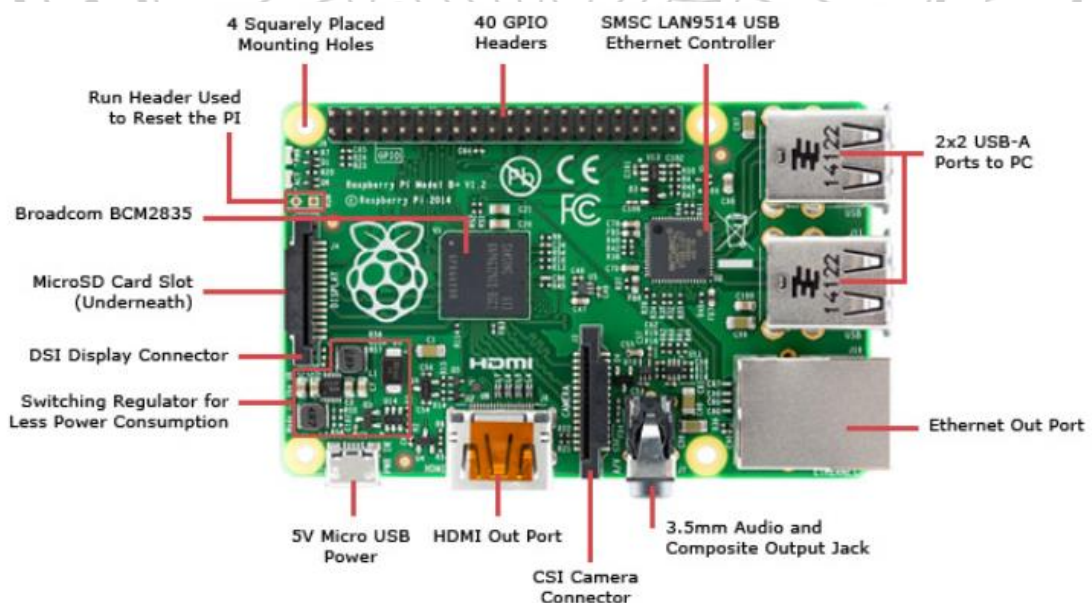
การใช้งานบอร์ดคอมพิวเตอร์ Raspberry Pi

การจะเริ่มต้นใช้งานบอร์ดคอมพิวเตอร์ Raoberry Pi จะต้องมีการเตรียมความพร้อมด้านอุปกรณ์ที่มีความจำเป็นดังต่อไปนี้ก่อน คือ

1. บอร์ด Raspberry Pi 3 Model B
2. Micro SD Card ขนาด 8 GB ขึ้นไป ถ้าได้ 16 GB จะดีมากครับ (แนะนำ class 10)

4. แหล่งจ่ายไฟ Adapter 5V 2A พร้อมสาย Micro USB
5. สาย HDMI (สำหรับหน้าจอแบบ HDMI) หรือ สาย VGA (สำหรับหน้าจอแบบ VGA)
6. หน้าจอแสดงผล (สามารถเชื่อมต่อแบบ HDMI หรือ VGA ได้)
7. USB Keyboard และ USB Mouse
8. สาย LAN สำหรับเชื่อมต่อ Internet หรือ USB Wifi
9. คอมพิวเตอร์ สำหรับลงโปรแกรมที่จำเป็นสำหรับการติดตั้งระบบปฏิบัติการ Linux เข้าสู่การ์ดหน่วยความจำ แบบ Micro SD Card
10. Card Reader สำหรับใช้ในการเขียนอิมเมจไฟล์ที่บรรจุระบบปฏิบัติการ Linux เข้าสู่การ์ดหน่วยความจำ แบบ Micro SD Card
11. ลำโพงแบบ USB ที่มีขั้วต่อลำโพง 3.5mm สำหรับใช้ในการทดสอบเสียง

หลังจากเตรียมความพร้อมด้านอุปกรณ์เสร็จแล้ว ต่อจากนั้นก็จะเป็นการเริ่มติดตั้งระบบปฏิบัติการ Linux เข้าสู่การ์ดหน่วยความจำ แบบ Micro SD Card โดยมีขั้นตอนเป็นไปตามภาคผนวก ข. การติดตั้งระบบปฏิบัติการ Linux และการตั้งค่าพื้นฐาน เมื่อทำการติดตั้งระบบปฏิบัติการ Linux เสร็จสิ้น จากนั้นให้ทำการเชื่อมต่อบอร์ด Raspberry Pi กับอุปกรณ์ที่จำเป็นต่าง ๆ ตามรายการข้างต้น ซึ่งต้องเชื่อมต่อให้ตรงตำแหน่งของอุปกรณ์นั้น ๆ โดยอ้างอิงตำแหน่งการเชื่อมต่อ ดังแสดงดังภาพที่ 3.2 แต่ต้องไม่เชื่อมต่อแหล่งจ่ายพลังงานไฟฟ้าให้กับตัวบอร์ด Raspberry Pi โดยมีขั้นตอนต่อไปดังนี้



ภาพที่ 3.4 ตำแหน่งอุปกรณ์บนบอร์ด Raspberry Pi

ที่มา: นิเทศ ประเดชญ, 2562

1. การใส่ Micro SD card ที่ติดตั้งระบบปฏิบัติการ Linux ไว้เรียบร้อยแล้ว ใส่เข้าไปที่บอร์ด Raspberry Pi 3 Model B+ โดยใส่ในช่อง Micro Card Slot ซึ่งอยู่ด้านล่างของตัวบอร์ด Raspberry Pi ดังแสดงตำแหน่งตามภาพที่ 3.2
2. ต่อ Power Adapter ที่มีแรงดันไฟฟ้า 5V และสามารถจ่ายกระแสไฟฟ้าได้อย่างน้อย 2A เข้ากับสาย Micro USB แล้วทำการต่อสาย Micro USB เข้าที่แหล่งจ่ายไฟของตัวบอร์ด Raspberry Pi 3 Model B+ ตรงตำแหน่ง 5V Micro USB Power ดังแสดงในภาพที่ 3.2
3. ทำการต่อสาย HDMI ของจอมอนิเตอร์แสดงผล เข้ากับช่องสัญญาณ HDMI Out Port ของบอร์ด Raspberry Pi (ถ้าจอมอนิเตอร์ที่มี Port VGA ก็สามารถใช้งานได้เช่นกัน แต่จะต้องมีตัวแปลงสัญญาณ HDMI To VGA ก่อนถึงจะใช้งานได้)
4. ต่อ USB Keyboard และ USB Mouse กับบอร์ด Raspberry Pi (แนะนำให้ใช้ Keyboard/Mouse แบบไร้สาย จะเพิ่มความสะดวกยิ่งขึ้น)
5. เชื่อมต่อสาย LAN สำหรับเชื่อมต่อ Internet หรือ USB Wifi ถ้าต้องการติดตั้งโปรแกรมเสริมเพิ่มเติม หรือถ้าต้องการอัปเดตข้อมูลของระบบปฏิบัติการ Linux
6. ทำการต่อลำโพงแบบ USB ที่มีหัว 3.5mm เข้าที่ Audio Composite Output Jack ของบอร์ด Raspberry pi ดังแสดงตามภาพที่ 3.2 ซึ่งมีขนาดของ Output Jack 3.5mm
7. ทำการต่อ Power Adapter เข้ากับแหล่งจ่ายไฟหลัก เพื่อทำการบูตระบบปฏิบัติการ Linux ที่ติดตั้งไว้บนหน่วยความจำ Micro SD Card และทำการกำหนดค่าพื้นฐานต่าง ๆ เพื่อให้พร้อมสำหรับการพัฒนาโปรแกรมภาษา Python ในลำดับต่อไป สามารถอ้างอิงการตั้งค่าพื้นฐานได้จาก ภาคผนวก ข. การติดตั้งระบบปฏิบัติการ Linux และการตั้งค่าพื้นฐาน

การสร้างไฟล์เสียง MP3 จากเว็บไซต์แปลภาษา

การออกแบบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดจำเป็นต้องมีเสียงเข้ามาเป็นส่วนประกอบในการออกแบบด้วย ซึ่งเสียงที่เราต้องสร้างขึ้นจะใช้เสียงสังเคราะห์เลียนแบบเสียงมนุษย์ และบันทึกเสียงเหล่านั้นไว้ในตำแหน่งของหน่วยความจำบนบอร์ด Raspberry Pi เพื่อให้โปรแกรมภาษา Python เรียกใช้งาน การสร้างไฟล์เสียงสามารถสร้างได้หลายวิธี เช่น การบันทึกเสียงผู้พูดเอง การบันทึกเสียงจากเว็บไซต์ เป็นต้น โดยในการบันทึกเสียงพูดเองนั้นจะต้องใช้เครื่องบันทึกเสียงหรือใช้โปรแกรมบันทึกเสียง อาทิเช่น โปรแกรมบันทึกและตัดต่อเสียงชื่อว่า Audacity เพื่อสร้างไฟล์เสียง แต่จะต้องดาวน์โหลดและติดตั้งโปรแกรม รวมถึงต้องซื้อตัวไมโครโฟน เพื่อบันทึกเสียงพูด ซึ่งมีความยุ่งยากและต้องเสียค่าใช้จ่ายเพิ่มเติม

วิธีการบันทึกเสียงจากเว็บไซต์ เป็นวิธีที่ง่ายและสะดวกประหยัดค่าใช้จ่าย โดยบันทึกเสียงจากเว็บไซต์แปลภาษาของบริษัท Google ซึ่งใช้งานได้ง่าย เพียงพิมพ์คำพูดที่

ต้องการลงไปแล้วแปลงเสียงพูดโดยการกดปุ่มเสียงพูด ตัวเว็บไซต์ก็จะสร้างไฟล์เสียงขึ้นมา มีเสียงพูดปรากฏขึ้นทันที จึงง่ายต่อการแก้ไขและสร้างไฟล์เสียงได้ตลอดเวลา แต่ต้องทำขณะที่เราเชื่อมต่อระบบอินเทอร์เน็ตเท่านั้น และเสียงพูดจะมีลักษณะเป็นเสียงของผู้หญิงเป็นเสียงพื้นฐานของระบบ ไม่สามารถทำการเปลี่ยนเสียงเป็นเพศอื่น ๆ ได้ เช่น เสียงเด็ก เสียงผู้ชาย เป็นต้น

งานวิจัยนี้จึงใช้วิธีการบันทึกเสียงจากเว็บไซต์แปลภาษา เพื่อดำเนินการสร้างไฟล์เสียงให้กับระบบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด จากนั้นบันทึกไฟล์เสียงที่ได้เก็บไว้ในตำแหน่งในหน่วยความจำ Micro SD Card ของระบบ และให้โปรแกรมภาษา Python ทำการเรียกใช้ไฟล์เสียงได้อย่างแม่นยำ และเรียบเรียงลำดับเสียงให้ถูกต้องตามค่าที่รับมาจากตัวมัลติมิเตอร์ระบบตัวเลข ยี่ UNI-T รุ่น UT61E โดยมีขั้นตอนในการสร้างไฟล์เสียงจากเว็บไซต์ดังนี้คือ

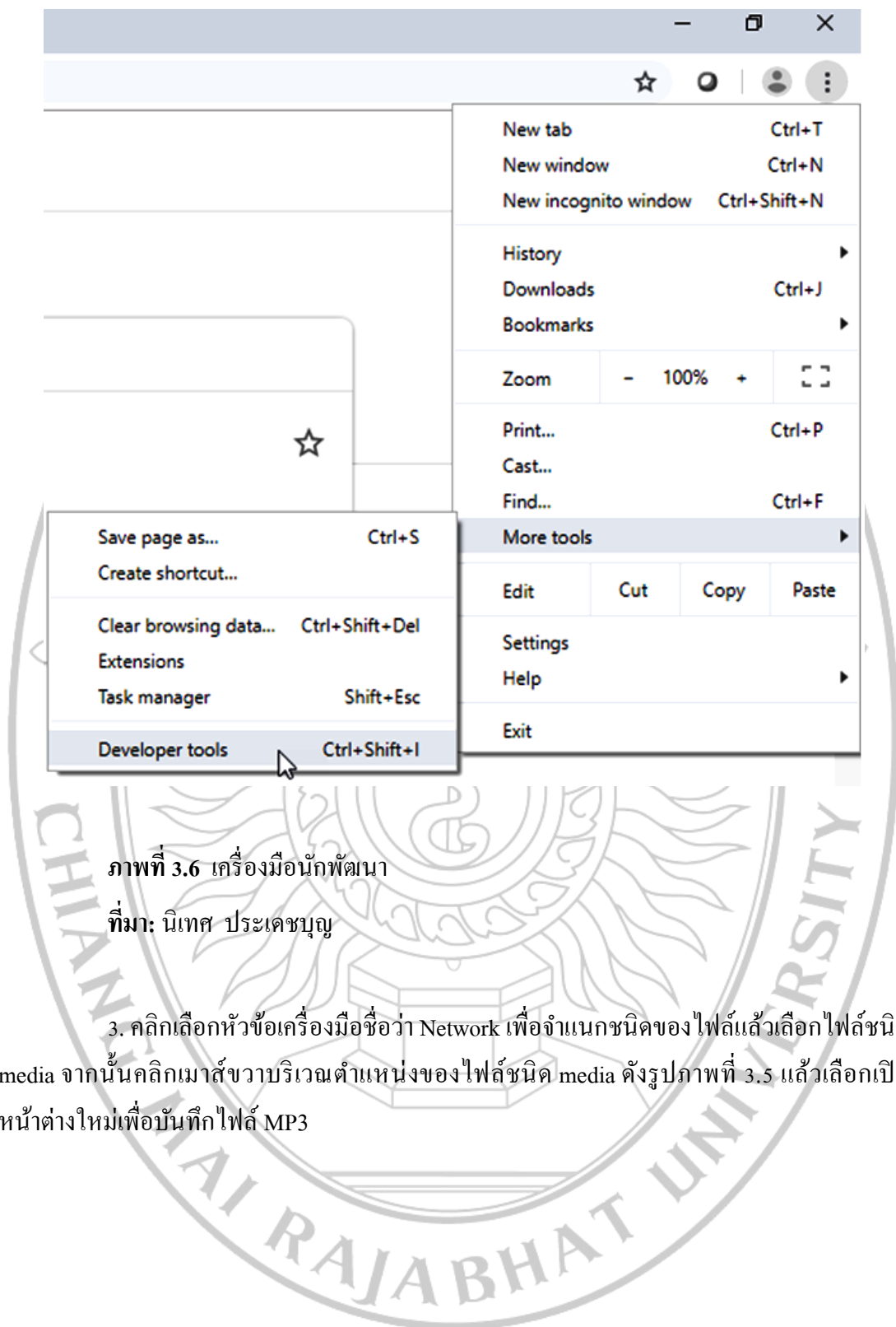
1. การสร้างไฟล์เสียงจากเว็บไซต์แปลภาษา โดยเราพิมพ์ลิงค์ลงไปในเว็บไซต์เบราว์เซอร์ชื่อว่า <https://translate.google.co.th/?hl=th> จากนั้นพิมพ์เสียงที่ต้องการลงไป ตามรูปภาพที่ 3.3



ภาพที่ 3.5 เว็บไซต์ google translate

ที่มา: นิเทศ ประเดชนุญ

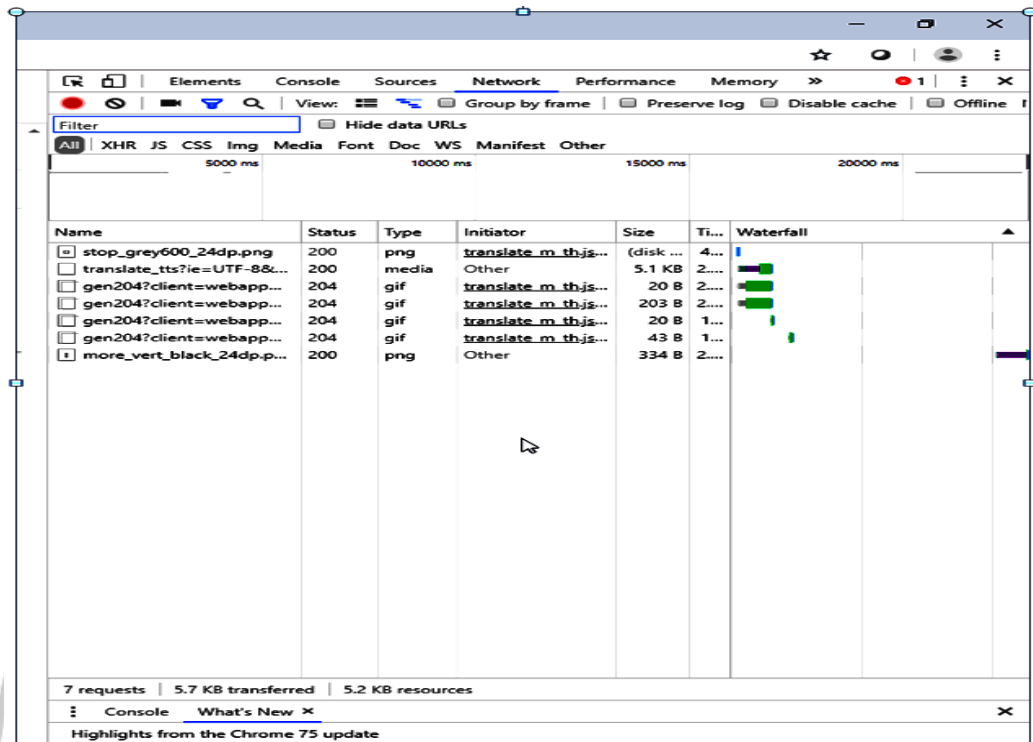
2. เข้าไปในโหมดการทำงานสำหรับเครื่องมือพัฒนา ดังรูปภาพที่ 3.4 เพื่อเลือกใช้งานเครื่องมือในการวิเคราะห์ชนิดของไฟล์



ภาพที่ 3.6 เครื่องมือนักพัฒนา

ที่มา: นิเทศ ประเดชนบุญ

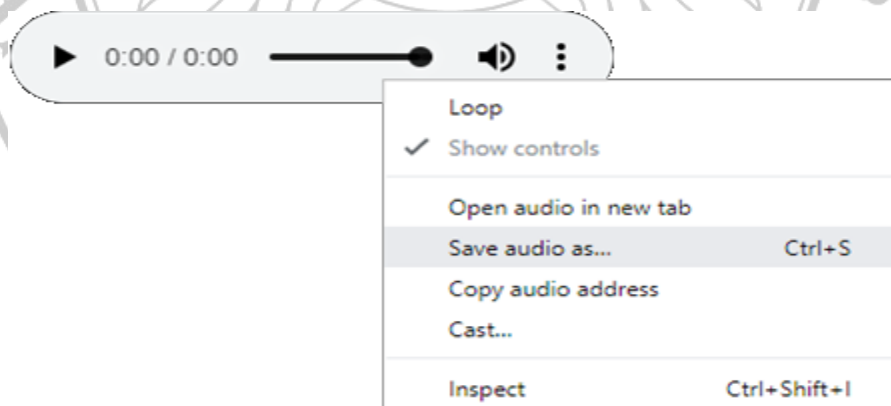
3. คลิกเลือกหัวข้อเครื่องมือชื่อว่า Network เพื่อจำแนกชนิดของไฟล์แล้วเลือกไฟล์ชนิด media จากนั้นคลิกเมาส์ขวาบริเวณตำแหน่งของไฟล์ชนิด media ดังรูปภาพที่ 3.5 แล้วเลือกเปิดหน้าต่างใหม่เพื่อบันทึกไฟล์ MP3



ภาพที่ 3.7 เปิดหน้าต่างใหม่เพื่อบันทึกไฟล์ .MP3

ที่มา: นิเทศ ประเดชญ

4. เข้าไปที่หน้าต่างวินโดว์ที่เปิดขึ้นมาใหม่ จากนั้นคลิกเมาส์ขวามบนแท็บของตัวเล่นเสียง ดังรูปภาพที่ 3.6



ภาพที่ 3.8 การบันทึกไฟล์ MP3

ที่มา: นิเทศ ประเดชญ, 2562

5. การปรากฏหน้าต่างแสดงเมนูจะปรากฏขึ้น จากนั้นเลือกที่เมนู Save audio as.. แล้วบันทึกไฟล์ .MP3 เก็บไว้เพื่อใช้งานสำหรับการเขียนโปรแกรม

6. สร้างชื่อไฟล์และทำการจัดเก็บไว้ในไดเรกทอรี ดังตารางที่ 3.10 เพื่อใช้หาความสัมพันธ์ของเสียงที่จะเรียกใช้งานจากโปรแกรมภาษา Python ซึ่งจะระบุตำแหน่งจัดเก็บของไฟล์เสียงนั้น ๆ ใน SD Card บนบอร์ด Raspberry Pi

ตารางที่ 3.10 ความสัมพันธ์ของไฟล์เสียงและตำแหน่งจัดเก็บ

ชื่อไฟล์	ตำแหน่งที่จัดเก็บ	ความหมายเสียง
0.mp3	/home/pi/MP34DMM/0.mp3	ศูนย์
1.mp3	/home/pi/MP34DMM/1.mp3	หนึ่ง
2.mp3	/home/pi/MP34DMM/2.mp3	สอง
3.mp3	/home/pi/MP34DMM/3.mp3	สาม
4.mp3	/home/pi/MP34DMM/4.mp3	สี่
5.mp3	/home/pi/MP34DMM/5.mp3	ห้า
6.mp3	/home/pi/MP34DMM/6.mp3	หก
7.mp3	/home/pi/MP34DMM/7.mp3	เจ็ด
8.mp3	/home/pi/MP34DMM/8.mp3	แปด
9.mp3	/home/pi/MP34DMM/9.mp3	เก้า
10.mp3	/home/pi/MP34DMM/10.mp3	สิบ
20.mp3	/home/pi/MP34DMM/20.mp3	ยี่สิบ
30.mp3	/home/pi/MP34DMM/30.mp3	สามสิบ
40.mp3	/home/pi/MP34DMM/40.mp3	สี่สิบ
50.mp3	/home/pi/MP34DMM/50.mp3	ห้าสิบ
60.mp3	/home/pi/MP34DMM/60.mp3	หกสิบ
70.mp3	/home/pi/MP34DMM/70.mp3	เจ็ดสิบ
80.mp3	/home/pi/MP34DMM/80.mp3	แปดสิบ
90.mp3	/home/pi/MP34DMM/90.mp3	เก้าสิบ

ตารางที่ 3.10 (ต่อ)

ชื่อไฟล์	ตำแหน่งที่จัดเก็บ	ความหมายเสียง
100.mp3	/home/pi/MP34DMM/100.mp3	หนึ่งร้อย
200.mp3	/home/pi/MP34DMM/200.mp3	สองร้อย
300.mp3	/home/pi/MP34DMM/300.mp3	สามร้อย
400.mp3	/home/pi/MP34DMM/400.mp3	สี่ร้อย
500.mp3	/home/pi/MP34DMM/500.mp3	ห้าร้อย
600.mp3	/home/pi/MP34DMM/600.mp3	หกร้อย
700.mp3	/home/pi/MP34DMM/700.mp3	เจ็ดร้อย
800.mp3	/home/pi/MP34DMM/800.mp3	แปดร้อย
900.mp3	/home/pi/MP34DMM/900.mp3	เก้าร้อย
1000.mp3	/home/pi/MP34DMM/1000.mp3	หนึ่งพัน
2000.mp3	/home/pi/MP34DMM/2000.mp3	สองพัน
3000.mp3	/home/pi/MP34DMM/3000.mp3	สามพัน
4000.mp3	/home/pi/MP34DMM/4000.mp3	สี่พัน
5000.mp3	/home/pi/MP34DMM/5000.mp3	ห้าพัน
6000.mp3	/home/pi/MP34DMM/6000.mp3	หกพัน
7000.mp3	/home/pi/MP34DMM/7000.mp3	เจ็ดพัน
8000.mp3	/home/pi/MP34DMM/8000.mp3	แปดพัน
9000.mp3	/home/pi/MP34DMM/9000.mp3	เก้าพัน
AC.mp3	/home/pi/MP34DMM/AC.mp3	กระแสสลับ
ampare.mp3	/home/pi/MP34DMM/ampare.mp3	แอมป์
capacitor.mp3	/home/pi/MP34DMM/capacitor.mp3	โหมควัดตัวเก็บประจุ
continuity.mp3	/home/pi/MP34DMM/continuity.mp3	โหมควัดความต่อเนื่องแบบเสียง
current.mp3	/home/pi/MP34DMM/current.mp3	โหมควัดกระแสไฟฟ้า
DC.mp3	/home/pi/MP34DMM/DC.mp3	กระแสตรง
delete.mp3	/home/pi/MP34DMM/delete.mp3	ลบ

ตารางที่ 3.10 (ต่อ)

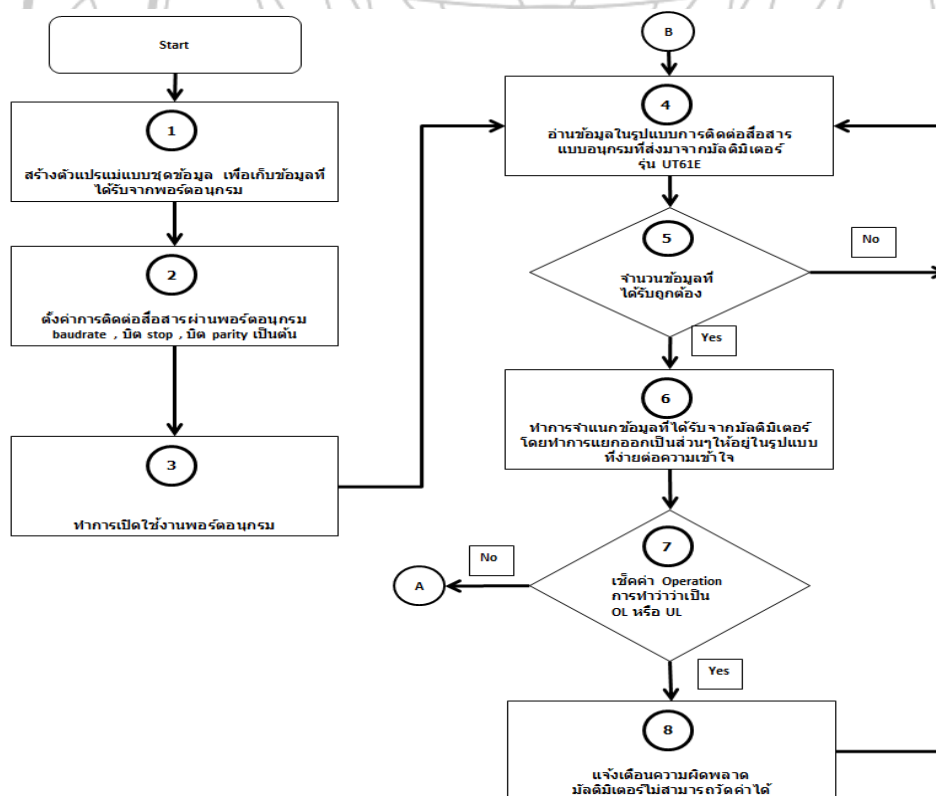
ชื่อไฟล์	ตำแหน่งที่จัดเก็บ	ความหมายเสียง
diode.mp3	/home/pi/MP34DMM/diode.mp3	ไดโอด
Duty_cycle.mp3	/home/pi/MP34DMM/duty_cycle.mp3	โหมควัดคิวตี้ไซเคิลความถี่
Ed.mp3	/home/pi/MP34DMM/Ed.mp3	เอ็ด
farad.mp3	/home/pi/MP34DMM/farad.mp3	ฟารัด
frequency.mp3	/home/pi/MP34DMM/frequency.mp3	โหมควัดความถี่
Hz.mp3	/home/pi/MP34DMM/Hz.mp3	เฮิร์ตซ์
kilo.mp3	/home/pi/MP34DMM/kilo.mp3	กิโล
Mega.mp3	/home/pi/MP34DMM/Mega.mp3	เมกะ
micro.mp3	/home/pi/MP34DMM/micro.mp3	ไมโคร
mili.mp3	/home/pi/MP34DMM/mili.mp3	มิลิ
nano.mp3	/home/pi/MP34DMM/nano.mp3	นาโน
ohm.mp3	/home/pi/MP34DMM/ohm.mp3	โอห์ม
overload.mp3	/home/pi/MP34DMM/overload.mp3	มัลติมิเตอร์ไม่สามารถวัดค่าได้
percent.mp3	/home/pi/MP34DMM/percent.mp3	เปอร์เซ็นต์
Pico.mp3	/home/pi/MP34DMM/ Pico.mp3	พิโก
point.mp3	/home/pi/MP34DMM/ point.mp3	เปอร์เซ็นต์
resistance.mp3	/home/pi/MP34DMM/ resistance.mp3	โหมควัดความต้านทาน
volt.mp3	/home/pi/MP34DMM/ volt.mp3	โวลท์
voltage.mp3	/home/pi/MP34DMM/ voltage.mp3	โหมควัดแรงดันไฟฟ้า

การเขียนโปรแกรมภาษา Python

การเขียนโปรแกรมภาษา Python เพื่ออ่านค่าจาก มัลติมิเตอร์รุ่น UT61E เพื่อสร้างเครื่องมือวัดทางไฟฟ้าแสดงผลด้วยเสียงนั้น จะมีขั้นตอนที่สอดคล้องและอ้างอิงกับรูปภาพที่ 3.7 ขั้นตอนการทำงานของโปรแกรมส่วน A และรูปภาพที่ 3.8 ขั้นตอนการทำงานของโปรแกรมส่วน B มีความสัมพันธ์กับชุดโปรแกรมภาษา Python ในภาคผนวก ค โปรแกรมภาษา Python สำหรับอ่านค่าจากมัลติมิเตอร์ โดยมีลำดับขั้นตอนดังต่อไปนี้

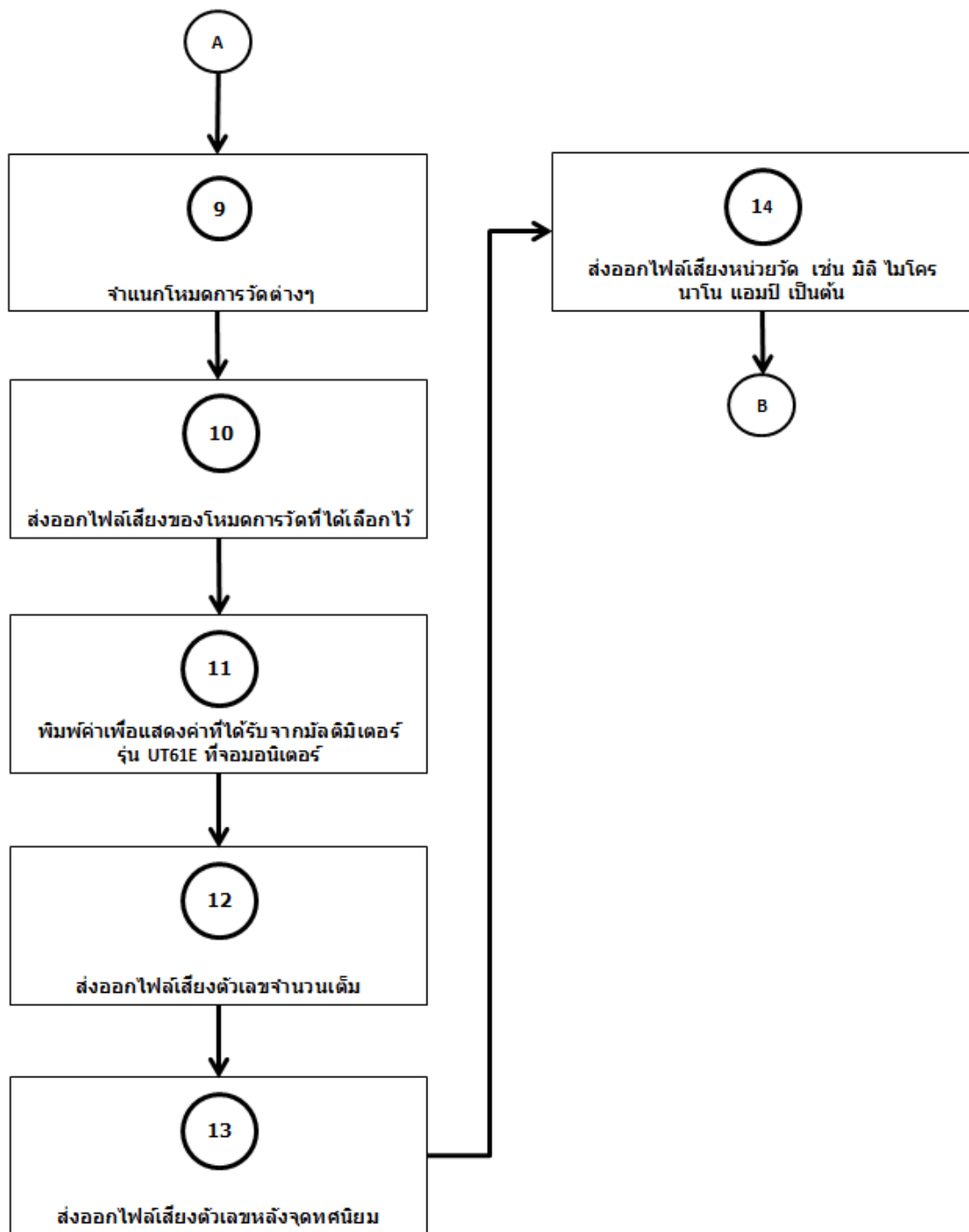
1. เริ่มต้นจากรูปภาพที่ 3.7 ขั้นตอนหมายเลข 1 การสร้างตัวแปรแม่แบบชุดข้อมูล เพื่อเก็บข้อมูลที่รับจากพอร์ตอนุกรม ขั้นตอนนี้ได้สร้างตัวแปรเป็นแม่แบบสำหรับใช้เก็บข้อมูลที่รับจากตัวมัลติมิเตอร์ ตัวแปรที่สร้างขึ้น อาทิเช่น ตัวแปร STATUS ตัวแปร OPTION1- OPTION4 ตัวแปร FUNCTION ตัวแปร RANGE_FREQUENCY เป็นต้น ซึ่งมีความสอดคล้องกับชุดโปรแกรมในภาคผนวก ก ตัวอย่างเช่น ตัวแปร STATUS ในบรรทัดที่ 144 และตัวแปร OPTION1 ในบรรทัดที่ 152 เป็นต้น

2. ขั้นตอนหมายเลข 2 ในรูปภาพที่ 3.7 เป็นการตั้งค่าการติดต่อสื่อสารผ่านพอร์ตอนุกรมของบอร์ด Raspberry Pi โดยจะต้องตั้งให้อยู่ในรูปแบบของตัวเลขที่กำหนดไว้เท่านั้น จึงจะสามารถทำการติดต่อสื่อสารกับตัวมัลติมิเตอร์ได้ จากชุดโปรแกรมในภาคผนวก ก ตั้งแต่บรรทัดที่ 420 ถึงบรรทัดที่ 425 จะเป็นชุดโปรแกรมตั้งค่าการติดต่อสื่อสารอนุกรม มีรายละเอียดดังนี้ คือ การตั้งค่า Baudrate เป็น 19200 มีค่า Stop bit เท่ากับ 1 บิต มีบิตข้อมูล 7 บิต และมีค่า parity เท่ากับ odd เป็นต้น



ภาพที่ 3.9 ขั้นตอนการทำงานของโปรแกรมส่วน A

ที่มา: นิเทศ ประเดชบุญ



ภาพที่ 3.10 ขั้นตอนการทำงานของโปรแกรมส่วน B

ที่มา: นิเทศ ประเดชญญ

3. หลังจากที่ทำกาที่ตั้งค่าการติดต่อสื่อสารพื้นฐานไปแล้ว ขั้นตอนหมายเลข 3 ในรูปภาพที่ 3.7 จะเป็นการทำการเปิดพอร์ตอนุกรมและทำการตั้งค่าขาสัญญาณ DTR และขาสัญญาณ RTS โดยข้อกำหนดด้านวงจรของการรับส่งอยู่ว่า ตัวขาสัญญาณ DTR จะต้องตั้งค่าลอจิกเป็น 1 และขาสัญญาณ RTS ต้องตั้งค่าลอจิกเป็น 0 การติดต่อสื่อสารระหว่างบอร์ด Raspberry Pi และมัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E ถึงจะเกิดขึ้น โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 429 ถึงบรรทัดที่ 430

4. ขั้นตอนหมายเลข 4 ในรูปที่ 3.7 เป็นขั้นตอนในการอ่านค่าจากพอร์ตอนุกรมโดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 436

5. ขั้นตอนหมายเลข 5 ในรูปที่ 3.7 เป็นขั้นตอนการตรวจเช็คจำนวนของข้อมูลที่เข้ามาจะต้องมีความยาว 12 ข้อมูล เพื่อตรวจสอบว่าข้อมูลที่ได้รับเข้ามาสมบูรณ์ และพร้อมจะนำไปวิเคราะห์ในขั้นตอนต่อไป โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 439

6. ขั้นตอนหมายเลข 6 ในรูปที่ 3.7 เป็นการจำแนกข้อมูลที่ได้รับมาจากมัลติมิเตอร์ โดยทำการแยกออกเป็นส่วน ๆ เพื่อให้ง่ายต่อความเข้าใจและวิเคราะห์ โดยในขั้นตอนนี้มีการเรียกใช้งานชุดฟังก์ชัน `parse()` ซึ่งภายในชุดโปรแกรมจะแตกข้อมูลออกเป็นก่อน ๆ ให้กับตัวแปรที่ได้ทำการประกาศไว้เพื่อเป็นตัวแทนของข้อมูลนั้น ๆ อาทิเช่น ข้อมูล STATUS ข้อมูล FUNCTION และอื่น ๆ เป็นต้น โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 442 เป็นการเรียกใช้งานฟังก์ชัน `parse()` และส่งออกค่าให้ตัวแปร `results` ส่วนตัวฟังก์ชัน `parse()` ถูกเขียนไว้อยู่ในบรรทัดที่ 185 ถึงบรรทัดที่ 283

7. ขั้นตอนหมายเลข 7 ในรูปที่ 3.7 เป็นการตรวจสอบว่ามัลติมิเตอร์อยู่ในสภาวะพร้อมใช้งานหรือไม่ หรืออยู่ในสภาวะที่ไม่สามารถวัดค่าได้ คือ สภาวะ Overload และสภาวะ Underload โดยตรวจเช็คตัวแปร `results['operation']` ว่าอยู่ในสภาวะดังกล่าวข้างต้นหรือไม่ โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 445

8. ขั้นตอนหมายเลข 8 ในรูปที่ 3.7 จะเป็นการแจ้งเตือนว่า ระบบไม่สามารถอ่านค่าได้เป็นเสี่ยงพูด ขั้นตอนนี้จะเกิดขึ้นก็ต่อเมื่อ ตัวแปร `results['operation']` อยู่ในสภาวะ Overload หรือ Underload โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค ตั้งแต่บรรทัดที่ 447 ถึงบรรทัดที่ 453

9. ขั้นตอนหมายเลข 9 ในรูปที่ 3.8 เป็นการจำแนกโหมดการวัด ว่าตอนนี้มัลติมิเตอร์กำลังใช้งานอยู่ในโหมดไหนอยู่ เช่น โหมดวัดกระแสไฟฟ้า โหมดวัดแรงดันไฟฟ้า โหมดวัดความต้านทาน โหมดวัดความถี่และโหมดวัดความจุ เป็นต้น โดยในการจำแนกโหมดจะทำการเรียกใช้งานฟังก์ชัน `output_csv()` โดยอ้างอิงจากชุดโปรแกรมในภาคผนวก ค บรรทัดที่ 455 เป็นการเรียกใช้งานฟังก์ชัน ส่วนตัวฟังก์ชัน `output_csv()` จะถูกเขียนไว้อยู่ในบรรทัดที่ 317 ถึงบรรทัดที่ 320

10. ขั้นตอนหมายเลข 10 ในรูปที่ 3.8 เป็นการส่งออกไฟล์เสียงของโหมดที่ใช้งานอยู่ในขณะนั้นออกมา โดยทำการโหลดไฟล์เสียงที่ถูกบันทึกไว้ในตำแหน่งเสียงของโหมดต่าง ๆ เข้าไปเก็บไว้ในตัวแปร ชื่อ `dir_sound` และจากนั้นก็ส่งตัวแปรให้กับโมดูลฟังก์ชัน `mixer.music.play()` ทำการเรียกใช้ไฟล์เสียงเพื่อส่งออกทางลำโพงต่อไป ขั้นตอนนี้ได้อ้างอิงจากชุดโปรแกรมในภาคผนวก ค. บรรทัดที่ 457 ถึงบรรทัดที่ 464 ซึ่งจะมีการหน่วงสัญญาณเสียงไว้ 3 วินาที ด้วยโมดูล `time` ฟังก์ชัน `sleep()` ในบรรทัดที่ 465

11. ขั้นตอนหมายเลข 11 ในรูปที่ 3.8 จะเป็นขั้นตอนในการพิมพ์ค่าที่ได้รับจากมัลติมิเตอร์เพื่อแจ้งให้ผู้พัฒนาทราบถึงข้อมูลที่รับเข้ามา เป็นการตรวจสอบการเขียนโปรแกรมเป็นอย่างดี โดยใช้ฟังก์ชัน `print()` และในขั้นตอนนี้ก็มีการเรียกใช้งานฟังก์ชัน `output_readable()` ที่ทำการส่งค่าออกมา 2 ค่า คือ ค่าที่อ่านได้จากมัลติมิเตอร์ขณะนั้น กับ ค่าของหน่วยวัดขณะนั้น ขั้นตอนนี้ได้อ้างอิงจากชุดโปรแกรมในภาคผนวก ค. บรรทัดที่ 467 ถึงบรรทัดที่ 470 และตัวฟังก์ชัน `output_readable()` ถูกเขียนให้เรียกใช้งานอยู่ในบรรทัดที่ 285 ถึงบรรทัดที่ 297

12. ขั้นตอนหมายเลข 12 ในรูปที่ 3.8 เป็นการส่งออกไฟล์เสียงของเลขจำนวนเต็มที่ถูกเก็บไว้ในตัวแปร ชื่อ `UT61E_DMM` และทำการแตกออกด้วยคำสั่ง `split()` จากนั้นก็จะนำค่าที่ทำการแตกเก็บไว้ในตัวแปร ชื่อ `Decimal_Number` และทศนิยมเก็บไว้ในตัวแปร ชื่อ `Float_Number` จากนั้นก็จะดำเนินการกับตัวแปร `Decimal_Number` เพื่อแยกตัวเลขออกเป็นหลักหน่วย หลักสิบ หลักร้อย และหลักพัน ขั้นตอนนี้ได้อ้างอิงจากชุดโปรแกรมในภาคผนวก ค. บรรทัดที่ 469 ถึงบรรทัดที่ 510 จากนั้นก็จะทำการส่งไฟล์เสียงออกทางลำโพงต่อไป โดยเรียกใช้ `mixer.music.play()` เพื่อเล่นไฟล์เสียง ดังแสดงในบรรทัดที่ 511

13. ขั้นตอนหมายเลข 13 ในรูปที่ 3.8 ดำเนินการกับตัวแปร `Float_Number` ในขั้นตอนที่ผ่านมาเพื่อแยกตัวเลขออกเป็นทีละตัว และทำการส่งค่าตัวเลขที่ได้ให้กับฟังก์ชัน `mixer.music.play()` เพื่อเล่นไฟล์เสียง ดังแสดงในบรรทัดที่ 517

14. ขั้นตอนหมายเลข 14 ในรูปที่ 3.8 เป็นการส่งออกไฟล์เสียงของหน่วยวัด อาทิเช่น ไมโคร นาโน เมกะ โอห์ม และอื่น ๆ เป็นต้น เพื่อให้ผู้ใช้ได้ทราบถึงหน่วยและค่าของสัญญาณที่กำลังวัดอยู่ได้อย่างถูกต้อง โดยทำการเปรียบเทียบและค้นหาเสียงได้จากตัวแปรชื่อ `UNIT_SOUNDS[]` ซึ่งได้รวบรวมหน่วยวัดไว้ ขั้นตอนนี้ได้อ้างอิงจากชุดโปรแกรมในภาคผนวก ค. บรรทัดที่ 519 ถึงบรรทัดที่ 523 เมื่อทำการส่งออกไฟล์เสียงของหน่วยวัดเสร็จสิ้น โปรแกรมภาษา Python จะวนกลับไปอ่านค่าพอร์ตคอนูกรมตลอด โดยมีการหน่วงเวลาของแต่ละรอบการอ่านไว้ที่ 3 วินาที ดังแสดงชุดโปรแกรมในภาคผนวก ค. ในบรรทัดที่ 524

การทำงานของระบบเครื่องวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด

ผู้วิจัยออกแบบการทำงานของระบบโดยใช้วงจรของมัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E เป็นตัววงจรหลักในชุดของระบบการวัดทั้งหมด อาทิเช่น การวัดแรงดันไฟฟ้า การวัดกระแสไฟฟ้า การวัดตัวต้านทาน การวัดตัวเก็บประจุ และการวัดความถี่ ซึ่งได้ทำการติดต่อสื่อสารกับตัวบอร์ด Raspberry Pi ที่ได้ติดตั้งระบบปฏิบัติการลินุกซ์ (Linux) และโปรแกรมภาษาไพธอน (Python) ไว้ผ่านทางตัวแปลงสัญญาณอนุกรม RS232 เป็นพอร์ต USB 2.0 โดยใช้โมดูลแปลงสัญญาณ รุ่น ET-USB/RS232 MINI ทำการรับค่าข้อมูลจากการวัดค่าตามโหมดที่ได้เลือกไว้ที่ตัวมัลติมิเตอร์ รุ่น UT61E ซึ่งสามารถเลือกโหมดการวัดจากสวิตช์ตัวเลือก (Selector Switch) ที่ถูกติดตั้งไว้ที่หน้าปัดตัวเครื่อง จากนั้นโปรแกรมภาษาไพธอน (Python) ที่ผู้วิจัยได้เขียนไว้จะทำการจำแนกสัญญาณออกเป็นส่วน ๆ เพื่อให้ง่ายต่อการวิเคราะห์ระบบต่าง ๆ และทำการเรียบเรียงเสียงพูดตามลำดับของข้อมูลที่ถูกจำแนกและส่งเสียงนั้นออกมาทางพอร์ตลำโพง 3.5 mm ของบอร์ด Raspberry Pi ให้กับลำโพงขนาดเล็กที่ได้รับแหล่งจ่ายไฟฟ้าจากพอร์ต USB 2.0 ของบอร์ด Raspberry Pi ทำให้ผู้ใช้ได้รับทราบโหมดการวัดและค่าที่กำลังวัดได้เป็นระยะ ๆ ตลอดเวลา

การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียงในโหมดการวัดแรงดันไฟฟ้ากระแสตรง

การวัดแรงดันไฟฟ้ากระแสตรง ผู้วิจัยใช้เครื่องจ่ายไฟฟ้ากระแสตรงปรับค่าได้ ตราสินค้า Skytoppower รุ่น STP3005 ซึ่งเป็นแหล่งจ่ายแรงดันไฟฟ้าแบบปรับค่าได้ระหว่าง 0-30V และสามารถจ่ายกระแสได้สูงถึง 5 แอมป์ ดังแสดงในภาพที่ 3.9 โดยจ่ายแรงดันออกเป็น 5 ค่าและบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าแรงดันที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi ดังตารางที่ 3.11

ตารางที่ 3.11 ความสัมพันธ์ของผลการวัดแรงดันไฟฟ้ากระแสตรง

แรงดันไฟฟ้า กระแสตรง	มัลติมิเตอร์ รุ่น UT61E	ที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
5V	5.001V	5.001V	5.001V	5.001V	5.001V
15V	15.02V	15.02V	15.02V	15.02V	15.02V
20V	19.941V	19.941V	19.94V	19.94V	19.94V
25V	24.82V	24.82V	24.82V	24.82V	24.82V
30V	29.81V	29.81V	29.81V	29.81V	29.81V

จากนั้นผู้วิจัยได้เขียนโปรแกรมโดยใช้ภาษาไพธอนเพื่อจำแนกค่าข้อมูลที่ได้รับจากจากมัลติมิเตอร์ รุ่น UT61E ทางพอร์ตอนุกรม และเรียบเรียงไฟล์เสียงตามลำดับตามข้อมูลที่จำแนกได้ แปลงสัญญาณจากพอร์ตอนุกรมเป็นพอร์ต USB ซึ่งใช้โมดูล ET-USB/RS232 MINI เสียบเข้าที่พอร์ต USB 2.0 ของบอร์ด Raspberry Pi และส่งออกสัญญาณเสียงทางช่องลำโพงขนาด 3.5mm ของบอร์ด ซึ่งได้เสียบติดตั้งตัวลำโพง เพื่อแสดงผลออกเสียงพูดให้ผู้ใช้งานได้รับทราบจากการทดลองทำให้ทราบว่าค่าแสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เมื่อทำการเปรียบเทียบกับค่าที่ได้รับของบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0

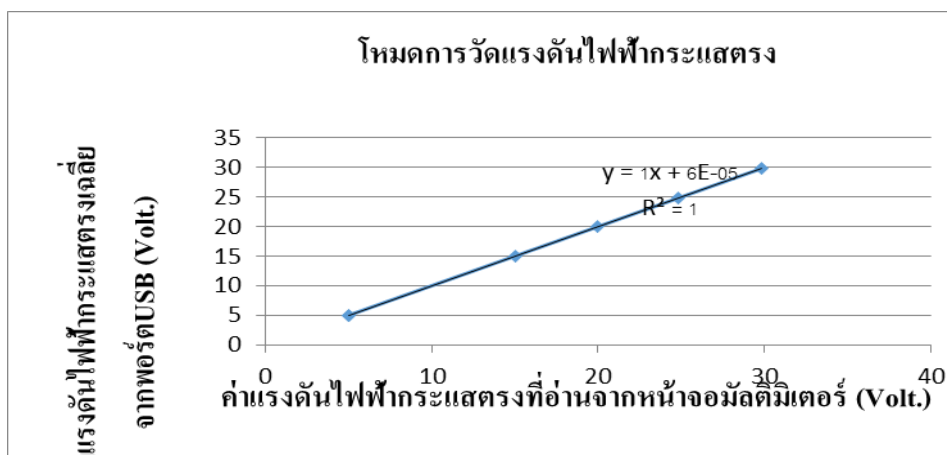


ภาพที่ 3.11 การต่ออุปกรณ์วัดแรงดันไฟฟ้ากระแสตรง
ที่มา: นิเทศ ประเดชญญ

การเชื่อมต่ออุปกรณ์สำหรับวัดแรงดันไฟฟ้ากระแสตรง แสดงดังภาพที่ 3.9 ซึ่งมีการจำแนกข้อมูลและเรียบเรียงลำดับไฟล์เสียงที่วัดได้จากแรงดันทดสอบ แสดงดังภาพที่ 3.10

```
pi@raspberrypi:~ $ sudo python3 UT61E_DMM.py
การจำแนกข้อมูลระบบ : ['5.001', 'V', 'voltage', 'DC', 'normal']
ส่งออกไฟล์เสียงโหมดการวัด : ['/home/pi/MP34DMM/voltage.mp3', '/home/pi/MP34DMM/DC.mp3']
สถานะการทำงานของเครื่อง : "normal"
ค่าที่วัดได้จาก UT61E : 5.001 V
ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : ['5.001', 'V']
ส่งออกไฟล์เสียงจำนวนเต็ม : ['/home/pi/MP34DMM/5.mp3', '/home/pi/MP34DMM/point.mp3']
ส่งออกไฟล์เสียงหลังจุดทศนิยม : ['/home/pi/MP34DMM/0.mp3', '/home/pi/MP34DMM/0.mp3', '/home/pi/MP34DMM/1.mp3']
ส่งออกไฟล์เสียงหน่วยวัด : ['/home/pi/MP34DMM/volt.mp3']
```

ภาพที่ 3.12 ผลค่าแรงดันไฟฟ้าและการเรียงลำดับไฟล์เสียงที่แรงดันทดสอบ 5V
ที่มา: นิเทศ ประเดชญญ



ภาพที่ 3.13 กราฟเปรียบเทียบระหว่างค่าแรงดันไฟฟ้าจากหน้าจอมัลติมิเตอร์กับค่าแรงดันไฟฟ้าเฉลี่ยจากพอร์ต USB

ที่มา: นิเทศ ประเดชบุญ

การวัดแรงดันไฟฟ้ากระแสตรง ผู้วิจัยได้ใช้เครื่องจ่ายไฟฟ้ากระแสตรงปรับค่าได้ โดยปรับแรงดันไฟฟ้าออกเป็น 6 ค่า และบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าแรงดันที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi และได้ทำการสร้างกราฟ เพื่อเปรียบเทียบแรงดันไฟฟ้าจากหน้าจอมัลติมิเตอร์กับค่าแรงดันไฟฟ้าเฉลี่ยจากพอร์ต USB ดังภาพที่ 3.11 พบว่ามีค่าตรงกัน

การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียงในโหมดการวัดความต้านทาน

ในโหมดการวัดความต้านทาน ผู้วิจัยได้ใช้ตัวต้านทานในการทดสอบแบ่งออกเป็น 8 ค่า ดังตารางที่ 3.12 และทำการบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าความต้านทานที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi

ตารางที่ 3.12 ความสัมพันธ์ของผลการวัดความต้านทานไฟฟ้า

ค่าความต้านทาน	มัลติมิเตอร์ รุ่น UT61E	ค่าที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
3 Ω	2.84 Ω	2.85 Ω	2.84 Ω	2.84 Ω	2.84 Ω
10 Ω	9.98 Ω	9.99 Ω	9.98 Ω	9.98 Ω	9.98 Ω

ตารางที่ 3.12 (ต่อ)

ค่าความ ต้านทาน	มัลติมิเตอร์ รุ่น UT61E	ค่าที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
20 Ω	20.06 Ω	20.06 Ω	20.06 Ω	20.06 Ω	20.06 Ω
100 Ω	98.78 Ω	98.78 Ω	98.77 Ω	98.78 Ω	98.78 Ω
500 Ω	0.4902 k Ω	0.4902 k Ω	0.4902 k Ω	0.4902 k Ω	0.4902 k Ω
1 k Ω	0.9849 k Ω	0.9849 k Ω	0.9850 k Ω	0.9848 k Ω	0.9849 k Ω
20 k Ω	20.11 k Ω	20.11 k Ω	20.11 k Ω	20.11 k Ω	20.11 k Ω
0.5 M Ω	0.5008 M Ω	0.5008 M Ω	0.5006 M Ω	0.5011 M Ω	0.5008 M Ω

จากนั้นผู้วิจัยได้เขียนโปรแกรมแสดงค่าของไฟล์เสียงเรียงตามลำดับตามข้อมูลของการวัดที่ได้รับจากมัลติมิเตอร์ รุ่น UT61E ทางพอร์ตอนุกรม โดยแปลงสัญญาณจากพอร์ตอนุกรมเป็นพอร์ต USB ซึ่งใช้โมดูล ET-USB/RS232 MINI เสียบเข้าที่พอร์ต USB 2.0 ของบอร์ด Raspberry Pi และส่งออกสัญญาณเสียงทางช่องลำโพงขนาด 3.5mm ของบอร์ด ซึ่งได้เสียบติดตั้งตัวลำโพง เพื่อแสดงผลออกเสียงพูดให้ผู้ใช้งานได้รับทราบ จากการทดลองทำให้ทราบว่าค่าแสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เมื่อเปรียบเทียบกับค่าที่ได้รับของบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0

ภาพที่ 3.14 การวัดความต้านทานขนาด 0.5M Ω

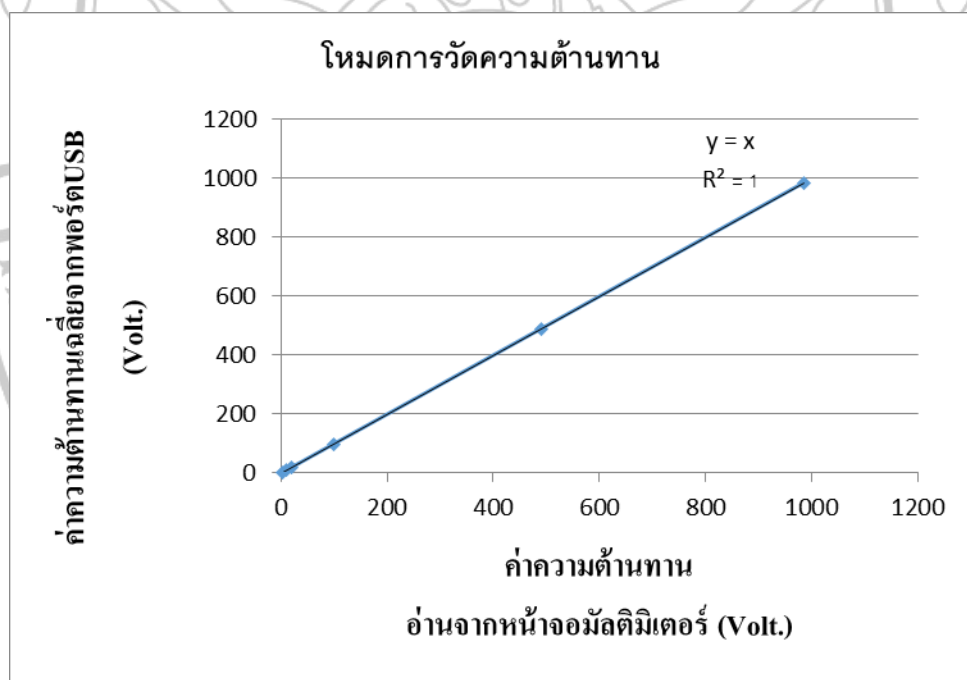
ที่มา: นิเทศ ประเดชญ

การเชื่อมต่ออุปกรณ์สำหรับวัดค่าความต้านทาน แสดงดังภาพที่ 3.12 ซึ่งมีการจำแนกข้อมูลและเรียบเรียงลำดับไฟล์เสียงที่วัดได้จากความต้านทานทดสอบ แสดงดังภาพที่ 3.13

```
pi@raspberrypi:~ $ sudo python3 UT61E_DMM.py
การจำแนกข้อมูลระบบ : ['500900.0', 'Ohm', 'resistance', '', 'normal']
ส่งออกไฟล์เสียงโหมดการวัด : ['/home/pi/MP34DMM/resistance.mp3']
สถานะการทำงานของเครื่อง : "normal"
ค่าที่วัดได้จาก UT61E : 0.5009 MO
ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : ['0.5009', 'MO']
ส่งออกไฟล์เสียงจำนวนเต็ม : ['/home/pi/MP34DMM/0.mp3', '/home/pi/MP34DMM/point.mp3']
ส่งออกไฟล์เสียงหลังจุดทศนิยม : ['/home/pi/MP34DMM/5.mp3', '/home/pi/MP34DMM/0.mp3',
'/home/pi/MP34DMM/0.mp3', '/home/pi/MP34DMM/9.mp3']
ส่งออกไฟล์เสียงหน่วยวัด : ['/home/pi/MP34DMM/Mega.mp3', '/home/pi/MP34DMM/ohm.mp3']
```

ภาพที่ 3.15 ผลค่าความต้านทานขนาด $0.5\text{M}\Omega$ และการเรียงลำดับไฟล์เสียง

ที่มา: นิเทศ ประเดชญญ



ภาพที่ 3.16 กราฟเปรียบเทียบระหว่างค่าความต้านทานจากหน้าจอมัลติมิเตอร์กับค่าความต้านทานเฉลี่ยจากพอร์ต USB

ที่มา: นิเทศ ประเดชญญ

ในโหมดการวัดความต้านทาน ผู้วิจัยได้ใช้ตัวต้านทานในการทดสอบแบ่งออกเป็น 4 ค่า ได้แก่ $0.5k\Omega$ $1k\Omega$ $20k\Omega$ และ $0.5M\Omega$ แล้วทำการบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของ มัลติมิเตอร์ เปรียบเทียบกับค่าความต้านทานที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi พบว่ามีค่าตรงกันดังแสดงไว้ในภาพที่ 3.14

การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียงใน โหมดการวัดตัวเก็บประจุ

ในโหมดการวัดตัวเก็บประจุ ผู้วิจัยได้ใช้ตัวเก็บประจุในการทดสอบแบ่งออกเป็น 3 ค่า ดังตารางที่ 3.13 และทำการบันทึกผลการทดลองจากค่าที่แสดงผลของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าตัวเก็บประจุที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi

ตารางที่ 3.13 ตารางความสัมพันธ์ของผลการวัดตัวเก็บประจุ

ค่าตัวเก็บ ประจุ	มัลติมิเตอร์ รุ่น UT61E	ค่าที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
0.1nF	0.406nF	0.406nF	0.406nF	0.406nF	0.406nF
1nF	1.313nF	1.313nF	1.313nF	1.313nF	1.313nF
2.2nF	2.568nF	2.568nF	2.570nF	2.570nF	2.569nF
100nF	102.38nF	102.38nF	102.67nF	102.32nF	102.46nF
1uF	1.0422nF	1.0422nF	1.0396nF	1.0423nF	1.0414nF
10uF	10.228nF	10.228nF	10.229nF	10.228nF	10.228nF
100uF	98.13nF	98.13nF	98.09nF	98.06nF	98.09nF
470uF	0.4321mF	0.4321mF	0.4320mF	0.4321mF	0.4321mF
2200uF	2.0687mF	2.0687mF	2.0691mF	2.0716mF	2.0698mF

จากนั้นผู้วิจัยได้เขียนโปรแกรมแสดงค่าของไฟล์เสียงเรียงตามลำดับตามข้อมูลของการวัดที่ได้รับจากมัลติมิเตอร์ รุ่น UT61E ทางพอร์ตอนุกรม โดยแปลงสัญญาณจากพอร์ตอนุกรมเป็นพอร์ต USB ซึ่งใช้โมดูล ET-USB/RS232 MINI เสียบเข้ากับพอร์ต USB 2.0 ของบอร์ด Raspberry Pi และส่งออกสัญญาณเสียงทางช่องลำโพงขนาด 3.5mm ของบอร์ด ซึ่งได้เสียบติดตั้งตัวลำโพง

เพื่อแสดงผลออกเสียงพูดให้ผู้ใช้งานได้รับทราบ จากการทดลองทำให้ทราบว่าค่าแสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เมื่อทำการเปรียบเทียบกับค่าที่ได้รับของบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0



ภาพที่ 3.17 การวัดตัวเก็บประจุขนาด 2200uF

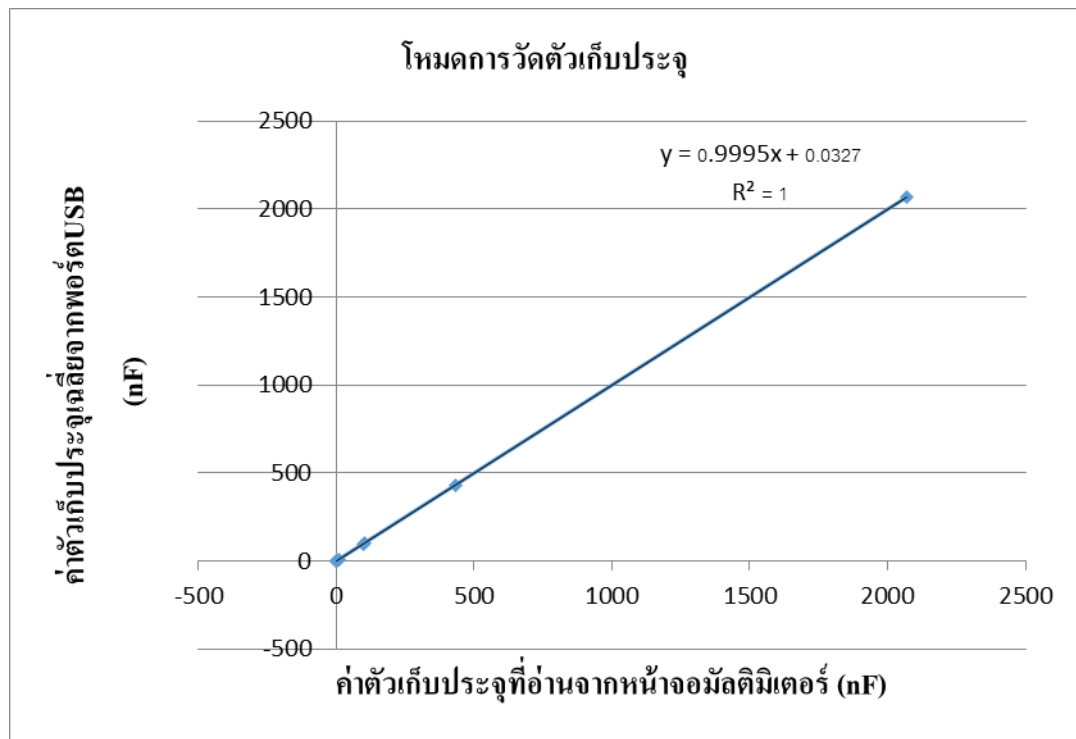
ที่มา: นิเทศ ประเดชญ

การเชื่อมต่ออุปกรณ์สำหรับวัดค่าตัวเก็บประจุไฟฟ้า แสดงดังภาพที่ 3.15 ซึ่งมีการจำแนกข้อมูลและเรียบเรียงลำดับไฟล์เสียงที่วัดได้จากตัวเก็บประจุทดสอบ แสดงดังภาพที่ 3.16

```
pi@raspberrypi:~ $ sudo python3 UT61E_DMM.py
การจำแนกข้อมูลระบบ : ['0.0020687', 'F', 'capacitance', '', 'normal']
ส่งออกไฟล์เสียงโหมดการวัด : ['/home/pi/MP34DMM/capacitor.mp3']
สถานะการทำงานของเครื่อง : "normal"
ค่าที่วัดได้จาก UT61E : 2.0687 mF
ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : ['2.0687', 'mF']
ส่งออกไฟล์เสียงจำนวนเต็ม : ['/home/pi/MP34DMM/2.mp3', '/home/pi/MP34DMM/point.mp3']
ส่งออกไฟล์เสียงหลังจุดทศนิยม : ['/home/pi/MP34DMM/0.mp3', '/home/pi/MP34DMM/6.mp3',
'/home/pi/MP34DMM/8.mp3', '/home/pi/MP34DMM/7.mp3']
ส่งออกไฟล์เสียงหน่วยวัด : ['/home/pi/MP34DMM/mili.mp3', '/home/pi/MP34DMM/farad.mp3']
```

ภาพที่ 3.18 การแสดงผลค่าตัวเก็บประจุขนาด 2200uF และการเรียงลำดับไฟล์เสียง

ที่มา: นิเทศ ประเดชญ



ภาพที่ 3.19 กราฟเปรียบเทียบระหว่างค่าตัวเก็บประจุจากหน้าจอมัลติมิเตอร์กับ
ค่าตัวเก็บประจุเฉลี่ยจากพอร์ต USB
ที่มา: นิเทศ ประเดชญญ

การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียงพูด ใน
โหมดการวัดตัวเก็บประจุ ผู้วิจัยได้ใช้ตัวเก็บประจุในการทดสอบแบ่งออกเป็น 9 ค่า ดังนี้คือ 0.1nF,
1nF, 2.2nF, 100nF, 1uF, 10uF, 100uF, 470uF และ 2200uF แล้วทำการบันทึกผลการทดลองจาก
ค่าที่แสดงผลของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าความตัวเก็บประจุที่ได้รับจากพอร์ต
USB 2.0 ของ Raspberry Pi ดังภาพที่ 3.17 พบว่ามีค่าตรงกัน

**การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียง ใน
โหมดการวัดความถี่**

ในโหมดการวัดความถี่ ผู้วิจัยได้ใช้เครื่องสร้างสัญญาณความถี่ (DDS signal Generator)
รุ่น KMOON FY6800 ในการจำลองสัญญาณความถี่เพื่อการทดสอบระบบ ดังตารางที่ 3.14 และทำ
การบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับ
ค่าความถี่ที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi

ตารางที่ 3.14 ตารางความสัมพันธ์ของผลการวัดความถี่ไฟฟ้า

ความถี่	มัลติมิเตอร์รุ่น UT61E	ค่าที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
50Hz	50.0Hz	50.0Hz	50.0Hz	50.0Hz	50.0Hz
100Hz	100.0Hz	100.0Hz	100.0Hz	100.0Hz	100.0Hz
10kHz	9.999kHz	9.999kHz	9.999kHz	9.999kHz	9.999kHz
20kHz	20.00kHz	20.00kHz	20.00kHz	20.00kHz	20.00kHz
30kHz	30.00kHz	30.00kHz	30.00kHz	30.00kHz	30.00kHz
10MHz	9.999MHz	9.999MHz	9.999MHz	9.999MHz	9.999MHz
20MHz	19.998MHz	19.998MHz	19.999MHz	19.999MHz	19.999MHz

จากนั้นผู้วิจัยได้เขียนโปรแกรมแสดงค่าของไฟล์เสียงเรียงตามลำดับตามข้อมูลของการวัดที่ได้รับจากมัลติมิเตอร์ รุ่น UT61E ทางพอร์ตอนุกรม โดยได้ทำการแปลงสัญญาณจากพอร์ตอนุกรมเป็นพอร์ต USB ซึ่งใช้โมดูล ET-USB/RS232 MINI เสียบเข้ากับพอร์ต USB 2.0 ของบอร์ด Raspberry Pi และทำการส่งออกสัญญาณเสียงทางช่องลำโพงขนาด 3.5mm ของบอร์ด ซึ่งได้เสียบติดตั้งตัวลำโพง เพื่อแสดงผลออกเสียงพูดให้ผู้ใช้งานได้รับทราบ จากการทดลองทำให้ทราบว่าค่าแสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เมื่อทำการเปรียบเทียบกับค่าที่ได้รับของบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0



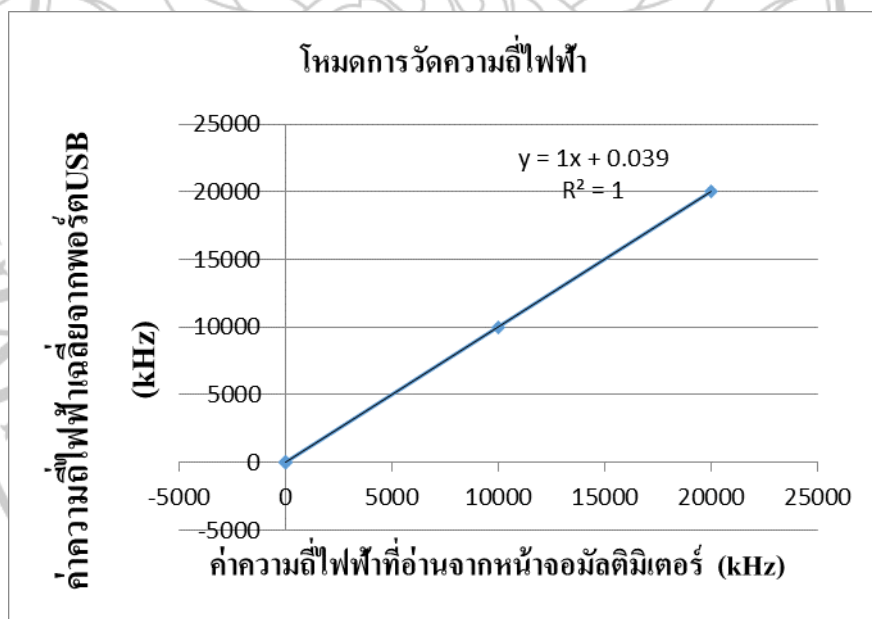
ภาพที่ 3.20 การทดสอบวัดความถี่ไฟฟ้า

ที่มา: นิเทศ ประเดชญญ

การเชื่อมต่ออุปกรณ์สำหรับวัดความถี่ไฟฟ้า แสดงดังภาพที่ 3.18 ซึ่งมีการจำแนกข้อมูล และเรียบเรียงลำดับไฟล์เสียงที่วัดได้จากความถี่ทดสอบ แสดงดังภาพที่ 3.19

```
pi@raspberrypi:~ $ sudo python3 UT61E_DMM.py
การจำแนกข้อมูลระบบ : ['19998000.0', 'Hz', 'frequency', '', 'normal']
ส่งออกไฟล์เสียงโหมดการวัด : ['/home/pi/MP34DMM/frequency.mp3']
สถานะการทำงานของเครื่อง : "normal"
ค่าที่วัดได้จาก UT61E : 19.998 MH
ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : ['19.998', 'MH']
ส่งออกไฟล์เสียงจำนวนเต็ม : ['/home/pi/MP34DMM/10.mp3', '/home/pi/MP34DMM/9.mp3',
'/home/pi/MP34DMM/point.mp3']
ส่งออกไฟล์เสียงหลังจุดทศนิยม : ['/home/pi/MP34DMM/9.mp3', '/home/pi/MP34DMM/9.mp3',
'/home/pi/MP34DMM/8.mp3']
ส่งออกไฟล์เสียงหน่วยวัด : ['/home/pi/MP34DMM/Mega.mp3', '/home/pi/MP34DMM/Hz.mp3']
```

ภาพที่ 3.21 การแสดงผลค่าวัดความถี่ 20MHz และการเรียงลำดับไฟล์เสียง
ที่มา: นิเทศ ประเดชญญ



ภาพที่ 3.22 กราฟเปรียบเทียบระหว่างค่าความถี่ไฟฟ้าจากหน้าจอมัลติมิเตอร์กับ
ค่าความถี่ไฟฟ้าเฉลี่ยจากพอร์ต USB

ที่มา: นิเทศ ประเดชญญ

ในโหมดการวัดความถี่ ผู้วิจัยได้ใช้เครื่องสร้างสัญญาณความถี่ ในการจำลองสัญญาณความถี่เพื่อการทดสอบระบบ โดยแบ่งความถี่ทดสอบออกเป็น 7 ค่าดังนี้คือ 50Hz, 100Hz, 10kHz, 20kHz, 30kHz, 10MHz และ 20MHz แล้วทำการบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่าความถี่ไฟฟ้าที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi ดังภาพที่ 3.20 พบว่ามีค่าตรงกัน

การทดสอบวัดผลการทำงานของเครื่องมือวัดทางไฟฟ้า แบบแสดงผลด้วยเสียงในโหมดการวัดกระแสไฟฟ้ากระแสตรง

ในโหมดการวัดกระแสไฟฟ้ากระแสตรง ผู้วิจัยได้ใช้เครื่องจ่ายไฟฟ้ากระแสตรงปรับค่าได้ ตราสินค้า Skytoppower รุ่น STP3005 ซึ่งเป็นแหล่งจ่ายแรงดันไฟฟ้าแบบปรับค่าได้ระหว่าง 0-30V และสามารถจ่ายกระแสได้สูงถึง 5 แอมป์ ในการจำลองเป็นแหล่งจ่ายให้กับโหลดความต้านทานค่าต่าง ๆ เพื่อทำการทดสอบระบบวัดกระแสไฟฟ้ากระแสตรง ได้ทำการทดลองปรับค่ากระแสไฟฟ้าของแหล่งจ่ายไฟฟ้าให้เป็นไปตามค่าดังตารางที่ 3.15 โดยทำการวัดกระแสไฟฟ้าซ้ำจำนวน 3 รอบและหาค่าเฉลี่ย และทำการบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เปรียบเทียบกับค่ากระแสไฟฟ้าที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi

ตารางที่ 3.15 ความสัมพันธ์ของผลการวัดกระแสไฟฟ้ากระแสตรง

แหล่งจ่าย กระแสไฟฟ้า	มัลติมิเตอร์ รุ่น UT61E	ค่าที่อ่านได้จากพอร์ต USB 2.0 ของ Raspberry Pi			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
30mA	27.27mA	27.27mA	27.26mA	27.26mA	27.26mA
40mA	36.10mA	36.10mA	36.10mA	36.10mA	36.10mA
50mA	45.12mA	45.12mA	45.12mA	45.12mA	45.12mA
60mA	53.81mA	53.81mA	53.81mA	53.81mA	53.81mA
70mA	63.02mA	63.02mA	63.02mA	63.01mA	63.02mA
1.0A	0.978mA	0.978mA	0.979mA	0.979mA	0.979mA
1.2A	1.244A	1.244A	1.243A	1.242A	1.243A

จากนั้นผู้วิจัยได้เขียนโปรแกรมแสดงค่าของไฟล์เสียงเรียงตามลำดับตามข้อมูลของการวัดที่ได้รับจากมัลติมิเตอร์ รุ่น UT61E ทางพอร์ตอนุกรม โดยได้ทำการแปลงสัญญาณจากพอร์ตอนุกรมเป็นพอร์ต USB ซึ่งใช้โมดูล ET-USB/RS232 MINI เสียบเข้าที่พอร์ต USB 2.0 ของบอร์ด Raspberry Pi และทำการส่งออกสัญญาณเสียงทางช่องลำโพงขนาด 3.5mm ของบอร์ด ซึ่งได้เสียบติดตั้งตัวลำโพง เพื่อแสดงผลออกเสียงพูดให้ผู้ใช้งานได้รับทราบ จากการทดลองทำให้ทราบว่าค่าแสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E เมื่อทำการเปรียบเทียบกับค่าที่ได้รับของบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0



ภาพที่ 3.23 การวัดกระแสไฟฟ้ากระแสตรง

ที่มา: นิเทศ ประเดชนบุญ

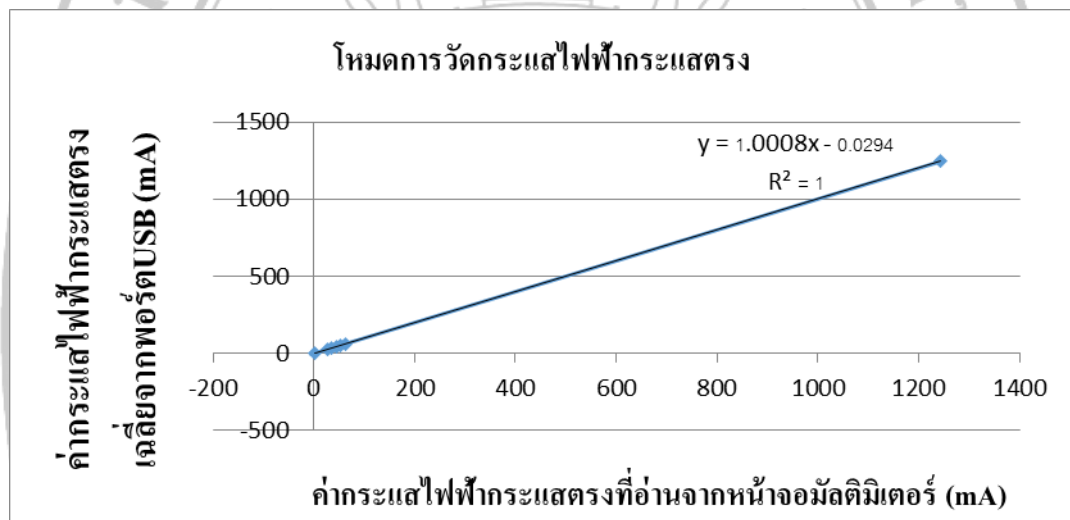
การเชื่อมต่ออุปกรณ์สำหรับวัดค่ากระแสไฟฟ้ากระแสตรง แสดงดังภาพที่ 3.21 ซึ่งมีการจำแนกข้อมูลและเรียบเรียงลำดับไฟล์เสียงที่วัดได้จากกระแสไฟฟ้าทดสอบแสดงดังภาพที่ 3.22

```

pi@raspberrypi:~ $ sudo python3 UT61E_DMM.py
การจำแนกข้อมูลระบบ : ['0.02727', 'A', 'current', 'DC', 'normal']
ส่งออกไฟล์เสียงโหมดการวัด : ['/home/pi/MP34DMM/current.mp3', '/home/pi/MP34DMM/DC.mp3']
สถานะการทำงานของเครื่อง : "normal"
ค่าที่วัดได้จาก UT61E : 27.27 mA
ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : ['27.27', 'mA']
ส่งออกไฟล์เสียงจำนวนเต็ม : ['/home/pi/MP34DMM/20.mp3', '/home/pi/MP34DMM/7.mp3',
'/home/pi/MP34DMM/point.mp3']
ส่งออกไฟล์เสียงหลังจุดทศนิยม : ['/home/pi/MP34DMM/2.mp3', '/home/pi/MP34DMM/7.mp3']
ส่งออกไฟล์เสียงหน่วยวัด : ['/home/pi/MP34DMM/mili.mp3', '/home/pi/MP34DMM/ampere.mp3']

```

ภาพที่ 3.24 ผลค่าการวัดกระแสไฟฟ้ากระแสตรง 30mA และการเรียงลำดับไฟล์เสียง
ที่มา: นิเทศ ประเดชญญ์



ภาพที่ 3.25 กราฟเปรียบเทียบระหว่างค่ากระแสไฟฟ้ากระแสตรงจากหน้าจอ
มัลติมิเตอร์กับค่ากระแสไฟฟ้ากระแสตรงเฉลี่ยจากพอร์ต USB
ที่มา: นิเทศ ประเดชญญ์

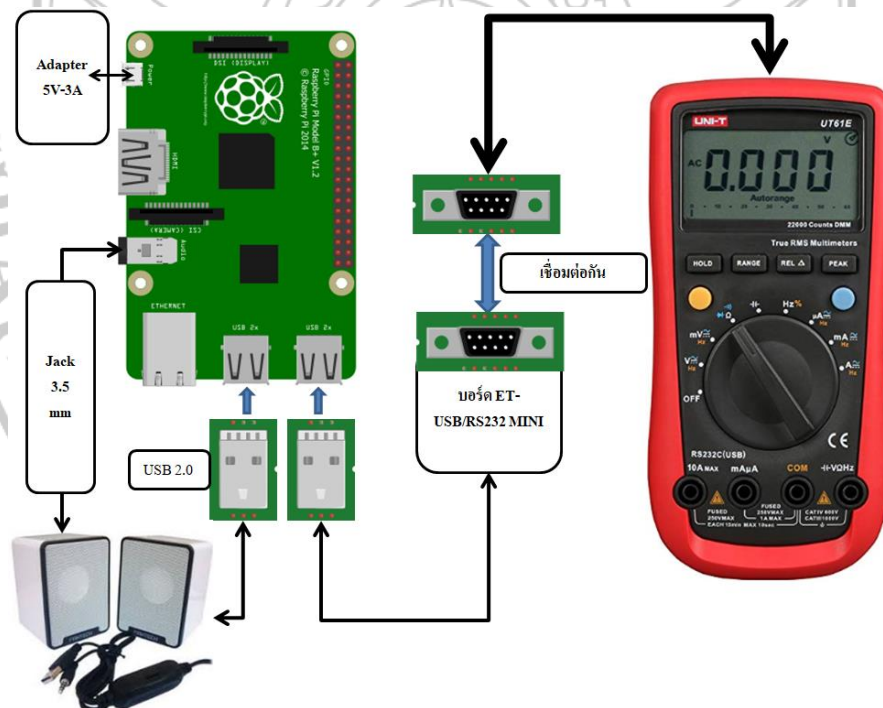
ในโหมดการวัดกระแสไฟฟ้ากระแสตรง ผู้วิจัยได้ใช้เครื่องจ่ายไฟฟ้ากระแสตรงปรับค่าได้จำลองเป็นแหล่งจ่ายให้กับโหลดความต้านทานค่าต่าง ๆ เพื่อทดสอบระบบวัดกระแสไฟฟ้ากระแสตรง ทดลองปรับค่ากระแสไฟฟ้าของแหล่งจ่ายไฟฟ้าให้เป็น 7 ค่าดังนี้คือ 30mA, 40mA, 50mA, 60mA, 70mA, 1.0A และ 1.2A โดยทำการวัดกระแสไฟฟ้าซ้ำ จำนวน 3 รอบ และหาค่าเฉลี่ย และทำการบันทึกผลการทดลองจากค่าที่แสดงผลทางหน้าจอของมัลติมิเตอร์ รุ่น UT61E

เปรียบเทียบกับค่ากระแสไฟฟ้าที่ได้รับจากพอร์ต USB 2.0 ของ Raspberry Pi ดังภาพที่ 3.23 พบว่ามีค่าตรงกัน

การสร้างและส่วนประกอบของตัวเครื่อง

หลังจากได้ทำการออกแบบและเขียนโปรแกรมในการติดต่อสื่อสารระหว่างตัวบอร์ด Raspberry Pi กับมัลติมิเตอร์ระบบตัวเลข ตราสินค้า UNI-T รุ่น UT61E จากนั้นได้ทำการประกอบตัวเครื่องเข้าด้วยกัน ซึ่งมีส่วนประกอบต่าง ๆ ดังรายการดังต่อไปนี้

- | | |
|---|-----------------|
| 1. กล่องเหล็กหุ้มด้วยสติ๊กเกอร์สีฟ้า | จำนวน 1 กล่อง |
| 2. อะแดปเตอร์แหล่งจ่ายไฟฟ้ากระแสตรง 5V 3A. | จำนวน 1 ชุด |
| 3. สายโพรบสีแดงและสีดำสำหรับการวัด | จำนวน 2 เส้น |
| 4. มัลติมิเตอร์ตราสินค้า UNI-T รุ่น UT61E | จำนวน 1 เครื่อง |
| 5. บอร์ด Raspberry Pi Model B+ | จำนวน 1 บอร์ด |
| 6. ลำโพงแบบ USB ตราสินค้า ASTHAS รุ่น GS733 | จำนวน 1 ชุด |
| 7. สายสัญญาณ RS232 Isolation | จำนวน 1 เส้น |
| 8. โมดูลแปลงสัญญาณ รุ่น ET-USB/RS232 MINI | จำนวน 1 โมดูล |



ภาพที่ 3.26 เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด

ที่มา: นิเทศ ประเดชนบุญ

การสร้างและออกแบบเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดครั้งนี้ได้มีการประกอบวงจรระบบสัญญาณทางไฟฟ้า สายสัญญาณ RS232 Isolation ลำโพงแบบ USB มัลติมีเตอร์ ตราสินค้า UNI-T รุ่น UT61E เข้ากับบอร์ด Raspberry Pi ซึ่งติดต่อสื่อสารกันผ่านทางพอร์ต USB 2.0 ดังแสดงดังภาพที่ 3.38



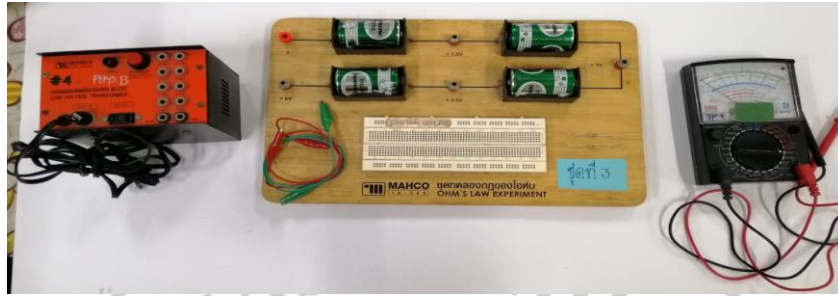
ภาพที่ 3.27 เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด

ที่มา: นิเทศ ประเดชญ

เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดที่ประกอบสำเร็จและพร้อมใช้งาน ดังแสดงในภาพที่ 3.25 เพื่อนำไปใช้เป็นส่วนประกอบในสื่อการสอนในการวัดค่าทางไฟฟ้า สำหรับการจัดการเรียนการสอนวิทยาศาสตร์ ซึ่งสามารถใช้ได้กับเด็กปกติทั่วไป และเหมาะสม สำหรับใช้กับเด็กที่บกพร่องทางการมองเห็น จากนั้นจะนำเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผล ด้วยเสียงพูดไปใช้ใน ชุดทดลอง เรื่อง กฎของโอห์ม สำหรับนักเรียนระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์ และวิทยาศาสตร์ทั่วไป เพื่อทำการประเมินความพึงพอใจในสื่อวัตกรรมการสอนต่อไป

ประเมินความพึงพอใจในสื่อวัตกรรมการสอน

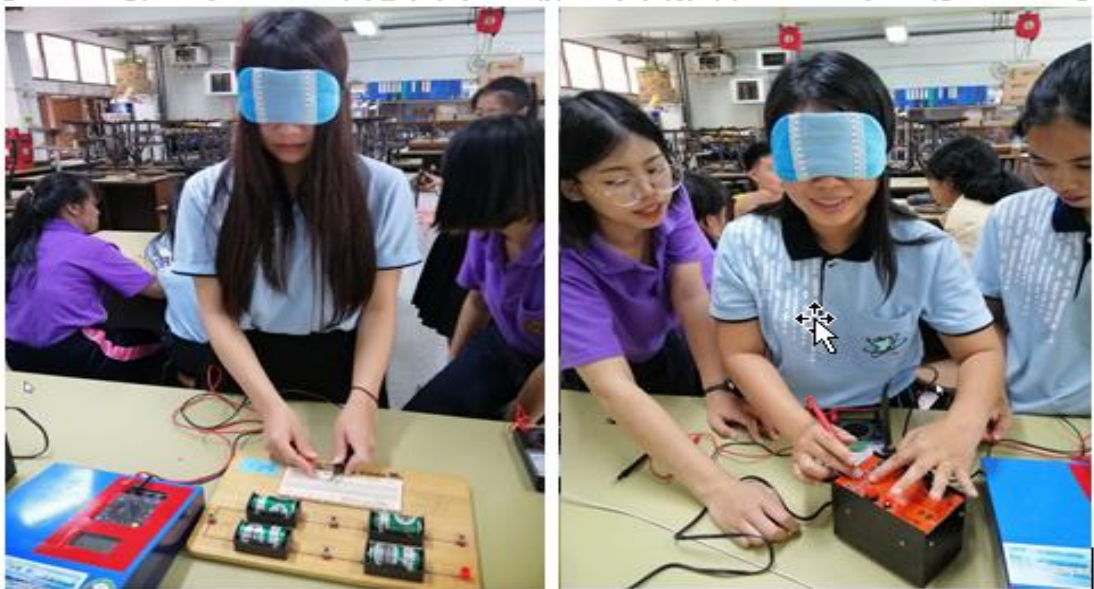
นำเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดนำไปใช้เป็นส่วนประกอบในสื่อ การสอนในการวัดค่าทางไฟฟ้า ในชุดทดลอง เรื่อง กฎของโอห์ม สำหรับนักเรียนระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป เพื่อประเมินความพึงพอใจในสื่อวัตกรรมการสอน โดยใน การจัดกิจกรรมในชุดทดลองเรื่อง กฎของโอห์มได้มีอุปกรณ์ที่ใช้ในการจัดกิจกรรมดังต่อไปนี้



ภาพที่ 3.28 ชุดทดลองกฎของโอห์ม

ที่มา: นิเทศ ประเดชญญ

เป็นอุปกรณ์ในชุดทดลอง เรื่อง กฎของโอห์ม สำหรับนักเรียนระดับชั้นปีที่ 4 ภาควิชา ฟิสิกส์และวิทยาศาสตร์ทั่วไป ประกอบไปด้วย แผงชุดทดลองกฎของโอห์ม ซึ่งประกอบด้วยเซลล์แบตเตอรี่ จำนวน 4 ก้อน และแผงต่อวงจรแบบอนุกรม-ขนาน แหล่งจ่ายไฟฟ้ากระแสตรง ตัวต้านทานไฟฟ้า มัลติมิเตอร์ชนิดเข็ม ทรานซิสเตอร์ SUNWA ดังแสดงในภาพที่ 3.26 และการใช้งาน เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด ดังแสดงในภาพที่ 3.27



ภาพที่ 3.29 ใช้งานเครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด

ที่มา: นิเทศ ประเดชญญ

การจัดกิจกรรมโดยใช้เครื่องมือวัดค่าทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด เข้าไปรวมกันกับชุดทดลอง เรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ซึ่งได้ดำเนินการทดลองตามเอกสารการทดลองชุดที่ 7 การทดลองกฎของโอห์ม ซึ่งมีขั้นตอนการทดลอง ตามเอกสารภาคผนวก ง. โดยมีการทำการทดลองของวงจรแบ่งออกเป็น 3 วงจร คือ การต่อวงจรสำหรับวัดปริมาณไฟฟ้าต่าง ๆ ตามกฎของโอห์ม การต่อตัวต้านทานในวงจรไฟฟ้า แบบอนุกรม (Serial Circuit) และแบบขนาน (Parallel Circuit) เพื่อให้นักศึกษาสามารถบอกความสัมพันธ์ของค่าความต่างศักย์ไฟฟ้าปริมาณกระแสไฟฟ้าและค่าความต้านทานไฟฟ้าตามกฎของโอห์มได้อย่างถูกต้อง และเพื่อให้นักศึกษามีทักษะการวัดค่าความต่างศักย์ไฟฟ้าและปริมาณกระแสไฟฟ้าในวงจรไฟฟ้าที่มีการต่อตัว ต้านทานแบบอนุกรมและแบบขนาน



บทที่ 4

ผลการวิเคราะห์ข้อมูล

การวิจัยครั้งนี้เป็นการวิจัย เรื่อง การออกแบบและพัฒนาเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพุดจากมัลติมิเตอร์ระบบตัวเลข ยี่ห้อ UNI-T รุ่น UT61E ร่วมกับบอร์ด Raspberry Pi Model B+ แล้วทดสอบการวัดค่าทางไฟฟ้าได้แก่ ค่าแรงดันไฟฟ้ากระแสตรง ค่ากระแสไฟฟ้าของไฟฟ้ากระแสตรง ค่าความต้านทานไฟฟ้าของตัวต้านทาน ค่าแรงดันไฟฟ้ากระแสสลับ ค่ากระแสไฟฟ้าของไฟฟ้ากระแสสลับ ค่าความถี่ไฟฟ้า และค่าตัวเก็บประจุ โดยทำการเปรียบเทียบค่าทางไฟฟ้าที่วัดได้กับค่าที่ได้รับจากบอร์ด Raspberry Pi ผ่านทางพอร์ต USB 2.0 พบว่าเครื่องมือวัดนี้สามารถแสดงผลเป็นเสียงพุดได้ถูกต้องตามค่าที่วัดได้ และจากนั้นได้ทำการเก็บข้อมูลเพื่อประเมินระดับความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อนวัตกรรมในการทดลองเรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป

ประชากรและกลุ่มตัวอย่าง

ประชากร นักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป มหาวิทยาลัยราชภัฏเชียงใหม่ จำนวน 8 คน

สถานที่ทดลอง

ห้องปฏิบัติการฟิสิกส์ชั้น 2 ห้อง 28206 มหาวิทยาลัยราชภัฏเชียงใหม่

ความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อนวัตกรรม



ภาพที่ 4.1 การจัดกิจกรรมการเรียนการสอนโดยใช้สื่อนวัตกรรม

ที่มา: นิเทศ ประเดชนบุญ

ในการวัดระดับความพึงพอใจของนักเรียนที่มีต่อการเรียนการสอนโดยใช้สื่อนวัตกรรม ดังแสดงในภาพที่ 4.1 โดยใช้ชุดการทดลองเรื่อง กฎของโอห์ม สำหรับนักเรียนระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป สามารถวัดได้จากแบบประเมินความพึงพอใจ โดยมีการประเมินด้านผลประโยชน์ที่ได้รับ ด้านสื่อการสอนและด้านผู้สอน ซึ่งเกณฑ์การประเมินระดับความพึงพอใจ ดังแสดงในตารางที่ 4.1 มีเกณฑ์การประเมินคือ 4.5-5 หมายถึง มากที่สุด 3.50-4.49 หมายถึง มาก 2.50-3.49 หมายถึง ปานกลาง 1.50-2.49 หมายถึง น้อย และ 1.00-1.49 หมายถึง น้อยที่สุด

ตารางที่ 4.1 เกณฑ์การประเมินระดับความพึงพอใจ

ค่าเฉลี่ย	ผลความพึงพอใจ
4.50-5.00	มากที่สุด
3.50-4.49	มาก
2.50-3.49	ปานกลาง
1.50-2.49	น้อย
1.00-1.49	น้อยที่สุด

ผลการประเมินความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อนวัตกรรมในการทดลองเรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป จำนวน 8 คน ได้ผลการประเมินในตารางที่ 4.2

ตารางที่ 4.2 ผลการประเมินระดับความพึงพอใจต่อสื่อการสอน

ตัวชี้วัดความพึงพอใจ	ความพึงพอใจ					เฉลี่ย	ระดับความพึงพอใจ
	1	2	3	4	5		
1. เนื้อหาของสื่อวัตกรรมการสอน มีความสอดคล้องกับวัตถุประสงค์รายวิชา						5	มากที่สุด
2. เนื้อหาของสื่อวัตกรรมการสอน สอดคล้องกับกิจกรรมการเรียนการสอน						5	มากที่สุด
3. สื่อวัตกรรมการสอนเหมาะกับสภาพแวดล้อมการเรียนรู้						4.38	มาก
4. พฤติกรรมและบุคลิกภาพของผู้สอน						5	มากที่สุด
5. สื่อวัตกรรมการสอน มีการออกแบบถูกต้องตามหลักการผลิตสื่อ						4.75	มากที่สุด
6. สื่อวัตกรรมการสอน ได้รับความสนใจให้เกิดการเรียนรู้ในเรื่องราวที่ต้องศึกษา						4.63	มากที่สุด
7. สื่อวัตกรรมการสอน มีความทันสมัยแปลกใหม่แตกต่างไปจากการเรียนปกติ						5	มากที่สุด
8. สื่อวัตกรรมการสอน เพิ่มบทบาทผู้เรียนในการเป็นผู้ปฏิบัติ						5	มากที่สุด
9. ผู้สอนมีการจัดกิจกรรมการเรียนรู้เพื่อส่งเสริมและกระตุ้นให้ผู้เรียนเกิดกระบวนการคิดวิเคราะห์						4.75	มากที่สุด

ตารางที่ 4.2 (ต่อ)

ตัวชี้วัดความพึงพอใจ	ความพึงพอใจ					เฉลี่ย	ระดับความพึงพอใจ
	1	2	3	4	5		
10. เป็นสื่อนวัตกรรมการสอนที่ส่งเสริมคุณธรรม จริยธรรม ให้แก่ผู้เรียนไม่ขัดต่อคุณธรรมและจริยธรรมที่พึงมีพึงปฏิบัติ						4.63	มากที่สุด
11. ผู้เรียนมีส่วนร่วมในการใช้สื่อ						5	มากที่สุด
12. สื่อนวัตกรรมการเรียน ผ่านการตรวจสอบจากผู้เชี่ยวชาญ						4.75	มากที่สุด
13. สื่อนวัตกรรมการเรียน ช่วยเชื่อมโยงความรู้เดิมกับความรู้ใหม่						4.75	มากที่สุด
14. ผู้สอนและผู้เรียนมีการสร้างองค์ความรู้ร่วมกันจากการใช้สื่อ						4.75	มากที่สุด
15. ติดตามผลจากการใช้สื่อของผู้เรียนได้ง่าย						4.38	มาก
16. สื่อที่ใช้สร้างความสนใจและส่งเสริมให้ผู้เรียนอยากเรียนรู้ด้วยตนเอง						4.75	มากที่สุด
17. สื่อนวัตกรรมการสอน สามารถถ่ายทอดเนื้อหาที่เป็นนามธรรมให้เป็นรูปธรรม						4.5	มากที่สุด
18. สื่อนวัตกรรมการสอน ราคาไม่แพง ต้นทุนการผลิตต่ำ คุ่มค่าต่อการใช้งาน						4.38	มาก
19. สื่อนวัตกรรมการสอน ช่วยส่งเสริมปฏิสัมพันธ์ทางสังคมระหว่างผู้สอนกับผู้เรียน						5	มากที่สุด
20. ท่านมีความพึงพอใจสื่อนวัตกรรมการสอนโดยรวมเป็นอย่างไร						4.38	มาก
รวมทั้งหมด						4.74	มากที่สุด

จากตารางที่ 4.2 สรุปผลการประเมินความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อวัตกรรมการทดลองเรื่อง กฎของโอห์ม สำหรับนักศึกษาระดับชั้นปีที่ 4 ภาควิชาฟิสิกส์และวิทยาศาสตร์ทั่วไป จำนวน 8 คน พบว่า นักเรียนส่วนใหญ่มีความพึงพอใจต่อการทดลองในการทดลองเรื่อง กฎของโอห์มในทุก ๆ ด้าน จากการประเมินโดยภาพรวมค่าเฉลี่ยที่ได้เท่ากับ 4.74 อยู่ในระดับความพึงพอใจมากที่สุด

ความคิดเห็นเพิ่มเติมในการปรับปรุงสื่อวัตกรรมการ

1. ลำโพงในตัวเครื่องอ่านคำมีเสียงเบา และช้า
2. ควรมีการเพิ่มอักษรเบลล์ลงบนสื่อวัตกรรมการนี้ ให้ผู้เรียนสามารถตั้งคำหรืออ่านคำ
ชี้แนะได้
3. ควรปรับปรุงตัวอ่านข้อมูลเรื่องความแม่นยำให้มากขึ้น
4. ควรมีให้เลือกหลายภาษา และสามารถปรับระดับความดังเสียงได้
5. เครื่องมีความรีโหลดช้าต่อการพูดออกเสียง และคำที่หน้าจอไม่ตรงกับคำที่พูด
6. ควรมีคู่มือการใช้งานขณะวัดว่าควรวัดอย่างไร เวลาใด

บทที่ 5

สรุป อภิปรายผล และข้อเสนอแนะ

สรุปผลการวิจัย

งานวิจัยได้ทำการออกแบบเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูดซึ่งตัวเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด สามารถพูดออกเสียงให้ผู้ใช้งานได้ถูกต้อง ตามค่าที่วัดได้ และสามารถวัดค่าต่าง ๆ ทางไฟฟ้า ได้แก่ การวัดแรงดันไฟฟ้า การวัดกระแสไฟฟ้า การวัดตัวต้านทาน การวัดตัวเก็บประจุ และการวัดความถี่ เป็นต้น จึงได้นำเครื่องมือวัดแบบแสดงผลด้วยเสียงพูด เข้าไปเป็นส่วนประกอบรวมในชุดทดลองด้านวิทยาศาสตร์ เพื่อประกอบการเรียนการสอนเป็นสื่อวัตกรรมการศึกษา และได้ทำการประเมินความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนการสอนโดยใช้สื่อวัตกรรมการเรียนการสอนนี้ ในการทดลอง เรื่อง กฎของโอห์ม โดยมีการทดลองใช้เครื่องมือวัดแบบแสดงผลด้วยเสียงพูด ทำการวัดค่าทางไฟฟ้าต่าง ๆ ตามขั้นตอนในแบบทดลอง เรื่องกฎของโอห์ม แยกออกเป็นทำการทดลอง 2 ตอน ได้แก่ องค์ประกอบพื้นฐานของวงจรไฟฟ้าตามกฎของโอห์ม การต่อความต้านทานในวงจรไฟฟ้าแบบอนุกรมและแบบขนาน

เครื่องมือที่ใช้ในการเก็บรวบรวมข้อมูลเป็นตารางแบบสอบถามเกี่ยวกับความพึงพอใจต่อการสอน โดยแยกออกเป็นตัวชี้วัดความพึงพอใจในด้านต่าง ๆ อาทิเช่น เนื้อหาของสื่อวัตกรรมการสอน มีความสอดคล้องกับวัตถุประสงค์รายวิชา สื่อวัตกรรมการสอน สอดคล้องกับกิจกรรมการเรียนการสอน สื่อวัตกรรมการสอน มีความทันสมัย แปลกใหม่ แตกต่างไปจากการเรียนปกติ สื่อวัตกรรมการสอน เพิ่มบทบาทผู้เรียนในการปฏิบัติ ผู้สอนมีการจัดกิจกรรมการเรียนรู้ เพื่อส่งเสริม และกระตุ้นให้ผู้เรียนเกิดกระบวนการคิดวิเคราะห์ และอื่น ๆ เป็นต้น โดยมีการวัดผลความพึงพอใจ มีเกณฑ์การประเมิน แยกออกเป็น 5 ระดับ คือ มากที่สุด มาก ปานกลาง น้อย และน้อยที่สุด

อภิปรายผล

จากการศึกษาและประเมินความพึงพอใจของผู้เรียนที่มีต่อสื่อวัตกรรมการสอน โดยอาศัยเครื่องมือวัดทางไฟฟ้าแสดงผลเป็นเสียงพูด เข้าไปมีส่วนร่วมในการจัดกิจกรรมการเรียน

การสอนด้านวิทยาศาสตร์ฟิสิกส์ เกี่ยวกับเรื่อง ไฟฟ้าเบื้องต้น โดยการจัดกิจกรรมการเรียนการสอน ในครั้งนี้ ได้ทดลองในหัวข้อ เรื่อง กฎของโอห์ม โดยมีการจัดผลพบว่า นักเรียนมีความพึงพอใจต่อ สื่อนวัตกรรมการสอนอยู่ใน ระดับความพึงพอใจมากที่สุด ซึ่งมีค่าเฉลี่ยความพึงพอใจอยู่ในระดับ ค่าเฉลี่ย 5 อยู่ด้วยกัน 7 ตัวชี้วัด ดังนี้คือ เนื้อหาของสื่อนวัตกรรมการสอน มีความสอดคล้องกับ วัตถุประสงค์รายวิชา เนื้อหาของสื่อนวัตกรรมการสอน สอดคล้องกับกิจกรรมการเรียนการสอน พฤติกรรมและบุคลิกภาพของผู้สอน สื่อนวัตกรรมการสอน มีความทันสมัย แปลกใหม่แตกต่างไป จากการเรียนปกติ สื่อนวัตกรรมการสอน เพิ่มบทบาทผู้เรียนในการเป็นผู้ปฏิบัติ ผู้เรียนมีส่วนร่วม ในการใช้สื่อและสื่อนวัตกรรมการสอน ช่วยส่งเสริมปฏิสัมพันธ์ทางสังคมระหว่างผู้สอนกับผู้เรียน

นักเรียนมีความพึงพอใจต่อสื่อนวัตกรรมการสอนอยู่ใน ระดับความพึงพอใจมาก ซึ่งมี ค่าเฉลี่ยความพึงพอใจอยู่ในระดับค่าเฉลี่ย 4.38 อยู่ด้วยกัน 1 ตัวชี้วัด ดังนี้คือ สื่อนวัตกรรมการสอน เหมาะกับสภาพแวดล้อมการเรียนรู้ เป็นต้น เนื่องจากพื้นที่ในการจัดในการจัดวางการทดลองในครั้ง นี้ข้อจำกัดอยู่บ้าง และตัวโต๊ะทดลองบ้างโต๊ะ บ้างจุดไม่มีการติดตั้งเต้าไฟฟ้าไว้ ทำให้ใช้งานเครื่อง อย่างสะดวก ซึ่งตัวเครื่องมือวัดทางไฟฟ้าแบบแสดงผลด้วยเสียงพูด จะต้องทำการเชื่อมต่อปลั๊ก แหล่งจ่ายของตัวเครื่องเข้ากับแหล่งจ่ายไฟฟ้าหลัก จึงจะสามารถใช้งานได้

จากการประเมินภาพรวมความพึงพอใจของผู้เรียนที่มีต่อสื่อนวัตกรรมการสอน โดยอาศัยเครื่องมือวัดทางไฟฟ้าแสดงผลเป็นเสียงพูด เข้าไปมีส่วนร่วมในการจัดกิจกรรมการเรียน การสอนมีค่าเฉลี่ยอยู่ที่เท่ากับ 4.74 อยู่ในระดับความพึงพอใจมากที่สุด เนื่องจากผู้เรียนสามารถ ที่จะเรียนรู้ด้วยตัวเอง จากสื่อการสอนที่หลากหลาย ทำให้ผู้เรียนเกิดความน่าสนใจและ กระตือรือร้นที่จะเรียนรู้ ผู้เรียนสามารถเลือกสื่อตามความสนใจและตามความถนัดของตนเอง สร้างบรรยากาศให้ผ่อนคลายก่อนเข้าสู่บทเรียน เพื่อกระตุ้นจิตใจของผู้เรียนให้มีความตื่นตัว ที่จะเรียนรู้และเพื่อลดความกลัวในตัวผู้เรียนให้ผู้เรียนรู้สึกผ่อนคลาย มีการเชื่อมโยงของสิ่งที่เรียนรู้ กับความเป็นจริงในชีวิตประจำวันทำให้ผู้เรียนจำได้และสนุกที่จะเรียนรู้

ข้อเสนอแนะ

1. งานวิจัยนี้ควรนำไปใช้กับกลุ่มผู้เรียนที่มีความบกพร่องทางการมองเห็นเพื่อประเมิน ประสิทธิภาพของสื่อนวัตกรรมที่ได้สร้างขึ้นอีกครั้ง
2. งานวิจัยนี้ได้ออกแบบและพัฒนาเครื่องมือวัดทางไฟฟ้าขึ้นเพื่อใช้ในการวัดค่าทาง ไฟฟ้าต่าง ๆ โดยให้แสดงผลเป็นเสียงพูด ความดังของเสียงที่แสดงออกมาควรมีการปรับให้สูงต่ำ ตามความต้องการได้ และสามารถใช้เป็นแนวทางในการพัฒนาให้สามารถวัดค่าอื่น ๆ เช่น อุณหภูมิ ความชื้น ความสว่างได้ต่อไป

3. ควรมีการเพิ่มปุ่มกด เพื่อใช้สำหรับเป็นตัวสั่งการให้เครื่องมือวัดทำการอ่านค่าการวัด ซึ่งจะทำให้การวัดค่าไม่มีเสียงพูดที่วนซ้ำกันตลอดเวลาและค่าที่แสดงทางหน้าจอกับค่าที่พูดจะมีความสอดคล้องกันมากขึ้น

4. ควรมีการเพิ่มอักษรเบลล์ลงบนสื่อนวัตกรรมนี้ ให้ผู้เรียนสามารถตั้งค่าหรืออ่านค่าได้



บรรณานุกรม

- ขเจนศักดิ์ ไตรธิเลน. (2553). การพัฒนากิจกรรมการเรียนการสอนโดยใช้รูปแบบการสอนที่เน้นทักษะปฏิบัติเรื่องการใช้อุปกรณ์และการต่อวงจรไฟฟ้า กลุ่มสาระการเรียนรู้การงานและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 1. (วิทยานิพนธ์ครุศาสตรมหาบัณฑิต, มหาวิทยาลัยราชภัฏสกลนคร).
- จิอริรัตน์ วงศ์จันทร์ชมพู่. (2558). ระบบควบคุมคุณภาพคลังยาโดยใช้บอร์ด Raspberry Pi. (วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, มหาวิทยาลัยเทคโนโลยีสุรนารี).
- ครุณี อ่อนน้อม. (2557). ระบบเฝ้าระวังสภาพแวดล้อมและตรวจสอบการเจริญเติบโตของไก่ในโรงเรือนเลี้ยงไก่ผ่านอินเทอร์เน็ต. (วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, มหาวิทยาลัยเทคโนโลยีสุรนารี).
- ญาคอย หอยสังข์. (2559). ไฟฟ้าและอิเล็กทรอนิกส์ มัธยมศึกษาปีที่ 3. สืบค้นจาก <http://weaerdekdee.blogspot.com>
- ทัศนัย ใจเย็น. (2554). การสร้างชุดการสอน เรื่องวงจรไฟฟ้า วิชางานไฟฟ้าและอิเล็กทรอนิกส์ เบื้องต้น หลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2546 สำนักงานคณะกรรมการการอาชีวศึกษา. (วิทยานิพนธ์ครุศาสตรบัณฑิต, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ).
- ภักธิมา เทียงตรง. (2553). ประสิทธิภาพในการใช้โปรแกรมคอมพิวเตอร์ฝึกพิมพ์ดีดสัมผัสสำหรับคนพิการ. วารสารวิทยาลัยราชสุดา มหาวิทยาลัยมหิดล, 6(1), 1-5.
- ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ. (2559). การเริ่มต้นใช้งานบอร์ด Raspberry Pi 3 เบื้องต้น. สืบค้นจาก http://cpre.kmutnb.ac.th/esl/learning/index.php?article=rpi3_quickstart
- มานพ ทนงค์ชัย. (2559). วงจรไฟฟ้า. สืบค้นจาก <http://webhtml.horhook.com/wbi/ec/index.htm>
- Datasheet 11000 Counts Muti-Channel ADC IC (ES51922). (2017). Retrieved from <http://www.cyrustek.com.tw/product-1-11000.htm#ES51922>

Alee R. (2556). *Reading data from a digital multimeter using a Raspberry pi.*

(Bachelor's thesis Information Technology Networking, TURKU UNIVERSITY).

Sangthong, S. (2558). *เริ่มต้นกับ Raspberry Pi.* สืบค้นจาก <http://www.sathittham.com/raspberry-pi/rpi-ep-2/>



ประวัติผู้วิจัย

ชื่อ-สกุล

นิเทศ ประเดชนบุญ

วัน เดือน ปีเกิด

27 มีนาคม 2525

ที่อยู่ปัจจุบัน

149 หมู่ 7 ตำบลจุน อำเภอจุน จังหวัดพะเยา

ประวัติการศึกษา

พ.ศ. 2549 ปริญญาตรี วิศวกรรมบัณฑิต วิศวกรรมไฟฟ้า

ประวัติการทำงาน

ปัจจุบัน วิศวกรไฟฟ้า สถาบันวิจัยและพัฒนาพลังงานนครพิงค์





ภาคผนวก

ภาคผนวก ก

คุณสมบัติและข้อจำกัดของมัลติมิเตอร์ ยี่ห้อ UNI-T รุ่น UT61E

คุณสมบัติและข้อจำกัดของตัวเครื่องมัลติมิเตอร์ ยี่ห้อ UNI-T รุ่น UT61E ในการวัดค่าทางไฟฟ้าใน
โหมดการวัดต่างๆดังนี้

1. DCV สำหรับการวัดความต่างศักย์ไฟฟ้ากระแสตรง มี 5 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ	แรงดันขาเข้าคงที่
220mV	0.01mV	+/- (0.1%+5)	1000VDC / 750VAC
2.2V	0.0001V	+/- (0.1%+2)	
22V	0.001V		
220V	0.01V		
1000V	0.1V	+/- (0.1%+5)	

2. ACV สำหรับการวัดความต่างศักย์ไฟฟ้ากระแสสลับ มี 5 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ		แรงดันขาเข้า คงที่
		45-1kHz	>1kHz-10kHz	
220mV	0.01mV	+/- (1.0%+10)	+/- (1.5%+50)	1000VDC / 750VAC
2.2V	0.0001V	+/- (0.8%+10)	+/- (1.2%+50)	
22V	0.001V		+/- (2.0%+50))	
220V	0.01V			
750V	0.1V	+/- (1.2%+10)	+/- (3.0%+50)	

3. DCA สำหรับการวัดปริมาณกระแสตรง มี 5 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ	ป้องกันโหลดเกิน
220 μ A	0.01 μ A	(0.5%+10)	ฟิวส์ 1A 240V
2200 μ A	0.1 μ A		
22mA	0.001mA		
220mA	0.01mA		
10A	0.001A	(1.2%+50)	ฟิวส์ 10A 240V

4. ACA สำหรับการวัดปริมาณกระแสสลับ มี 5 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ		ป้องกันโหลดเกิน
		45-1kHz	>1kHz-10kHz	
220μA	0.01μA	(0.8%+10)	(1.2%+50)	ฟิวส์ 1A 240V
2200μA	0.1μA			
22mA	0.001mA	(1.2%+10)	(1.5%+50)	
220mA	0.01mA			
10A	0.01V	(1.5%+10)	>1kHz-5kHz	ฟิวส์ 10A 240V
			(2.0%+50)	

5. สำหรับการวัดความต้านทาน มี 7 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ	แรงดันขาเข้าคงที่
220Ω	0.01Ω	+/- (0.5%+10)	1000VDC / 750VAC
2.2kΩ	0.0001kΩ		
22kΩ	0.001kΩ		
220kΩ	0.01kΩ		
2.2MΩ	0.0001MΩ	+/- (0.8%+10)	
22MΩ	0.001MΩ	+/- (1.5%+10)	
220MΩ	0.01MΩ	+/- (3.0%+50)	

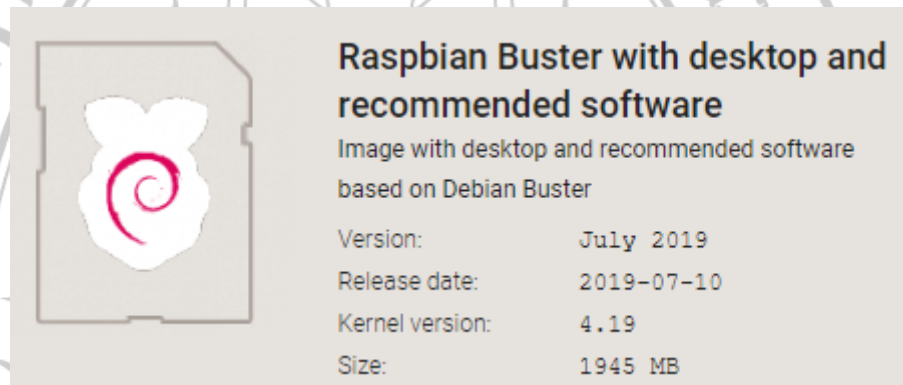
6. สำหรับการวัดความจุไฟฟ้า มี 8 ช่วงการวัด

ช่วงการวัด	ความละเอียด	ความแม่นยำ	แรงดันขาเข้าคงที่
22nF	0.001nF	+/- (0.5%+10)	1000VDC / 750VAC
220nF	0.01nF		
2.2μF	0.0001μF		
22μF	0.001μF		
220μF	0.01μF	+/- (0.8%+10)	
2.2mF	0.0001mF	+/- (1.5%+10)	
22mF	0.001mF	ไม่เจาะจง	
220mF	0.01mF		

ภาคผนวก ข

การติดตั้งระบบปฏิบัติการ Linux และการตั้งค่าพื้นฐาน

1. การติดตั้งระบบปฏิบัติการของบอร์ด Raspberry Pi เข้าไปดาวน์โหลดระบบปฏิบัติการ Linux ในเว็บไซต์ <https://www.raspberrypi.org/downloads/raspbian/> ดังในตัวอย่างในภาพที่ 1 และทำการแตกไฟล์ที่ดาวน์โหลดมาเพื่อทำขั้นตอนในการลงระบบปฏิบัติการในการ์ดแบบ SD CARD



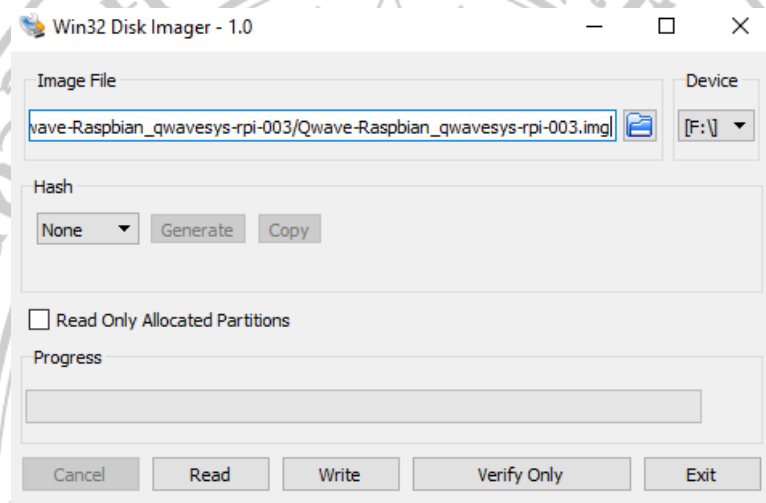
ภาพที่ 1 ดาวน์โหลดไฟล์ .img

1. ทำการติดตั้งโปรแกรมจากเว็บไซต์ <http://sourceforge.net/projects/win32diskimager/> ชื่อว่า โปรแกรม Win32DiskImager



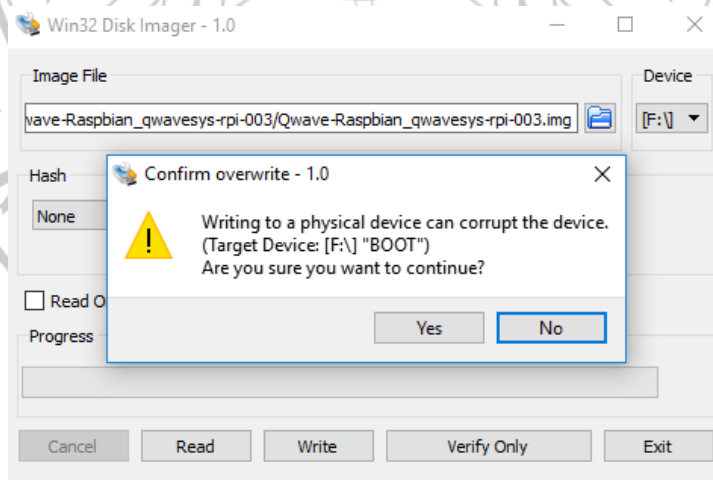
ภาพที่ 2 ดาวน์โหลดไฟล์ Win32DiskImager

2. โปรแกรม Win32 Disk Imager เป็นโปรแกรมที่ใช้เขียนอิมเมจไฟล์ลงใน SD CARD
 ดังแสดงในภาพที่ 2 ซึ่งโปรแกรม Win32 Disk Imager เป็นโปรแกรมฟรี ไม่เสียค่าใช้จ่ายใด ๆ
3. ให้นำ SD CARD (แนะนำให้ใช้ขนาด 8GB ขึ้นไป) ที่จะลงระบบปฏิบัติการไปเสียบเข้า CARD
 READER จากนั้นให้เปิดโปรแกรม Win32DiskImager คลิกที่ไอคอนโฟลเดอร์เลือกไปที่
 อิมเมจไฟล์ที่ดาวน์โหลดมา ดังภาพที่ 3



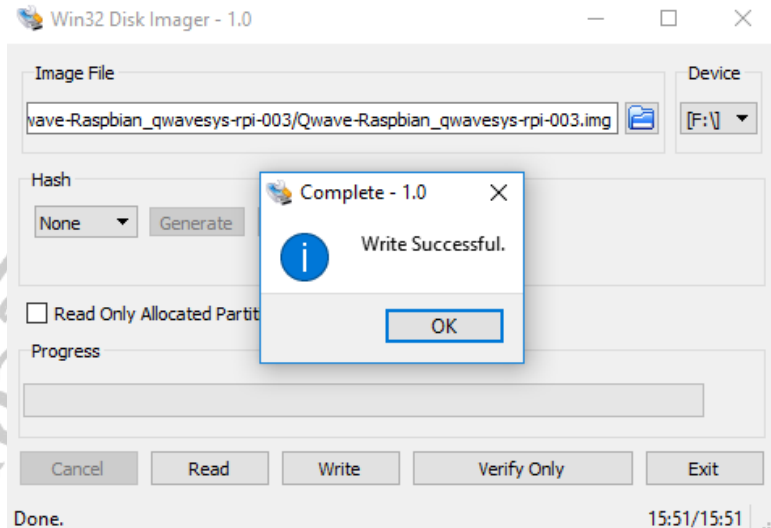
ภาพที่ 3 เลือกไฟล์อิมเมจที่จะเขียนลง SD CARD

4. ทำการคลิกปุ่ม Write เพื่อเริ่มการเขียนอิมเมจไฟล์เข้าสู่ SD CARD ซึ่งโปรแกรมจะถามยืนยัน
 การเขียนให้คลิก Yes ดังภาพที่ 4



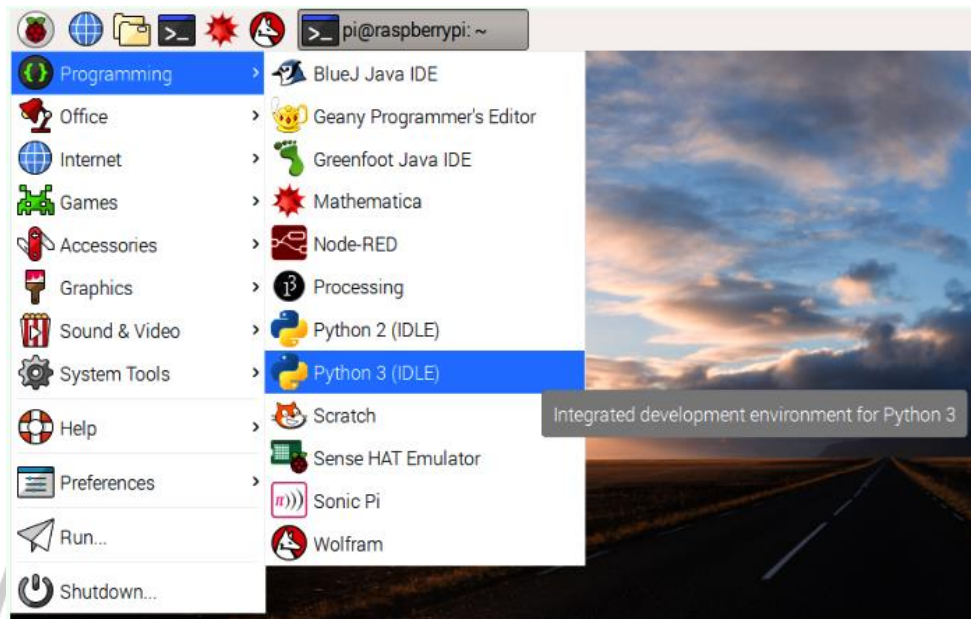
ภาพที่ 4 ยืนยันการเขียนไฟล์อิมเมจ

5. รอนการเขียนอิมเมจไฟล์เสร็จสมบูรณ์ จะแสดงดังภาพที่ 5 และพร้อมที่จะนำ SD CARD ที่ได้ไปเสียบเข้าบอร์ดคอมพิวเตอร์ Raspberry Pi ในขั้นตอนต่อไป



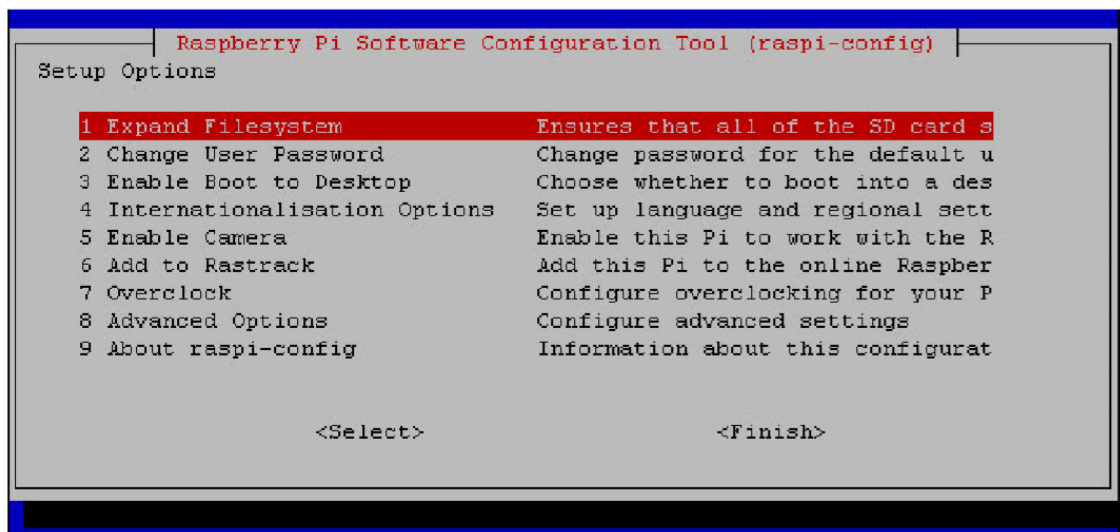
ภาพที่ 5 การเขียนไฟล์อิมเมจเสร็จสมบูรณ์

6. ทำการเชื่อมต่อคีย์บอร์ดและเมาส์ที่ช่อง USB 2.0 จากนั้นเสียบสาย HDMI และต่อจอภาพ LCD เสียบสายอะแดปเตอร์แหล่งจ่ายไฟให้แก่บอร์ด Raspberry Pi ผ่านทางช่องเสียบ Micro USB จากนั้นบอร์ด Raspberry Pi จะแสดงข้อความการบูตเข้าสู่ระบบปฏิบัติการ Linux ซึ่งตัวระบบปฏิบัติการจะทำการอ่านและติดตั้งค่าเริ่มต้น เพื่อให้ระบบสามารถเริ่มทำงานได้แบบอัตโนมัติ และเมื่อเข้าระบบสมบูรณ์จะแสดงหน้าต่าง เมนูของโปรแกรมต่างๆ ที่ได้ติดตั้งมาพร้อมกับตัวอิมเมจไฟล์ใน SD CARD ดังรูปภาพภาพที่ 6



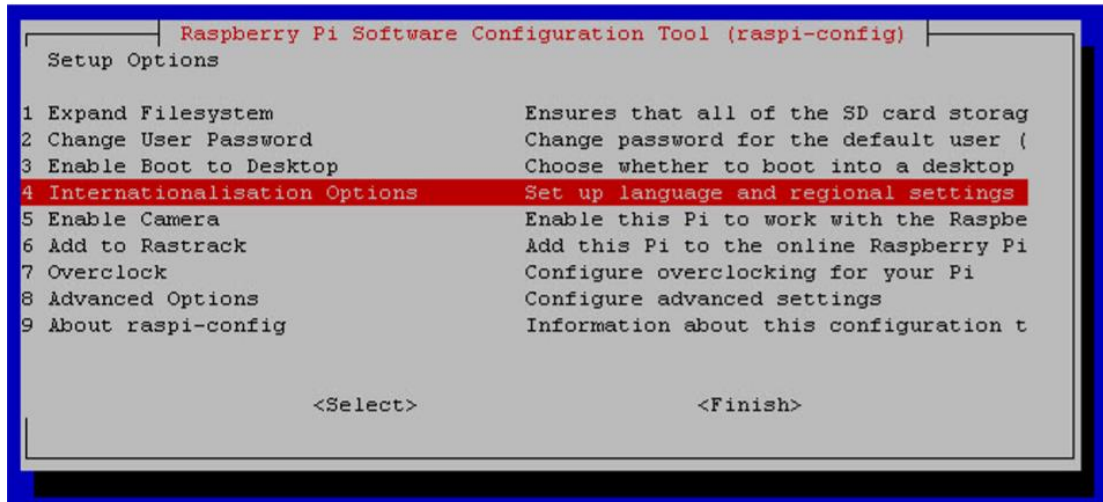
ภาพที่ 6 ระบบปฏิบัติการ Linux ของบอร์ด Raspberry Pi

7. การเข้าไปตั้งค่าระบบที่มีความจำเป็นพื้นฐานของบอร์ดผ่านทางคำสั่ง `sudo raspi-config` ในหน้าต่าง console แล้ว ENTER หน้าจอจะแสดง ดังรูปภาพที่ 7



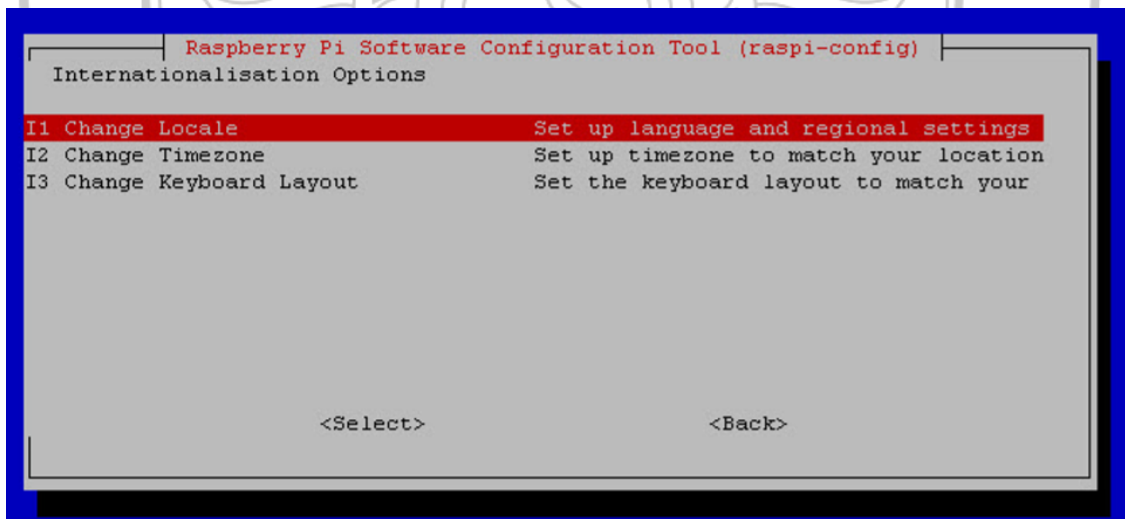
ภาพที่ 7 หน้าจอแสดงโหมดตั้งค่าพื้นฐาน

8. จากนั้นกดปุ่มลูกศรเลือกที่เมนู Internationalisation Options แล้วกดปุ่ม ENTER ดังภาพที่ 8



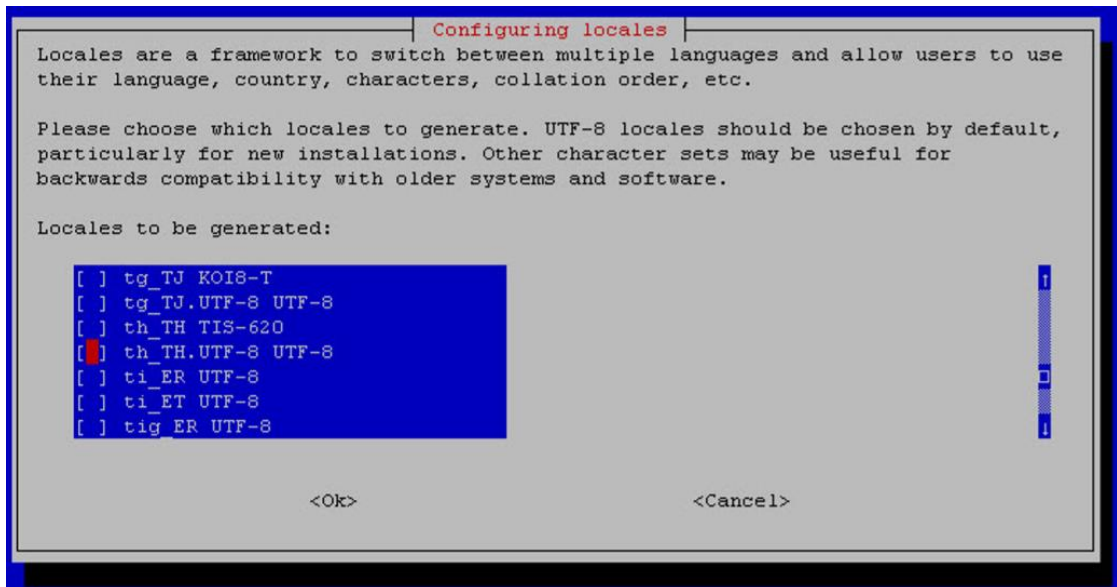
ภาพที่ 8 เมนู Internationalisation Options

9. เลือกที่เมนู Change Locale และกดปุ่ม ENTER ดังภาพที่ 9



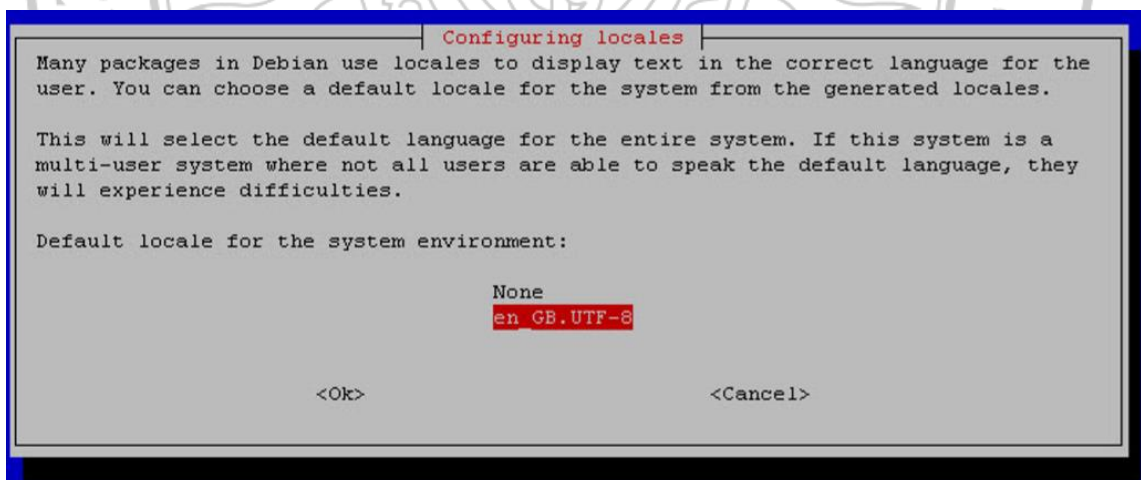
ภาพที่ 9 เมนู Change Locale

10. เลื่อนแถบสีแดงโดยใช้ปุ่มลูกศร เลือกที่ th_TH.UTF-8 UTF8 ดังภาพที่ 10



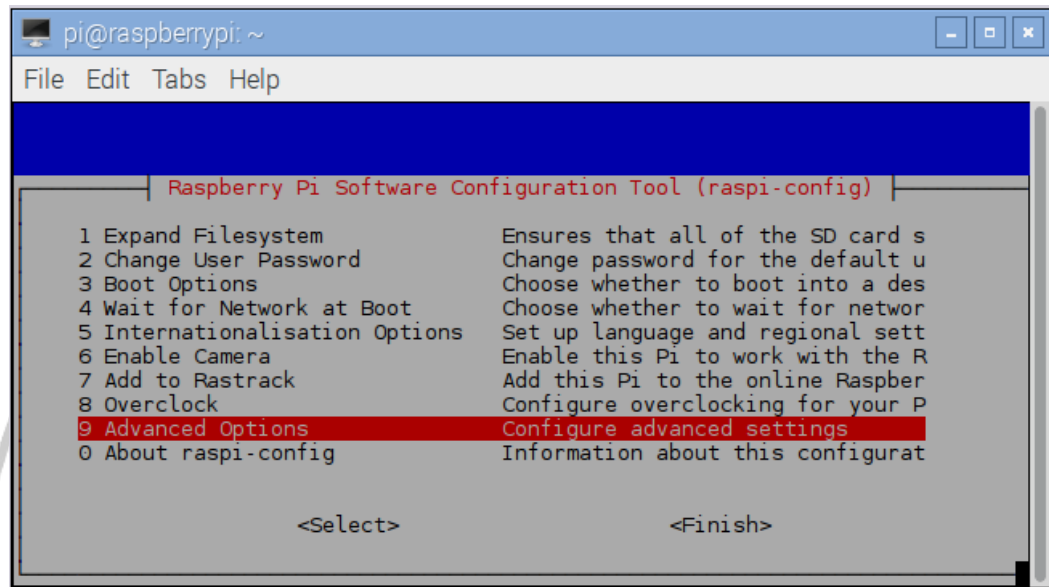
ภาพที่ 10 เมนู th_TH.UTF-8 UTF8

11. จากนั้นทำการกดปุ่ม ENTER อีกครั้ง ดังภาพที่ 11



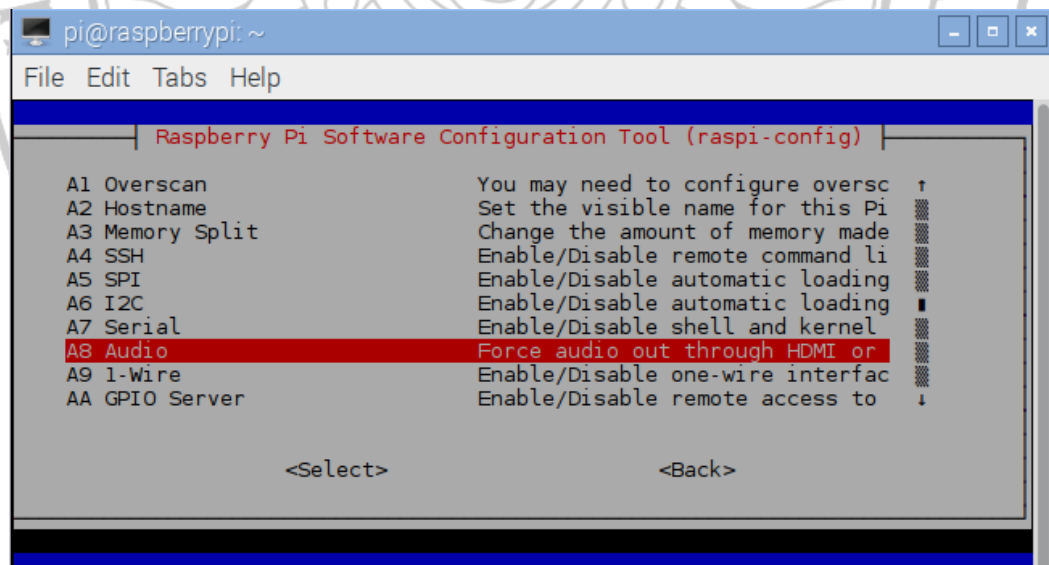
ภาพที่ 11 เมนู en_GB.UTF-8

12. กลับไปที่เมนู configuration tool อีกครั้ง เลื่อนแถบสีแดงมาที่ Advanced Options แล้วกดปุ่ม ENTER ดังภาพที่ 12



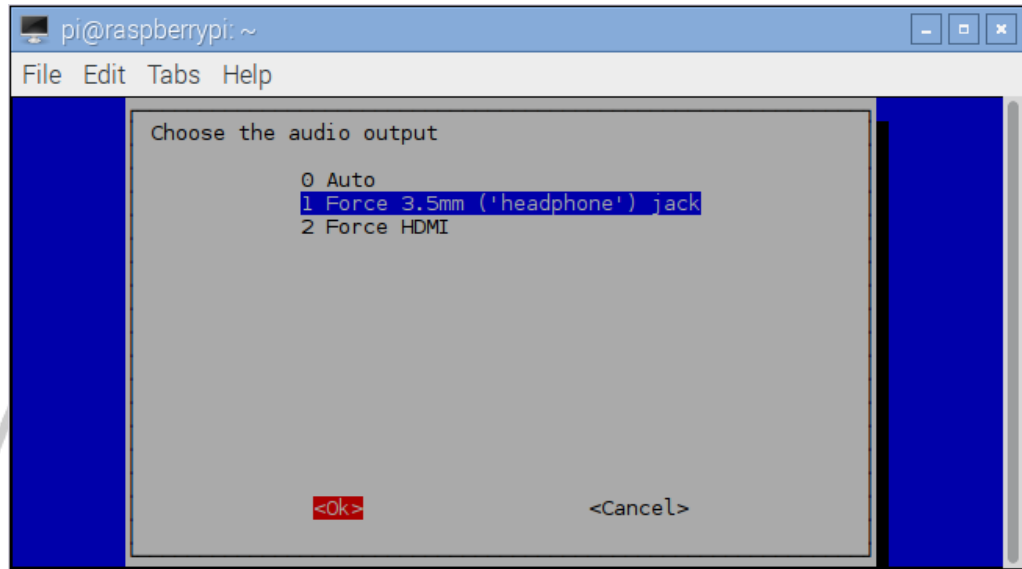
ภาพที่ 12 เมนู Advanced Options

13. จากนั้นเลื่อนแถบสีแดงมาที่ Audio แล้วกดปุ่ม ENTER ดังภาพที่ 13



ภาพที่ 13 เมนู Audio

14. จากนั้นจะมีเมนูขึ้นมาให้เลือก Force 3.5mm ('headphone') jack ดังภาพที่ 14



ภาพที่ 14 เมนู Force Headphone Jack

15. กลับไปที่เมนู Configuration Tool แล้วเลื่อนแถบสีแดงมาที่ปุ่ม Finish แล้วกดปุ่ม ENTER เป็นการเสร็จสิ้นการตั้งค่าพื้นฐานให้กับระบบของบอร์ด Raspberry Pi

ภาคผนวก ค

โปรแกรมภาษา Python สำหรับอ่านค่าจาก มัลติมิเตอร์ ตราสินค้า UNI-T รุ่น UT61E

```

1  from __future__ import print_function    #เรียกใช้งานโมดูล __future__
2                                          #คำสั่ง print_function
3  import serial                            #เรียกใช้งานโมดูล serial
4  import sys                              #เรียกใช้งานโมดูล sys
5  from decimal import Decimal             #เรียกใช้งานโมดูล decimal คำสั่ง Decimal
6  import struct                            #เรียกใช้งานโมดูล struct
7  import logging                           #เรียกใช้งานโมดูล logging
8  import datetime                          #เรียกใช้งานโมดูล datetime
9  from pygame import mixer                 #เรียกใช้งานโมดูล pygame คำสั่ง mixer
10
11 import time                              #เรียกใช้งานโมดูล time
12 import math                              #เรียกใช้งานโมดูล math
13
14 # http://wiki.python.org/moin/BitManipulation
15 def test_bit(int_type, offset):           #ฟังก์ชัน test_bit
16     mask = 1 << offset                    #เลื่อนบิตตามจำนวนของ offset
17     return bool(int_type & mask)           #ส่งออกค่าบูลีน และออกจากฟังก์ชัน
18
19 def get_bits(int_type, template):         #ฟังก์ชัน get_bits
20     bits = {}                             #สร้างตัวแปรชนิด Dictionary ชื่อ bits
21     for i in range(7):
22         bit = test_bit(int_type, i)       #เรียกฟังก์ชัน test_bit()

```

```

23     bit_name = template[6-i]           #กำหนดค่าตัวแปร bit_name
24     if bit_name in (0,1) and bit==bit_name: #ตรวจสอบข้อมูลว่าแต่ละบิตถูกต้องหรือไม่
25         continue                       #ทำงานรอบต่อไป
26     elif bit_name in (0,1):             # ถ้า bit_name เป็น 0,1
27         raise ValueError                #แจ้งความผิดพลาด ถ้า bit_name ไม่ใช่ 0,1
28
29     else:
30         bits[bit_name] = bit            #กำหนดค่าให้กับตัวแปร ชื่อ bits
31     return bits
32
33     RANGE_VOLTAGE = {                   #ตัวแปรกำหนด RANGE ในโหมดวัดแรงดัน
34         0b0110000: (1e0, 4, "V"),       # ช่วงแสดงผลเต็มสเกลที่ 2.2000V
35         0b0110001: (1e0, 3, "V"),       # ช่วงแสดงผลเต็มสเกลที่ 22.000V
36         0b0110010: (1e0, 2, "V"),       # ช่วงแสดงผลเต็มสเกลที่ 220.00V
37         0b0110011: (1e0, 1, "V"),       # ช่วงแสดงผลเต็มสเกลที่ 2200.0V
38         0b0110100: (1e-3, 2, "mV"),     # ช่วงแสดงผลเต็มสเกลที่ 220.00mV
39     }
40     RANGE_CURRENT_AUTO_UA = {           #ตัวแปรกำหนด RANGE
41                                         #ในโหมดวัดกระแส Auto  $\mu$ A
42         0b0110000: (1e-6, 2, "uA"),     # ช่วงแสดงผลเต็มสเกลที่ 220.00 $\mu$ A
43         0b0110001: (1e-6, 1, "uA"),     # ช่วงแสดงผลเต็มสเกลที่ 2200.0 $\mu$ A
44     }
45     RANGE_CURRENT_AUTO_MA = {           #ตัวแปรกำหนด RANGE
46                                         #ในโหมดวัดกระแส Auto mA
47         0b0110000: (1e-3, 3, "mA"),     # ช่วงแสดงผลเต็มสเกลที่ 22.000mA
48         0b0110001: (1e-3, 2, "mA"),     # ช่วงแสดงผลเต็มสเกลที่ 220.00mA

```

```

49     }
50     RANGE_CURRENT_AUTO = {           #ตัวแปรกำหนด RANGE
51                                     #ในโหมดวัดกระแส Auto
52         0b0110000: "Lower Range (IVSL)", #อินพุตการวัดกระแสในช่วง 220μA, 22mA.
53         0b0110001: "Higher Range (IVSH)" #อินพุตการวัดกระแสในช่วง 2200μA, 220mA, 22A.
54     }
55
56
57     RANGE_CURRENT_22A = {           #ตัวแปรกำหนด RANGE ในโหมดวัดกระแส 22A
58         0b0110000: (1e0, 3, "A")      #ช่วงแสดงผลเต็มสเกลที่ 22.000 A
59     }
60     RANGE_CURRENT_MANUAL = {        #ตัวแปรกำหนด RANGE
61                                     #ในโหมดวัดกระแส Manual
62         0b0110000: (1e0, 4, "A"),      #ช่วงแสดงผลเต็มสเกลที่ 2.2000A
63         0b0110001: (1e0, 3, "A"),      #ช่วงแสดงผลเต็มสเกลที่ 22.000A
64         0b0110010: (1e0, 2, "A"),      #ช่วงแสดงผลเต็มสเกลที่ 220.00A
65         0b0110011: (1e0, 1, "A"),      #ช่วงแสดงผลเต็มสเกลที่ 2200.0A
66         0b0110100: (1e0, 0, "A"),      #ช่วงแสดงผลเต็มสเกลที่ 22000A
67     }
68     RANGE_ADAP = {                 #ตัวแปรกำหนด RANGE ในโหมด ADP
69         0b0110000: "ADP4",             #ช่วงแสดงผลเต็มสเกลที่ 2.2000
70         0b0110001: "ADP3",             #ช่วงแสดงผลเต็มสเกลที่ 22.000
71         0b0110010: "ADP2",             #ช่วงแสดงผลเต็มสเกลที่ 220.00A
72         0b0110011: "ADP1",             #ช่วงแสดงผลเต็มสเกลที่ 2200.0
73         0b0110100: "ADP0",             #ช่วงแสดงผลเต็มสเกลที่ 22000
74     }

```

```

75     RANGE_RESISTANCE = {                                #ตัวแปรกำหนด RANGE
76                                                         #ในโหมดวัดตัวต้านทาน
77         0b0110000: (1e0, 2, "O"),                        #ช่วงแสดงผลเต็มสเกลที่ 220.00Ω
78         0b0110001: (1e3, 4, "kO"),                      #ช่วงแสดงผลเต็มสเกลที่ 2.2000KΩ
79         0b0110010: (1e3, 3, "kO"),                      #ช่วงแสดงผลเต็มสเกลที่ 22.000KΩ
80         0b0110011: (1e3, 2, "kO"),                      #ช่วงแสดงผลเต็มสเกลที่ 220.00KΩ
81         0b0110100: (1e6, 4, "MO"),                      #ช่วงแสดงผลเต็มสเกลที่ 2.2000MΩ
82         0b0110101: (1e6, 3, "MO"),                      #ช่วงแสดงผลเต็มสเกลที่ 22.000MΩ
83         0b0110110: (1e6, 2, "MO"),                      #ช่วงแสดงผลเต็มสเกลที่ 220.00MΩ
84     }
85     RANGE_FREQUENCY = {                                #ตัวแปรกำหนด RANGE
86                                                         #ในโหมดวัดความถี่
87         0b0110000: (1e0, 1, "H"),                        #ช่วงแสดงผลเต็มสเกลที่ 22.00Hz
88         0b0110001: (1e0, 1, "H"),                      #ช่วงแสดงผลเต็มสเกลที่ 220.0Hz
89         #0b0110010
90         0b0110011: (1e3, 3, "kH"),                      #ช่วงแสดงผลเต็มสเกลที่ 22.000KHz
91         0b0110100: (1e3, 2, "kH"),                      #ช่วงแสดงผลเต็มสเกลที่ 220.00KHz
92         0b0110101: (1e6, 4, "MH"),                      #ช่วงแสดงผลเต็มสเกลที่ 2.2000MHz
93         0b0110110: (1e6, 3, "MH"),                      #ช่วงแสดงผลเต็มสเกลที่ 22.000MHz
94         0b0110111: (1e6, 2, "MH"),                      #ช่วงแสดงผลเต็มสเกลที่ 220.00MHz
95     }
96     RANGE_CAPACITANCE = {                             #ตัวแปรกำหนด RANGE
97                                                         #ในโหมดวัดตัวเก็บประจุ
98         0b0110000: (1e-9, 3, "nF"),                    #ช่วงแสดงผลเต็มสเกลที่ 22.000nF
99         0b0110001: (1e-9, 2, "nF"),                    #ช่วงแสดงผลเต็มสเกลที่ 220.00nF
100        0b0110010: (1e-6, 4, "uF"),                     #ช่วงแสดงผลเต็มสเกลที่ 2.2000μF

```

```

101      0b0110011: (1e-6, 3, "uF"),          #ช่วงแสดงผลเต็มสเกลที่ 22.000µF
102      0b0110100: (1e-6, 2, "uF"),          #ช่วงแสดงผลเต็มสเกลที่ 220.00µF
103      0b0110101: (1e-3, 4, "mF"),          #ช่วงแสดงผลเต็มสเกลที่ 2.2000mF
104      0b0110110: (1e-3, 3, "mF"),          #ช่วงแสดงผลเต็มสเกลที่ 22.000mF
105      0b0110111: (1e-3, 2, "mF"),          #ช่วงแสดงผลเต็มสเกลที่ 220.00mF
106  }
107  RANGE_DIODE = {                          #ตัวแปรกำหนด RANGE
108                                          #ในโหมดวัดตัว Diode
109      0b0110000: (1e0, 4, "V"),            #ช่วงแสดงผลเต็มสเกลที่ 2.2000V
110  }
111  RANGE_CONTINUITY = {                    #ตัวแปรกำหนด RANGE
112                                          #ในโหมดวัดความต่อเนื่อง
113      0b0110000: (1e0, 2, "O"),            #ช่วงแสดงผลเต็มสเกลที่ 220.00Ω <Ohm>
114  }
115
116  FUNCTION = {                            #ตัวแปรกำหนด FUNCTION
117                                          #ทุกโหมดเรียงตามลำดับ
118      # (function, subfunction, unit)
119      0b0111011: ("voltage", RANGE_VOLTAGE, "V"),          #โหมดแรงดันไฟฟ้า
120      0b0111101: ("current", RANGE_CURRENT_AUTO_UA, "A"),  #โหมดกระแสไฟฟ้า
121      #Auto µA Current / Auto µA Current / Auto 220.00A/2200.0A
122      0b0111111: ("current", RANGE_CURRENT_AUTO_MA, "A"),  #โหมดกระแส Auto
123  mA
124      #Auto mA Current  Auto mA Current  Auto 22.000A/220.00A
125      0b0110000: ("current", RANGE_CURRENT_22A, "A"),      #โหมดกระแส 22A

```

```

126      0b0111001: ("current", RANGE_CURRENT_MANUAL, "A"),      #โหมดกระแส
127  Manual
128      0b0110011: ("resistance", RANGE_RESISTANCE, "Ohm"),      #โหมดความ
129  ด้านทาน
130      0b0110101: ("continuity", RANGE_CONTINUITY, "Ohm"),      #โหมดวัดความ
131  ต่อเนื่อง
132      0b0110001: ("diode", RANGE_DIODE, "V"),      #โหมดวัด Diode
133
134      0b0110010: ("frequency", RANGE_FREQUENCY, "Hz"),      #โหมดวัดความถี่
135  ไฟฟ้า
136      0b0110110: ("capacitance", RANGE_CAPACITANCE, "F"),      #โหมดวัดความประจุ
137      0b0110100: ("temperature", None, "deg"),      #โหมดวัดอุณหภูมิ
138      0b0111110: ("ADP", RANGE_ADP, ""),      #โหมด ADP
139  }
140  DIGITS = {      #ตัวแปรกำหนด DIGITS ตัวแทนตัวเลขดิจิทัล
141      0b0110000: 0,      #แทนตัวเลข 0
142      0b0110001: 1,      #แทนตัวเลข 1
143      0b0110010: 2,      #แทนตัวเลข 2
144      0b0110011: 3,      #แทนตัวเลข 3
145      0b0110100: 4,      #แทนตัวเลข 4
146      0b0110101: 5,      #แทนตัวเลข 5
147      0b0110110: 6,      #แทนตัวเลข 6
148      0b0110111: 7,      #แทนตัวเลข 7
149      0b0111000: 8,      #แทนตัวเลข 8
150      0b0111001: 9,      #แทนตัวเลข 9
151  }

```

```

152     STATUS = [                                #ตัวแปรกำหนด STATUS
153                                           #กำหนดรูปแบบ DMM STATUS
154     0, 1, 1,                                #บิตคงที่ ใน DMM STATUS
155     "Judge", # 1-°C, 0-°F.                  #ชื่อบิต Judge < 1-°C, 0-°F>
156     "Sign",                                #ชื่อบิต Sign < 1-minus sign, 0-no sign>
157     "BATT",                                #ชื่อบิต BATT <1-battery low>
158     "OL",                                  #ชื่อบิต OL <input overflow>
159 ]
160     OPTION1 = [                             #ตัวแปรกำหนด OPTION1
161                                           #กำหนดรูปแบบ DMM OPTION1
162     0, 1, 1,                                #บิตคงที่ ใน DMM OPTION1
163     "MAX",                                  #ชื่อบิต MAX < maximum >
164     "MIN",                                  #ชื่อบิต MIN < minimum >
165     "REL",                                  #ชื่อบิต REL < relative/zero mode >
166     "RMR",                                  #ชื่อบิต RMR <current value >
167 ]
168     OPTION2 = [                             #ตัวแปรกำหนด OPTION2
169                                           #กำหนดรูปแบบ DMM OPTION2
170     0, 1, 1,                                #บิตคงที่ ใน DMM OPTION2
171     "UL",                                  #ชื่อบิต UL < under load >
172     "PMA",                                  #ชื่อบิต PMA < maximum peak value >
173     "PMIN",                                #ชื่อบิต PMIN < minimum peak value >
174     0,                                      #บิตคงที่ ใน DMM OPTION2
175 ]
176
177     OPTION3 = [                             #ตัวแปรกำหนด OPTION3

```

```

"AC", #ข้อบิต AC เหม็ดวัด Vac , lac
"AUTO", #ข้อบิต AUTO < 1-automatic mode, 0-manual >
"VAHZ", #ข้อบิต VAHZ
]
OPTION4 = [
    #ตัวแปรกำหนด OPTION4
    #กำหนดรูปแบบ DMM OPTION4
    0, 1, 1, 0, #บิตลงที่ ใน DMM OPTION4
    "VBAR", #ข้อบิต AUTO <1-VBAR pin is connected to V>
    "Hold", #ข้อบิต Hold < hold mode >
    "LPF",.. #ข้อบิต LPF < low-pass-filter feature is activated >
]
def parse(packet):
    #ฟังก์ชัน ชื่อ parse() ใช้จำแนกแตกข้อมูล packet
    d_range, \
    d_digit4, d_digit3, d_digit2, d_digit1, d_digit0, \
    d_function, d_status, \
    d_option1, d_option2, d_option3, d_option4 = struct.unpack("B"*12, packet) #จำแนก
ข้อมูล
mode = FUNCTION[d_function][0] #กำหนดตัวแปร mode
m_range = FUNCTION[d_function][1][d_range] #กำหนดตัวแปร m_range แทน range
unit = FUNCTION[d_function][2] #กำหนดตัวแปร unit แทนหน่วย

```

```

203     d_options = (d_status, d_option1, d_option2, d_option3, d_option4)      #กำหนดตัว
204     แปล
205     OPTIONS = (STATUS, OPTION1, OPTION2, OPTION3, OPTION4)      #กำหนดตัว
206     แปล
207     for d_option, OPTION in zip(d_options, OPTIONS):
208         bits = get_bits(d_option, OPTION)      #เรียกใช้ฟังก์ชัน get_bits()
209         options.update(bits)      #ทำการอัปเดตตัวแปร options
210         current = None      #กำหนดตัวแปรชื่อ current แทนโหมดกระแสไฟฟ้า
211         if options["AC"] and options["DC"]:
212             raise ValueError
213         elif options["DC"]:      #ตรวจสอบว่าเป็นโหมดการวัดไฟฟ้ากระแสตรง DC
214             current = "DC"      #กำหนดค่าตัวแปรชื่อ current แทนโหมดกระแสไฟฟ้า
215         elif options["AC"]:      #ตรวจสอบว่าเป็นโหมดการวัดไฟฟ้ากระแสสลับ AC
216             current = "AC"      #กำหนดค่าตัวแปรชื่อ current แทนโหมดกระแสสลับ
217         operation = "normal"
218         if options["UL"]:      #ตรวจสอบว่าเป็น under load
219             operation = "underload"      #กำหนดค่าตัวแปรชื่อ operation ด้วย
220         underload
221         elif options["OL"]:      #ตรวจสอบว่าเป็น over load
222             operation = "overload"      #กำหนดค่าตัวแปรชื่อ operation ด้วย overload
223         if options["AUTO"]:      #ตรวจสอบว่าเป็น AUTO
224             mrange = "auto"      #กำหนดค่าตัวแปรชื่อ mrange ด้วย auto
225         else:
226             mrange = "manual"      #กำหนดค่าตัวแปรชื่อ mrange ด้วย manual
227         if options["BATT"]:      #ตรวจสอบว่าเป็น BATT
228             battery_low = True      #กำหนดค่าตัวแปรชื่อ battery_low ด้วย true

```

```

229     else:
230         battery_low = False           #กำหนดค่าตัวแปรชื่อ battery_low ด้วย False
231     # relative measurement mode
232     if options["REL"]:                 #ตรวจสอบว่าเป็น REL
233         relative = True               #กำหนดค่าตัวแปรชื่อ relative ด้วย true
234     else:
235         relative = False              #กำหนดค่าตัวแปรชื่อ relative ด้วย False
236     # data hold mode
237     if options["Hold"]:                #ตรวจสอบว่าเป็น Hold
238         hold = True                   #กำหนดค่าตัวแปรชื่อ hold ด้วย true
239     else:
240         hold = False                  #กำหนดค่าตัวแปรชื่อ hold ด้วย False
241         peak = None                  #กำหนดค่าตัวแปรชื่อ peak ด้วย None
242     if options["MAX"]:                 #ตรวจสอบว่าเป็น options["MAX"]:
243         peak = "max"                 #กำหนดค่าตัวแปรชื่อ peak ด้วย max
244     elif options["MIN"]:              #ตรวจสอบว่าเป็น options["MIN "]:
245         peak = "min"                 #กำหนดค่าตัวแปรชื่อ peak ด้วย min
246     if mode == "current" and options["VBAR"]: #ถ้าอยู่ในวัดกระแส และ VBAR
247         pass                         #ผ่านไป
248     elif mode == "current" and not options["VBAR"]: #ถ้าอยู่ในวัดกระแส และ ไม่มี VBAR
249         pass                         #ผ่านไป
250
251     if options["VAHZ"] and not options["Judge"]: #ถ้ามี options["VAHZ"]
252                                                #และไม่มี options["Judge"]
253         mode = "frequency"           #กำหนดค่าตัวแปรชื่อ peak ด้วย frequency
254         unit = "Hz"                  #กำหนดค่าตัวแปรชื่อ unit ด้วย Hz

```

```

255         m_range = (1e0, 1, "H")           #กำหนดค่าตัวแปรชื่อ m_range ด้วยช่วง
256     2200.0
257     elif (options["VAHZ"] or mode == "frequency") and options["Judge"]:
258         mode = "duty_cycle"                 #กำหนดค่าตัวแปรชื่อ mode ด้วย duty_cycle
259         unit = "%"                          #กำหนดค่าตัวแปรชื่อ unit ด้วย %
260         m_range = (1e0, 1, "%")            #กำหนดค่าตัวแปรชื่อ m_range ด้วยช่วง
261     2200.0
262
263     if mode == "temperature" and options["VBAR"]: #ถ้าอยู่ในโหมดวัดอุณหภูมิ
264         m_range = (1e0, 1, "deg")           #กำหนดค่าตัวแปรชื่อ m_range ด้วยช่วง 2200.0
265     elif mode == "temperature" and not options["VBAR"]:
266         m_range = (1e0, 2, "deg")           #กำหนดค่าตัวแปรชื่อ m_range ด้วยช่วง 220.00°C and
267     °F
268
269     digits = [d_digit4, d_digit3, d_digit2, d_digit1, d_digit0] #กำหนดค่าให้ตัวแปรแทนตัวเลข
270     digits = [DIGITS[digit] for digit in digits]                 #ทำการตรวจสอบตัวเลข
271     display_value = 0
272     for i, digit in zip(range(5), digits):
273         display_value += digit*(10**(4-i))   #รวมค่าตัวเลขให้ตัวแปร display_value
274     if options["Sign"]:
275         display_value = display_value * -1   #กำหนดค่าติดลบให้ตัวแปร display_value
276     display_value = Decimal(display_value) / 10**m_range[1]      #ทำการหาค่าตามสเกล range
277     display_value = display_value.quantize(Decimal(1)/10**m_range[1])
278     display_unit = m_range[2]                 #กำหนดค่าหน่วยวัดให้ตัวแปร display_unit
279     value = float(display_value) * m_range[0] #กำหนดค่าที่วัดได้ให้ตัวแปร value
280

```

```

281     if operation != "normal": #ตรวจสอบการทำงานว่าปกติ
282         display_value = "" #การทำงานผิดปกติ เคลียร์ค่าให้ตัวแปร display_value
283         value = "" #การทำงานผิดปกติ เคลียร์ค่าให้ตัวแปร value
284     results = { #กำหนดค่าตัวแปรชื่อ results เพื่อใช้วิเคราะห์
285         "value": value,
286         "unit": unit,
287         "display_value": display_value,
288         "display_unit": display_unit,
289         "mode": mode,
290         "current": current,
291         "peak": peak,
292         "relative": relative,
293         "hold": hold,
294         "range": mrange,
295         "operation": operation,
296         "battery_low": battery_low
297     }
298     return results
299
300 def output_readable(results): #ฟังก์ชัน output_readable()
301     operation = results["operation"] #กำหนดค่าตัวแปรชื่อ operation
302     print('สถานะการทำงานของเครื่อง :{}'.format(operation)) #ส่งออกตัวแปรชื่อ operation
303     battery_low = results["battery_low"] #กำหนดค่าตัวแปรชื่อ battery_low
304     if operation == "normal": #ถ้าการทำงานปกติ
305         display_value = results["display_value"] #กำหนดค่าตัวแปรชื่อ display_value
306         display_unit = results["display_unit"] #กำหนดค่าตัวแปรชื่อ display_unit

```

```

307         line = "{value} {unit}".format(value=display_value, unit=display_unit) #กำหนดค่าตัว
308     แปล
309     else:
310         line = "-, การวัดค่าของมัลติมิเตอร์ {operation}ed!".format(operation=operation)
311     if battery_low:
312         line.append(" Battery low!")      #อัปเดตข้อความ Battery low! ให้ตัวแปร line
313     return line
314
315     CSV_FIELDS = ["value", "unit", "mode", "current", "operation"]      #กำหนดค่าตัวแปร
316                                     #ชื่อ CSV_FIELDS
317     def format_field(results, field_name):      #ฟังก์ชัน format_field
318         value = results[field_name]
319         if field_name == "value":      # field_name เป็น value
320             if results["operation"]=="normal":      #การทำงานปกติ
321                 return str(value)      #ส่งค่าตัว value ออกไปให้ภายนอกเรียกใช้
322             else:
323                 return ""      #ส่งค่าว่างเปล่าออกไปให้ภายนอกเรียกใช้
324             if value==None:      #ตรวจสอบ ถ้า value เป็น None
325                 return ""      #ส่งค่าว่างเปล่าออกไปให้ภายนอกเรียกใช้
326             elif value==True:      #ตรวจสอบ ถ้า value เป็น True
327                 return "1"      #ส่งค่า 1 ออกไปให้ภายนอกเรียกใช้
328             elif value==False:      #ฟังก์ชัน format_field
329                 return "0"      #ส่งค่า 0 ออกไปให้ภายนอกเรียกใช้
330         else:
331             return str(value)      #ส่งค่าตัว value ออกไปให้ภายนอกเรียกใช้
332

```

```

333 def output_csv(results):                                #ฟังก์ชัน output_csv
334     field_data = [format_field(results, field_name) for field_name in CSV_FIELDS]
335     line = " ".join(field_data)                        #ทำการต่อข้อมูลให้ตัวแปร line โดยมี ; คั่นไว้
336     return line
337
338 DIGIT_0_SOUNDS = {                                     #กำหนดค่าตัวแปร ชื่อ DIGIT_0_SOUNDS
339     '-': '/home/pi/MP34DMM/delete.mp3',               #ตำแหน่งไฟล์เสียงแทนเสียงพูด ลบ
340     '.': '/home/pi/MP34DMM/point.mp3',               #ตำแหน่งไฟล์เสียงแทนเสียงพูด จุด
341     '0': '/home/pi/MP34DMM/0.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 0
342     '1': '/home/pi/MP34DMM/1.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 1
343     '2': '/home/pi/MP34DMM/2.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 2
344     '3': '/home/pi/MP34DMM/3.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 3
345     '4': '/home/pi/MP34DMM/4.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 4
346     '5': '/home/pi/MP34DMM/5.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 5
347     '6': '/home/pi/MP34DMM/6.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 6
348     '7': '/home/pi/MP34DMM/7.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 7
349     '8': '/home/pi/MP34DMM/8.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 8
350     '9': '/home/pi/MP34DMM/9.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 9
351 }
352
353 DIGIT_1_SOUNDS = {                                     #กำหนดค่าตัวแปร ชื่อ DIGIT_1_SOUNDS
354     '-': '/home/pi/MP34DMM/delete.mp3',               #ตำแหน่งไฟล์เสียงแทนเสียงพูด ลบ
355     '.': '/home/pi/MP34DMM/point.mp3',               #ตำแหน่งไฟล์เสียงแทนเสียงพูด จุด
356     '0': '/home/pi/MP34DMM/0.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 0
357     '1': '/home/pi/MP34DMM/Ed.mp3',                  #ตำแหน่งไฟล์เสียงแทนเสียงพูด เอ็ด
358     '2': '/home/pi/MP34DMM/2.mp3',                   #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 2

```

```

359 '3': '/home/pi/MP34DMM/3.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 3
360 '4': '/home/pi/MP34DMM/4.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 4
361 '5': '/home/pi/MP34DMM/5.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 5
362 '6': '/home/pi/MP34DMM/6.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 6
363 '7': '/home/pi/MP34DMM/7.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 7
364 '8': '/home/pi/MP34DMM/8.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 8
365 '9': '/home/pi/MP34DMM/9.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 9
366 }
367 DIGIT_2_SOUNDS = {                    #กำหนดค่าตัวแปร ชื่อ DIGIT_2_SOUNDS
368     '-': '/home/pi/MP34DMM/delete.mp3', #ตำแหน่งไฟล์เสียงแทนเสียงพูด ลบ
369     '.': '/home/pi/MP34DMM/point.mp3',  #ตำแหน่งไฟล์เสียงแทนเสียงพูด จุด
370     '1': '/home/pi/MP34DMM/10.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 10
371     '2': '/home/pi/MP34DMM/20.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 20
372     '3': '/home/pi/MP34DMM/30.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 30
373     '4': '/home/pi/MP34DMM/40.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 40
374     '5': '/home/pi/MP34DMM/50.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 50
375     '6': '/home/pi/MP34DMM/60.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 60
376     '7': '/home/pi/MP34DMM/70.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 70
377     '8': '/home/pi/MP34DMM/80.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 80
378     '9': '/home/pi/MP34DMM/90.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 90
379 }
380 DIGIT_3_SOUNDS = {                    #กำหนดค่าตัวแปร ชื่อ DIGIT_3_SOUNDS
381     '-': '/home/pi/MP34DMM/delete.mp3', #ตำแหน่งไฟล์เสียงแทนเสียงพูด ลบ
382     '.': '/home/pi/MP34DMM/point.mp3',  #ตำแหน่งไฟล์เสียงแทนเสียงพูด จุด
383     '1': '/home/pi/MP34DMM/100.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 100
384     '2': '/home/pi/MP34DMM/200.mp3',     #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 200

```

```

385 '3': '/home/pi/MP34DMM/300.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 300
386 '4': '/home/pi/MP34DMM/400.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 400
387 '5': '/home/pi/MP34DMM/500.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 500
388 '6': '/home/pi/MP34DMM/600.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 600
389 '7': '/home/pi/MP34DMM/700.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 700
390 '8': '/home/pi/MP34DMM/800.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 800
391 '9': '/home/pi/MP34DMM/900.mp3',      #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 900
392 }
393 DIGIT_4_SOUNDS = {                      #กำหนดค่าตัวแปร ชื่อ DIGIT_4_SOUNDS
394     '-': '/home/pi/MP34DMM/delete.mp3', #ตำแหน่งไฟล์เสียงแทนเสียงพูด ลบ
395     '.': '/home/pi/MP34DMM/point.mp3',   #ตำแหน่งไฟล์เสียงแทนเสียงพูด จุด
396     '1': '/home/pi/MP34DMM/1000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 1000
397     '2': '/home/pi/MP34DMM/2000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 2000
398     '3': '/home/pi/MP34DMM/3000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 3000
399     '4': '/home/pi/MP34DMM/4000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 4000
400     '5': '/home/pi/MP34DMM/5000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 5000
401     '6': '/home/pi/MP34DMM/6000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 6000
402     '7': '/home/pi/MP34DMM/7000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 7000
403     '8': '/home/pi/MP34DMM/8000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 8000
404     '9': '/home/pi/MP34DMM/9000.mp3',    #ตำแหน่งไฟล์เสียงแทนเครื่องหมาย 9000
405 }
406 UNIT_SOUNDS = {                        #กำหนดค่าตัวแปร ชื่อ UNIT_SOUNDS
407     'n': '/home/pi/MP34DMM/nano.mp3',    #ตำแหน่งไฟล์เสียงแทนเสียงพูด นาโน
408     'u': '/home/pi/MP34DMM/micro.mp3',   #ตำแหน่งไฟล์เสียงแทนเสียงพูด ไมโคร
409     'm': '/home/pi/MP34DMM/mili.mp3',    #ตำแหน่งไฟล์เสียงแทนเสียงพูด มิลิ
410     'k': '/home/pi/MP34DMM/kilo.mp3',    #ตำแหน่งไฟล์เสียงแทนเสียงพูด กิโล

```

```

411      'M': '/home/pi/MP34DMM/Mega.mp3',      #ตำแหน่งไฟล์เสียงแทนเสียงพูด เมกะ
412      'A': '/home/pi/MP34DMM/ampere.mp3',    #ตำแหน่งไฟล์เสียงแทนเสียงพูด แอมป์
413      'V': '/home/pi/MP34DMM/volt.mp3',      #ตำแหน่งไฟล์เสียงแทนเสียงพูด โวลท์
414      'O': '/home/pi/MP34DMM/ohm.mp3',       #ตำแหน่งไฟล์เสียงแทนเสียงพูด โอห์ม
415      'F': '/home/pi/MP34DMM/farad.mp3',     #ตำแหน่งไฟล์เสียงแทนเสียงพูด ฟารัด
416      'H': '/home/pi/MP34DMM/Hz.mp3',       #ตำแหน่งไฟล์เสียงแทนเสียงพูด เฮิร์ต
417      '%': '/home/pi/MP34DMM/percent.mp3',   #ตำแหน่งไฟล์เสียงแทนเสียงพูด เปอร์เซ็นต์
418  }
419  ERROR_SOUNDS = {                          #กำหนดค่าตัวแปร ชื่อ ERROR_SOUNDS
420      'OV': '/home/pi/MP34DMM/overload.mp3', #เสียงแทนเสียงพูด มัลติมิเตอร์ไม่สามารถวัด
421  ค่าได้
422      'msg1': '/home/pi/MP34DMM/error_msg1.mp3', #ไม่ได้ใช้งาน
423  }
424  MODE_SOUNDS = {                          #กำหนดค่าตัวแปร ชื่อ MODE_SOUNDS
425      'capacitance': '/home/pi/MP34DMM/capacitor.mp3', #เสียงแทนเสียงพูด โหมดตัวเก็บประจุ
426      'frequency': '/home/pi/MP34DMM/frequency.mp3', #เสียงแทนเสียงพูด โหมดวัดความถี่
427      'duty_cycle': '/home/pi/MP34DMM/duty_cycle.mp3', #เสียงแทนเสียงพูด โหมดวัด duty
428  cycle
429      'continuity': '/home/pi/MP34DMM/continuity.mp3', #เสียงแทนเสียงพูด โหมดวัดต่อเนื่อง
430      'diode': '/home/pi/MP34DMM/diode.mp3',      #เสียงแทนเสียงพูด โหมด Diode
431      'resistance': '/home/pi/MP34DMM/resistance.mp3', #เสียงแทนเสียงพูด โหมดตัวต้านทาน
432      'voltage': '/home/pi/MP34DMM/voltage.mp3',  #เสียงแทนเสียงพูด โหมดวัด
433  แรงดันไฟฟ้า
434      'current': '/home/pi/MP34DMM/current.mp3',  #เสียงแทนเสียงพูด โหมดวัด
435  กระแสไฟฟ้า
436      'DC': '/home/pi/MP34DMM/DC.mp3',           #เสียงแทนเสียงพูด กระแสตรง

```

```

437     'AC': '/home/pi/MP34DMM/AC.mp3',          #เสียงแทนเสียงพูด กระแสกลับ
438 }
439 try:
440     ser = serial.Serial(port = '/dev/ttyUSB0',    #ทำกำหนดรูปแบบการติดต่อสื่อสาร
441                          baudrate = 19200,        #ทำกำหนด baudrate เป็น 19200
442                          bytesize=serial.SEVENBITS, #ทำกำหนด bytesize เป็น 7 บิต
443                          stopbits = serial.STOPBITS_ONE, #ทำกำหนด stopbits เป็น 1 บิต
444                          parity = serial.PARITY_ODD,   #ทำกำหนด parity เป็น odd
445                          timeout=1.0)              # default timeout
446 except serial.SerialException as e:
447     sys.exit(e)                                #ออกจากระบบถ้าไม่สามารถเปิดพอร์ตอนุกรม
448 ได้
449
450 ser.setDTR(True)                             #ตั้งค่า DTR ให้เป็นลอจิก 1
451 ser.setRTS(False)                            #ตั้งค่า RTS ให้เป็นลอจิก 0
452 #sound initial
453 mixer.init()                                #กำหนดค่าเริ่มต้นให้กับตัวเล่นเสียง
454 while True:
455     f_Minus = False
456     ser.setDTR(True)                         #ตั้งค่า DTR ให้เป็นลอจิก 1
457     line = ser.readline().strip()            #อ่านค่าจากพอร์ตอนุกรม RS232 ที่ละบรรทัด
458     ser.flushInput()                          #ทำการเคลียร์บัฟเฟอร์อินพุต
459     ser.setDTR(False)                        #ตั้งค่า DTR ให้เป็นลอจิก 0
460     if len(line)==12:                         #ตรวจสอบจำนวนข้อมูลที่รับเข้ามาทางพอร์ต
461
462     try:

```

```

463         results = parse(line)                #จำแนกข้อมูลให้ตัวแปร results
464     except Exception as e:
465         print('Error "{}" packet from multimeter: {}'.format(e,line)) #แสดงผล Error
466     if results['operation'] == 'overload' or results['operation'] == 'underload':
467         #เช็คว่า overload/underload
468         dir_sound = []                        #สร้างตัวแปรเก็บตำแหน่งเสียง ชื่อ dir_sound
469         dir_sound.append(ERROR_SOUNDS['OV'])  #กำหนดเสียง Error ให้ตัวแปร
470     dir_sound
471     print('ส่งออกไฟล์เสียงการแจ้งเตือน : {}'.format(dir_sound)) #แสดงผลแจ้งเตือน
472     for j in range(len(dir_sound)):
473         mixer.music.load(dir_sound[j])        #ทำการโหลดไฟล์เสียงเข้าระบบ
474         mixer.music.play()                    #ทำการส่งออกเสียงไปลำโพง
475         time.sleep(1)                         #หยุดรอ 1 วินาที
476     else:
477         d_Join = output_csv(results).split(';') #เรียกใช้ฟังก์ชัน output_csv() และแตก
478         print('การจำแนกข้อมูลระบบ : {}'.format(d_Join)) #พิมพ์แสดงผลการจำแนกข้อมูล
479     ระบบ
480         dir_sound = []                        #สร้างตัวแปรเก็บตำแหน่งเสียง ชื่อ dir_sound
481         dir_sound.append(MODE_SOUNDS[d_Join[2]]) #ทำการโหลดไฟล์เสียงเข้าโหมดวัด
482         if d_Join[2] == 'current' or d_Join[2] == 'voltage':
483             dir_sound.append(MODE_SOUNDS[d_Join[3]])
484         print('ส่งออกไฟล์เสียงโหมดการวัด : {}'.format(dir_sound)) #พิมพ์แสดงผลโหมดการ
485     วัด
486         for j in range(len(dir_sound)):
487             mixer.music.load(dir_sound[j])        #ทำการโหลดไฟล์เสียงเข้าระบบ
488             mixer.music.play()                    #ทำการส่งออกเสียงไปลำโพง

```

```

489         time.sleep(3)
490
491     UT61E_DMM = output_readable(results)      #เรียกใช้ฟังก์ชัน output_readable()
492     Print ('ค่าที่วัดได้จาก UT61E : {}'.format(UT61E_DMM)) #แสดงผลค่าที่วัดได้
493     d_sound = UT61E_DMM.split()              #กำหนดค่าให้ตัวแปร d_sound
494     print('ส่งออกไฟล์เสียงค่าที่วัดได้จาก UT61E : {}'.format(d_sound)) #แสดงผลไฟล์เสียง
495     dir_sound = []
496     if float(d_sound[0]) < 0 :                 #ตรวจสอบว่าค่าติดลบ
497         dir_sound.append(DIGIT_1_SOUNDS['-']) #ส่งเสียงติดลบ
498         f_Minus = True                        #กำหนดค่า f_Minus เป็น True
499         N_Buff = d_sound[0].split('.')        #ทำการแตกค่าข้อมูล
500         Decimal_Number = N_Buff[0]            #จำนวนเต็ม
501         Float_Number = N_Buff[1]             #จำนวนทศนิยม
502         Digital_Number = int(Decimal_Number)
503         if f_Minus == True :
504             Digital_Number = Digital_Number * (-1) #ทำให้เป็นค่าบวก
505             f_Minus = False
506             Digital_N3 = int(Digital_Number / 1000) #ตัวแปร Digital_N3 เลขหลักพัน
507             M_Digital_N3 = Digital_Number % 1000
508             Digital_N2 = int(M_Digital_N3 / 100)   #ตัวแปร Digital_N3 เลขหลักร้อย
509             M_Digital_N2 = M_Digital_N3 % 100
510             Digital_N1 = int(M_Digital_N2 / 10)    #ตัวแปร Digital_N3 เลขหลักสิบ
511             Digital_N0 = int(M_Digital_N2 % 10)
512             Digital_N3 = str(Digital_N3)           #ตัวแปร Digital_N3 เลขหลักหน่วย
513             if Digital_N3 != '0':
514                 dir_sound.append(DIGIT_4_SOUNDS[Digital_N3]) #ทำการโหลดไฟล์หลัก(1000)

```

```

515     Digital_N2 = str(Digital_N2)
516     if Digital_N2 != '0':
517         dir_sound.append(DIGIT_3_SOUNDS[Digital_N2]) #ทำการโหลดไฟล์หลัก(100)
518     Digital_N1 = str(Digital_N1)
519     if Digital_N1 != '0':
520         dir_sound.append(DIGIT_2_SOUNDS[Digital_N1]) #ทำการโหลดไฟล์หลัก(10)
521     if Digital_Number != 0:
522         Digital_N0 = str(Digital_N0)
523         if Digital_Number < 10:
524             if Digital_N0 != '0':
525                 dir_sound.append(DIGIT_0_SOUNDS[Digital_N0]) #ทำการโหลดหลักเพิ่มเติม
526             else:
527                 if Digital_N0 != '0':
528                     dir_sound.append(DIGIT_1_SOUNDS[Digital_N0]) #ทำการโหลดหลักเพิ่มเติม
529                 else:
530                     dir_sound.append(DIGIT_1_SOUNDS['0']) #ทำการโหลดหลักหน่วย
531                     dir_sound.append(DIGIT_1_SOUNDS['.']) #ทำการโหลดเสียงจุด
532                 print('ส่งออกไฟล์เสียงจำนวนเต็ม : {}'.format(dir_sound)) #พิมพ์ออกไฟล์เสียง
533                 for i in range(len(dir_sound)):
534                     mixer.music.load(dir_sound[i]) #ทำการโหลดไฟล์เสียงเข้าระบบ
535                     mixer.music.play() #ส่งออกไฟล์เสียงไปลำโพง
536                     time.sleep(1.2) #หยุดรอ 1.2 วินาที
537                 dir_sound = [DIGIT_0_SOUNDS[x] for x in Float_Number] #ไฟล์เสียงทศนิยม
538                 print('ส่งออกไฟล์เสียงหลังจุดทศนิยม : {}'.format(dir_sound)) #พิมพ์ออกไฟล์เสียง
539                 for i in range(len(dir_sound)):
540                     mixer.music.load(dir_sound[i]) #ทำการโหลดไฟล์เสียงเข้าระบบ

```

```

541 mixer.music.play()                #ส่งออกไฟล์เสียงไปลำโพง
542 time.sleep(1)                     #หยุดรอ 1 วินาที
543 dir_sound = [UNIT_SOUNDS[x] for x in d_sound[1]] #กำหนดไฟล์เสียงหน่วยวัด
544 print('ส่งออกไฟล์เสียงหน่วยวัด : {}'.format(dir_sound)) #พิมพ์ออกไฟล์เสียง
545 for i in range(len(dir_sound)):
546     mixer.music.load(dir_sound[i]) #ทำการโหลดไฟล์เสียงเข้าระบบ
547     mixer.music.play()             #ส่งออกไฟล์เสียงไปลำโพง
548     time.sleep(1)                  #หยุดรอ 1 วินาที
549     ser.flushInput()               #เคลียร์บัฟเฟอร์อินพุตพอร์ตอนุกรม
550     time.sleep(3)                  #หยุดรอ 3 วินาที ก่อนอ่านค่ารอบต่อไป
551     ser.close()                    #ทำการปิดพอร์ตอนุกรม

```

ภาคผนวก ง

แบบทดลอง เรื่อง กฎของโอห์ม

รายงานผลการทดลองครั้งที่.....

ผู้ร่วมทำการทดลอง

1. รหัสประจำตัว.....

2. รหัสประจำตัว.....

3. รหัสประจำตัว.....

4. รหัสประจำตัว.....

วันที่..... Labsection..... กลุ่มที่.....

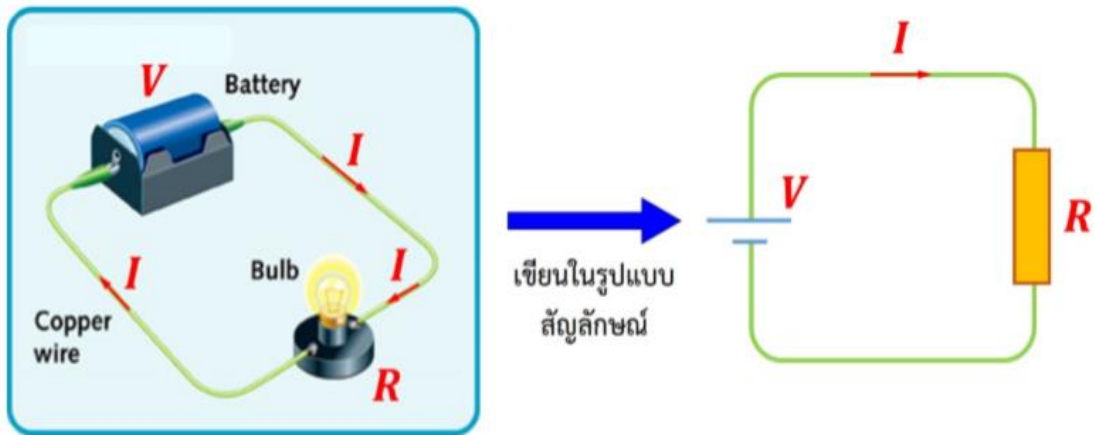
การทดลองที่ 7 กฎของโอห์ม วัตถุประสงค์ (Objectives)

1.

2.

ข้อมูลการทดลอง

ตอนที่ 1: กฎของโอห์ม



ภาพที่ 2 การต่อวงจรสำหรับวัดปริมาณไฟฟ้าต่าง ๆ ตามกฎของโอห์ม

แถบสีที่ระบุบนตัวต้านทาน คือ.....-.....-.....

ค่าความต้านทานที่ใช้ (R_{th}) = Ω

ตารางที่ 1 ตารางแสดงการวัดค่าความต่างศักย์ (V) และกระแสไฟฟ้า (I) ตามกฎของโอห์ม

จำนวน อ่านไฟฉาย (ก้อน)	ความต่างศักย์ของ อ่านไฟฉาย (V)	ความต่างศักย์ V (V)	กระแสไฟฟ้า I (mA)	ความต้านทาน R (Ω)

จากตารางที่ 1

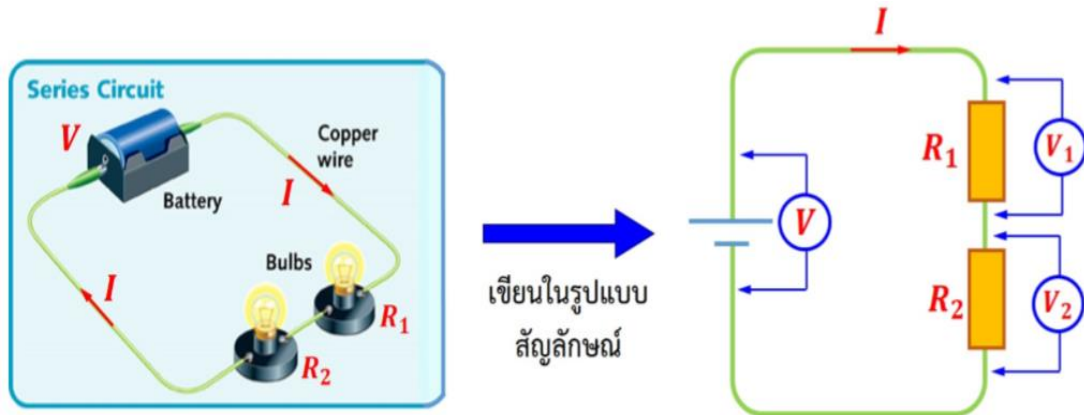
ค่าความต้านทานเฉลี่ยจากการทดลอง (R_{exp}) = Ω

ค่าความต้านทานทางทฤษฎี (R_{th}) = Ω

ร้อยละความคลาดเคลื่อนของค่าความต้านทาน R (%error) = %

ตอนที่ 2: การต่อตัวต้านทานในวงจรไฟฟ้า

ก. วงจรแบบอนุกรม (Serial circuit)

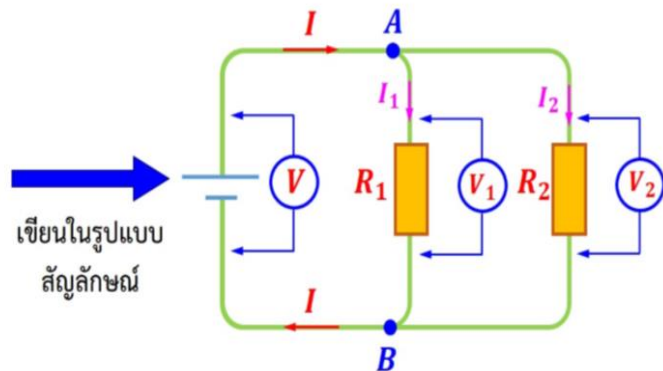
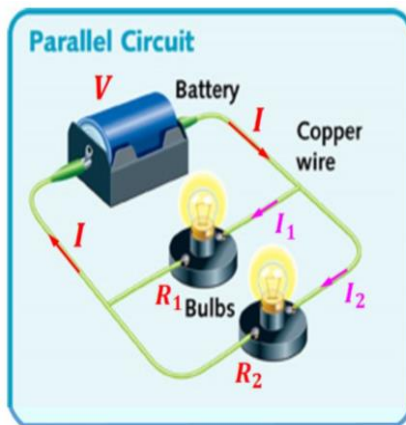


แถบสีที่ระบุบนตัวต้านทานตัวที่ 1 คือ
 ค่าความต้านทานที่ใช้ตัวที่ 1 (R_1 (th)) = Ω
 แถบสีที่ระบุบนตัวต้านทานตัวที่ 2 คือ
 ค่าความต้านทานที่ใช้ตัวที่ 2 (R_2 (th)) = Ω
 ค่าความต้านทานรวม ($R_{รวม}$ (th)) = Ω
 ค่ากระแสไฟฟ้าที่วัดได้ (I) = mA

ตารางที่ 2 ตารางแสดงการวัดค่าความต่างศักย์ (V) ในวงจรอนุกรม

ตัวต้านทาน R_i	ความต่างศักย์ที่วัดได้ V_i (V)	ความต้านทานที่ คำนวณได้ $R_{i(exp)} (\Omega)$	%error (%)
$R_{(i=1)}$			
$R_{(i=2)}$			
รวม			

ข. วงจรแบบขนาน (Parallel circuit)



เขียนในรูปแบบ
สัญลักษณ์

แถบสีที่ระบุบนตัวต้านทานตัวที่ 1 คือ
 ค่าความต้านทานที่ใช้ตัวที่ 1 ($R_1(th)$) = Ω
 แถบสีที่ระบุบนตัวต้านทานตัวที่ 2 คือ
 ค่าความต้านทานที่ใช้ตัวที่ 2 ($R_2(th)$) = Ω
 ค่าความต้านทานรวม ($R_{รวม}(th)$) = Ω
 ค่าความต่างศักย์ที่วัดได้ (V) = V

ตารางที่ 3 ตารางแสดงการวัดค่ากระแสไฟฟ้า (I) ในวงจรอนุกรมขนาน

ตัวต้านทาน R_i	กระแสไฟฟ้าที่วัดได้ I_i (mA)	ความต้านทานที่ คำนวณได้ $R_i(exp) (\Omega)$	%error (%)

สรุปและวิเคราะห์ผลการทดลอง

.....

