

ระบบสั่งงานด้วยเสียง
Voice Command System

ศรียะ พรหมสมันต์
SIRIYA PHROMSANTH

อิทธิ เตือนแจริญ
ITTI THUENCHAREEN

เอกวิทย์ วงษ์ไพฑูริย์
EAKAVIT WONGWAIPIRIT

ปริญญาบัตรฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2558

ระบบสั่งงานด้วยเสียง

Voice Command System

โดย

นายสุริยะ พรหมเสนีย์

นายอิทธิ เกื้อนเจริญ

นายเอกวิทย์ ว่องไวพนานิจ

อาจารย์ที่ปรึกษา

อาจารย์ ชินภัทร นันทจิวารักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสั่งงานด้วยเสียง

Voice Command System

ผู้จัดทำ นายสุริยะ พรหมเสนีย์ รหัสนักศึกษา 53011793

นายอิทธิ เตื่อนเจริญ รหัสนักศึกษา 53011936

นายเอกวิทย์ ว่องไวพนานิจ รหัสนักศึกษา 53011963

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

.....
(อาจารย์ ชินภัทร นันทจิวารชัย)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	ระบบสั่งงานด้วยเสียง		
นักศึกษา	นายสุริยะ	พรหมเสนีย์	รหัสประจำตัว 53011793
	นายอิทธิ	เถื่อนเจริญ	รหัสประจำตัว 53011936
	นายเอกวิทย์	ว่องไวพนานิจ	รหัสประจำตัว 53011963
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์		
ปีการศึกษา	2556		
อาจารย์ที่ปรึกษาปริญญานิพนธ์	อาจารย์ชินภัทร นันทจิวารักษ์		

บทคัดย่อ

โครงการนี้ อธิบายการออกแบบและสร้างระบบสั่งงานด้วยเสียงพูด โดยมีส่วนประกอบหลักอยู่ 3 ส่วน ได้แก่ 1.ส่วนรับรู้เสียงและประมวลเสียงพูด 2.ส่วนประมวลผลลำดับคำสั่งเสียง 3.ส่วนควบคุมอุปกรณ์ไฟฟ้า สามารถควบคุมอุปกรณ์ไฟฟ้าดังนี้ 1.โทรศัพท์บ้าน สามารถควบคุมการโทรออก-รับสาย ผ่านเสียงได้ 2.หลอดไฟ สามารถควบคุมการเปิด-ปิด บริเวณต่างๆ ของตัวบ้าน ผ่านเสียงได้ และยังออกแบบเต้าเสียบสำหรับอุปกรณ์ไฟฟ้าอื่นๆ โดยสามารถควบคุมการเปิด-ปิดของอุปกรณ์ไฟฟ้านั้นได้

Thesis Title	Voice Command System
Student	Mr. Suriya Phromsenee Student ID 53011793 Mr. Itti Thuencharern Student ID 53011936 Mr. Eakavit Wongwaipananij Student ID 53011963
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2013
Thesis Advisor	Mr. Chinnapat Nantajiwakornchai

Abstract

This project describes the design and construction of voice command system. The system divides in three parts 1.Voice Recognize and Voice Process Section. 2.Sequence of Voice Commands Process Section. 3.Controlled Electric Devices Section. The system can be controlled the electric devices including 1.Telephone can make a call-receive via voice command. 2.Lamps can be controlled turn on-off in areas of the house via voice command. A controlled electrical outlet for electric devices, which can be controlled turn on-off.

กิตติกรรมประกาศ

ปริญญาานิพนธ์นี้สำเร็จคล่องไปได้ด้วยดี ทั้งนี้เพราะได้รับคำแนะนำและคำปรึกษาจาก อาจารย์ ชินภัทร นันทจิรากรชัย ซึ่งเป็นอาจารย์ที่ปรึกษาในการทำปริญญาานิพนธ์ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

นอกจากนี้ขอขอบพระคุณทุกๆ ท่านที่ให้ความช่วยเหลือ ตลอดจนให้คำแนะนำต่างๆจนทำให้ปริญญาานิพนธ์นี้สำเร็จโดยสมบูรณ์ได้

คุณค่าและประโยชน์อันพึงมีในปริญญาานิพนธ์นี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

สุวิภา พรหมเสนีย์

นายสุริยะ พรหมเสนีย์
อภีร์ สีอ่อน ๑๐๖

นายอิทธิ เถื่อนเจริญ
เอกวิทย์ ว่องไวพนานิจ

นายเอกวิทย์ ว่องไวพนานิจ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูปภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตของงานวิจัย.....	1
1.4 ผลที่คาดว่าจะได้รับ.....	1
1.5 ส่วนประกอบของปริญญานิพนธ์.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1 UART, Universal Asynchronous Receiver Transmitter.....	3
2.1.1 UART เบอร์ 8250	3
2.1.2 UART เบอร์ 16450.....	4
2.2 ไมโครคอนโทรลเลอร์.....	4
2.2.1 โครงสร้างทั่วไป.....	4
2.2.1.1 หน่วยประมวลผลกลางหรือซีพียู.....	4
2.2.1.2 หน่วยความจำ.....	4
2.2.1.3 ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต.....	4
2.2.1.4 ช่องทางเดินของสัญญาณ หรือบัส.....	5
2.2.1.5 วงจรกำเนิดสัญญาณนาฬิกา.....	5
2.3 อุปกรณ์รู้จำเสียง.....	5
2.3.1 อุปกรณ์วิเคราะห์เสียงพูด.....	5
2.4 การรู้จำเสียงพูด.....	6
2.4.1 หลักการพื้นฐานของการรู้จำเสียง.....	7
2.4.2 โมเดลภาษา และกฎไวยากรณ์.....	8
2.4.3 การใช้งานจริงในแบบไม่เจาะจงผู้พูด.....	10
2.4.4 เทคโนโลยีที่เกี่ยวข้อง.....	10

	หน้า
2.4.4.1 DSR.....	11
2.4.4.2 การระบุตัวตนด้วยเสียง.....	11
2.5 รีเลย์.....	12
2.5.1 หลักการเบื้องต้น.....	12
บทที่ 3 การออกแบบและการคำนวณ.....	13
3.1 การเลือกใช้อุปกรณ์.....	13
3.1.1 อุปกรณ์ Voice Recognition.....	13
3.1.2 คุณสมบัติของ EasyVR Arduino Shield.....	13
3.1.3 Arduino Mega 2560.....	14
3.1.4 ULN3803APG.....	15
3.1.5 รีเลย์.....	16
3.1.6 หลอดไฟ.....	16
3.1.7 เต้าเสียบ.....	17
3.1.8 โทรศัพท์บ้าน.....	17
3.2 การออกแบบวงจรควบคุมหลอดไฟ และเต้าเสียบ.....	18
3.3 การออกแบบวงจรควบคุมโทรศัพท์.....	19
3.4 การออกแบบกลุ่มคำสั่ง.....	21
3.4.1 โปรแกรมในการควบคุมอุปกรณ์ไฟฟ้า.....	23
3.4.2 โปรแกรมในการโทรศัพท์.....	25
3.4.3 คำสั่งโปรแกรมในส่วนตรวจสอบการทำงานของโปรแกรม.....	27
3.4.3.1 โปรแกรมตรวจสอบการทำงานของโปรแกรมในด้านเวลา.....	27
บทที่ 4 ผลการทดลอง.....	28
4.1 จุดประสงค์การทดลอง.....	28
4.2 วิธีการทดลอง.....	28
4.2.1 การทดลองสั่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน.....	28
4.2.2 การทดลองสั่งด้วยเสียงเพื่อโทรออก.....	29
4.2.3 การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ.....	29
4.2.4 การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล.....	30
4.2.5 การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ.....	30
4.2.6 การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ.....	31
4.3 ผลการทดลอง.....	31
4.3.1 การทดลองสั่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน.....	31

หน้า

4.3.2 การทดลองส่งด้วยเสียงเพื่อโทรออก.....	32
4.3.3 การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ.....	32
4.3.4 การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล.....	34
4.3.4.1 การทดลองสั่งให้ปิดหลอดไฟที่ตำแหน่งนั้นได้.....	34
4.3.4.2 การทดลองสั่งงานเปิดพัดลม.....	34
4.3.4.3 การทดลองรับสายของโทรศัพท์.....	34
4.3.5 การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ.....	35
4.3.6 การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ.....	35
บทที่ 5 สรุปผลการทดลอง.....	36
บรรณานุกรม.....	40
ภาคผนวก.....	41
ภาคผนวก ก. Source Code ของโปรแกรม.....	42
- Source Code ของโปรแกรม Arduino.....	43
ภาคผนวก ข. วงจรที่ใช้.....	72
- วงจรควบคุมการเปิดปิดของหลอดไฟและอุปกรณ์ไฟฟ้า.....	73
- วงจรควบคุมการโทรออกและรับสายของโทรศัพท์.....	74

สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงความสัมพันธ์ระหว่างวงจรควบคุมหลอดไฟและตำแหน่งทำงาน.....	18
ตารางที่ 3.2 แสดงความสัมพันธ์ระหว่างวงจรควบคุมโทรศัพท์และวงจรในโทรศัพท์.....	20
ตารางที่ 3.3 แสดงรายชื่อกลุ่มคำสั่งและสมาชิกในกลุ่มคำสั่ง.....	21
ตารางที่ 4.1 แสดงการทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ.....	32
ตารางที่ 4.2 แสดงผลการทดลองสั่งให้ปิดหลอดไฟที่ตำแหน่งบันได.....	34
ตารางที่ 4.3 แสดงผลการทดลองสั่งงานเปิดพัดลม.....	34
ตารางที่ 4.4 แสดงผลการทดลองรับสายของโทรศัพท์.....	34
ตารางที่ 4.5 แสดงผลการทดลองประสิทธิภาพในระยะต่างๆ.....	35
ตารางที่ 4.6 แสดงผลการทดลองประสิทธิภาพในสภาพแวดล้อมต่างๆ.....	35

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 หลักการพื้นฐานของการรู้จำเสียง.....	8
รูปที่ 2.2 แสดงรูปร่างและสัญลักษณ์ของรีเลย์.....	12
รูปที่ 3.1 แสดงรูปร่างภายนอกของ EasyVR Arduino Shield.....	13
รูปที่ 3.2 แสดงรูปร่างภายนอกของ Arduino Mega 2560.....	15
รูปที่ 3.3 แสดงรูปร่างภายนอกของ ULN2803AP.....	15
รูปที่ 3.4 แสดงขาต่างๆและการเชื่อมต่อกันภายใน.....	16
รูปที่ 3.5 แสดงรูปร่างภายนอกของ HRS4-S-DC5V.....	16
รูปที่ 3.6 หลอดไฟขนาด 220/230V 3-5 W.....	17
รูปที่ 3.7 เต้าเสียบ.....	17
รูปที่ 3.8 โทรศัพท์บ้าน.....	17
รูปที่ 3.9 แสดงวงจรขับหลอดไฟ และเต้าเสียบ.....	19
รูปที่ 3.10 วงจรภายในโทรศัพท์.....	20
รูปที่ 3.11 แสดงวงจรควบคุมโทรศัพท์.....	21
รูปที่ 3.12 แสดงโฟลว์ชาร์ทการทำงานของระบบสั่งงานด้วยเสียง.....	23
รูปที่ 3.13 แสดงลำดับคำสั่งการเปิดหลอดไฟหน้าบ้าน.....	23
รูปที่ 3.14 แสดงลำดับคำสั่งการโทรออกของโทรศัพท์.....	25
รูปที่ 4.1 แสดงลำดับคำสั่งการเปิดหลอดไฟหน้าบ้าน.....	28
รูปที่ 4.2 แสดงลำดับคำสั่งการโทรออกของโทรศัพท์.....	29
รูปที่ 4.3 แสดงหน้าต่างโปรแกรม EasyVR Commander.....	30
รูปที่ 4.4 แสดงผลของการสั่งเปิดหลอดไฟหน้าบ้าน.....	31
รูปที่ 4.5 แสดงผลของตัวเลขบนจอโทรศัพท์.....	32
รูปที่ 5.1 แผนภูมิแสดงประสิทธิภาพในการรับคำสั่งในแต่ละคำ.....	37

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันมนุษย์เราใช้เทคโนโลยีเพื่ออำนวยความสะดวกในการดำเนินชีวิต ชีวิตประจำวันของเรามีเทคโนโลยีเข้าไปเกี่ยวข้องในทุกๆ ด้าน ในที่นี้จะกล่าวถึงระบบการสั่งการด้วยเสียง ซึ่งในปัจจุบันระบบนี้ได้มีการนำไปใช้งานบ้างแล้ว แต่ยังไม่เป็นที่แพร่หลายมากนัก ทางเราจึงได้นำเทคโนโลยีการสั่งการด้วยเสียงนี้มาศึกษา ในระบบการสั่งการด้วยเสียงนี้อยู่ 2 ส่วนใหญ่ๆ คือ 1. ส่วนของการรู้จำเสียง 2. ส่วนการควบคุมอุปกรณ์โดยใช้ไมโครคอนโทรลเลอร์

1.2 วัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อศึกษาการทำงานของ Easy VR Module
- 1.2.2 เพื่อศึกษาการควบคุมอุปกรณ์โดยใช้ไมโครคอนโทรลเลอร์

1.3 ขอบเขตของงานวิจัย

- 1.3.1 สามารถสั่งงานหลอดไฟให้เปิดปิดตามคำสั่งเสียง
- 1.3.2 สามารถสั่งงานอุปกรณ์ไฟฟ้าให้เปิดปิดตามคำสั่งเสียง
- 1.3.3 สามารถสั่งงานโทรศัพท์ให้โทรออกและรับสายตามคำสั่งเสียง

1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 ได้รับประสบการณ์ ในการเขียนโปรแกรมภาษาซี
- 1.4.2 ได้รับทักษะและประสบการณ์พื้นฐานเกี่ยวกับอิเล็กทรอนิกส์
- 1.4.3 รู้และเข้าใจระบบสั่งงานด้วยเสียง

1.5 ส่วนประกอบของปฏิญญานิพนธ์

ในปฏิญญานิพนธ์ฉบับนี้ แสดงถึงรายละเอียดของงานวิจัยที่จัดทำขึ้นมา ซึ่งประกอบด้วยหัวข้อต่างๆ รวมทั้งหมด 6 บท แต่ละบทกล่าวถึงดังต่อไปนี้

บทที่ 1 เป็นบทนำของปฏิญญานิพนธ์ ได้กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงการ ขอบเขตของงานวิจัย ผลที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎี และแนวความคิดที่ได้ศึกษา เพื่อเป็นข้อมูลที่ใช้ในการจัดทำโครงการระบบสั่งงานด้วย

บทที่ 3 กล่าวถึงการออกแบบ และการคำนวณค่าต่างๆ ที่ใช้ในการออกแบบระบบสั่งงานด้วยเสียง

บทที่ 4 ผลการทดลอง

บทที่ 5 สรุปและวิเคราะห์ผลการทดลอง

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 UART, Universal Asynchronous Receiver Transmitter

UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึง อุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบน คอมพิวเตอร์แล้ว UART ถือเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ ในรูปแบบอนุกรมแบบอะซิงโครนัสแล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะ ซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังทำการแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์ทราบด้วย เช่น อัตรา ความเร็วในการรับส่งข้อมูล, รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายถอดข้อมูล , เพรมข้อมูล เป็นต้น

ภายในUARTจะมีส่วนของวงจรสร้างอัตราการถ่ายถอดข้อมูลแบบโปรแกรมได้ (Programmable Baudrate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิตดังนั้นจึงกำหนดตัวหารให้อยู่ในช่วง 10 – 65,535

UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex) และฟูลดูเพล็กซ์ (Full Duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์ เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้น สามารถรับและส่งข้อมูลได้ในคราวเดียวกัน ชนิดของ UART ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ ใช้กันอยู่ 2 เบอร์ คือ

2.1.1 UART เบอร์ 8250

เป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่ง ข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆรุ่นจะต้อง สนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

2.1.2 UART เบอร์ 16450

ความสามารถรับส่งข้อมูลได้ด้วยความเร็ว 115, 200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนี้ยังเพิ่มส่วนของซีพียูรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่าเช่น เบอร์ TL 16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5 V และ +3 V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ด้วยความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน

2.2.1 โครงสร้างทั่วไป

โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

2.2.1.1 หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2.2.1.2 หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำ (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

2.2.1.3 ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ต

อินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิทช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

2.2.1.4 ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

2.2.1.5 วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการทำงานจังหวะ หากสัญญาณนาฬิกาที่มีความถี่สูงจังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

2.3 อุปกรณ์รู้จำเสียง (Voice Recognition Device)

การสื่อสารกับเครื่องคอมพิวเตอร์ในรูปของเสียงเป็นอีกขั้นตอนของการพัฒนาทางเทคโนโลยี ถึงแม้ในปัจจุบันนี้ยังมีปัญหาอยู่บ้างก็ตาม อุปกรณ์ที่ใช้ เช่น

2.3.1 อุปกรณ์วิเคราะห์เสียงพูด (Speech Recognition Device)

เป็นอุปกรณ์ที่พัฒนาโดยนักคอมพิวเตอร์และนักภาษาศาสตร์ เพื่อใช้รับสัญญาณเสียงที่มนุษย์พูด และแปลงเป็นสัญญาณดิจิทัลเก็บข้อมูลไว้ในคอมพิวเตอร์ ปัญหาที่สำคัญของอุปกรณ์ชนิดนี้คือผู้พูดแต่ละคนพูดด้วยน้ำเสียงและสำเนียงเฉพาะของแต่ละบุคคล จึงได้มีการแก้ปัญหาโดยให้คอมพิวเตอร์ได้เรียนรู้น้ำเสียงของผู้ที่ต้องการใช้งานในระยะเวลาหนึ่งก่อน เพื่อเก็บรูปแบบของน้ำเสียงและสำเนียงไว้ ซึ่งวิธีการนี้ทำให้อัตราการตีความเสียงผิดพลาดลดลงอย่างมาก ในระบบรับข้อมูลเสียงรุ่นแรก ๆ สามารถจดจำคำได้เพียงไม่กี่สิบคำเท่านั้น แต่ปัจจุบัน บริษัทไอบีเอ็มได้มีการสร้างอุปกรณ์ชื่อว่า Voice Type ซึ่งประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ที่สามารถจดจำคำได้ถึง 32,000 คำจากเสียงหลาย ๆ เสียง นิยมนำมาใช้กับผู้ที่ทำงานแล้วมือไม่ว่างพอที่จะกดแป้นพิมพ์ หรือผู้พิการ เช่น คนตาบอด ก็สามารถใช้น้ำเสียงทำงานกับคอมพิวเตอร์ได้โดยไม่ต้องใช้แป้นพิมพ์

ในปัจจุบัน สามารถแบ่งวิธีวิเคราะห์เสียงพูดได้เป็น 4 ประเภทใหญ่ ๆ ตามลักษณะออกเสียงคือ

2.3.1.1 Speaker Dependent Isolated Word Recognition วิเคราะห์เฉพาะผู้พูดที่เจาะจงและเป็นคำโดด

2.3.1.2 Speaker Dependent Continuous Word Recognition วิเคราะห์เฉพาะผู้พูดที่เจาะจงและเป็นคำต่อเนื่อง

2.3.1.3 Speaker Independent Isolated Word Recognition วิเคราะห์แบบไม่เจาะจงและเป็นคำโดด

2.3.1.4 Speaker Independent Continuous Word Recognition วิเคราะห์แบบไม่ที่เจาะจงและเป็นคำต่อเนื่อง

ความยากในการสร้างระบบวิเคราะห์จะเพิ่มขึ้นตามลำดับข้อที่กล่าวมา นอกจากนี้ ยังมีองค์ประกอบอื่น ๆ เช่น อารมณ์และน้ำเสียงของผู้พูดในขณะนั้น ซึ่งจะทำให้คำคำเดียวกันมีการออกเสียงที่ต่างกันไปได้อีกด้วย

2.4 การรู้จำเสียงพูด (Speech Recognition)

การรู้จำเสียงพูด (Speech Recognition) คือ การที่คอมพิวเตอร์สามารถรับรู้เสียงของมนุษย์ได้โดยอัตโนมัติ ในกรณีทั่วไปแล้วการรู้จำเสียงพูดจะหมายถึง การแปลงข้อมูลนำเข้าในรูปแบบเสียงพูดให้กลายเป็นข้อความตัวอักษร สำหรับมนุษย์ทั่วไป การที่เราสามารถฟังเสียงพูดและเข้าใจความหมายของบทสนทนา นั้น โดยปกติไม่ใช่เรื่องที่ยากลำบาก ดังนั้น เรามักจะคิดว่าทำให้คอมพิวเตอร์รับรู้และเข้าใจเสียงพูด น่าจะเป็นเรื่องที่ทำได้ไม่ยากเช่นเดียวกัน

แต่ในความเป็นจริงแล้ว การรู้จำเสียงพูดนั้นต้องอาศัยเทคโนโลยีขั้นสูง บางคนอาจคิดว่าเพียงแค่แปลงเสียงพูดที่เข้ามาให้กลายเป็นข้อความตัวอักษรง่ายๆ แค่นี้ก็น่าจะใช้งานได้แล้ว แต่ในสถานการณ์จริงนั้น การออกเสียงในบางครั้งก็เป็นที่ยากแม้แต่กับมนุษย์ด้วยกันเอง เช่น ในสถานการณ์การพูดคุยกันทางโทรศัพท์เกี่ยวกับตัวเลข เมื่อพูดถึงวันที่สามสิบเอ็ด มกราคม สองพันห้าร้อยยี่สิบเอ็ด (31 มกราคม 2521) เราก็อาจจะฟังผิดเป็น สองพันห้าร้อยยี่สิบเจ็ด (2527) ได้ เพราะเสียง “เอ็ด” กับเสียง “เจ็ด” มีความคล้ายคลึงกันมาก ในอีกกรณีหนึ่ง คือ เวลาเราฟังการประกาศเวลาทางวิทยุ เวลา “11:21:00 นาฬิกา” จะไม่ออกเสียงเป็น “สิบเอ็ดนาฬิกา ยี่สิบเอ็ดนาที ศูนย์วินาที” ตามปกติ แต่จะออกเสียงเป็น “สิบหนึ่งนาฬิกา ยี่สิบหนึ่งนาที ศูนย์วินาที” เพื่อหลีกเลี่ยงไม่ให้ฟังผิดเป็น “17:27:00 นาฬิกา” ทำให้มีโอกาสที่จะพบการออกเสียงที่ไม่ปกติได้ ดังนั้น ถึงแม้จะใช้เทคนิค “การแปลงเสียงตัวเป็นอักษร” แล้ว ก็ยังมีความยุ่งยากในการทำความเข้าใจความหมายของเสียงพูดอยู่

อีกตัวอย่างหนึ่งที่ใกล้เคียงการสื่อสารในชีวิตประจำวัน คือ หากมีการบอกเลขที่บัตร “3 2 5 5 5 7 9 0 0 6” โดยพูดเป็น “สาม สอง สองห้า เจ็ด เก้า ศูนย์สองตัว หก” การที่สามารถจะเข้าใจว่า

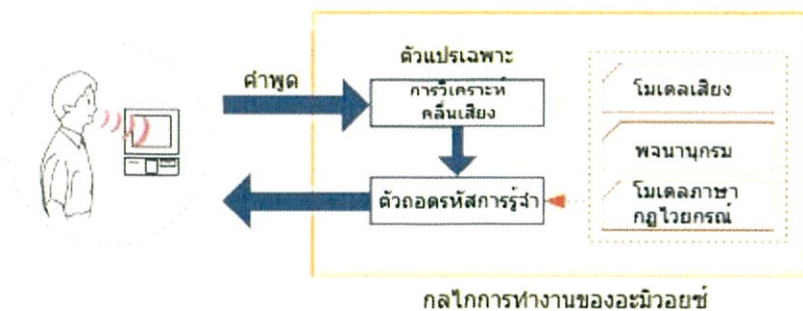
“ตอหง้า” คือ “5 5 5” และ “ศูนย์สองตัว” คือ “0 0” เพื่อให้สามารถให้ได้ผลลัพธ์เป็น “3 2 5 5 5 7 9 0 0 6” อย่างถูกต้องนั้น จะมีความยากขึ้นอีกระดับหนึ่ง ดังนั้นการนำการรู้จำเสียงไปใช้งานจริงนั้น ถือว่าเป็นเทคนิคขั้นสูงเลยทีเดียว

2.4.1 หลักการพื้นฐานของการรู้จำเสียง

หลักการพื้นฐานของการรู้จำเสียง จะคล้ายกับหลักการในการทำความเข้าใจภาษาที่มนุษย์เราใช้ในการสนทนาทั่วไป ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้ (1) การวิเคราะห์คลื่นเสียง (2) ตัวถอดรหัสการรู้จำ (3) โมเดลคลื่นเสียง (4) พจนานุกรม (5) โมเดลภาษา (กฎไวยากรณ์)

การวิเคราะห์คลื่นเสียงเป็นการวิเคราะห์เสียงที่ถูกส่งเข้ามาเพื่อให้ได้ค่าลักษณะเฉพาะที่เรียกว่า compact เพื่อแปลงเป็นข้อมูลเพิ่มเติมที่มีประโยชน์ต่อการรู้จำเสียง ตัวอย่างเช่น การใช้วิเคราะห์ทางคณิตศาสตร์แบบฟูเรียร์ในเทคโนโลยีการประมวลผลสัญญาณ ในกรณีของมนุษย์เรา เสียงที่เข้าไปในหูของเรา (การสั่นสะเทือนของเยื่อแก้วหู) จะถูกแปลงเป็นสัญญาณไฟฟ้าส่งไปยังสมอง แต่เสียงที่ได้รับมาก็ไม่ได้ถูกส่งไปยังสมองตามลักษณะดั้งเดิมทั้งหมด จะมีเพียงสัญญาณที่ถูกคัดเลือกจากระบบประสาทเท่านั้นที่จะถูกส่งไปยังสมองเพื่อนำไปใช้วิเคราะห์ในขั้นตอนต่อไป ซึ่งด้วยลักษณะกลไกการทำงานแบบเดียวกันนี้ การรู้จำเสียงจะแยกเอาลักษณะเฉพาะจากข้อมูลเสียงเฉพาะส่วนที่จำเป็นเท่านั้น ซึ่งขั้นตอนการเปลี่ยนสัญญาณเสียงที่เข้ามาให้เป็นข้อมูลที่มีประโยชน์ คือ การวิเคราะห์คลื่นเสียงนั่นเอง

ตัวถอดรหัสการรู้จำ เป็นส่วนประกอบที่เป็นใจกลางของระบบการรู้จำเสียง ซึ่งทำหน้าที่แปลงลักษณะเฉพาะของเสียงให้เป็นข้อความตัวอักษร หลักการสำคัญของการทำงานในส่วนนี้คือ “การตัดสินใจบนองค์ประกอบรวมของข้อมูลคลื่นเสียง และข้อมูลภาษา” ยกตัวอย่างเช่น วลี “กลบ เกลือ่นความผิด” คำว่า “กลบ” นั้นมีการออกเสียงคล้ายกับคำว่า “กบ” ซึ่งถึงแม้ว่าจะมีการออกเสียงผิดเป็น “กบเกลือ่น” มนุษย์เราก็จะยังสามารถฟังเข้าใจได้อย่างถูกต้อง โดยอาศัยข้อมูลภาษาและบริบทรอบข้างเข้ามาช่วยได้



รูปที่ 2.1 หลักการพื้นฐานของการรู้จำเสียง

การตัดสินใจบนองค์ประกอบรวมของข้อมูลของคลื่นเสียงและข้อมูลของภาษา เป็นวิธีการบนพื้นฐานของสถิติความน่าจะเป็น โดยในส่วนของข้อมูลคลื่นเสียงจะอาศัยโมเดลคลื่นเสียงเพื่อวิเคราะห์ลักษณะเฉพาะของคลื่นเสียงเพื่อหารายการสัญลักษณ์การออกเสียง และหาค่าความน่าจะเป็นของสัญลักษณ์การออกเสียงแต่ละตัว เช่น จากตัวอย่างก่อนหน้านี้ การออกเสียงคำว่า "กลบเกลื่อน" อาจวิเคราะห์ได้เป็นสัญลักษณ์การออกเสียง "กลบเกลื่อน" "กบเกลื่อน" "กลบเกื่อน" และ "กบเกื่อน" เป็นต้น โดยที่สัญลักษณ์การออกเสียงแต่ละตัวจะมีค่าความน่าจะเป็นกำกับอยู่ หลังจากนั้นจะนำสัญลักษณ์การออกเสียงที่ได้ไปเปรียบเทียบกับคำที่มีใน "พจนานุกรม" เพื่อตัดสัญลักษณ์การออกเสียงที่ไม่ตรงกับคำในภาษาที่พูดออกไป ในตัวอย่างนี้ จะเหลือเพียง "กลบเกลื่อน" และ "กบเกลื่อน" เท่านั้น เพราะคำว่า "เกื่อน" ไม่มีในพจนานุกรม ส่วน "โมเดลภาษา (หรือกฎไวยากรณ์)" จะช่วยในการวิเคราะห์ความน่าจะเป็นของผลลัพธ์ในบริบทของบทสนทนา เช่น ในตัวอย่างข้างต้น หากผู้พูดออกเสียง ล ลิง ในคำว่า "กลบ" ในวลี "กลบเกลื่อนความผิด" ไม่ชัดเจน เมื่อวิเคราะห์ข้อมูลคลื่นเสียงแล้วอาจได้เป็น "กบเกลื่อนความผิด" ด้วยความน่าจะเป็น 70 % และ "กลบเกลื่อนความผิด" ด้วยความน่าจะเป็น 30% แต่เมื่อนำสัญลักษณ์การออกเสียงมาวิเคราะห์เทียบกับโมเดลภาษาแล้ว อาจจะได้ผลลัพธ์เป็น "กลบเกลื่อนความผิด" 95% และ "กบเกลื่อนความผิด" 5% ซึ่งเมื่อเปรียบเทียบค่าความน่าจะเป็นแล้ว ทำให้ได้ผลลัพธ์เป็น "กลบเกลื่อนความผิด"

2.4.2 โมเดลภาษา และกฎไวยากรณ์

ความน่าจะเป็นของข้อมูลทางภาษานั้นจะมีค่าเปลี่ยนแปลงไปตามการใช้งานที่แตกต่างกันออกไป (ตามสถานการณ์ในการใช้งานการรู้จำเสียง) ยกตัวอย่างเช่น คำที่ออกเสียงว่า "โจด" อาจหมายถึง "โจทย์" ที่แปลว่า คำถามในวิชาคณิตศาสตร์ หรือ "โจทก์" ที่แปลว่า ผู้ฟ้องร้องในศาล หรือ "โจษ" ที่มาจากคำว่า "โจษจัน" ก็ได้ ซึ่งหากคำนี้มาจากบทสนทนาที่เกี่ยวกับศาลหรือกฎหมาย ก็มีความน่าจะเป็นสูงกว่าคำที่พูดนี้เป็นคำว่า "โจทก์" มากกว่าคำอื่นๆ แต่หากเป็นบทสนทนาเกี่ยวกับการ

เรียนการศึกษา โดยเฉพาะวิชาคณิตศาสตร์ ก็มีความน่าจะเป็นสูงว่าคำนี้คือคำว่า “โจทท์” ซึ่งการจะหาค่าความน่าจะเป็นตามลักษณะการใช้งานนี้ จะต้องอาศัย “โมเดลภาษา” และ “กฎไวยากรณ์”

กฎไวยากรณ์ ใช้สำหรับระบุวิธีการพูดที่มีขอบเขตจำกัดในการแปลงข้อมูลเสียงพูดในระบบการรู้จำเสียง ยกตัวอย่างเช่น การใช้งานระบบการรู้จำเสียงสำหรับการจองเที่ยวบินทางโทรศัพท์ด้วยเสียงพูด ซึ่งต้องระบุชื่อของท่าอากาศยานต้นทางและปลายทาง ก็อาจจะวิเคราะห์การพูดในรูปแบบ “จาก (ชื่อท่าอากาศยาน) ถึง (ชื่อท่าอากาศยาน)” โดยระบบรู้จำเสียงจะต้องมีรายชื่อและวิธีการออกเสียงชื่อท่าอากาศยานต่างๆ ที่จะใช้ในการพูดอยู่ จึงจะสามารถเข้าใจวิธีการพูดในลักษณะ “จากสุวรรณภูมิถึงเชียงใหม่” ได้ ซึ่งการแปลงข้อมูลในระบบการรู้จำเสียงโดยอาศัยกฎไวยากรณ์นั้น การพูดจะต้องตรงตามลักษณะไวยากรณ์ที่กำหนดไว้เท่านั้น ระบบจึงจะสามารถเข้าใจและแปลงข้อมูลได้อย่างถูกต้อง เช่น ในตัวอย่างนี้ หากพูดว่า “ไปสุวรรณภูมิออกจากเชียงใหม่” ระบบก็จะไม่สามารถแปลงข้อมูลได้

โมเดลภาษาสามารถรองรับการวิเคราะห์การพูดที่ไม่มีรูปแบบตายตัวได้ ซึ่งในการแปลงข้อมูลจะไม่ได้เทียบกับรูปแบบของการพูดเหมือนการใช้กฎไวยากรณ์ แต่จะแปลงข้อมูลเสียงพูดทั้งหมดออกมาเป็นตัวอักษร โดยอาศัยความน่าจะเป็นของคำที่ประกอบกันขึ้นมาเป็นประโยคเข้ามาช่วย การรู้จำเสียงที่ใช้โมเดลภาษาแปลงคำพูดทั้งหมดออกมาเป็นตัวอักษรเรียกว่า Dictation (การเขียนตามคำบอก) การประยุกต์ใช้งานระบบรู้จำเสียงพูดในลักษณะของ Dictation นั้น นอกจากจะสามารถนำไปใช้ในการแปลงเสียงพูดเป็นตัวอักษรลงในโปรแกรมประมวลผลคำ (Word Processor) อย่าง Microsoft Word แล้ว ยังสามารถนำไปใช้ในด้านอื่นๆ ได้ด้วย เช่น การนำไปใช้ในระบบจองเที่ยวบินทางโทรศัพท์ดังตัวอย่างข้างต้น ซึ่งจะทำให้สามารถรองรับรูปแบบการพูดที่หลากหลาย และยืดหยุ่นเป็นธรรมชาติมากกว่า

ค่าความน่าจะเป็นจากการวิเคราะห์ด้วยโมเดลภาษาจะช่วยบอกแนวโน้มของคำที่น่าจะปรากฏอยู่ในผลลัพธ์ โดยอาศัยความรู้ความเข้าใจทางภาษาที่ป้อนให้กับโมเดลภาษา เช่นตัวอย่างข้างต้นที่สามารถระบุได้ว่า การออกเสียง “โจด” ในเสียงพูดหมายถึงคำใดนั้น ต้องใช้การหาค่าความน่าจะเป็นของคำที่เป็นไปได้ คือ “โจทท์” “โจทก์” และ “โจษ” ซึ่งต้องอาศัยบริบทของการสนทนาช่วยในการวิเคราะห์ บริบทของการสนทนาจะวิเคราะห์จากคำอื่นๆ ที่แวดล้อมคำที่กำลังพิจารณา เช่น หากพูดว่า “คำให้การพยานโจด” คำที่อยู่แวดล้อมคือ “คำให้การ” และ “พยาน” ซึ่งจะช่วยระบุถึงบริบท ทำให้รู้ได้ว่า การออกเสียง “โจด” น่าจะมีความน่าจะเป็นว่าเป็นคำว่า “โจทก์” มากกว่าคำอื่นๆ การที่สามารถวิเคราะห์เช่นนี้ได้ ก็เนื่องจากโมเดลภาษามีการเก็บสถิติว่าคำใดมักปรากฏอยู่ร่วมกับคำใดบ้าง

2.4.3 การใช้งานจริงในแบบไม่เจาะจงผู้พูด

จะเห็นว่าการรู้จำเสียงนั้นอาศัยค่าความน่าจะเป็นจาก 2 ส่วนมาประกอบกัน คือ จากการวิเคราะห์คลื่นเสียง และจากตัวถอดรหัสการรู้จำ ซึ่งการทำงานของทั้งสองส่วนนี้จะเป็นการทำงานโดยอาศัยข้อมูลสถิติ นั่นคือ การวิเคราะห์คลื่นเสียงจะอาศัยข้อมูลสถิติของลักษณะคลื่นเสียงของการออกเสียงในแบบต่างๆ ส่วนตัวถอดรหัสการรู้จำจะอาศัยข้อมูลสถิติของคำที่ใช้ในภาษาที่วิเคราะห์ ด้วยลักษณะการทำงานเช่นนี้ ทำให้สามารถจำแนกการทำงานของระบบรู้จำเสียงได้เป็น 2 แบบ คือ การทำงานแบบเจาะจงผู้พูด และการทำงานแบบไม่เจาะจงผู้พูด

ในการทำงานแบบเจาะจงผู้พูด ผู้พูดจะต้องลงทะเบียนเสียงของตนกับระบบก่อน ระบบจึงจะสามารถเข้าใจเสียงพูดได้ การลงทะเบียนเสียงพูดไว้ล่วงหน้าทำให้มีสถิติคลื่นเสียงของผู้พูดเพื่อนำมาวิเคราะห์เปรียบเทียบ ซึ่งเป็นการวิเคราะห์ที่เทียบกับเสียงบุคคลเดียวกัน ดังนั้นจึงมีความแม่นยำสูงกว่าการวิเคราะห์ที่เทียบกับสถิติกลางที่ไม่ใช่เสียงของผู้พูดเอง แต่หากพิจารณาถึงความสะดวกและเหมาะสมในการใช้งานแล้ว การรู้จำเสียงแบบเจาะจงผู้พูดนั้นอาจจะไม่ใช่แนวทางที่ดีเสมอไป ในลักษณะงานที่ต้องวิเคราะห์เสียงที่ผู้พูดมีการสลับเปลี่ยนหมุนเวียนกันเป็นจำนวนมาก การต้องลงทะเบียนเสียงสำหรับผู้พูดทุกๆ คน เป็นเรื่องที่ยุ่งยากและไม่สะดวกในการใช้งาน การรู้จำเสียงแบบไม่เจาะจงผู้พูดจึงเป็นทางเลือกที่เหมาะสมกว่า เทคโนโลยีของอะมิวอยซ์จัดอยู่ในกลุ่มการรู้จำเสียงแบบไม่เจาะจงผู้พูด ซึ่งทำงานโดยอาศัยการวิเคราะห์คลื่นเสียงด้วยเทคโนโลยีขั้นสูงที่สามารถวิเคราะห์ลักษณะคลื่นเสียงได้แม่นยำ แม้จะไม่มีสถิติคลื่นเสียงของผู้พูดปัจจุบัน และใช้โมเดลภาษาที่มีข้อมูลสถิติที่รวบรวมมาจากข้อมูลการใช้ภาษาจำนวนมาก ที่เรียกว่า corpus ซึ่งข้อมูลสถิติที่ใช้ใน corpus นั้นจะเลือกมาตามขอบเขตและลักษณะการใช้งานของการรู้จำเสียง เช่น สำหรับระบบจองเที่ยวบินทางโทรศัพท์ ข้อมูลที่รวบรวมมาใช้ใน corpus ก็จะเป็นข้อมูลที่เกี่ยวข้องกับการเดินทาง การจองเที่ยวบิน เส้นทางการเดินทาง การระบุวันเวลา ฯลฯ ซึ่งจะทำให้โมเดลภาษาสามารถวิเคราะห์เสียงพูดที่มีเนื้อหาในด้านนั้นได้อย่างแม่นยำ ลักษณะการทำงานเช่นนี้ทำให้สามารถปรับแต่งโมเดลภาษาเพื่อให้ระบบรู้จำเสียงสามารถตอบสนองการใช้งานเฉพาะด้านได้อย่างมีประสิทธิภาพได้

2.4.4 เทคโนโลยีที่เกี่ยวข้อง

จากที่ได้อธิบายมาข้างต้น จะเห็นว่ามีการใช้เทคโนโลยีขั้นสูงในการรู้จำเสียง เทคโนโลยีเหล่านี้ช่วยให้สามารถนำระบบรู้จำเสียงพูดไปใช้งานจริงได้อย่างสะดวกและเกิดประโยชน์ต่อการทำงานและการดำรงชีวิตเป็นอย่างมาก เทคโนโลยีเหล่านี้ได้แก่

2.4.4.1 DSR (Distributed Speech Recognition)

ในขณะที่ โบายคอมพิวเตอร์ (Mobile Computing) มีความแพร่หลายมากขึ้น ก็มี การใช้โทรศัพท์เคลื่อนที่ในงานด้านต่างๆ ที่หลากหลายตามมา แอปพลิเคชันบนอุปกรณ์ อิเล็กทรอนิกส์แบบพกพาเหล่านี้ อาจอาศัยระบบการรู้จำเสียงช่วยในการรับข้อมูลเข้าจาก ผู้ใช้งาน ซึ่งทำให้ผู้ใช้ได้รับความสะดวกในการป้อนข้อมูลในกรณีที่ผู้ใช้พิมพ์ไม่ถนัดเนื่องจาก ตัวโทรศัพท์ที่ไม่มีแป้นพิมพ์ หรือในกรณีที่ผู้ใช้ไม่สามารถพิมพ์ได้เนื่องจากมือไม่ว่าง แต่ เนื่องจากการประมวลผลการรู้จำเสียงต้องใช้ความสามารถของ CPU และ memory สูงมาก เพื่อให้ได้ผลลัพธ์รวดเร็วและแม่นยำ การใช้งานระบบการรู้จำเสียงบนอุปกรณ์อิเล็กทรอนิกส์ แบบพกพาซึ่งมักจะมีสมรรถนะในการทำงานจำกัดจึงเป็นเรื่องที่ยากมาก ด้วยเหตุผลนี้เอง จึงมีการออกแบบให้การรู้จำเสียงทำงานโดยวิเคราะห์คลื่นเสียงบนอุปกรณ์อิเล็กทรอนิกส์ แบบพกพา แล้วส่งข้อมูลลักษณะเฉพาะของคลื่นเสียงไปยังเซิร์ฟเวอร์เพื่อประมวลผลการรู้จำ เสียง แล้วจึงส่งผลกลับไปยังอุปกรณ์อิเล็กทรอนิกส์แบบพกพา วิธีการนี้เรียกว่า “DSR” (Distributed Speech Recognition) หากเปรียบเทียบกับ การส่งข้อมูลเสียงทั้งหมดไปบน คู่สายโทรศัพท์แล้วประมวลผลการรู้จำเสียงบนเซิร์ฟเวอร์ วิธี DSR จะมีข้อดีที่เหนือกว่า คือ (1) ข้อมูลลักษณะเฉพาะของคลื่นเสียง เป็นการเลือกลักษณะเฉพาะของคลื่นเสียงต้นฉบับ แล้วเปลี่ยนเป็นข้อมูลทางดิจิทัล ซึ่งไม่ได้เป็นการบีบอัดข้อมูลโดยการลดทอนคุณภาพเสียง ดังนั้น จึงทำให้ได้ผลการรู้จำเสียงที่มีความละเอียดสูง (2) ขนาดของข้อมูลลักษณะเฉพาะจะมีขนาดเล็กกว่าข้อมูลเสียงทั้งหมด ทำให้ใช้เวลาในการส่งข้อมูลน้อยลง (3) มีค่าใช้จ่ายในการรับส่งข้อมูลลดลง

2.4.4.2 การระบุตัวตนด้วยเสียง

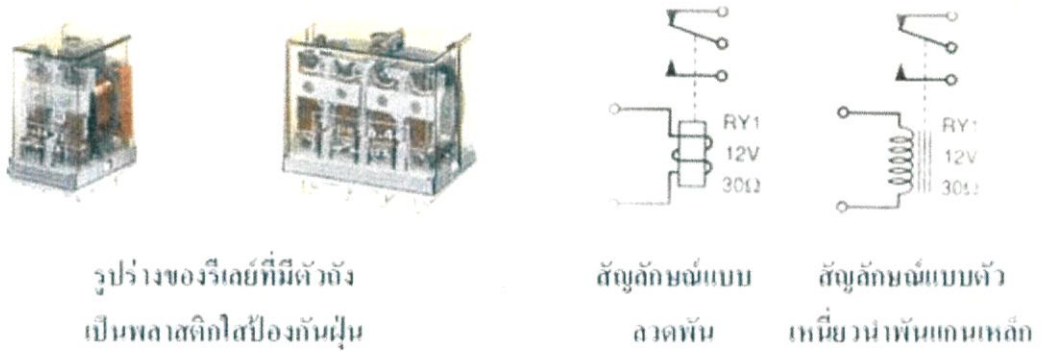
ระบบรู้จำเสียงพูดที่สามารถทำงานได้แบบไม่เจาะจงผู้พูด จะต้องคัดเลือก ลักษณะเฉพาะของคลื่นเสียงโดยเลือกเฉพาะลักษณะคลื่นที่มาจากคำพูด และจัดลักษณะ คลื่นที่เกิดจากตัวผู้พูดออกไป นั่นคือ เลือกลักษณะเฉพาะที่มีเสถียรภาพไม่เปลี่ยนแปลงตาม ตัวผู้พูดไปวิเคราะห์เปรียบเทียบ เพื่อหาว่าลักษณะเฉพาะนี้คล้ายกับการออกเสียงคำใด ในทางกลับกัน การคัดเลือกลักษณะเฉพาะที่เกิดจากตัวผู้พูดก็สามารถทำได้เช่นกัน ซึ่งการนำ ลักษณะเฉพาะเช่นนี้ไปวิเคราะห์เปรียบเทียบ จะทำให้สามารถทราบได้ว่าลักษณะเฉพาะของ เสียงนี้ใกล้เคียงกับลักษณะเฉพาะของเสียงใคร การทำงานในลักษณะนี้ สามารถนำไป ประยุกต์ใช้กับการระบุหรือยืนยันตัวตนด้วยเสียงพูดได้ ซึ่งเป็นอีกทางเลือกหนึ่งที่สามารถ นำไปใช้ในระบบรักษาความปลอดภัยได้

2.5 รีเลย์ (Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการ ดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสถานะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิด หรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุม วงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

2.5.1 หลักการเบื้องต้น

รีเลย์เป็นอุปกรณ์ที่นิยมนำมาทำเป็นสวิตช์ทางด้านอิเล็กทรอนิกส์ โดยจะต้องป้อน กระแสไฟฟ้าให้ไหลผ่านขดลวดจำนวนหนึ่ง เพื่อนำไปควบคุมวงจรกำลังงานสูง ๆ ที่ต่ออยู่กับ หน้าสัมผัสหรือคอนแทคของรีเลย์ รูปที่ 2.5 แสดงรูปร่างและสัญลักษณ์ของรีเลย์



รูปร่างของรีเลย์ที่มีตัวถัง เป็นพลาสติกใสป้องกันฝุ่น

สัญลักษณ์แบบ ลวดพัน

สัญลักษณ์แบบตัว เหนี่ยวนำพันแทนเหล็ก

รูปที่ 2.2 แสดงรูปร่างและสัญลักษณ์ของรีเลย์

บทที่ 3

การออกแบบและการคำนวณ

3.1 การเลือกใช้อุปกรณ์

3.1.1 อุปกรณ์ Voice Recognition

ในการเลือกใช้อุปกรณ์ที่ทำหน้าที่ Voice Recognition ทางผู้จัดทำได้เลือกใช้ EasyVR Arduino Shield จากบริษัท VeeaR เนื่องจากทางผู้จัดทำมีประสบการณ์ในการเขียนโปรแกรมในตัว Arduino

ซึ่งความสามารถของอุปกรณ์ EasyVR Arduino Shield สามารถทำงานร่วมกับ Microcontroller ตัวใดก็ได้ โดยอาศัยการเชื่อมต่อแบบ UART โดยทำงานอยู่ที่ 3.3V – 5V โดยมีโหมดการทำงาน 4 แบบ ได้แก่ UP, PC, HW, SW ซึ่งจะกล่าวต่อไปในหัวข้อต่อไป



รูปที่ 3.1 แสดงรูปร่างภายนอกของ EasyVR Arduino Shield

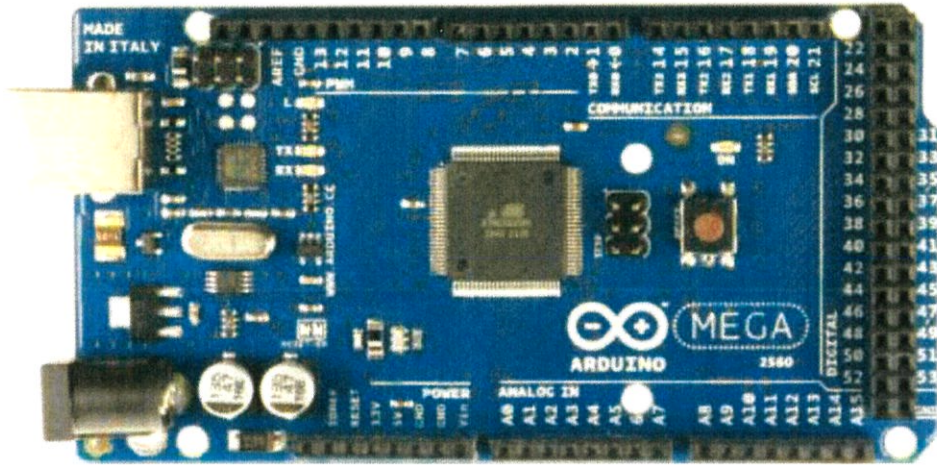
3.1.2 คุณสมบัติของ EasyVR Arduino Shield

- ชุดคำสั่งพื้นฐานแบบ Speaker Independent (SI) โดยรองรับสำหรับภาษาดังต่อไปนี้
 - อังกฤษ (US)

- ญี่ปุ่น
 - อิตาลี
 - เยอรมัน
 - สเปน
 - ฝรั่งเศส
- รองรับการสั่งงานแบบ Speaker Dependent (SD) สำหรับทุกภาษา
 - รองรับคำสั่งรหัส ยืนยันตัวบุคคล
 - 3 พอร์ต GPIO
 - Port PWM Output สำหรับลำโพงขนาด 8 โอห์ม
 - Jack สำหรับหูฟัง
 - LED สำหรับทดลองเขียนโปรแกรม
 - รองรับ Sound Playback
 - รองรับโปรแกรม Voice Commands
 - สามารถทำงานร่วมกับ Microcontroller ใดๆ ผ่านการสื่อสารแบบ UART (ทำงานที่ 3.3V – 5V)
 - ใช้งานง่ายและมีโปรโตคอลที่มีประสิทธิภาพในการเข้าถึงโปรแกรมโมดูล

3.1.3 Arduino Mega 2560

ในส่วนของ Microcontroller ทางผู้จัดทำได้เลือกใช้งาน Microcontroller สำเร็จรูปรุ่น Mega 2560 ของบริษัท Arduino เนื่องจากเป็น Microcontroller สำเร็จรูปที่มีราคาถูกสะดวกต่อการเขียนโปรแกรมโดยใช้ภาษา C เป็นพื้นฐาน และเป็นที่นิยมในการเลือกงานของบุคคลทั่วไป

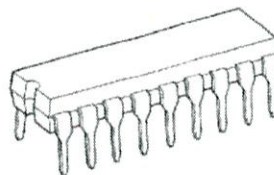


รูปที่ 3.2 แสดงรูปร่างภายนอกของ Arduino Mega 2560

ซึ่ง Arduino Mega 2560 เป็น Microcontroller โดยใช้ ATmega2560 ในการประมวลผล ซึ่งตัว Arduino Mega 2560 จะมี 54 พอร์ตสำหรับ Digital Input/Output 16 พอร์ตสำหรับ Analog Input 4 พอร์ตสำหรับ UART และใช้ Crystal Oscillator ที่ 16 MHz มีพอร์ตเชื่อมต่อ USB แจ็คสำหรับจ่ายไฟ ICSP Header และปุ่ม Reset

3.1.4 ULN3803APG

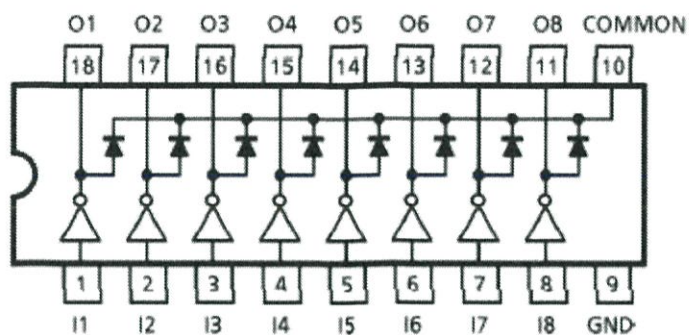
ULN2803APG เป็น IC Drive วงจร ควบคุมทั้งบอร์ดควบคุมการเปิดปิดหลอดไฟและอุปกรณ์ไฟฟ้า และบอร์ดควบคุมการโทรออกและรับสายของโทรศัพท์ โดยมีกระแสเอาท์พุทมากที่สุด 500 mA ในตัว IC จะมีไดโอดเชื่อมระหว่างขาอินพุทและขาเอาท์พุท โดยไดโอดจะมีแรงดันย้อนกลับ 50 V และมีกระแสตรง 500 mA ตัว IC จะจ่ายกระแสจากขาเอาท์พุทไปสู่รีเลย์เพื่อควบคุมการทำงานของหลอดไฟ อุปกรณ์ไฟฟ้า และโทรศัพท์



DIP18-P-300-2.54F

รูปที่ 3.3 แสดงรูปร่างภายนอกของ ULN2803AP

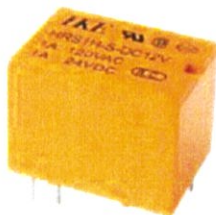
PIN CONNECTION (TOP VIEW)



รูปที่ 3.4 แสดงขาต่างๆและการเชื่อมต่อกันภายใน

3.1.5 รีเลย์ (Relay)

รีเลย์ที่เลือกใช้งานสำหรับระบบสั่งงานด้วยเสียงคือ HRS4-S-DC5V และ HRS1H-S-DC5V ซึ่งเป็นรีเลย์ขนาด 5 V ทั้งสองตัวใช้ในการควบคุมหลอดไฟ อุปกรณ์ไฟฟ้า และโทรศัพท์ โดย HRS4-S-DC5V จะมีกำลังของขดลวด 450 mW ส่วน HRS1H-S-DC5V จะมีกำลังของขดลวด 360 mW



รูปที่ 3.5 แสดงรูปร่างภายนอกของ HRS4-S-DC5V

3.1.6 หลอดไฟ

ในส่วนหลอดไฟนั้น ผู้จัดทำได้เลือกใช้หลอดไฟขนาด 220/230V 3-5 W ทั้ง 5 สี ได้แก่ สีฟ้า สีแดง สีเหลือง สีเขียว และสีวอร์ม



รูปที่ 3.6 หลอดไฟขนาด 220/230V 3-5 W

3.1.7 เต้าเสียบ

ใช้เต้าเสียบแบบ 3 พอร์ต ต่อกับเครื่องใช้ไฟฟ้า ดังนี้ กาน้ำ แทนชาร์ตโทรศัพท์ และพัดลม



รูปที่ 3.7 เต้าเสียบ

3.1.8 โทรศัพท์บ้าน



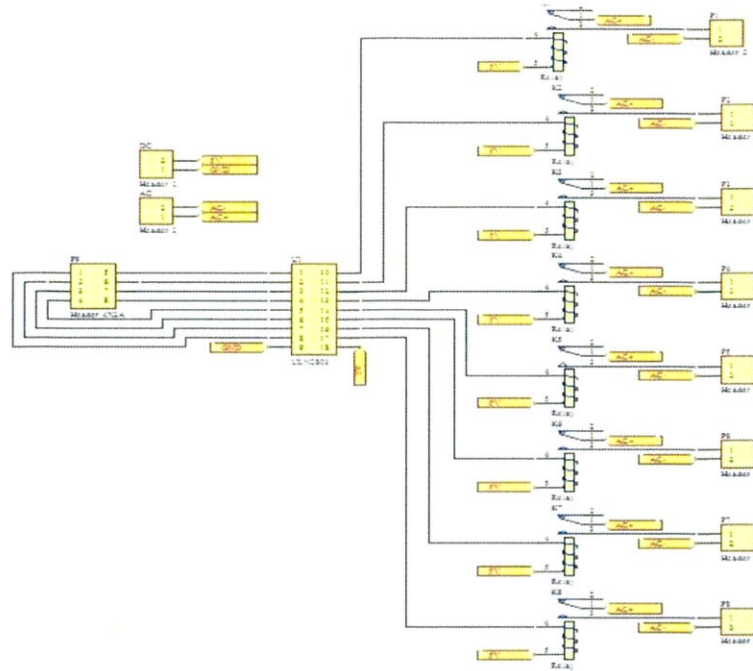
รูปที่ 3.8 โทรศัพท์บ้าน

3.2 การออกแบบวงจรควบคุมหลอดไฟ และเต้าเสียบ

โดยหน้าที่ของบอร์ดนี้จะทำการรับคำสั่งจากบอร์ด Arduino จากนั้นนำสัญญาณเข้าสู่ IC ULN2803APG โดยตัว ULN2803APG จะทำการกลับสัญญาณที่ได้ เช่น ค่า Input เป็น 5V ที่ Output จะมีค่าเป็น 0V ซึ่งจะเห็นได้ว่า เราได้จ่ายไฟ 5V ให้กับ Relay ไว้ก่อนแล้ว ซึ่งถ้าได้รับแรงดัน 0V จาก ULN2803APG จะทำให้ Coil ใน Relay มีความต่างศักย์ เป็น 5V ก็จะทำให้ขา Common ของ Relay จากเดิมซึ่งสัมผัสกับขา NC เปลี่ยนมาสัมผัสกับขา NO ทำให้หลอดไฟได้รับไฟครบวงจร ดังแสดงในตารางที่ 3.1

วงจรควบคุมหลอดไฟ และเต้าเสียบ Input พอร์ตที่	รับคำสั่งจากบอร์ด Arduino พอร์ตที่	ตำแหน่งทำงานที่ Output
1	27	ไฟหน้าบ้าน
2	29	ไฟหลังบ้าน
3	31	ไฟรั้ว
4	33	ไฟบรรได
5	32	ไฟห้องนอน
6	30	พัดลม
7	28	ที่ชาร์ต
8	26	กาน้ำร้อน

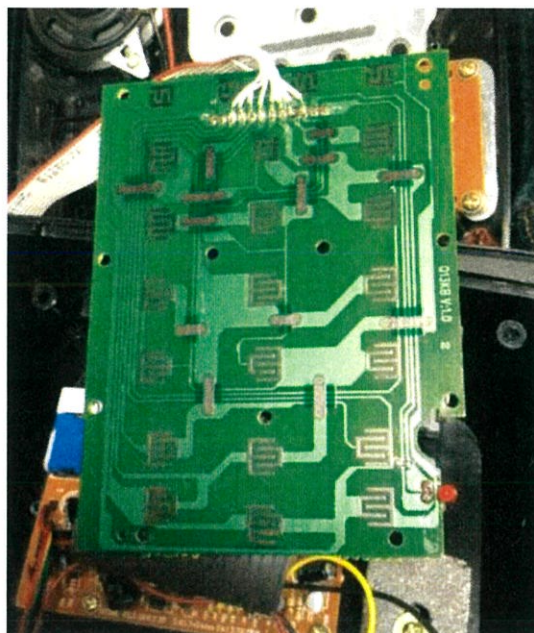
ตารางที่ 3.1 แสดงความสัมพันธ์ระหว่างวงจรควบคุมหลอดไฟและตำแหน่งทำงาน



รูปที่ 3.9 แสดงวงจรขับหลอดไฟ และเต้าเสียบ

3.3 การออกแบบวงจรควบคุมโทรศัพท์

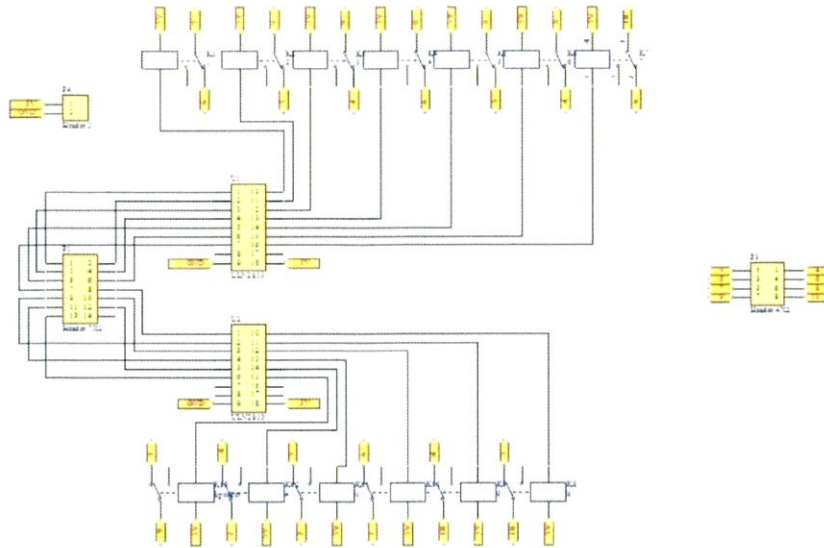
โดยหน้าที่ของบอร์ดนี้จะทำการรับคำสั่งจากบอร์ด Arduino จากนั้นนำสัญญาณเข้าสู่ IC ULN2803APG โดยตัว ULN2803APG จะทำการกลับสัญญาณที่ได้ เช่น ค่า Input เป็น 5V ที่ Output จะมีค่าเป็น 0V ซึ่งจะเห็นได้ว่า เราได้จ่ายไฟ 5V ให้กับ Relay ไว้ก่อนแล้ว ซึ่งถ้าได้รับแรงดัน 0V จาก ULN2803APG จะทำให้ Coil ใน Relay มีความต่างศักย์ เป็น 5V ก็จะทำให้ขา Common ของ Relay จากเดิมซึ่งสัมผัสกับขา NC เปลี่ยนมาสัมผัสกับขา NO ทำให้วงจรปุ่มกดบนโทรศัพท์ได้รับไฟครบวงจร และแสดงผลบนจอโทรศัพท์



รูปที่ 3.10 วงจรภายในโทรศัพท์

วงจรควบคุมโทรศัพท์ พอร์ตที่	รับคำสั่งจากบอร์ด Arduino พอร์ตที่	วงจรในโทรศัพท์ พอร์ตA และ B เชื่อม กัน	ผลที่ได้คือการกดปุ่ม
1	38	9,6	1
2	39	9,5	2
3	40	9,4	3
4	41	8,6	4
5	42	8,5	5
6	43	8,4	6
7	44	10,6	7
8	45	10,5	8
9	46	10,4	9
10	48	7,5	0
11	49	7,6	*
12	47	7,4	#
13	50	9,3	Speaker

ตารางที่ 3.2 แสดงความสัมพันธ์ระหว่างวงจรควบคุมโทรศัพท์และวงจรในโทรศัพท์



รูปที่ 3.11 แสดงวงจรควบคุมโทรศัพท์

3.4 การออกแบบกลุ่มคำสั่ง

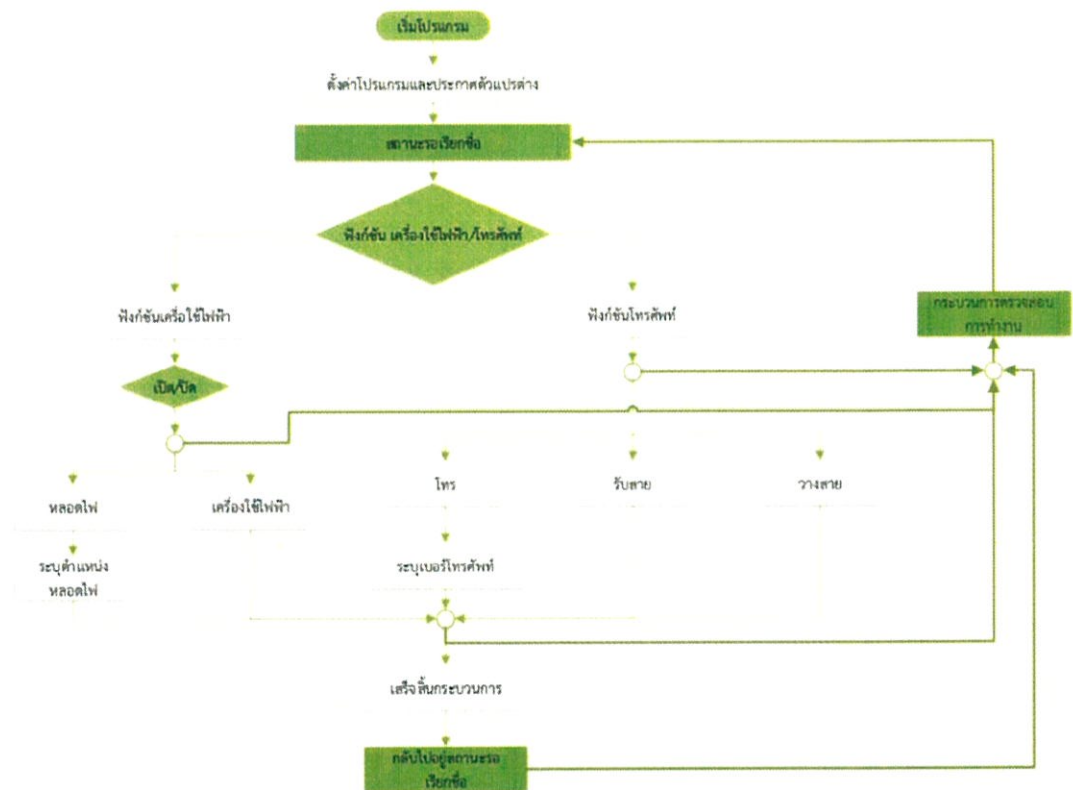
เราได้ออกแบบกลุ่มคำสั่งทั้งหมด จำนวน 5 กลุ่มคำสั่ง ได้แก่ ชื่อ คำสั่ง อุปกรณ์ ตำแหน่งของอุปกรณ์ โทรศัพท์ ซึ่งจากกลุ่มทั้งหมดเราได้ทำการออกแบบคำพูดเพื่อตอบสนองทุกการใช้งานเป็นคำพูดทั้งหมด 29 คำ ซึ่งจะแสดงได้ดังตารางด้านล่างนี้

กลุ่มคำสั่ง	สมาชิกในกลุ่มคำสั่ง
ชื่อ	สายัณฑ์
คำสั่ง	เปิด
	ปิด
	โทร
	รับสาย
	วางสาย
อุปกรณ์	หลอดไฟ
	พัดลม
	ที่ชาร์จ
	กาน้ำร้อน
	ทั้งหมด

ตำแหน่งอุปกรณ์	หน้าบ้าน
	หลังบ้าน
	รู
	บันได
	ห้องนอน
	ทั้งหมด
โทรศัพท์	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	*
	#
	ยกเลิก

ตารางที่ 3.3 แสดงรายชื่อกลุ่มคำสั่งและสมาชิกในกลุ่มคำสั่ง

ต่อไปจะแสดงการทำงานของระบบสั่งงานด้วยเสียงโดยแผนผังโฟลว์ชาร์ท



รูปที่ 3.12 แสดงโฟลว์ชาร์ทการทำงานของระบบสั่งงานด้วยเสียง

ซึ่งในตัวโปรแกรมนั้นได้ทำการออกแบบส่วนตรวจสอบการทำงานของโปรแกรม ซึ่งจะอธิบายในส่วนต่อไป

3.4.1 โปรแกรมในการควบคุมอุปกรณ์ไฟฟ้า

โดยจะยกตัวอย่างการสั่งให้เปิดหลอดไฟบริเวณหน้าบ้าน



รูปที่ 3.13 แสดงลำดับคำสั่งการเปิดหลอดไฟหน้าบ้าน

```
case G0_SAYAN:
```

```
    group = GROUP_1;
```

```
break;
```

จากโค้ดด้านบนจะเป็นกลุ่มของการเรียกชื่อซึ่งถ้าได้รับคำสั่งเรียกชื่อแล้วโปรแกรมจะข้ามไปกลุ่มคำสั่งที่ 1

```
case GROUP_1:
```

```
    switch (idx)
```

```
        {case G1_OPEN:
```

```
            x=1;
```

```
                group = GROUP_2;
```

```
        break;
```

จากโค้ดด้านบนหลังจากที่เรียกชื่อแล้วก็จะเป็คำสั่งเปิด ซึ่งหลังจากได้รับคำสั่งแล้ว จะให้ค่า x=1 แล้วจึงจะข้ามไปยังกลุ่มคำสั่งที่ 3

```
case GROUP_2:
```

```
    switch (idx)
```

```
        {case G2_BULB:
```

```
            group = GROUP_3;
```

```
        break;
```

จากโค้ดด้านบนหลังจากให้คำสั่งเปิดแล้วโปรแกรมก็จะรอให้ ผู้ใช้ระบุอุปกรณ์ไฟฟ้า ซึ่งจากโค้ดเป็นโค้ดของหลอดไฟหลังจากได้คำสั่งแล้วจึงจะข้ามไปยังกลุ่มคำสั่งที่

```
case GROUP_3:
```

```
    switch (idx)
```

```
        { case G3_FRONT_HOUSE:
```

```

if(x==1){
    digitalWrite(front,HIGH);}

else if(x==0){
    digitalWrite(front,LOW);}

group = GROUP_0;

break;

```

จากขั้นตอนด้านบนที่เป็นการเลือกอุปกรณ์ไฟฟ้า ซึ่งตามตัวอย่างเป็นหลอดไฟ และจากโค้ดด้านบน จะเป็นการรับคำสั่งระบุตำแหน่งของหลอดไฟที่บริเวณหน้าบ้าน แล้วยังมีการตรวจสอบอีกครั้งว่าเป็นการเปิดหรือปิด โดยจากข้างบนเป็นการเปิดจึงกำหนดให้ $x=1$ หลังจากตรวจสอบก็จะไปกำหนดให้ตัวแปร front ให้มีค่าเป็น 5V เพื่อให้หลอดไฟเปิด จากนั้นโปรแกรมจะกลับไปอยู่สถานะรอเรียกชื่ออีกครั้ง

3.4.2 โปรแกรมในการโทรศัพท์

โดยจะยกตัวอย่างการให้โทรออกตามเบอร์ 081-234-5678



รูปที่ 3.14 แสดงลำดับคำสั่งการโทรออกของโทรศัพท์

```

case G0_SAYAN:
    group = GROUP_1;

break;

```

จากโค้ดด้านบนจะเป็นกลุ่มของการเรียกชื่อซึ่งถ้าได้รับคำสั่งเรียกชื่อแล้วโปรแกรมจะเข้าไปกลุ่มคำสั่งที่ 2

```

case G1_CALL:

    digitalWrite(but_spk,HIGH);

    delay(50);

    digitalWrite(but_spk,LOW);

    spk=1;

    group = GROUP_4;

break;

```

หลังจากรับคำสั่งเรียกชื่อแล้ว ก็จะต้องระบุคำสั่งจากตัวอย่างจะเป็นการโทร โดยโปรแกรม หลังจากรับคำสั่งแล้ว จะให้ตัวแปร but_spk เป็น 5V แล้วคงค่าไว้เป็นเวลา 50 mSec หลังจากนั้นจึง ให้ค่าตัวแปร but_spk เป็น 0V หลังจากนั้นจะกำหนดตัวแปร spk = 1 เพื่อใช้ในการตรวจสอบการทำงาน of โปรแกรม จากนั้นจึงข้ามไปยังกลุ่มคำสั่งที่ 5

```

case G4_0:

    if(p<=9){

        digitalWrite(but_0,HIGH);

        delay(50);

        digitalWrite(but_0,LOW);

        p++;

        group = GROUP_4;}

    else if(p==10){

        digitalWrite(but_0,HIGH);

        delay(50);

        digitalWrite(but_0,LOW);

        p=1;

```

```
group = GROUP_0;}
```

```
break;
```

หลังจากได้รับคำสั่งโทรแล้วจะเข้าสู่กลุ่มระบุเบอร์โทรศัพท์โดยตัวอย่างโค้ดด้านบนเป็นของตัวเลข 0 โดยมีขั้นตอนดังนี้ ถ้าหาก p มีค่าน้อยกว่าหรือเท่ากับ 9 โปรแกรมจะกำหนดค่าตัวแปร but_0 เป็น 5V และคงค่าเป็นเวลา 50 mSec จากนั้นจึงให้ตัวแปรกลับมาเป็นค่า 0V จากนั้นจึงให้ p มีค่าเพิ่มขึ้นอีกหนึ่ง และรอคำสั่งระบุเบอร์โทรศัพท์ต่อไป ซึ่งโปรแกรมจะวนรับเบอร์โทรศัพท์จนครบ 10 ตัว หลังจากนั้นจะกำหนดให้ p=1 อีกครั้งจากนั้นจึงจะกลับไปอยู่สถานะรอเรียกชื่ออีกครั้ง

3.4.3 คำสั่งโปรแกรมในส่วนตรวจสอบการทำงานของโปรแกรม

3.4.3.1 โปรแกรมตรวจสอบการทำงานของโปรแกรมในด้านเวลา

```
if (easyvr.isTimeout()){
    t_o++;
    Serial.println("Timed out, try again...");
    if(t_o==3){
        if(spkr==1&&p<9){
            digitalWrite(but_spk,HIGH);
            delay(50);
            digitalWrite(but_spk,LOW);
            spk=0;}
        t_o=0;
    }
    group = GROUP_0;}}
```

จากโค้ดทางด้านบนจะเป็นการตรวจสอบโปรแกรมทางด้านเวลา ซึ่งได้ตั้งเวลาในการรอรับคำสั่งเป็นเวลา 5 วินาที โดยถ้าภายใน 5 วินาทีไม่มีการส่งคำสั่งระบบจะทำการส่งข้อความมาเตือนดังนี้ “Time out, try again...” และโปรแกรมจะวนลักษณะนี้เป็นจำนวน 3 ครั้ง ถ้ายังไม่ได้รับคำสั่ง โปรแกรมจะกลับไปสู่สถานะรอเรียกชื่อ

บทที่ 4

ผลการทดลอง

4.1 จุดประสงค์การทดลอง

- เพื่อตรวจสอบการทำงานของระบบ
- เพื่อวัดคุณสมบัติเฉพาะของระบบ
- เพื่อวัดประสิทธิภาพของระบบ

4.2 วิธีการทดลอง

โดยเราจะทำการแบ่งการทดลองออกเป็น 5 แบบ เพื่อทดสอบประสิทธิภาพของระบบ โดยจะมีการทดลองแบบต่างๆดังนี้

4.2.1 การทดลองส่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน

จะทำการทดลองโดยกำหนดให้ผู้ฝึกสอนสั่งงานให้กับระบบ โดยจะสั่งให้ระบบทำการเปิดหลอดไฟที่ตำแหน่งหน้าบ้าน มีขั้นตอนการสั่งงานดังนี้



รูปที่ 4.1 แสดงลำดับคำสั่งการเปิดหลอดไฟหน้าบ้าน

โดยมีระยะห่างระหว่างผู้ฝึกสอนกับระบบอยู่ที่ 1 เมตร และสั่งงานด้วยระดับเสียง 90 เดซิเบล

4.2.2 การทดลองสั่งด้วยเสียงเพื่อโทรออก

จะทำการทดลองโดยกำหนดให้ผู้ฝึกสอนสั่งงานให้กับระบบ โดยจะสั่งให้ระบบทำการโทรออกไปยังหมายเลข 081-234-5678 มีขั้นตอนการสั่งงานดังนี้

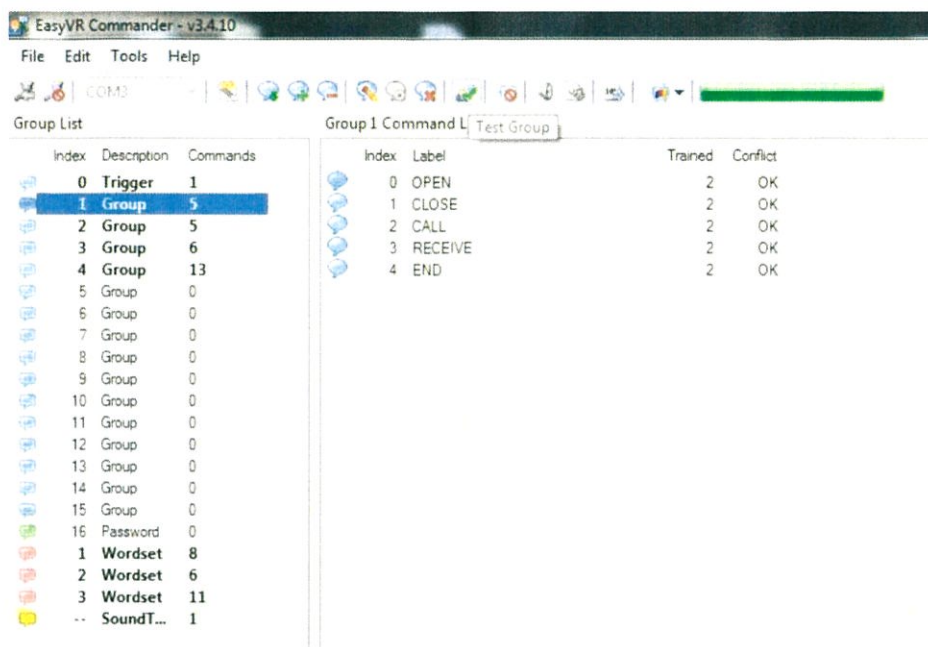


รูปที่ 4.2 แสดงลำดับคำสั่งการโทรออกของโทรศัพท์

โดยมีระยะห่างระหว่างผู้ฝึกสอนกับระบบอยู่ที่ 1 เมตร และสั่งงานด้วยสั่งงานด้วยระดับเสียง 90 เดซิเบล

4.2.3 การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ

จะทำการทดลองโดยกำหนดให้ผู้ฝึกสอนสั่งงานให้กับระบบ ผ่านโปรแกรม EasyVR Commander โดยจะใช้ฟังก์ชัน “Test Group” ในการทดลอง โดยจะทดลองสั่งทั้ง 30 คำที่ได้ทำการสอนให้กับระบบ คำละ 10 ครั้งแล้วบันทึกผล แล้วจึงนำผลการทดลองแต่ละคำมาหาเปอร์เซ็นต์ความสำเร็จในการสั่งงาน โดยมีระยะห่างระหว่างผู้ฝึกสอนกับระบบอยู่ที่ 1 เมตร และสั่งงานด้วยระดับเสียง 90 เดซิเบล



รูปที่ 4.3 แสดงหน้าต่างโปรแกรม EasyVR Commander

4.2.4 การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล

จะทำการทดลองโดยกำหนดให้ ผู้ฝึกสอนและบุคคลอื่นอีก 4 คน โดยแบ่งเป็นเพศชาย 2 คน เพศหญิง 2 คน สั่งงานให้กับระบบในคำสั่งเดียวกันเป็นจำนวน 3 คำสั่ง

- คำสั่งปิดหลอดไฟตำแหน่งบันได
- คำสั่งเปิดพัดลม
- คำสั่งรับสาย

แล้วบันทึกผลการทดลองและหาเปอร์เซ็นต์ความสำเร็จในการสั่งงานโดยมีระยะห่างระหว่างผู้สั่งงานกับระบบอยู่ที่ 1 เมตร และสั่งงานด้วยระดับเสียง 90 เดซิเบล

4.2.5 การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ

จะทำการทดลองโดยกำหนดให้ ผู้ฝึกสอนสั่งงานให้กับระบบ โดยใช้คำสั่งเปิดหลอดไฟ ตำแหน่งห้องนอน เป็นจำนวน 10 ครั้ง โดยให้ระยะห่างระหว่างระบบกับผู้ฝึกสอนเป็นระยะ 1 เมตร

หลังจากนั้นเปลี่ยนระยะห่างระหว่างระบบกับผู้ฝึกสอนเป็น 2 เมตร และ 3 เมตรตามลำดับพร้อมทั้งจดบันทึกผลการทดลอง โดยสั่งงานด้วยระดับเสียง 90 เดซิเบล

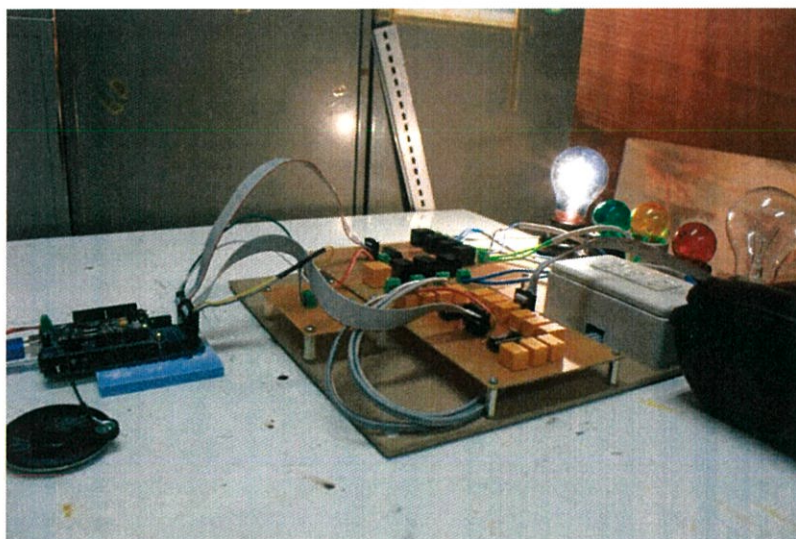
4.2.6 การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ

จะทำการทดลองโดยกำหนดให้ ผู้ฝึกสอนสั่งงานให้กับระบบ ในสภาพแวดล้อมปกติ โดยใช้คำสั่งปิดกาน้ำร้อน เป็นจำนวน 10 ครั้ง หลังจากนั้นจึงทดลองในสภาพแวดล้อมที่เปิดเครื่องปรับอากาศในขณะที่สั่งงานให้กับระบบ เป็นจำนวน 10 ครั้ง หลังจากนั้นจึงทดลองในสภาพแวดล้อมที่เปิดเครื่องปรับอากาศและเปิดเพลงในขณะที่สั่งงานให้กับระบบ เป็นจำนวน 10 ครั้ง แล้วจึงทำการบันทึกผลการทดลอง โดยมีระยะห่างระหว่างผู้สั่งงานกับระบบอยู่ที่ 1 เมตร และสั่งงานด้วยระดับเสียง 90 เดซิเบล

4.3 ผลการทดลอง

4.3.1 การทดลองสั่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน

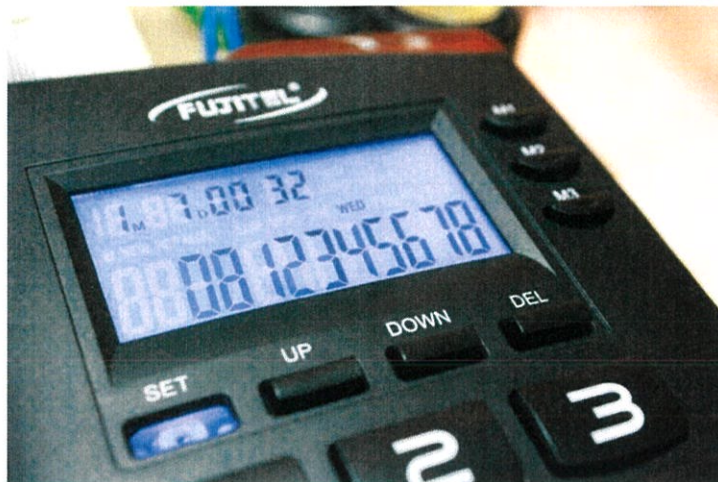
โดยการทดลองสั่งงานด้วยเสียงให้เปิดหลอดไฟตำแหน่งหน้าบ้าน สามารถทำงานได้ดีดังรูป



รูปที่ 4.4 แสดงผลของการสั่งเปิดหลอดไฟหน้าบ้าน

4.3.2 การทดลองสั่งด้วยเสียงเพื่อโทรออก

โดยการทดลองสั่งงานด้วยเสียงให้ระบบทำการโทรออกไปยังหมายเลข 081-234-5678 สามารถทำงานได้ดีดังรูป



รูปที่ 4.5 แสดงผลของตัวเลขบนจอโทรศัพท์

4.3.3 การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ

หมายเหตุ 0 = คำสั่งไม่สำเร็จ 1 = คำสั่งสำเร็จ

คำสั่ง	คำสั่งครั้งที่										เปอร์เซ็นต์สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
สายยนต์	1	1	1	1	0	1	1	1	1	1	90
เปิด	0	1	1	1	1	1	0	1	1	1	80
ปิด	1	1	0	0	1	1	1	1	0	1	70
โทร	1	1	1	1	1	1	1	1	1	1	100
รับสาย	0	1	1	1	1	1	1	1	1	1	90
วางสาย	0	0	1	1	1	1	1	1	1	1	80
หลุดไฟ	0	1	1	1	1	1	1	0	1	1	80
พัดลม	1	0	1	1	0	0	1	1	1	1	70
ที่ชาร์จ	1	1	1	1	1	0	1	1	1	1	90

กาน้ำร้อน	1	1	1	1	1	1	1	1	1	1	100
ทั้งหมด(อุปกรณ์)	0	0	1	1	1	1	1	1	1	1	80
หน้าบ้าน	1	1	1	0	1	1	1	1	1	1	90
หลังบ้าน	1	0	0	0	0	1	0	1	0	1	40
รู	0	1	1	1	1	0	0	1	0	0	50
บันได	1	0	1	1	1	1	0	1	1	1	80
ห้องนอน	0	1	1	1	1	1	1	1	1	1	90
ทั้งหมด(ตำแหน่งอุปกรณ์)	0	1	0	0	0	0	1	1	1	1	50
ศูนย์	1	1	1	1	0	1	1	1	1	1	90
หนึ่ง	1	1	0	1	1	1	1	1	1	1	90
สอง	0	1	1	1	0	1	1	1	1	0	70
สาม	1	1	1	1	1	0	1	1	1	0	80
สี่	1	1	0	1	1	1	0	1	0	1	70
ห้า	0	0	1	1	1	1	1	1	1	1	80
หก	1	1	1	1	1	1	1	0	0	1	80
เจ็ด	1	1	1	1	0	1	1	1	1	1	90
แปด	1	1	1	1	0	0	1	1	1	0	70
เก้า	1	1	1	1	1	1	1	1	1	1	100
ดอกจัน	1	1	1	1	1	1	1	0	1	1	90
สี่เหลี่ยม	1	1	1	0	0	1	1	1	0	1	70
ยกเล็ก	0	1	1	1	1	0	1	1	1	1	80

ตารางที่ 4.1 แสดงการทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ

4.3.4 การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล

4.3.4.1 การทดลองสั่งให้ปิดหลอดไฟที่ตำแหน่งบันได

ผู้ส่งงาน	คำสั่งครั้งที่										เปอร์เซ็นต์ สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
ผู้ฝึกสอน	0	1	1	1	1	1	1	0	1	1	80
ผู้ชายคนที่ 1	0	0	0	0	0	0	0	0	0	0	0
ผู้ชายคนที่ 2	0	0	1	0	1	0	0	0	0	0	20
ผู้หญิงคนที่ 1	0	0	0	0	0	0	1	0	0	0	10
ผู้หญิงคนที่ 2	0	0	0	0	0	0	0	0	0	0	0

ตารางที่ 4.2 แสดงผลการทดลองสั่งให้ปิดหลอดไฟที่ตำแหน่งบันได

4.3.4.2 การทดลองสั่งงานเปิดพัดลม

ผู้ส่งงาน	คำสั่งครั้งที่										เปอร์เซ็นต์ สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
ผู้ฝึกสอน	1	1	1	0	1	1	1	1	1	0	80
ผู้ชายคนที่ 1	0	1	0	0	0	0	0	0	0	0	10
ผู้ชายคนที่ 2	0	0	0	0	0	0	0	0	0	0	0
ผู้หญิงคนที่ 1	0	0	0	0	1	0	0	0	0	0	10
ผู้หญิงคนที่ 2	0	0	0	0	0	0	0	0	1	0	10

ตารางที่ 4.3 แสดงผลการทดลองสั่งงานเปิดพัดลม

4.3.4.3 การทดลองรับสายของโทรศัพท์

ผู้ส่งงาน	คำสั่งครั้งที่										เปอร์เซ็นต์ สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
ผู้ฝึกสอน	0	1	1	1	1	1	1	1	1	1	90
ผู้ชายคนที่ 1	0	1	0	1	1	1	1	1	1	0	70
ผู้ชายคนที่ 2	0	1	0	0	1	0	1	1	1	0	50
ผู้หญิงคนที่ 1	0	1	0	1	0	0	1	0	0	0	30
ผู้หญิงคนที่ 2	0	0	0	1	0	0	0	1	0	0	20

ตารางที่ 4.4 แสดงผลการทดลองรับสายของโทรศัพท์

4.3.5 การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ

ระยะทาง	คำสั่งครั้งที่										เปอร์เซ็นต์ สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
1 เมตร	1	0	1	1	1	0	0	1	1	1	70
2 เมตร	1	1	1	1	0	1	1	0	1	0	70
3 เมตร	1	1	1	1	0	0	1	1	1	1	80
4 เมตร	0	0	1	1	1	1	1	1	0	1	70

ตารางที่ 4.5 แสดงผลการทดลองประสิทธิภาพในระยะต่างๆ

4.3.6 การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ

ระดับเสียง รบกวน	คำสั่งครั้งที่										เปอร์เซ็นต์ สำเร็จ
	1	2	3	4	5	6	7	8	9	10	
ปกติ (58dB)	1	1	1	1	1	1	1	1	1	1	100
เปิดแอร์ (69dB)	1	1	1	1	1	1	1	1	1	1	100
เปิดเพลง (80dB)	0	0	0	0	0	0	0	1	0	0	10

ตารางที่ 4.6 แสดงผลการทดลองประสิทธิภาพในสภาพแวดล้อมต่างๆ

บทที่ 5

สรุปผลการทดลอง

จากโครงการนี้ซึ่งเป็นโครงการที่ศึกษาในเรื่อง ระบบสั่งงานด้วยเสียง ที่ผู้จัดทำได้ออกแบบให้สามารถควบคุมอุปกรณ์ไฟฟ้า 3 ประเภท ได้แก่ หลอดไฟ เครื่องใช้ไฟฟ้า และโทรศัพท์ ซึ่งมีได้ออกแบบคำสั่งดังนี้ “เปิด” “ปิด” สำหรับหลอดไฟ และเครื่องใช้ไฟฟ้า ส่วนคำสั่ง “โทร” “รับสาย” “วางสาย” เป็นคำสั่งสำหรับโทรศัพท์ ซึ่งทางผู้จัดทำทำการออกแบบการทดลองทั้งหมด 6 การทดลองเพื่อทดสอบประสิทธิภาพของ ระบบสั่งด้วยเสียง ได้แก่

1. การทดลองสั่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน
2. การทดลองสั่งด้วยเสียงเพื่อโทรออก
3. การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ
4. การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล
5. การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ
6. การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ

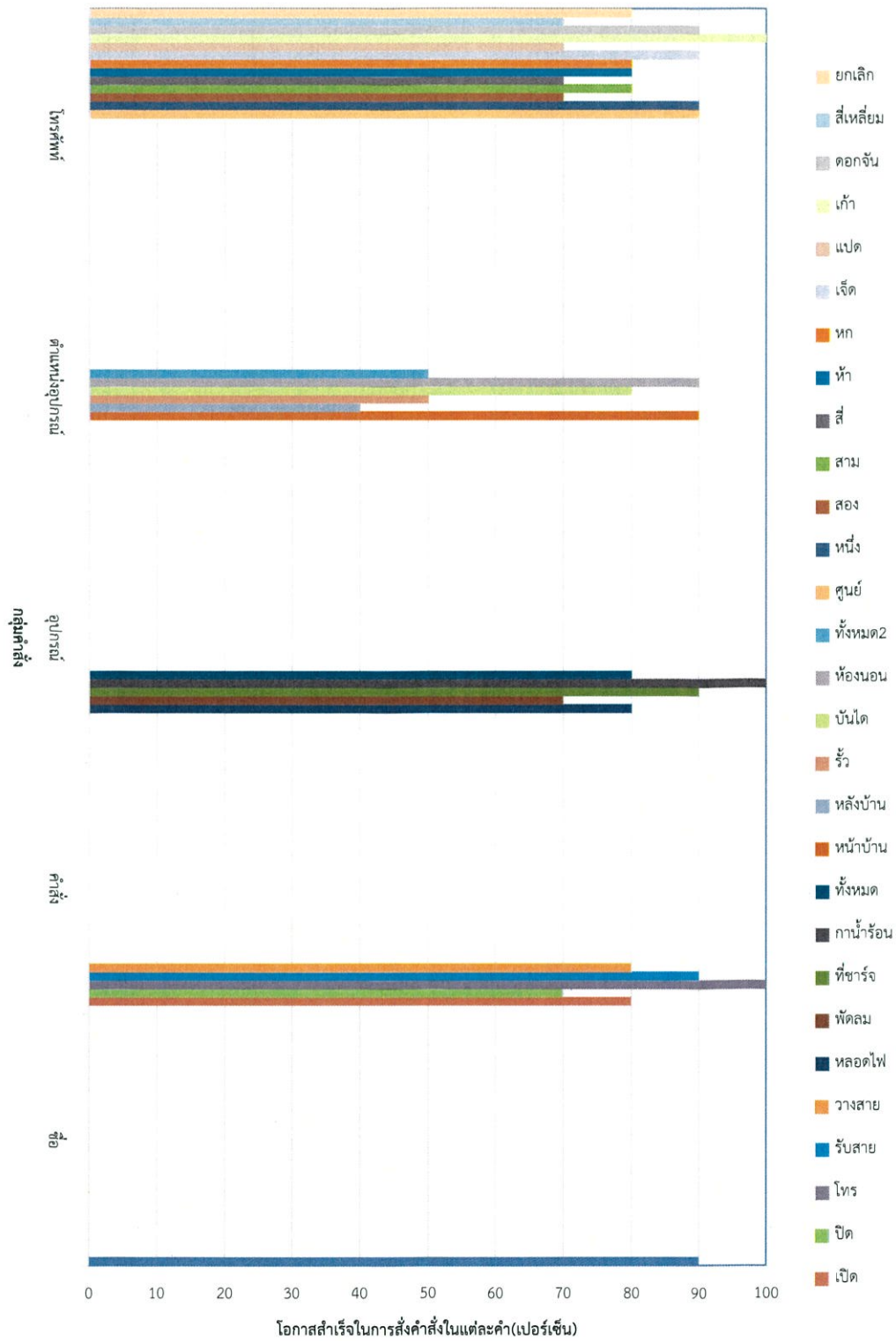
โดยจะสรุปผลการทดลองของแต่ละการทดลองดังนี้

การทดลองสั่งด้วยเสียงเพื่อเปิดหลอดไฟตำแหน่งหน้าบ้าน ในการทดลองนี้ระบบสามารถตอบสนองคำสั่งได้เป็นอย่างดี ดังรูปที่ 4.4 ในบทที่ 4

การทดลองสั่งด้วยเสียงเพื่อโทรออก ในการทดลองนี้ระบบสามารถตอบสนองคำสั่งได้น่าอย่างพอใจสามารถสั่งงานให้ระบบ ส่งสัญญาณไฟฟ้า แทนการกดปุ่มโทรศัพท์ จนสามารถให้โทรศัพท์โทรออกหรือรับสายได้ แต่เนื่องจากการส่งสัญญาณไฟฟ้าเพื่อควบคุมโทรศัพท์นั้นอาจจะใช้ระยะเวลาาน เนื่องจากกระบวนการตรวจสอบคำสั่งเสียงในกลุ่มโทรศัพท์นั้นมีจำนวน 13 คำสั่ง และอาจจะเกิดความผิดพลาดในรับคำสั่งจึงต้องการสั่งคำสั่งซ้ำ จึงทำให้การส่งสัญญาณไฟฟ้าเพื่อควบคุมโทรศัพท์เกิดความล่าช้าไม่ทันช่วงรอรับเบอร์โทรศัพท์ของโทรศัพท์

การทดลองประสิทธิภาพในการรับคำสั่งในแต่ละคำ ในการทดลองนี้จะเป็นการทดลองการรับคำสั่งแต่ละคำสั่งที่ได้สอนไป ว่ามีประสิทธิภาพในการรับคำสั่งเป็นอย่างไร แสดงดังแผนภูมิด้านล่าง

แผนภูมิแสดงประสิทธิภาพในการรับคำสั่งในแต่ละคำ



รูปที่ 5.1 แผนภูมิแสดงประสิทธิภาพในการรับคำสั่งในแต่ละคำ

จากการทดลองจะพบว่าค่าที่มีโอกาสสำเร็จสูงถึง 100 เปอร์เซ็นต์ ได้แก่คำว่า “โทร” “กาน้ำร้อน” “เก้า” และค่าที่โอกาสสำเร็จต่ำสุด 40 เปอร์เซ็นต์ ได้แก่คำว่า “หลังบ้าน” และกลุ่มคำสั่งที่มีโอกาสสำเร็จสูงสุดได้แก่ “กลุ่มชื่อ” 90 เปอร์เซ็นต์ และกลุ่มคำสั่งที่มีโอกาสสำเร็จน้อยสุดได้แก่ “กลุ่มตำแหน่งอุปกรณ์” 66 เปอร์เซ็นต์

ทั้งนี้ผลการทดลองที่ได้ถือว่าเป็นแนวโน้มยังไม่สิ้นสุด ยังต้องการปริมาณการทดลองจำนวนมากเพื่อที่จะได้ผลการทดลองที่สามารถสรุปได้

การทดลองประสิทธิภาพในการรับคำสั่งจากหลายบุคคล ในการทดลองนี้เป็นการนำหลายบุคคลมาทดลองสั่งคำสั่งเดียวกัน เพื่อดูประสิทธิภาพการรับคำสั่ง

ผลการทดลองที่ได้ในคำสั่งปิดหลอดไฟตำแหน่งบันได ผู้ฝึกสอนมีเปอร์เซ็นต์ความสำเร็จอยู่ที่ 80 เปอร์เซ็นต์ ส่วนบุคคลอื่นๆ มีเปอร์เซ็นต์ความสำเร็จไม่เกิน 20 เปอร์เซ็นต์ทั้งเพศชายและเพศหญิง เนื่องจากในคำสั่งนี้อาจมีขั้นตอนในการสั่งงานซับซ้อน หรือค่าที่ใช้สั่งงานมีความเป็นเอกลักษณ์สูง

ผลการทดลองที่ได้ในคำสั่งเปิดพัดลม ผู้ฝึกสอนมีเปอร์เซ็นต์ความสำเร็จอยู่ที่ 80 เปอร์เซ็นต์ ส่วนบุคคลอื่นๆ มีเปอร์เซ็นต์ความสำเร็จไม่เกิน 10 เปอร์เซ็นต์ทั้งเพศชายและหญิง เช่นกัน คำสั่งนี้อาจมีขั้นตอนในการสั่งงานซับซ้อน หรือค่าที่ใช้สั่งงานมีความเป็นเอกลักษณ์สูง

ผลการทดลองที่ได้ในคำสั่งรับสายโทรศัพท์ ผู้ฝึกสอนมีเปอร์เซ็นต์ความสำเร็จอยู่ที่ 90 เปอร์เซ็นต์ ในเพศชายมีเปอร์เซ็นต์ความสำเร็จเฉลี่ยเป็น 60 เปอร์เซ็นต์ ในเพศหญิงมีเปอร์เซ็นต์ความสำเร็จเฉลี่ยเป็น 25 เปอร์เซ็นต์ อันเนื่องมาจากในคำสั่งนี้เป็นคำสั่งที่ไม่ซับซ้อน และต้องพูดคำสั่งเป็นจำนวน 2 คำสั่ง ก็สามารถสั่งงานได้

การทดลองประสิทธิภาพในการรับคำสั่งที่ระยะต่างๆ ในการทดลองนี้เป็นการทดลองการรับคำสั่งในระยะห่างต่างๆ ซึ่งระยะในการทดลองนั้นอยู่ที่ 1-4 เมตร ผลที่ได้จากการทดลอง มีเปอร์เซ็นต์ความสำเร็จเฉลี่ยอยู่ที่ 72.5 เปอร์เซ็นต์ จากค่าที่ได้นั้นอยู่ในเกณฑ์ดี

การทดลองประสิทธิภาพในการรับคำสั่งในสภาพแวดล้อมต่างๆ ในการทดลองนี้เป็นการทดลองการรับคำสั่งในสภาพแวดล้อมที่แตกต่างกัน โดยผู้ทดสอบจะสั่งงานในระดับเสียงประมาณ 90 dB จากผลการทดลองจะได้ว่าในสภาพแวดล้อมปกติและสภาพแวดล้อมขณะเปิดเครื่องปรับอากาศที่มีระดับเสียงรบกวนประมาณ 58-69 dB ซึ่งผลการทดลองที่ได้มีเปอร์เซ็นต์ความสำเร็จเฉลี่ยที่ 100 เปอร์เซ็นต์ และการทดลองในสภาพแวดล้อมขณะเปิดเพลงซึ่งมีระดับเสียงรบกวนประมาณ 80 dB ผลการทดลองที่ได้เปอร์เซ็นต์ความสำเร็จอยู่ที่ 10 เปอร์เซ็นต์ อันเนื่องมาจากเสียงรบกวนนั้นมีค่า

ความดังที่ใกล้เคียงกับคำสั่งที่ใช้สั่งงานและยังมีจังหวะการรบกวนคล้ายเคียงกับคำสั่งที่ใช้สั่งงานจึงทำให้เปอร์เซ็นต์ความสำเร็จมีค่าต่ำ

จากการทดลองทั้งหมดทำให้เราได้ทราบว่าระบบสั่งงานด้วยเสียงที่ได้ออกแบบ สามารถสั่งงานด้วยเสียงเพื่อ เปิด ปิด อุปกรณ์ไฟฟ้า และสามารถสั่งงานด้วยเสียงเพื่อ โทรออก รับสาย โทรศัพท์ได้ ซึ่งก่อนการสั่งงานอุปกรณ์นั้นผู้ใช้ควรจะมีฝึกฝนคำสั่งด้วยเสียงของผู้ใช้เอง เพื่อประสิทธิภาพในการรับคำสั่งที่ดี ซึ่งระยะห่างที่เหมาะสมในการสั่งงานควรมีระยะอยู่ประมาณ 4 เมตร และควรมีสภาพแวดล้อมในขณะที่สั่งงานมีระดับเสียงรบกวนประมาณ 69 dB ซึ่งระบบสามารถทำงานได้อย่างมีประสิทธิภาพ

บรรณานุกรม

<http://www.veear.eu>

<http://www.arduino.in.th>

<http://kpp.ac.th/elearning/elearning3/book-09.html>

http://tvis.nectec.or.th/speech/index.php?option=com_content&view=article&id=61&temid=83&lang=th

http://www.acentech.net/cms/index.php?option=com_content&task=view&id=567&Itemid=205

<http://www.amivoicethai.com/ami-engine>

ภาคผนวก

ภาคผนวก ก.

Source Code ของโปรแกรม

Source Code ของโปรแกรม Arduino

```
#if defined(ARDUINO) && ARDUINO >= 100

#include "Arduino.h"

#include "SoftwareSerial.h"

SoftwareSerial port(12,13);

#else // Arduino 0022 - use modified NewSoftSerial

#include "WProgram.h"

#include "NewSoftSerial.h"

NewSoftSerial port(12,13);

#endif

#include "EasyVR.h"

EasyVR easyvr(port);

//Groups and Commands

enum Groups

{

GROUP_0 = 0,

GROUP_1 = 1,
```

```
GROUP_2 = 2,  
  
GROUP_3 = 3,  
  
GROUP_4 = 4,  
  
};
```

```
enum Group0  
  
{  
  
    G0_SAYAN = 0,  
  
};
```

```
enum Group1  
  
{  
  
    G1_OPEN = 0,  
  
    G1_CLOSE = 1,  
  
    G1_CALL = 2,  
  
    G1_RECEIVE = 3,  
  
    G1_END = 4,  
  
};
```

```
enum Group2
```

```
{  
  
    G2_BULB = 0,  
  
    G2_FAN = 1,  
  
    G2_CHANGER = 2,  
  
    G2_KETTLE = 3,  
  
    G2_ALL_ELECTRIC = 4,  
  
};
```

```
enum Group3
```

```
{  
  
    G3_FRONT_HOUSE = 0,  
  
    G3_BACK_HOUSE = 1,  
  
    G3_LADDER = 2,  
  
    G3_FENCE = 3,  
  
    G3_BEDROOM = 4,  
  
    G3_ALL_LAMP = 5,  
  
};
```

```
enum Group4
```

```
{
```

```
G4_0 = 0,  
  
G4_1 = 1,  
  
G4_2 = 2,  
  
G4_3 = 3,  
  
G4_4 = 4,  
  
G4_5 = 5,  
  
G4_6 = 6,  
  
G4_7 = 7,  
  
G4_8 = 8,  
  
G4_9 = 9,  
  
G4_STAR = 10,  
  
G4_SQR = 11,  
  
G4_CANCEL = 12,  
  
};
```

```
EasyVRBridge bridge;
```

```
int8_t group, idx;
```

```
int x;  
  
int p=1;  
  
int a=0;  
  
int e_r=0;  
  
int t_o=0;  
  
int spk=0;  
  
  
//group bulb position  
  
int front = 27;  
  
int back = 29;  
  
int ladder = 31;  
  
int fence = 33;  
  
int bedroom = 32;  
  
  
  
//group electric  
  
int fan = 30;  
  
int charger = 28;  
  
int kettle = 26;
```

```
//group telephone
```

```
int but_0 = 48;
```

```
int but_1 = 38;
```

```
int but_2 = 39;
```

```
int but_3 = 40;
```

```
int but_4 = 41;
```

```
int but_5 = 42;
```

```
int but_6 = 43;
```

```
int but_7 = 44;
```

```
int but_8 = 45;
```

```
int but_9 = 46;
```

```
int but_sqr = 49;
```

```
int but_star = 47;
```

```
int but_spk = 50;
```

```
void setup()
```

```
{
```

```
    pinMode(front, OUTPUT);
```

```
    pinMode(back, OUTPUT);
```

```
    pinMode(ladder, OUTPUT);
```

```
pinMode(fence, OUTPUT);

pinMode.bedroom, OUTPUT);

pinMode(fan, OUTPUT);

pinMode(charger, OUTPUT);

pinMode(kettle, OUTPUT);

pinMode(but_0, OUTPUT);

pinMode(but_1, OUTPUT);

pinMode(but_2, OUTPUT);

pinMode(but_3, OUTPUT);

pinMode(but_4, OUTPUT);

pinMode(but_5, OUTPUT);

pinMode(but_6, OUTPUT);

pinMode(but_7, OUTPUT);

pinMode(but_8, OUTPUT);

pinMode(but_9, OUTPUT);

pinMode(but_sqr, OUTPUT);

pinMode(but_star, OUTPUT);

pinMode(but_spk, OUTPUT);

// bridge mode?

if (bridge.check())
```

```
{  
  
  cli();  
  
  bridge.loop(0, 1, 12, 13);  
  
}  
  
// run normally  
  
Serial.begin(9600);  
  
port.begin(9600);  
  
if (!easyvr.detect())  
  
{  
  
  Serial.println("EasyVR not detected!");  
  
  for (;;) ;  
  
}  
  
easyvr.setPinOutput(EasyVR::IO1, LOW);  
  
Serial.println("EasyVR detected!");  
  
easyvr.setTimeout(5);  
  
easyvr.setLanguage(0);  
  
group = EasyVR::TRIGGER; //<-- start group (customize)
```

```
}
```

```
void action();
```

```
void loop()
```

```
{
```

```
    easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
```

```
    Serial.print("Say a command in Group ");
```

```
    Serial.println(group);
```

```
    easyvr.recognizeCommand(group);
```

```
    do
```

```
    {
```

```
        // can do some processing while waiting for a spoken command
```

```
    }
```

```
    while (!easyvr.hasFinished());
```

```
    easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
```

```
idx = easyvr.getWord();

if (idx >= 0)

{

    // built-in trigger (ROBOT)

    // group = GROUP_X; <-- jump to another group X

    return;

}

idx = easyvr.getCommand();

if (idx >= 0)

{

    // print debug message

    uint8_t train = 0;

    char name[32];

    Serial.print("Command: ");

    Serial.print(idx);

    if (easyvr.dumpCommand(group, idx, name, train))

    {

        Serial.print(" = ");

        Serial.println(name);

    }

}
```

```
else

    Serial.println();

    easyvr.playSound(0, EasyVR::VOL_DOUBLE);

    // perform some action

    action();

}

else // errors or timeout

{

    if (easyvr.isTimeout()){

        t_o++;

        Serial.println("Timed out, try again...");

    }

    if(t_o==3){

        if(spkr==1&&p<9){

            digitalWrite(but_spk,HIGH);

            delay(50);

            digitalWrite(but_spk,LOW);

            spk=0;}

        t_o=0;

        group = GROUP_0;}}
```

```
int16_t err = easyvr.getError();

if (err >= 0){

    Serial.print("Error ");

    Serial.println(err, HEX);

}

}

}
```

```
void action()

{

    switch (group)

    {

    case GROUP_0:

        switch (idx)

        {

        case GO_SAYAN:

            group = GROUP_1;

        }

        break;
```

```
}  
  
break;  
  
case GROUP_1:  
  
    switch (idx)  
  
    {  
  
    case G1_OPEN:  
  
        x=1;  
  
        group = GROUP_2;  
  
        break;  
  
    case G1_CLOSE:  
  
        x=0;  
  
        group = GROUP_2;  
  
        break;  
  
    case G1_CALL:  
  
        digitalWrite(but_spk,HIGH);  
  
        delay(50);  
  
        digitalWrite(but_spk,LOW);  
  
        spk=1;
```

```
group = GROUP_4;

break;

case G1_RECEIVE:

digitalWrite(but_spk,HIGH);

delay(50);

digitalWrite(but_spk,LOW);

group = GROUP_0;

break;

case G1_END:

digitalWrite(but_spk,HIGH);

delay(50);

digitalWrite(but_spk,LOW);

group = GROUP_0;

break;

}

break;

case GROUP_2:
```

```
switch (idx)

{

case G2_BULB:

group = GROUP_3;

break;

case G2_FAN:

if(x==1){

digitalWrite(fan,HIGH);}

else if(x==0){

digitalWrite(fan,LOW);}

group = GROUP_0;

break;

case G2_CHANGER:

if(x==1){

digitalWrite(charger,HIGH);}

else if(x==0){

digitalWrite(charger,LOW);}

group = GROUP_0;
```

```
break;

case G2_KETTLE:

if(x==1){

    digitalWrite(kettle,HIGH);}

else if(x==0){

    digitalWrite(kettle,LOW);}

group = GROUP_0;

break;

case G2_ALL_ELECTRIC:

if(x==1){

    digitalWrite(fan,HIGH);

    digitalWrite(charger,HIGH);

    digitalWrite(kettle,HIGH);}

else if(x==0){

    digitalWrite(fan,LOW);

    digitalWrite(charger,LOW);

    digitalWrite(kettle,LOW);}

group = GROUP_0;
```

```
        break;
    }

    break;

case GROUP_3:

    switch (idx)
    {

    case G3_FRONT_HOUSE:

        if(x==1){

            digitalWrite(front,HIGH);}

        else if(x==0){

            digitalWrite(front,LOW);}

        group = GROUP_0;

        break;

    case G3_BACK_HOUSE:

        if(x==1){

            digitalWrite(back,HIGH);}

        else if(x==0){

            digitalWrite(back,LOW);}
```

```
group = GROUP_0;
```

```
    break;
```

```
case G3_LADDER:
```

```
if(x==1){
```

```
    digitalWrite(ladder,HIGH);} 
```

```
else if(x==0){
```

```
    digitalWrite(ladder,LOW);} 
```

```
group = GROUP_0;
```

```
    break;
```

```
case G3_FENCE:
```

```
if(x==1){
```

```
    digitalWrite(fence,HIGH);} 
```

```
else if(x==0){
```

```
    digitalWrite(fence,LOW);} 
```

```
group = GROUP_0;
```

```
    break;
```

```
case G3_BEDROOM:
```

```
if(x==1){

    digitalWrite(bedroom,HIGH);}

else if(x==0){

    digitalWrite(bedroom,LOW);}

group = GROUP_0;

    break;

case G3_ALL_LAMP:

if(x==1){

    digitalWrite(front,HIGH);

    digitalWrite(back,HIGH);

    digitalWrite(ladder,HIGH);

    digitalWrite(fence,HIGH);

    digitalWrite(bedroom,HIGH);}

else if(x==0){

    digitalWrite(front,LOW);

    digitalWrite(back,LOW);

    digitalWrite(ladder,LOW);

    digitalWrite(fence,LOW);

    digitalWrite(bedroom,LOW);}
```

```
group = GROUP_0;

    break;

}

break;

case GROUP_4:

    switch (idx)

    {

    case G4_0:

        if(p<=9){

            digitalWrite(but_0,HIGH);

            delay(50);

            digitalWrite(but_0,LOW);

            p++;

            group = GROUP_4;}

        else if(p==10){

            digitalWrite(but_0,HIGH);

            delay(50);

            digitalWrite(but_0,LOW);

            p=1;
```

```
    group = GROUP_0;}

    break;

case G4_1:

    if(p<=9){

        digitalWrite(but_1,HIGH);

        delay(50);

        digitalWrite(but_1,LOW);

        p++;

        group = GROUP_4;}

    else if(p==10){

        digitalWrite(but_1,HIGH);

        delay(50);

        digitalWrite(but_1,LOW);

        p=1;

        group = GROUP_0;}

    break;

case G4_2:

    if(p<=9){
```

```
digitalWrite(but_2,HIGH);

delay(50);

digitalWrite(but_2,LOW);

p++;

group = GROUP_4;}

else if(p==10){

digitalWrite(but_2,HIGH);

delay(50);

digitalWrite(but_2,LOW);

p=1;

group = GROUP_0;}

break;

case G4_3:

if(p<=9){

digitalWrite(but_3,HIGH);

delay(50);

digitalWrite(but_3,LOW);

p++;

group = GROUP_4;}
```

```
else if(p==10){

    digitalWrite(but_3,HIGH);

    delay(50);

    digitalWrite(but_3,LOW);

    p=1;

    group = GROUP_0;}

break;

case G4_4:

if(p<=9){

    digitalWrite(but_4,HIGH);

    delay(50);

    digitalWrite(but_4,LOW);

    p++;

    group = GROUP_4;}

else if(p==10){

    digitalWrite(but_4,HIGH);

    delay(50);

    digitalWrite(but_4,LOW);

    p=1;
```

```
    group = GROUP_0;}

    break;

case G4_5:

    if(p<=9){

        digitalWrite(but_5,HIGH);

        delay(50);

        digitalWrite(but_5,LOW);

        p++;

        group = GROUP_4;}

    else if(p==10){

        digitalWrite(but_5,HIGH);

        delay(50);

        digitalWrite(but_5,LOW);

        p=1;

        group = GROUP_0;}

    break;

case G4_6:

    if(p<=9){
```

```
digitalWrite(but_6,HIGH);

delay(50);

digitalWrite(but_6,LOW);

p++;

group = GROUP_4;}

else if(p==10){

digitalWrite(but_6,HIGH);

delay(50);

digitalWrite(but_6,LOW);

p=1;

group = GROUP_0;}

break;

case G4_7:

if(p<=9){

digitalWrite(but_7,HIGH);

delay(50);

digitalWrite(but_7,LOW);

p++;

group = GROUP_4;}
```

```
else if(p==10){

    digitalWrite(but_7,HIGH);

    delay(50);

    digitalWrite(but_7,LOW);

    p=1;

    group = GROUP_0;}

break;

case G4_8:

if(p<=9){

    digitalWrite(but_8,HIGH);

    delay(50);

    digitalWrite(but_8,LOW);

    p++;

    group = GROUP_4;}

else if(p==10){

    digitalWrite(but_8,HIGH);

    delay(50);

    digitalWrite(but_8,LOW);

    p=1;
```

```
    group = GROUP_0;}

    break;

case G4_9:

    if(p<=9){

        digitalWrite(but_9,HIGH);

        delay(50);

        digitalWrite(but_9,LOW);

        p++;

        group = GROUP_4;}

    else if(p==10){

        digitalWrite(but_9,HIGH);

        delay(50);

        digitalWrite(but_9,LOW);

        p=1;

        group = GROUP_0;}

    break;

case G4_STAR:

    if(p<=9){
```

```
digitalWrite(but_star,HIGH);

delay(50);

digitalWrite(but_star,LOW);

p++;

group = GROUP_4;}

else if(p==10){

digitalWrite(but_star,HIGH);

delay(50);

digitalWrite(but_star,LOW);

p=1;

group = GROUP_0;}

break;

case G4_SQR:

if(p<=9){

digitalWrite(but_sqr,HIGH);

delay(50);

digitalWrite(but_sqr,LOW);

p++;

group = GROUP_4;}
```

```
else if(p==10){

    digitalWrite(but_sqr,HIGH);

    delay(50);

    digitalWrite(but_sqr,LOW);

    p=1;

    group = GROUP_0;}

break;

case G4_CANCEL:

    digitalWrite(but_spk,HIGH);

    delay(50);

    digitalWrite(but_spk,LOW);

    p=1;

    group = GROUP_0;

break;

}

break;

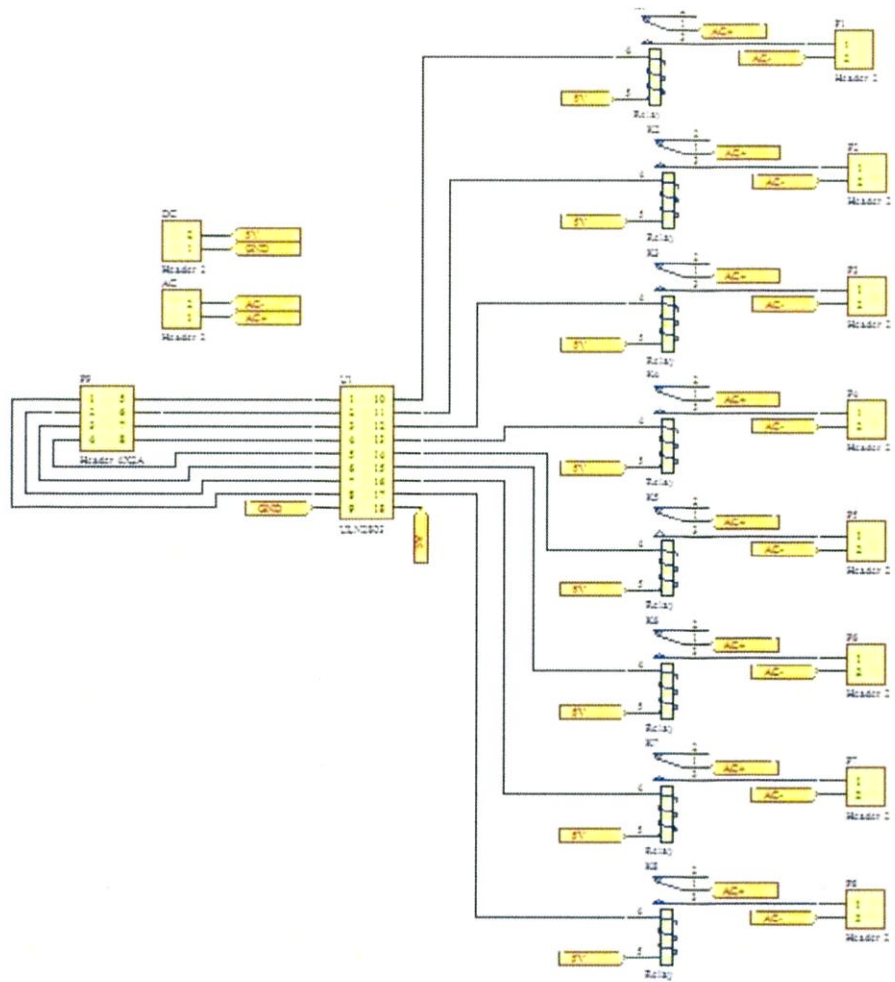
}

}
```

ภาคผนวก ข.

วงจรถ่าย

วงจรควบคุมการเปิดปิดของหลอดไฟและอุปกรณ์ไฟฟ้า



วงจรควบคุมการโทรออกและรับสายของโทรศัพท์

