ไม้เท้าอัจฉริยะ
INTELLIGENT STAFF

โดย

นายกันต์ธีร์ มะรอแม

นายชัชชวิศ โชติมาโนช

นายจำรัส ศรีจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ไม้เท้าอัจฉริยะ

INTELLIGENT STAFF

โดย

นายกันต์ธีร์	มะรอแม	57010060
นายจำรัส	ศรีจันทร์	57010174
นายชัชชวิศ	โชติมาโนช	57010289

อาจารย์ที่ปรึกษา

รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

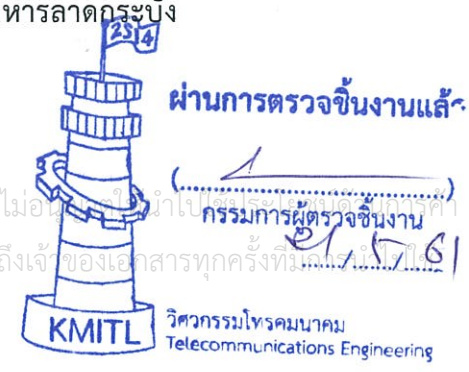
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไม้เท้าอัจฉริยะ

INTELLIGENT STAFF

ผู้จัดทำ

1. นายกันต์ธีร์ มะรอแม 57010060
2. นายจำรัส ศรีจันทร์ 57010174
3. นายชัชชวิศ โชติมาโนช 57010289

..... อาจารย์ที่ปรึกษา

(รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือของ รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี อาจารย์ที่ปรึกษาโครงงาน ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ อันเป็นประโยชน์อย่างยิ่งในการทำโครงงาน อีกทั้งยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการดำเนินงานอีกด้วย

สุดท้ายนี้ ผู้วิจัยขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจเสมอมา

นายกันต์ธีร์ มะรอแม
นายจำรัส ศรีจันทร์
นายชัชชวิศ โชติมาโนช
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม้เท้าอัจฉริยะ

INTELLIGENT STAFF

โดย นายกันต์ธีร์ มะรอแม 57010060
นายจำรัส ศรีจันทร์ 57010174
นายชัชชวิศ โชติมาโนช 57010289

อาจารย์ที่ปรึกษา รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี

บทคัดย่อ

ปริญญานิพนธ์นี้ได้นำเสนอเรื่องไม้เท้าอัจฉริยะสำหรับผู้พิการทางสายตา โดยมีวัตถุประสงค์เพื่อศึกษาการใช้งานเซนเซอร์ชนิดต่างๆและเขียนโปรแกรมเพื่อควบคุมการทำงานเซนเซอร์ ไม้เท้าที่พัฒนาขึ้นมานี้สามารถตรวจจับสิ่งกีดขวางได้ภายในระยะ 3 เมตร และแจ้งเตือนผู้ใช้งานด้วยการสั่นและเสียง หากสิ่งกีดขวางอยู่ในระดับสูง ระดับกลาง และระดับต่ำก็จะส่งเสียงเตือนที่แตกต่างกัน รวมทั้งไม้เท้าจะส่งเสียงเตือนถี่มากขึ้นเมื่อสิ่งกีดขวางอยู่ใกล้มากขึ้น อีกทั้งยังสามารถบอกเวลา ณ ขณะนั้นเมื่อผู้ใช้งานกดสวิตช์ โดยมีการส่งเสียงผ่านลำโพง ไม้เท้าอัจฉริยะนี้สามารถนำไปพัฒนาต่อยอดเพื่อเพิ่มความสะดวกสบายให้กับผู้ใช้งานได้มากยิ่งขึ้น

ABSTRACT

This thesis presents the intelligent staff for the blind. The objective is to study the various types of sensors and write a program to control the sensors. The staff developed can detect obstacles within a distance of 3 meters and alert the user via sound and vibration. In case of the obstruction is high medium and low, it will alarm differently. The staff will also sound more often when the obstruction is closer. It also tells the time now when the user press switch and remind via loudspeaker. The intelligent staff can be used to develop to increase the convenience for the user even more.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VIII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 บอร์ดอาดูโน่ (Arduino MEGA 2560 R3)	2
	2.2 Servo Motor S3003	5
	2.3 มอเตอร์สั่น (Coin Vibration Motor)	7
	2.4 SD card module	8
	2.5 Ultrasonic Range Finder LV-MaxSonar-EZ0	10
	2.6 Passive Buzzer	11
	2.7 โมดูลนาฬิกา (RTC-DS3231)	13
	2.8 โปรแกรม Arduino	15
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	16
	3.1 การออกแบบ	16
	3.2 เครื่องมือที่ใช้ในการทดลอง	27
	3.3 การจัดเก็บผลการทดลอง	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**35**
	4.1 การทดสอบการตรวจจับสิ่งกีดขวาง	35
	4.2 การทดสอบการแสดงวัน เวลาและอุณหูมิ	44
	4.3 การทดสอบเสียงของ Passive Buzzer	45
	4.4 การทดสอบเสียงเวลาผ่านลำโพง	46
	4.5 การเชื่อมต่อโดยรวม	50
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**48**
	5.1 สรุปผล	48
	5.2 ข้อเสนอแนะ	48
บรรณานุกรม		**49**
ภาคผนวก	โค้ดคำสั่ง	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Arduino MEGA2560 R3	3
2.2	Arduino Board (Model: Arduino MEGA2560 R3)	3
2.3	Servo motor s3003	5
2.4	ส่วนประกอบภายใน Servo Motor	6
2.5	การหมุนมุมของ Servo Motor โดยแบ่งพัลส์	6
2.6	มอเตอร์สั่นขนาดเล็ก	7
2.7	ภายในมอเตอร์สั่น (Vibration motor)	7
2.8	SD Card Module	8
2.9	Ultrasonic Range Finder LV-MaxSonar-EZ0	10
2.10	Passive Buzzer	11
2.11	วงจรภายใน Passive Buzzer	12
2.12	Real Time Clock (RTC)	13
2.13	การใช้งาน Arduino IDE	15
3.1	บล็อกไดอะแกรมของระบบการประมวลผลของไม้เท้าอัจฉริยะ	16
3.2	Flow Chart แสดงขั้นตอนการตรวจจับสิ่งกีดขวางและการแจ้งเตือนเวลา	17
3.3	Flow chart การแจ้งเตือนด้วยเสียงเตือนโทนต่างๆเมื่อพบสิ่งกีดขวาง	18
3.4	โปรแกรม Arduino IDE	19
3.5	หน้าต่างเครื่องมือของโปรแกรม ARDUINO IDE	19
3.6	โค้ดการทำงานของ Ultrasonic LV-Maxsonar EZ0	20
	โค้ดการทำงานของ Ultrasonic LV-Maxsonar EZ0(ต่อ)	21
3.7	การแปลงไฟล์สำหรับ SD Card Module	22
3.8	โค้ดการทำงานของ SD Card Module	23
3.9	คำสั่งการทำงานของ Real Time Clock (RTC) เพื่อหาเวลา	24
3.10	โค้ดการทำงานของ Passive Buzzer	25
3.11	โค้ดการทำงานของ Vibration Motor	25
3.12	โค้ดการทำงานของ Servo Motor	26
3.13	การเชื่อมต่อ Ultrasonic LV-Maxsonar EZ0 กับ Arduino Mega2560 R3	27
3.14	การเชื่อมต่อ Real Time Clock กับ Arduino Mega2560 R3	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่

3.15	การเชื่อมต่อ SD Card Module กับ Arduino Mega2560 R3	29
3.16	การเชื่อมต่อ Passive Buzzer กับ Arduino Mega2560 R3	30
3.17	การเชื่อมต่อ Servo Motor กับ Arduino Mega2560 R3	31
3.18	การเชื่อมต่อ Vibration motor กับ Arduino UNO	32
3.19	การเชื่อมต่อ ลำโพง กับ Arduino Mega2560 R3	33
4.1	Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 60 องศา	35
4.2	Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 60 องศา	36
4.3	Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 60 องศา	36
4.4	Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 80 องศา	37
4.5	Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 80 องศา	37
4.6	Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 80 องศา	38
4.7	Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 100 องศา	38
4.8	Serial monitor เทียบกับ Oscilloscopeที่ระยะกลางมุม 100 องศา	39
4.9	Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 100 องศา	39
4.10	Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 120 องศา	40
4.11	Serial monitor เทียบกับ Oscilloscopeที่ระยะกลางมุม 120 องศา	40
4.12	Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 120 องศา	41
4.13	Serial monitor เทียบกับ Oscilloscope ในระยะที่ไม่สามารถตรวจจับสิ่งกีดขวางได้	41
4.14	การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่สามารถก้าวผ่านได้ในระยะใกล้	42
4.15	การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่ไม่สามารถก้าวผ่านได้ในระยะกลาง	42
4.16	การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่ไม่สามารถผ่านได้	43
4.17	การใช้งานจริงเมื่อตรวจจับไม่เจอสิ่งกีดขวาง	43
4.18	วันและเวลาโดยแสดงผ่าน Serial monitor	44
4.19	สัญญาณของ Passive Buzzer ที่ความถี่ 200Hz	45
4.20	สัญญาณของ Passive Buzzer ที่ความถี่ 1000Hz	45
4.21	สัญญาณของ Passive Buzzer ที่ความถี่ 2000Hz	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่

4.22 สัญญาณเสียงเมื่อไม่มีเสียงแจ้งเตือนผ่านลำโพง 46

4.23 สัญญาณเสียงเมื่อมีเสียงแจ้งเตือนผ่านลำโพง 47

4.24 การเชื่อมต่ออุปกรณ์บนไม้เท้าโดยรวม 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 ข้อมูลจำเพาะ Arduino Mega2560 R3 2

2.2 การทำงานของแต่ละพอร์ตใน Arduino Mega2560 R3 4

2.3 ข้อมูลจำเพาะ มอเตอร์สั่น(Coin Vibration Motor) 7

2.4 พารามิเตอร์การเชื่อมต่อของ SD Card Module 9

2.5 ขาการทำงานของ Ultrasonic sensor 11

2.6 ขาการทำงานของ Passive Buzzer 12

2.7 ขอการทำงานของ Real Time Clock 13

3.1 การเชื่อมต่อขาของ Ultrasonic sensor กับ Arduino Mega2560 R3 27

3.2 การเชื่อมต่อขาของ Real Time Clock กับ Arduino Mega2560 R3 28

3.3 การเชื่อมต่อขาของ SD Card module กับ Arduino Mega2560 R3 29

3.4 การเชื่อมต่อขาของ Passive buzzer กับ Arduino Mega2560 R3 30

3.5 การเชื่อมต่อขาของ Servo motor กับ Arduino Mega2560 R3 31

3.6 การเชื่อมต่อขาของ Vibration motor กับ Arduino Mega2560 R3 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันสังคมไทยประสบปัญหามีผู้พิการเพิ่มมากขึ้น และผู้พิการที่เราพบได้บ่อยในชีวิตประจำวันก็คือผู้พิการทางสายตา ผู้จัดทำได้มองเห็นถึงปัญหาและความต้องการของผู้พิการทางสายตา เนื่องจากผู้พิการทางสายตาไม่สามารถดำรงชีวิตได้เหมือนคนทั่วไป จะพบปัญหาต่างๆ มากมาย เช่น การมองเห็น การได้ยิน ความจำ จึงจำเป็นต้องพัฒนาเทคโนโลยีเพื่อเพิ่มความปลอดภัยและอำนวยความสะดวกต่อผู้พิการทางสายตารวมถึงเพิ่มความมั่นใจแก่ผู้ดูแลผู้พิการทางสายตาด้วย

จากความสำคัญข้างต้นจึงเป็นที่มาของปริญญานิพนธ์ไม้เท้าอัจฉริยะ เพื่อให้ผู้พิการทางสายตาที่มีความต้องการใช้งานไม้เท้าอยู่แล้ว ได้รับข้อมูลที่สามารถอำนวยความสะดวกและเพิ่มความปลอดภัยได้มากขึ้น เช่น สามารถตรวจจับและรับรู้ถึงสิ่งกีดขวางได้โดยการสั่นและเสียงเตือน รวมทั้งการที่ผู้พิการทางสายตาที่ไม่สามารถทราบถึงเวลา ก็สามารถรับรู้ได้เช่นกัน

1.2 วัตถุประสงค์

1) เพื่อศึกษาการใช้งานเซนเซอร์ได้แก่ Ultrasonic Range Finder LV-Maxsonar-EZ0, RTC DS3231, SD Card Module, Passive Buzzer Module, Servo Motor และ Arduino Mega2560 R3 และศึกษาการเขียนคำสั่งควบคุม
2) เพื่อทำการวัดระยะทางของสิ่งกีดขวางและเขียนโปรแกรมควบคุม Ultrasonic Range Finder LV-Maxsonar-EZ0 และทำการแจ้งเตือนโดยการสั่นผ่าน Vibration Motor และมีเสียงเตือน เมื่อผู้ใช้งานพบสิ่งกีดขวางภายในระยะห่างที่กำหนด รวมถึงสามารถบอกถึงระดับของสิ่งกีดขวางได้
3) เพื่อบอกเวลาแก่ผู้ใช้งานด้วยระบบเสียง

1.3 ขอบเขตของปริญญานิพนธ์

1) สามารถนำเซนเซอร์มาประยุกต์ใช้งานกับไม้เท้า
2) ไม้เท้าสามารถสั่นและส่งเสียงเตือนเพื่อบอกระยะของสิ่งกีดขวางในระดับต่างๆได้
3) ผู้ใช้สามารถทราบเวลาผ่านลำโพงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

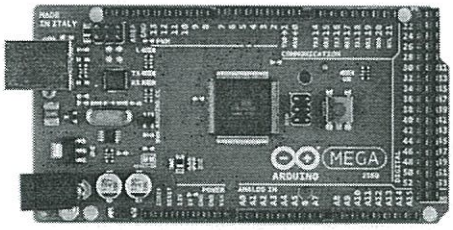
2.1 บอร์ดอาดูโน่ (Arduino MEGA 2560 R3)

Arduino MEGA2560 R3 เป็นบอร์ด Arduino สำหรับงานที่ต้องใช้ I/O มากกว่าบอร์ด Arduino Uno R3 ซึ่งขาสัญญาณ Input-Output ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Arduino Mega2560 R3 ยังมีหน่วยความจำแบบ Flash มากกว่าบอร์ด Arduino Uno R3 สามารถเขียนโปรแกรมได้มากกว่าในความเร็วของ MCU ที่เท่ากัน แสดงดังรูปที่ 2.1 และข้อมูลจำเพาะแสดงดังตารางที่ 2.1

ตารางที่ 2.1 ข้อมูลจำเพาะ Arduino Mega2560 R3 [1]

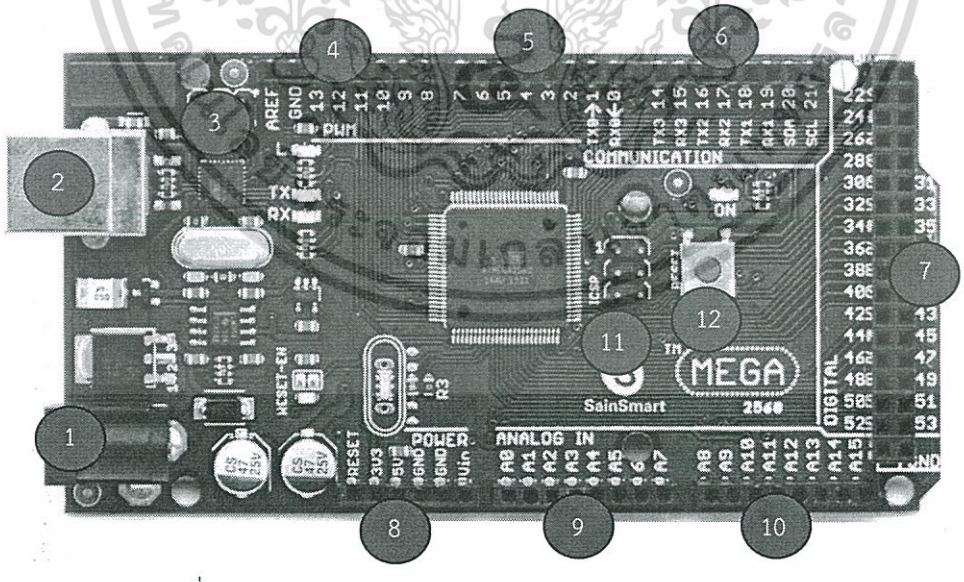
ชิปไอซีไมโครคอนโทรลเลอร์	ATmega2560
แรงดันไฟฟ้าที่ใช้งาน	5V
รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ)	7-12V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6-20V
จำนวนพอร์ต Digital I/O	54 พอร์ต (มี 14 พอร์ต PWM output)
จำนวนพอร์ต Analog Input	16 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40mA
กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	50mA
พื้นที่โปรแกรมภายใน	พื้นที่โปรแกรม 256kB 8kB ถูกใช้โดย Bootloader
พื้นที่แรม	8kB
พื้นที่หน่วยความจำถาวร (EEPROM)	4KB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 Arduino MEGA 2560 R3 [1]

Arduino Mega 2560 บอร์ดขนาดใหญ่ในของตระกูล Arduino ซึ่งมีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 ใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB พื้นที่แรม 8 KB ใช้ไฟเลี้ยงตั้งแต่ 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มีจำนวนพอร์ต Digital Input / Output มากถึง 54 พอร์ต (เป็น PWM 14 ขา) มีจำนวนพอร์ต Analog Input 16 พอร์ต มี Serial UART 4 ชุด, I2C 1 ชุด, SPI 1 ชุด สามารถเขียนโปรแกรมบน Arduino IDE และอัพโหลดโปรแกรมผ่านสาย USB บอร์ด Arduino Mega 2560 R3 เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้การพัฒนาไมโครคอนโทรลเลอร์ที่ต้องการบอร์ด Arduino ที่มีหน่วยความจำและจำนวนพอร์ตสัญญาณให้ใช้งานมากขึ้น มีโครงสร้างการทำงานของบอร์ดแสดงดังรูปที่ 2.2 แบ่งส่วนการทำงานแสดงดังตารางที่ 2.2



รูปที่ 2.2 Arduino Board (Model: Arduino Mega2560 R3) [1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 โครงสร้างบอร์ด Arduino Mega2560 R3 [1]

หมายเลข	คำอธิบาย
1	ขั้วต่อแหล่งจ่ายไฟภายนอก ใช้ได้ทั้งแบบ DC และ AC พร้อมวงจร Bridge Rectifier และ Regulate
2	ขั้วต่อ USB สำหรับเชื่อมต่ออุปกรณ์ USB
3	ICSP สำหรับ 16U2 USB Interface
4	ขาสัญญาณDigital D8-D13
5	ขาสัญญาณDigital D0-D7
6	ขาสัญญาณDigital D14-D21
7	ขาสัญญาณDigital D22-D53
8	ขั้วต่อ Power
9	ขาสัญญาณAnalog A0-A7
10	ขาสัญญาณAnalog A8-A15
11	ICSP สำหรับ ATmega2560
12	สวิตช์ RESET

จากหมายเลข 4-7 ขาสัญญาณ Digital Input-Output แบบ TTL ทำหน้าที่เป็น Input หรือ Output ตามคำสั่งจากโปรแกรม และขาสัญญาณ D20,D21 มีหน้าที่เป็น SDA,SCL ของ I2C Bus ของ I2C ได้ด้วย

จากหมายเลข 8 พอร์ต 3.3V เป็นแหล่งจ่ายไฟขนาด +3.3V ที่ได้จากวงจร Regulate สามารถจ่ายกระแสได้สูงสุด 500mA พอร์ต 5V เป็นแหล่งจ่ายไฟของบอร์ดโดยจะจ่ายกระแสได้สูงสุดไม่เกิน 500mA

จากหมายเลข 9-10 ขาสัญญาณ Analog Input แบบ ADC มีความละเอียด 10บิต รองรับแรงดัน Analog Input ได้ 0-5VDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Servo Motor S3003

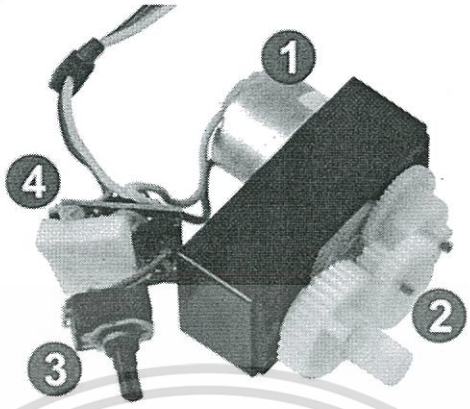
Servo Motor คือ มอเตอร์ไฟฟ้ากระแสตรง DC MOTOR แสดงดังรูปที่ 2.3 เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุนและส่ง Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ

รูปที่ 2.3 Servo motor s3003 [2]

ส่วนประกอบภายใน Servo Motor แสดงดังรูปที่ 2.4

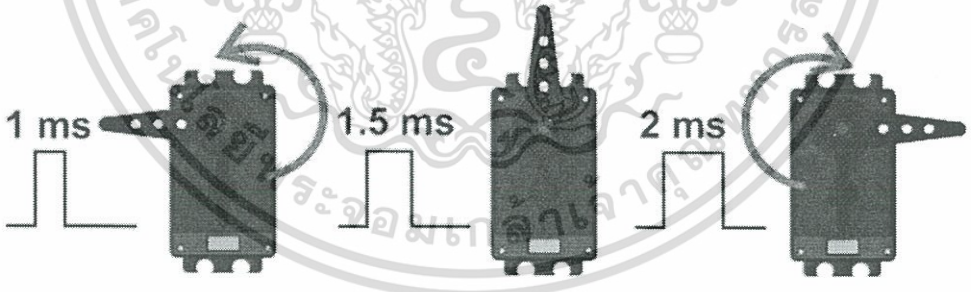
1) Motor เป็นส่วนของตัวมอเตอร์
2) Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
3) Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
4) Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ส่วนประกอบภายใน Servo Motor [2]

การทำงานโดยการหมุนมุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากความกว้างของพัลส์ที่ใช้ใน Servo Motor จะอยู่ในช่วง 1-2 ms หรือ 0.5-2.5 ms เช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางด้ายซ้ายจนสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี แสดงดังรูปที่ 2.5



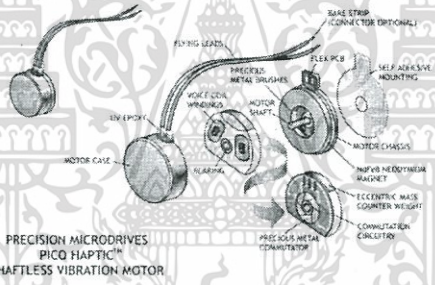
รูปที่ 2.5 การหมุนมุมของ Servo Motor โดยแบ่งพัลส์ [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มอเตอร์สั่น (Coin Vibration Motor)

มอเตอร์สั่น (Vibration motor) ขนาด 14mmx3.4mm ขนาดประมาณเหรียญ 50 สตางค์ (coin motor) ใช้งานกับแรงดัน 3 โวลต์ ใช้กับขาสัญญาณ PWM ของบอร์ด Arduino แสดงดังรูปที่ 2.6 เพื่อปรับระดับแรงดันให้ลดลงภายในมอเตอร์สั่นแสดงดังรูปที่ 2.7 ข้อมูลจำเพาะแสดงดังตารางที่ 2.3

รูปที่ 2.6 มอเตอร์สั่นขนาดเล็ก [3]



รูปที่ 2.7 ภายในมอเตอร์สั่น (Vibration motor) [3]

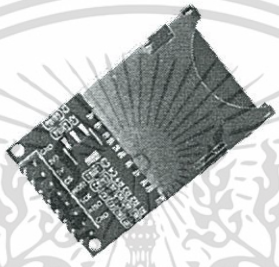
ตารางที่ 2.3 ข้อมูลจำเพาะของมอเตอร์สั่น(Coin Vibration Motor) [3]

ชนิดมอเตอร์	Micro Motor
แรงบิด	2.68 g.cm
กระแสไฟฟ้าที่ใช้	55mA
กำลังส่งออก	165mW
แรงดันไฟฟ้า	3V
ขนาดเส้นผ่าศูนย์กลาง	14mm
ความหนา	3.4mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 SD Card Module

โมดูล SD Card เป็นโมดูลที่จะทำให้บอร์ด Arduino สามารถบันทึกข้อมูลลงใน SD Card ได้ง่ายต่อการใช้งาน แสดงดังรูปที่ 2.8 ผู้ใช้งานสามารถนำไฟล์ที่บันทึกอยู่ใน SD Card ไปใช้งานต่อได้ โดยโปรแกรม Arduino IDE จะมีไลบารี่ของ SD Card ให้เสร็จสมบูรณ์พร้อมใช้งาน ข้อมูลจำเพาะของ SD card Module แสดงดังตารางที่ 2.4



รูปที่ 2.8 SD Card Module [4]

คุณสมบัติโมดูลมีดังนี้

1. รองรับการ์ด Micro SD และ Micro SDHC Card (การ์ดความเร็วสูง)
2. รองรับแรงดันไฟฟ้าที่ 3.3V หรือ 5V
3. วงจรบอร์ดสื่อสารโดยใช้อินเตอร์เฟซ Serial Peripheral Protocol (SPI) มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ข้อมูลจำเพาะของ SD Card Module [4]

โครงการ	น้อยสุด	ค่าทั่วไป	สูงสุด	หน่วย
แรงดันไฟฟ้าที่แหล่งจ่ายไฟ VCC	4.5	5	5.5	V
กระแสไฟฟ้า	0.2	80	200	mA
ระดับแรงดันการเชื่อมต่อ	3.3 หรือ 5			V
ประเภท card ที่รองรับ	การ์ด Micro SD (<= 2G), Micro SDHC Card (<= 32G)			-
ขนาด	42 X 24 X 12			มิลลิเมตร
น้ำหนัก	5			กรัม

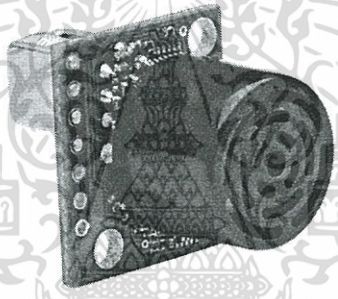
รายละเอียด :

ขาสัญญาณมีทั้งหมด 6 ขา ได้แก่ GND, VCC (3.3V และ 5.5V), MISO, MOSI, SCK และ CS โดยที่ขาสัญญาณ VCC จะเชื่อมต่อกับแหล่งจ่ายไฟ, ขาสัญญาณ MISO, MOSI และ SCK เป็นบัส SPI และขาสัญญาณ CS เป็นชิปเลือกขาสำหรับส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Ultrasonic Range Finder LV-MaxSonar-EZ0

Ultrasonic Range Finder LV-MaxSonar EZ0 เป็น Ultrasonic เซนเซอร์ขนาดเล็ก แสดงดังรูปที่ 2.9 มีความแม่นยำสูงและใช้งานง่าย โดยจะวัดค่าระยะสิ่งกีดขวางได้อย่างแม่นยำในช่วง 15 - 632 เซนติเมตร และแทบจะไม่มี dead zone เลย วัดค่าโดยใช้คลื่นอัลตราโซนิกซึ่งเป็นคลื่นเสียงที่มีความถี่สูงเกินกว่าการได้ยินของมนุษย์ เซนเซอร์นี้สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำ เหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติหรืองานด้านหุ่นยนต์ หลักการทำงานจะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของค้างคาว โดยมีขาการทำงานแสดงดังตารางที่ 2.5



รูปที่ 2.9 Ultrasonic Range Finder LV-MaxSonar-EZ0 [5]

คุณสมบัติการทำงานแสดงดังต่อไปนี้

1) ความละเอียดในการการวัดระยะสิ่งกีดขวางตั้งแต่ 15-500 เซนติเมตร

2) อัตราการอ่านค่าเซนเซอร์ 20Hz

3) ใช้กระแสไฟฟ้าที่ 2 mA

4) ใช้คลื่นความถี่สูง 42 KHz

5) ใช้แรงดันไฟเลี้ยงได้ทั้ง +3.3V หรือ +5V

6) มีเอาท์พุตจากเซนเซอร์ได้ทั้งหมด 3 แบบ ได้แก่ Analog Voltage, RS232 Serial, Pulse Width

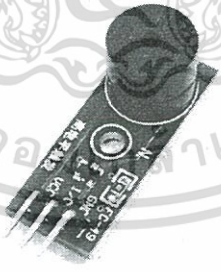
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 ขาการทำงานของ Ultrasonic sensor [5]

ขา	คำอธิบาย
BW	อินพุต
PW	เอาท์พุตดิจิตอลแบบ Pulse
AN	เอาท์พุตแบบอนาล็อก
RX	อินพุตสำหรับ Serial
TX	เอาท์พุตสำหรับ Serial
VCC	สำหรับแรงดันไฟเลี้ยง +3.3V หรือ +5V
GND	ground

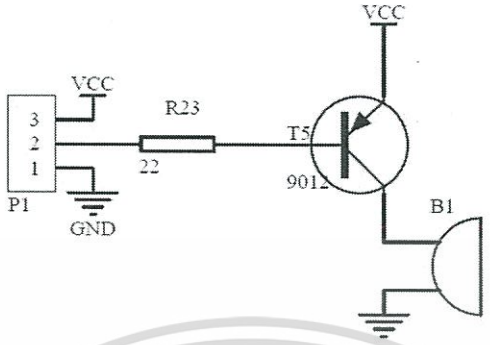
2.6 Passive Buzzer Module

พาสซีฟบัสเซอร์(Passive Buzzer) เป็นอุปกรณ์ทำหน้าที่เป็นลำโพงออกแบบมาเพื่อใช้ในการส่งสัญญาณเสียงความถี่สูงแสดงดังรูปที่ 2.10 บนบอร์ดมีทรานซิสเตอร์สำหรับช่วยขับจึง สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้โดยตรง มีวงจรภายในแสดงดังรูปที่ 2.11 สามารถใช้งานโดยเขียนโค้ดคำสั่งควบคุม โดยใน Arduino สามารถใช้ฟังก์ชั่น tone() ได้เลย ข้อดีของบัสเซอร์แบบพาสซีฟคือสามารถกำหนดความถี่เสียงที่ต้องการได้ มีคุณสมบัติการทำงานแสดงดังตารางที่ 2.6



รูปที่ 2.10 Passive Buzzer [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 วงจรภายใน Passive Buzzer [6]

ข้อมูลจำเพาะของ Passive Buzzer มีดังนี้

1) ไม่มีวงจรสร้างสัญญาณเสียงภายใน รองรับสัญญาณภายนอกได้ตั้งแต่ 500Hz - 5KHz
2) มีทรานซิสเตอร์ช่วยขับบนโมดูล
3) ทำงานได้ที่แรงดันไฟฟ้า 3.3V หรือ 5V
4) ขนาด 3.3cm*1.3cm

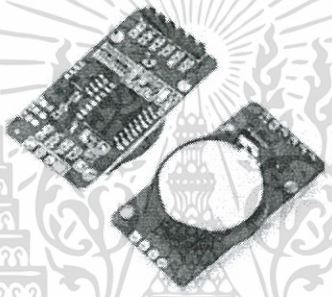
ตารางที่ 2.6 ขาการทำงานของ Passive Buzzer Module [6]

ขา	คำอธิบาย
VCC	แรงดันไฟฟ้า 3.5-5 volt
I/O	Interface of SCM
GND	Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 โมดูลนาฬิกา (RTC-DS3231)

โมดูล Real Time Clock (RTC) คืออุปกรณ์ที่ให้ค่าเวลาตามเวลาจริง แสดงดังรูปที่ 2.12 ซึ่งทำงานโดยการจับสัญญาณนาฬิกาที่ได้มาจาก Crystal ในบางรุ่นจะมีถ่านสำรองมาให้ โมดูลนี้ทำหน้าที่ในการบันทึกเวลาอย่างต่อเนื่องถึงแม้ในกรณีที่ไม่มีไฟเลี้ยงเข้ามาที่ตัวบอร์ด Arduino โมดูลเวลาก็ยังคงสามารถนับเวลาได้ต่อ ทำให้ไม่ต้องเสียเวลามาตั้งเวลาใหม่หลังจากที่หยุดจ่ายไฟเลี้ยง การทำงานแต่ละขาของโมดูลนี้แสดงดังตารางที่ 2.7 โมดูลRTCนี้จำเป็นสำหรับกับการใช้งานที่ต้องมีการบันทึกเวลา (Time Stamp) เช่น อุปกรณ์ Data logger



รูปที่ 2.12 Real Time Clock (RTC) [7]

ตารางที่ 2.7 ขาการทำงานของ Real Time Clock [7]

ขา	คำอธิบาย
VCC	สำหรับแรงดันไฟเลี้ยง +3.3V
SCL	เป็นขา I/O สำหรับสัญญาณ SCL (serial clock) สำหรับ I2C
SDA	เป็นขา I/O สำหรับสัญญาณ SDA (serial data) สำหรับ I2C
GND	Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Board มีตัวจับเวลา เช่น millis แต่ที่ต้องใช้โมดูลนาฬิกานี้ เพราะไมโครโปรเซสเซอร์ที่เป็นหัวใจในการทำงานของ Arduino board ทั้งหลายนั้น ต้องทำงานหลากหลาย เช่น คำสั่งพื้นฐาน รวมถึงการติดต่อกับอุปกรณ์ภายนอก คำสั่งจะทำงานแบบอนุกรม (Serial) คือทำทีละบรรทัด ส่งผลให้การทำงานของคำสั่งจับเวลาถูกรบกวนไปด้วย จึงไม่สามารถนำมาเป็นเวลาตามจริงที่ต้องการบันทึกไปพร้อมกับค่าอื่นๆ ที่ต้องการวัดได้

ดังนั้น ในการประยุกต์ใช้กับงานที่ต้องการเวลาที่แม่นยำและเป็นเวลาตามนาฬิกา ที่สามารถบอก วันที่ เดือน ปี ชั่วโมง นาที วินาที ก็เลยจำเป็นต้องใช้อุปกรณ์ที่ทำหน้าที่จับเวลาแยก ทำให้ต้องมีสัญญาณนาฬิกาจาก Crystal แยกต่างหาก วิธีคือ ติดต่อผ่านไปที่บอร์ดโดยใช้การสื่อสารแบบ I2C หรือ Inter - Integrated Circuit ที่ใช้ SDA SCL VCC และ GND เท่านั้น

โมดูล RTC มีอยู่หลายแบบ เช่น DS3231, DS1302, Tiny RTC I2C 24C32 DS1307 แต่ละรุ่นมีความต่างกันในความละเอียดในการจับเวลา โมดูลที่ใช้ในปริญญานิพนธ์คือ RTC DS3231 ข้อดีคือ มีการชดเชยการเปลี่ยนแปลงของสัญญาณนาฬิกา เนื่องจากการเปลี่ยนแปลงของอุณหภูมิแวดล้อม เวลาที่อุณหภูมิเปลี่ยนจะทำให้สัญญาณนาฬิกาจากCrystalเปลี่ยนตาม ซึ่งทำให้เวลามีคลาดเคลื่อน แต่โมดูลนาฬิกานี้จะทำการวัดค่าอุณหภูมิพร้อมทั้งชดเชยความเปลี่ยนแปลงนี้ไปด้วย ทำให้เวลาที่ได้มีความแม่นยำสูง

ข้อมูลจำเพาะของ RTC DS3231

1) ขนาด : 38mm(กว้าง)*22mm(สูง)
2) น้ำหนัก : 8 g
3) แรงดันไฟฟ้าที่ใช้งานได้ : 3v – 5v
4) ชิบความจำ : AT24C32 (32k)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

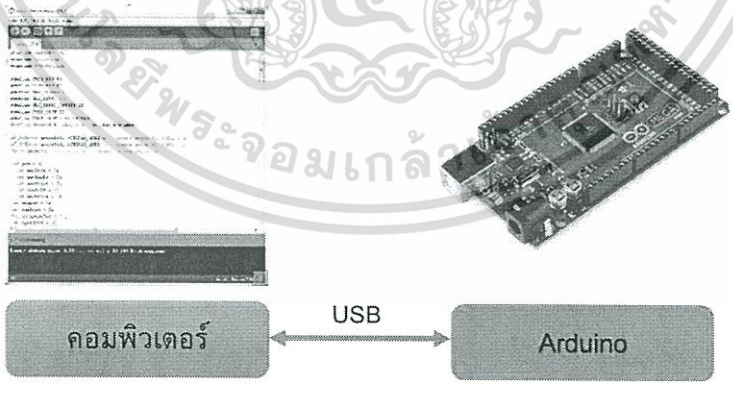
2.8 โปรแกรม Arduino

เป็นโปรแกรมบนบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software แสดงดังรูปที่ 2.13 ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด เพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield, Motor Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2. มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
3. Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
4. ราคาไม่แพง
5. Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้



รูปที่ 2.13 การใช้งาน Arduino IDE [8]

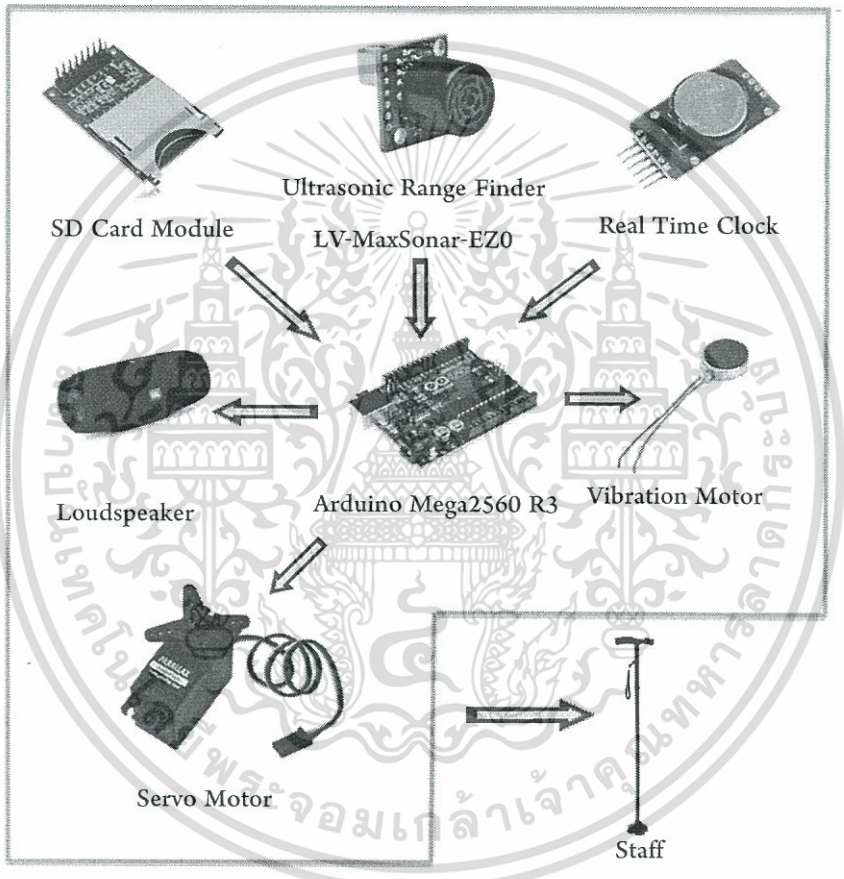
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำโครงงาน

3.1 การออกแบบ

1. ภาพรวมของโครงงาน

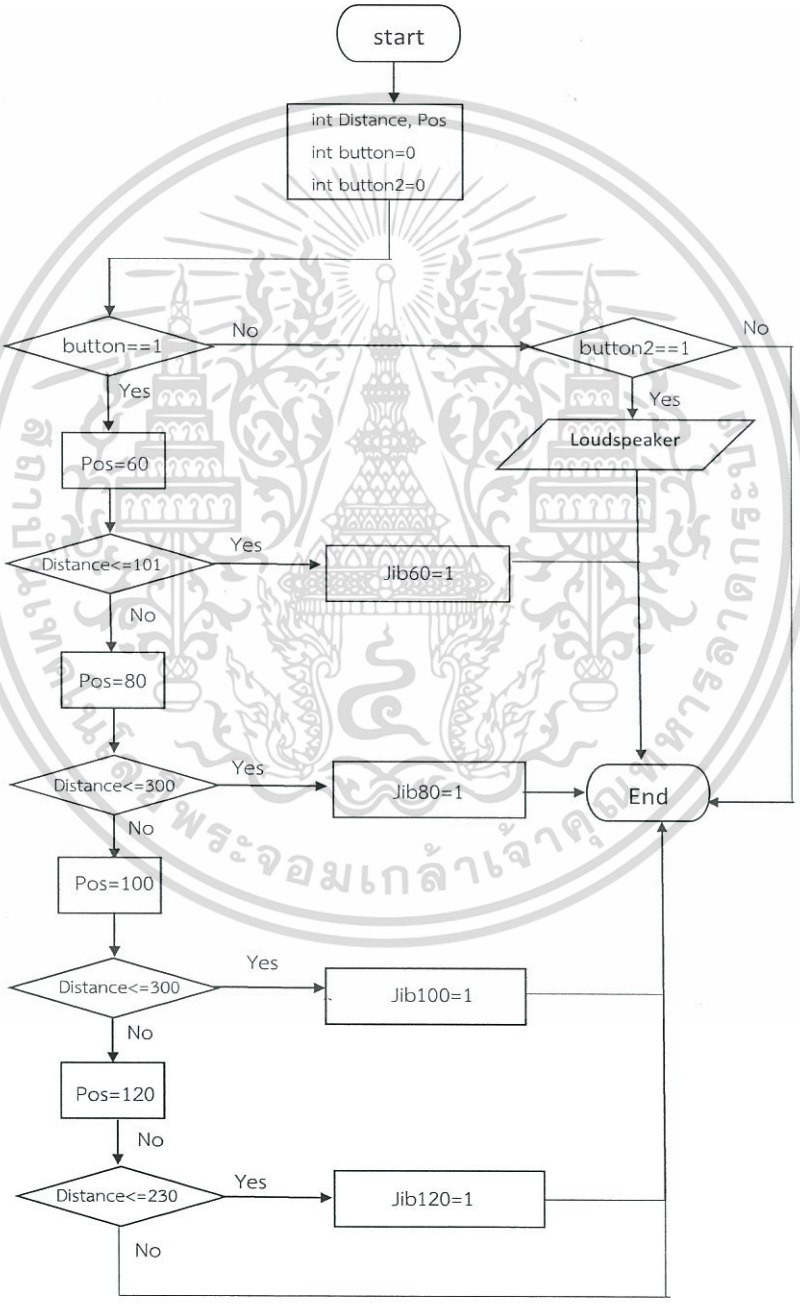


รูปที่ 3.1 บล็อกไดอะแกรมของระบบการประมวลผลของไม้เท้าอัจฉริยะ

หลักการทำงานคือไม้เท้าอัจฉริยะนี้มี Ultrasonic Range Finder LV-Maxsonar EZ0 ซึ่งใช้สำหรับตรวจจับระยะทางวัตถุในองศาต่างๆ ตามการหมุนของ Servo Motor หากตรวจพบสิ่งกีดขวางในระดับต่างๆ ไม้เท้าจะส่งเสียงแจ้งเตือนผ่าน Passive Buzzer โดยจะมีหลายโทนเสียงเพื่อให้ผู้ใช้งานได้ทราบถึงระดับที่แตกต่างกันของสิ่งกีดขวาง นอกจากนั้นยังมีการแจ้งเตือนผ่าน Vibration Motor เพื่อป้องกันสำหรับสถานที่ที่มีเสียงรบกวนมาก มีสวิตช์สำหรับกดเพื่อทราบค่าเวลา ณ ขณะนั้นๆจาก Real Time Clock ออกมาเป็นเสียงผ่านลำโพงโดยมีไฟล์เสียงที่เก็บข้อมูลไว้ใน SD Card Module

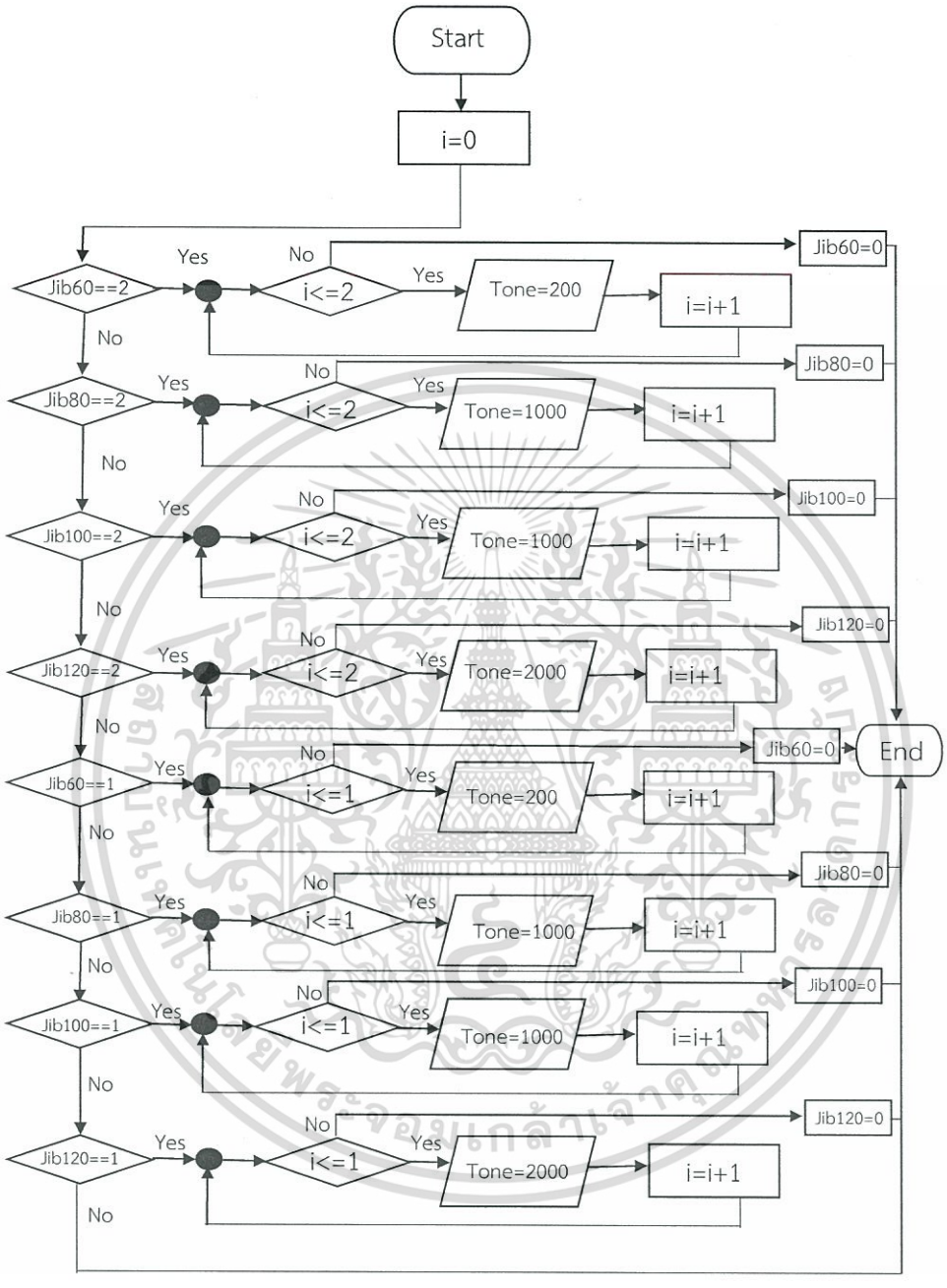
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบจะแบ่งเป็น 2 ส่วนคือ ส่วนของการกดสวิตซ์ตัวแรกเพื่อรับค่าเวลา ณ ขณะนั้นแล้วแจ้งเตือนผ่านลำโพง และส่วนที่สองคือการรับค่าจากสวิตซ์อีกตัวเมื่อได้รับการกดจะทำการตรวจจับสิ่งกีดขวางที่ระดับต่างๆ แสดงดังรูปที่ 3.2 และแจ้งเตือนด้วยเสียงโทนต่างๆ แสดงดังรูปที่ 3.3



รูปที่ 3.2 Flow Chart แสดงขั้นตอนการตรวจจับสิ่งกีดขวางและการแจ้งเตือนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

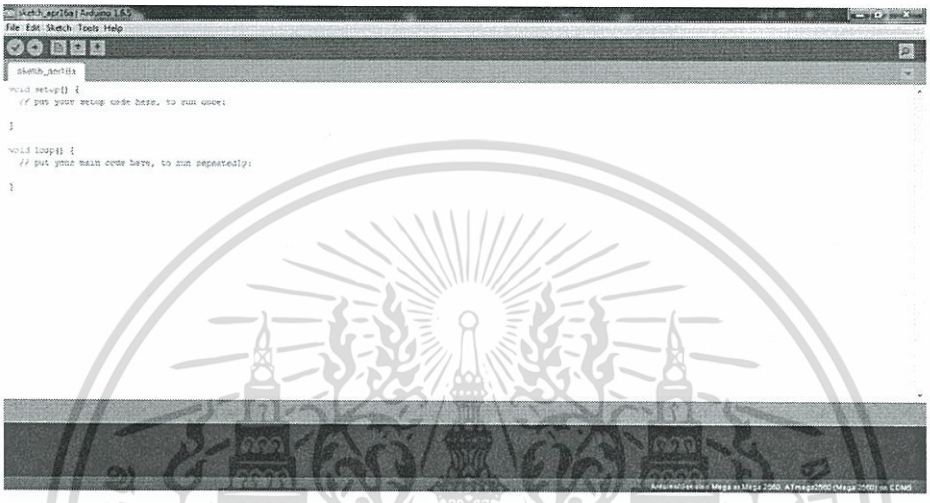


รูปที่ 3.3 Flow chart การแจ้งเตือนด้วยเสียงเตือนโทนต่างๆเมื่อพบสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

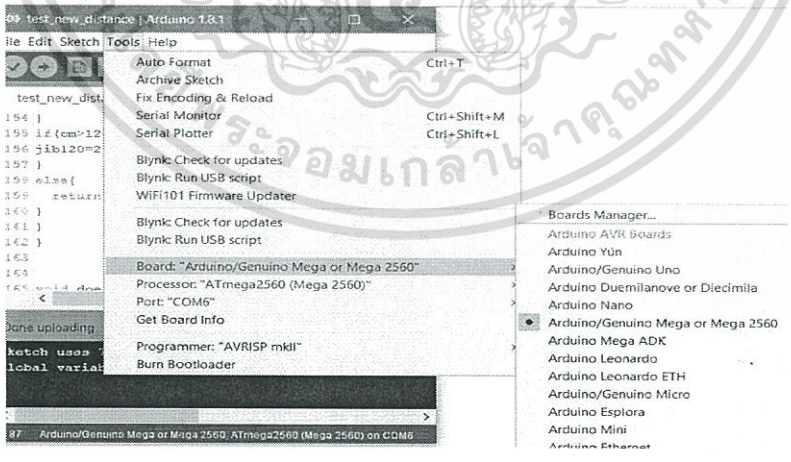
3.1.1 การออกแบบโปรแกรม Arduino Mega2560 R3 เพื่อควบคุมเซนเซอร์ต่างๆ

1) ทำการติดตั้งโปรแกรม Arduino IDE ที่ใช้ในการเขียนคำสั่งของบอร์ด Arduino Mega2560 R3 โดยสามารถดาวน์โหลดได้จาก https://www.arduino.cc/en/Main/Software แสดงดังรูปที่ 3.4



รูปที่ 3.4 โปรแกรม Arduino IDE

2) จากนั้นคลิกไปที่ Menu Tools เลือกไปที่ Board: "xxxxxx" และเลือกไปที่ Board Manager แล้วเลือก type เป็น Arduino Mega2560 R3 แสดงดังรูปที่ 3.5



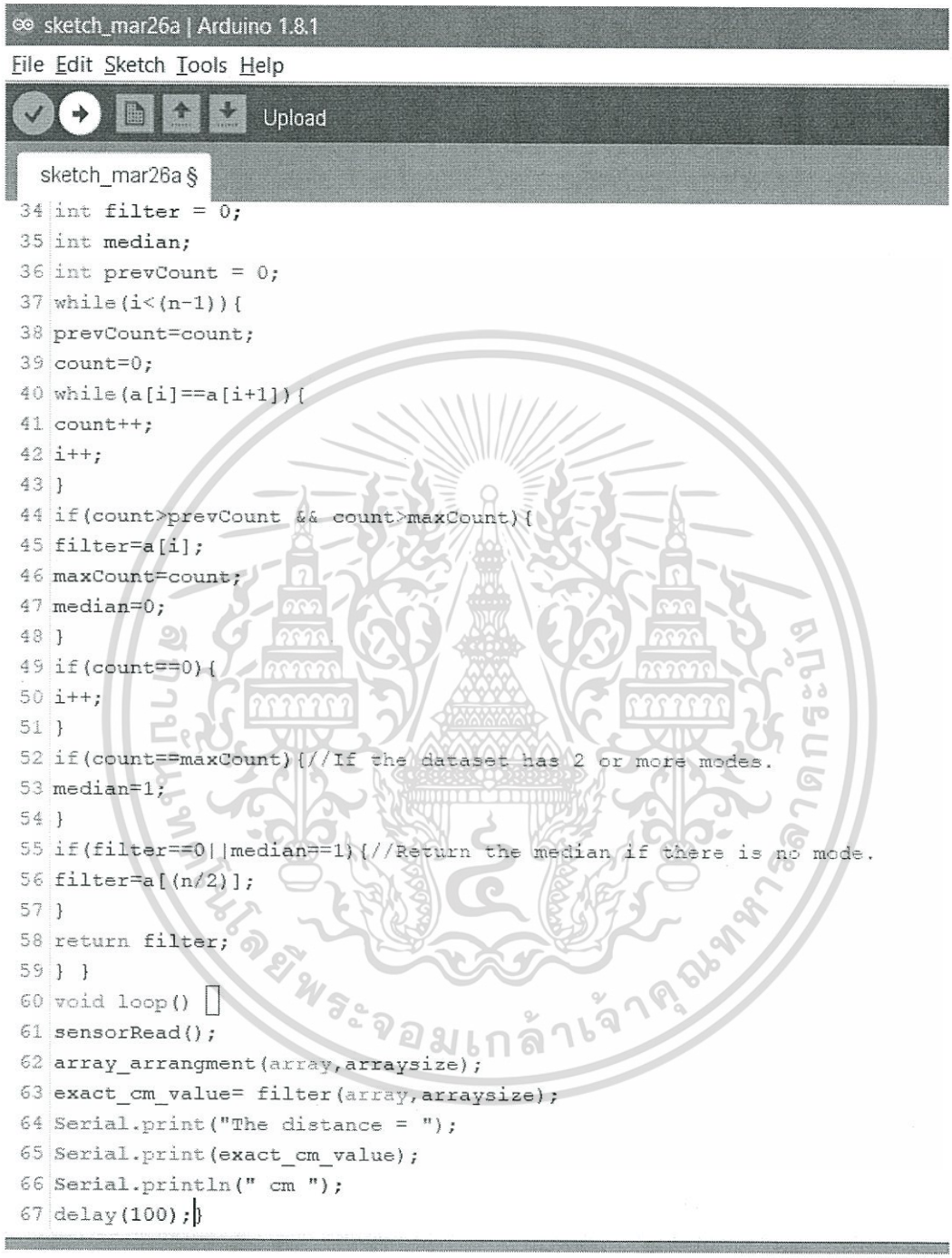
รูปที่ 3.5 หน้าต่างเครื่องมือของโปรแกรม ARDUINO IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ทำการต่อ Ultrasonic Range Finder LV-Maxsonar EZ0 เข้ากับ Arduino Mega2560 R3 แล้วป้อนคำสั่งการทำงานเพื่อตรวจวัดระยะทางของสิ่งกีดขวางโดยแสดงผลเป็นหน่วยเซนติเมตร แสดงดังรูปที่ 3.6

รูปที่ 3.6 โค้ดการทำงานของ Ultrasonic LV-Maxsonar EZ0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

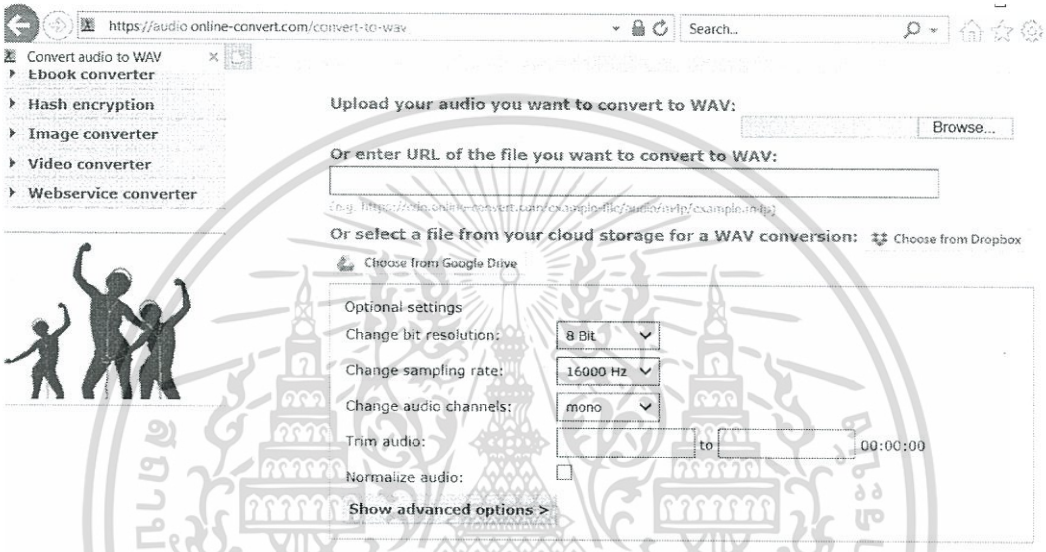



```
int filter = 0;
int median;
int prevCount = 0;
while(i<(n-1)){
prevCount=count;
count=0;
while(a[i]==a[i+1]){
count++;
i++;
}
if(count>prevCount && count>maxCount){
filter=a[i];
maxCount=count;
median=0;
}
if(count==0){
i++;
}
if(count==maxCount){//If the dataset has 2 or more modes.
median=1;
}
if(filter==0||median==1){//Return the median if there is no mode.
filter=a[(n/2)];
}
return filter;
} }
void loop()
sensorRead();
array_arrangment(array,arraysize);
exact_cm_value= filter(array,arraysize);
Serial.print("The distance = ");
Serial.print(exact_cm_value);
Serial.println(" cm ");
delay(100);}
```

รูปที่ 3.6 โค้ดการทำงานของ Ultrasonic LV-Maxsonar EZ0(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการต่อ SD Card Module เข้ากับบอร์ด Arduino Mega2560 R3 โดยมีการเก็บไฟล์เสียงที่ทำการบันทึกไว้ในแต่ละช่วงเวลา โดยจะมีการแปลงไฟล์เพื่อให้อยู่ในสกุลไฟล์ .wav ที่มี bit resolution 8 bit, sampling rate 16000 Hz และ audio channel เป็น Mono แสดงดังรูปที่ 3.7



รูปที่ 3.7 การแปลงไฟล์เสียงสำหรับ SD Card Module

เมื่อมีการกดสวิตช์เพื่อต้องการทราบ จะมีการแจ้งเตือนผ่านลำโพงโดยดึงข้อมูลจาก SD Card Module ที่ได้ทำการเก็บข้อมูลไว้ ซึ่งเขียนโปรแกรมแสดงดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

test_sound | Arduino 1.8.1

File Edit Sketch Tools Help

test_sound §

```
#include <SD.h>
#include <TMRpcm.h>

#define SD_ChipSelectPin 53

TMRpcm tmrpcm;

void setup(){

  tmrpcm.speakerPin = 46;


  if (!SD.begin(SD_ChipSelectPin)) {
    return;
  }

  tmrpcm.setVolume(6);
  tmrpcm.play("1.wav");
}

void loop(){}
```

รูปที่ 3.8 โค้ดการทำงานของ SD Card Module

5) ทำการต่อ Real Time Clock (RTC) เข้ากับ Arduino Mega2560 R3 โดยมีการบอกเวลาผ่านลำโพงเพื่อให้ผู้ใช้งานสามารถทราบค่าเวลา ณ ขณะนั้นได้ จากนั้นป้อนคำสั่งการทำงานของ Real Time Clock เพื่อตั้งค่าวันและเวลาแสดงดังรูปที่ 3.9

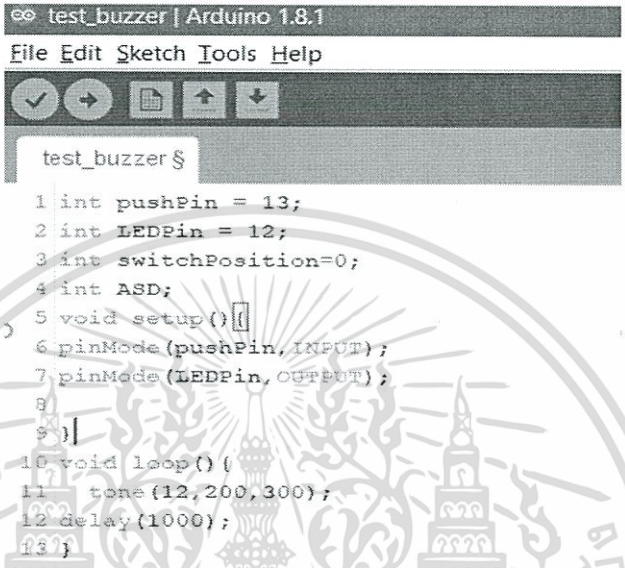
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Date, Time and Alarm functions using a DS3231 RTC connected via I2C and Wire lib

#include "Wire.h"
#include "SPI.h"  // not used here, but needed to prevent a RTClib compile error
#include "RTClib.h"
#include <SPI.h>
#include <Wire.h>

#define OLED_RESET 4

RTC_DS3231 RTC;
int x;

void setup () {
  Serial.begin(9600);
    Wire.begin();
  RTC.begin();
  RTC.adjust(DateTime(__DATE__, __TIME__));

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }

  DateTime now = RTC.now();
  // ตั้งค่าเวลาปลุกที่ต้องการ (ชั่วโมง, นาที)
    RTC.setAlarm1Simple(23, 9);
    if (RTC.checkIfAlarm(1)) {
    Serial.println("Alarm Triggered!");
    }*/
  RTC.setAlarm1Simple(23, 9);

  RTC.turnOnAlarm(1);

  RTC.turnOnAlarm(1);

  if (RTC.checkAlarmEnabled(1)) {
    Serial.println("Alarm Enabled");
  }
}
void loop () {

  DateTime now = RTC.now();
  Serial.print(now.day(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  x=now.second(),DEC;

  if (RTC.checkIfAlarm(1)) {
  Serial.println();
  Serial.print("Tempeature = ");
  Serial.print(RTC.getTemperature()); // คำสั่งดึงอุณหภูมิออกมาแสดง
  Serial.println(" C");
  Serial.println();
      delay(1000);
  }
```

รูปที่ 3.9 คำสั่งการทำงานของ Real Time Clock (RTC) เพื่อหาเวลา

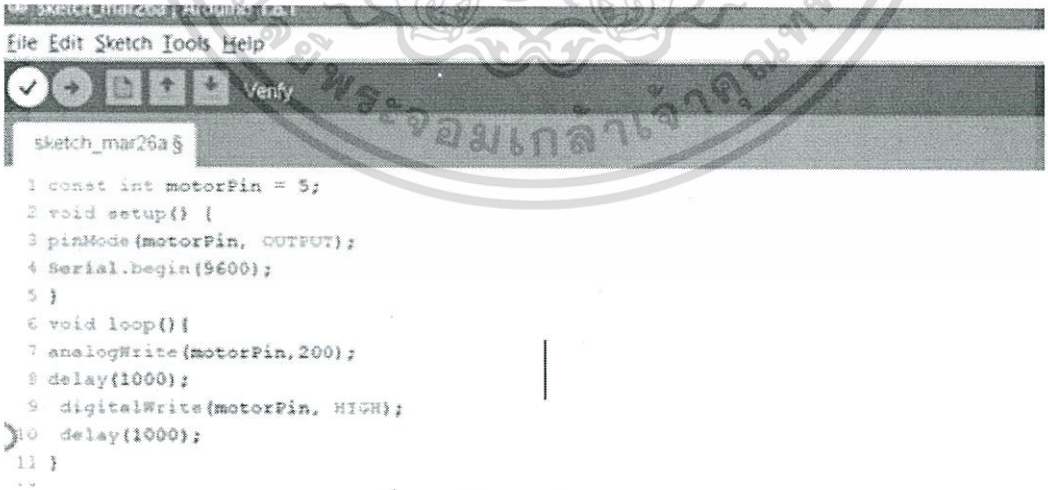
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ทำการต่อ Passive Buzzer กับ Arduino Mega2560 R3 แล้วป้อนคำสั่งการทำงานโดยให้ Passive Buzzer ส่งเสียงแจ้งเตือน แสดงดังรูปที่ 3.10



รูปที่ 3.10 โค้ดการทำงานของ Passive Buzzer

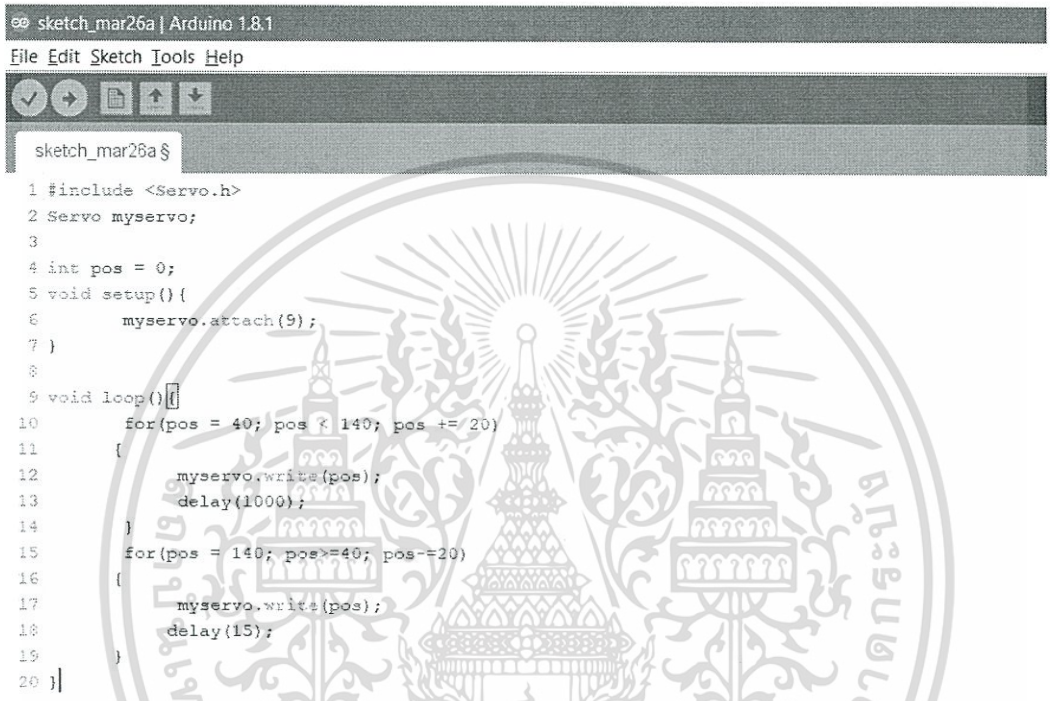
7) ทำการต่อ Vibration Motor กับ Arduino Mega2560 R3 แล้วป้อนคำสั่งการทำงานโดยให้ Vibration Motor มีการสั่นเมื่อตรวจพบสิ่งกีดขวางแสดงดังรูปที่ 3.11



รูปที่ 3.11 โค้ดการทำงานของ Vibration Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ทำการต่อ Servo Motor กับ Arduino Mega2560 R3 แล้วป้อนคำสั่งการทำงานโดย Servo Motor หมุนตั้งแต่ 40-140 องศา โดยหมุนเพิ่มทีละ 20 องศา เมื่อครบรอบจะทำการหมุนลงมาที่ 40 องศาใหม่ แสดงดังรูปที่ 3.12



รูปที่ 3.12 โค้ดการทำงานของ Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

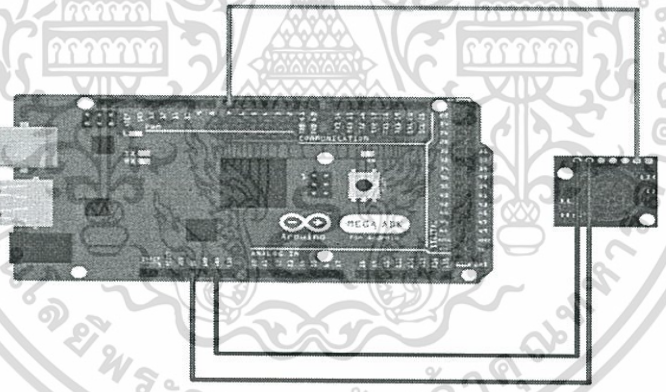
3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Ultrasonic Range Finder LV-Maxsonar EZ0

การตรวจสอบสิ่งกีดขวางสามารถทำได้โดยใช้ Ultrasonic Range Finder LV-Maxsonar EZ0 เป็นอุปกรณ์ที่ใช้ในการตรวจจับสิ่งกีดขวางโดยจะใช้หลักการส่งสัญญาณเสียงไปที่ความถี่ 20 KHz ซึ่งสามารถตรวจจับการสะท้อนของเสียงได้ตั้งแต่ 15-600 cm การเชื่อมต่อแสดงดังรูปที่ 3.13 และขาการเชื่อมต่อแสดงดังตารางที่ 3.1

ตารางที่ 3.1 การเชื่อมต่อขาของ Ultrasonic sensor กับ Arduino Mega2560 R3

Ultrasonic EZ0	Arduino Mega2560 R3
PW	7
Vcc	+5V
GND	GND



รูปที่ 3.13 การเชื่อมต่อ Ultrasonic LV-Maxsonar EZ0 กับ Arduino Mega2560 R3

กล่าวคือ ขาอินพุตที่เชื่อต่อกับ Arduino Mega2560 R3 จะถูกป้อนคำสั่งเพื่อส่งคลื่นอัลตร้าโซนิคที่ความถี่ 20 KHz และถูกแปลงออกเป็นความกว้างของพัลส์ที่ส่งออกไปที่ขาเอาต์พุตที่เชื่อมต่อกับ Arduino Mega2560 R3 และจะถูกนำไปประมวลผลเพื่อแสดงค่าออกมาเป็นระยะทางซึ่งมีหน่วยเป็นเซนติเมตร

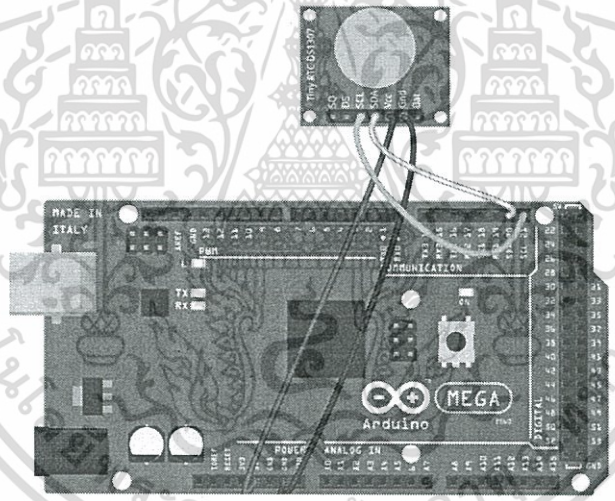
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Real Time Clock (RTC-DS3231)

อุปกรณ์ที่ใช้ในการเชื่อมต่อเพื่อทำการตรวจวัดอุณหภูมิ และตั้งค่าวัน เวลา โดยมีการเชื่อมต่อแสดงดังรูปที่ 3.14 และขาการเชื่อมต่อแสดงดังตารางที่ 3.2

ตารางที่ 3.2 การเชื่อมต่อขาของ Real Time Clock กับ Arduino Mega2560 R3

Real Time Clock	Arduino Mega2560 R3
SCL (serial clock)	SCL 21
SDA (serial data)	SDA 20
Vcc	+5V
GND	GND



รูปที่ 3.14 การเชื่อมต่อ Real Time Clock กับ Arduino Mega2560 R3

Real Time Clock จะทำหน้าที่ในการตรวจวัดเวลาและอุณหภูมิ โดยจะมีการนำผลของเวลาไปแสดงและใช้งานร่วมกับ SD Card Module และ ลำโพง เพื่อใช้ในการบอกเวลาด้วยเสียง

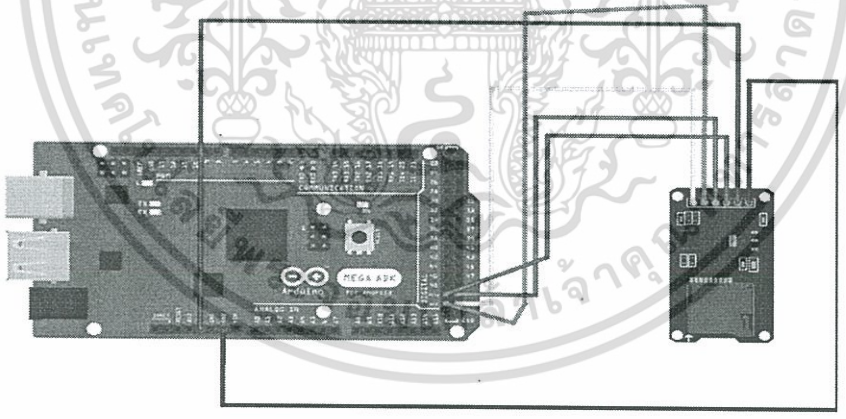
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 SD Card Module

การส่งสัญญาณแจ้งเตือนออกมาในรูปแบบเสียง ณ เวลาต่างๆจำเป็นต้องมีอุปกรณ์ที่ใช้ในการจัดเก็บไฟล์เสียง ณ เวลานั้น เพื่อให้ผู้ใช้งานสามารถทราบเวลาได้โดยจะดึงข้อมูลไฟล์เสียงจาก SD Card Module โดยแต่ละขาเชื่อมต่อกับ Arduino Mega2560 R3 แสดงดังรูปที่ 3.15 และตารางที่ 3.3

ตารางที่ 3.3 การเชื่อมต่อขาของ SD Card Module กับ Arduino Mega2560 R3

SD Card Module	Arduino Mega2560 R3
+5V	+5V
CS	53
MOSI	51
SCK	52
MISO	50
GND	GND



รูปที่ 3.15 SD Card Module กับ Arduino Mega2560 R3

SD Card Module จะทำหน้าที่ในการเก็บไฟล์เสียงที่มีนามสกุล .wav เพื่อนำไปใช้แสดงผลเวลาผ่านเสียง ณ เวลาต่างๆ

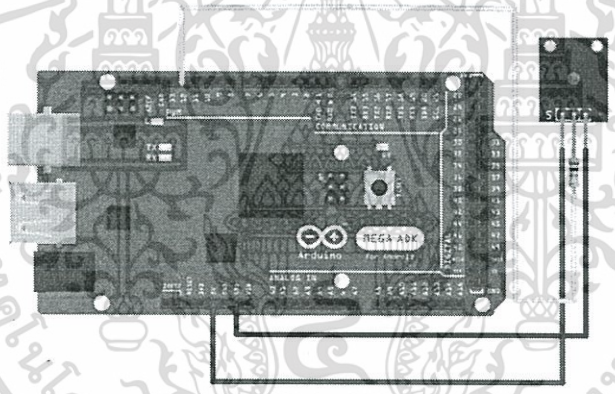
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Passive Buzzer

การตรวจสอบสิ่งกีดขวางจำเป็นต้องมีการแจ้งเตือน จึงได้มีการนำ Passive Buzzer มาใช้ในการแจ้งเตือนเพื่อให้ทราบถึงระยะของสิ่งกีดขวางโดยสามารถปรับโทนเสียงได้หลายโทน แสดงการเชื่อมต่อดังรูปที่ 3.16 และ ตารางที่ 3.4

ตารางที่ 3.4 การเชื่อมต่อขาของ Passive Buzzer กับ Arduino Mega2560 R3

Passive Buzzer	Arduino Mega2560 R3
VCC	+5V
I/O	12
GND	GND



รูปที่ 3.16 การเชื่อมต่อ Passive Buzzer กับ Arduino Mega2560 R3

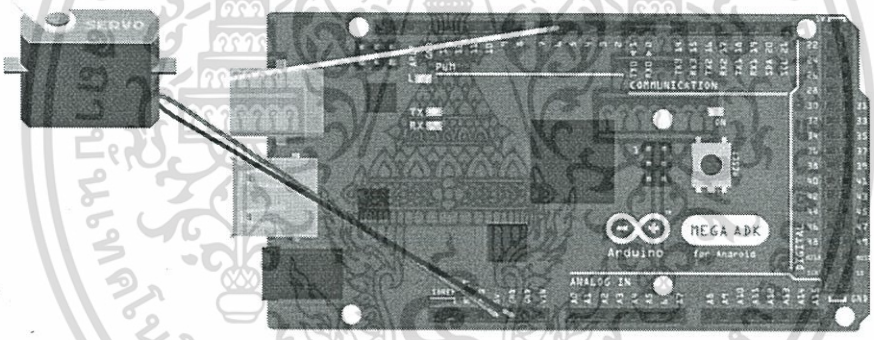
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Servo Motor

การที่จะใช้งาน Ultrasonic sensor เพื่อตรวจสอบสิ่งกีดขวางในบางครั้งอาจจะตรวจจับได้ไม่ทั่วถึงจึงจำเป็นต้องนำ Servo Motor มาประยุกต์ใช้งานเพื่อให้ Ultrasonic มีการหมุนเพื่อตรวจสอบสิ่งกีดขวางได้ในระดับต่างๆ โดยแสดงการเชื่อมต่อดังรูปที่ 3.17 และตารางที่ 3.5

ตารางที่ 3.5 การเชื่อมต่อขาของ Servo Motor กับ Arduino Mega2560 R3

Servo Motor	Arduino Mega2560 R3
VCC	+5V
I/O	D6
GND	GND



รูปที่ 3.17 การเชื่อมต่อ Servo Motor กับ Arduino Mega2560 R3

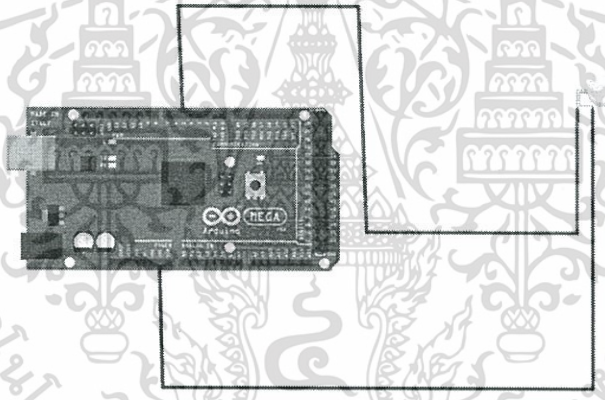
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 Vibration Motor

ในปริญญานิพนธ์นี้ทางผู้จัดทำเลือกใช้ Vibration motor ซึ่งทำหน้าที่แตกต่างจากมอเตอร์ทั่วไปเนื่องจากมอเตอร์ทั่วไปจะใช้การหมุนของขดลวดเพื่อบังคับอุปกรณ์ต่างๆ แต่ Vibration motor จะทำหน้าที่แค่สั่นเมื่อมีกระแสไฟเข้าตัวมอเตอร์ โดยสามารถสั่งการโดยใช้โปรแกรม Arduino มาควบคุมการทำงานของ Vibration motor ให้มีระดับการสั่นตามต้องการได้ โดยแสดงการเชื่อมต่อดังรูปที่ 3.18 และตารางที่ 3.6

ตารางที่ 3.6 การเชื่อมต่อขาของ Vibration motor กับ Arduino Mega2560 R3

Vibration Motor	Arduino Mega2560 R3
สายสีแดง	+5V
สายสีน้ำเงิน	D9

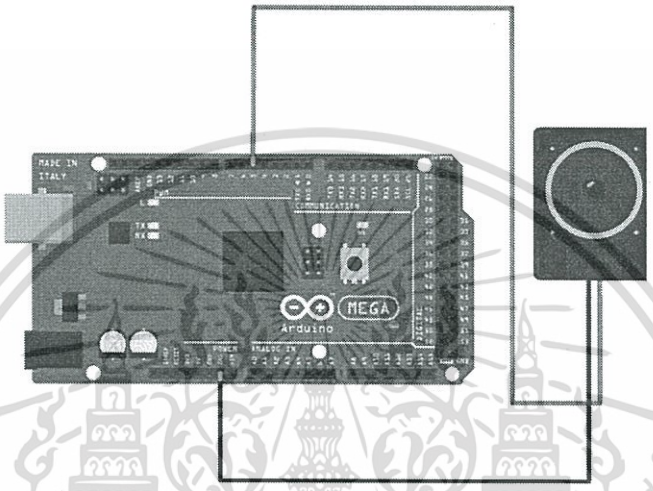


รูปที่ 3.18 การเชื่อมต่อ Vibration motor กับ Arduino Mega2560 R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 ลำโพง

ในการแจ้งเตือนด้วยเสียงเพื่อให้ผู้ใช้งานทราบจำเป็นต้องมีลำโพงเพื่อให้ง่ายต่อการใช้งาน โดยมีลักษณะการเชื่อมต่อกับ Arduino Mega2560 R3 แสดงดังรูปที่ 3.19



รูปที่ 3.19 การเชื่อมต่อ ลำโพง กับ Arduino Mega2560 R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

ในการจัดเก็บผลการทดลองเพื่อทดสอบการทำงานของไม้เท้าจะใช้เครื่องคอมพิวเตอร์ที่มีโปรแกรม Arduino เพื่อดูผลผ่าน Serial monitor จากนั้นจะกำหนดให้แสดงผลเพื่อให้ผู้ใช้งานสามารถทราบค่าต่างๆได้ผ่านลำโพงและ Passive Buzzer

3.3.1 เก็บผลการทดลองการใช้งาน Ultrasonic Range Finder LV-Maxsonar EZ0

ทำการตรวจสอบโดยนำวัตถุมาวางที่ระยะต่างๆซึ่งตัว Ultrasonic Range Finder LV-Maxsonar EZ0 จะแสดงค่าระยะทางเป็นหน่วยเซนติเมตร

3.3.2 เก็บผลการทดลองจากการใช้งาน Real Time Clock

ทำการรับค่าวันและเวลาจาก Real Time Clock โดยค่าจะมีความสามารถเก็บค่าต่อเนื่องแม้กระทั่งไม่มีการจ่ายไฟให้กับอุปกรณ์

3.3.3 เก็บผลการทดลองของ SD Card Module

เรียกไฟล์เสียงจาก SD Card ที่บันทึกไว้มาใช้โดยสกุลไฟล์เป็น .wav

3.3.4 เก็บผลการทดลองของ Servo Motor

กำหนดองศาของการหมุน Servo Motor โดยกำหนดให้หมุนตั้งแต่ 40-140 องศาโดยให้หมุนทีละ 20 องศา

3.3.5 การควบคุมสวิตซ์ให้แสดงเวลาผ่านลำโพง

กดสวิตซ์เมื่อต้องการทราบค่าเวลา ณ ขณะนั้นผ่านลำโพง

3.3.6 การควบคุมสวิตซ์ควบคุมการทำงานของ Ultrasonic Range Finder LV-Maxsonar EZ0

กดสวิตซ์เมื่อต้องการตรวจสอบสิ่งกีดขวางที่ระดับต่างๆ ตามการหมุนของ Servo Motor โดยแจ้งเตือนผ่าน Passive Buzzer ในโทนเสียงที่ต่างกันสำหรับระดับความสูงต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

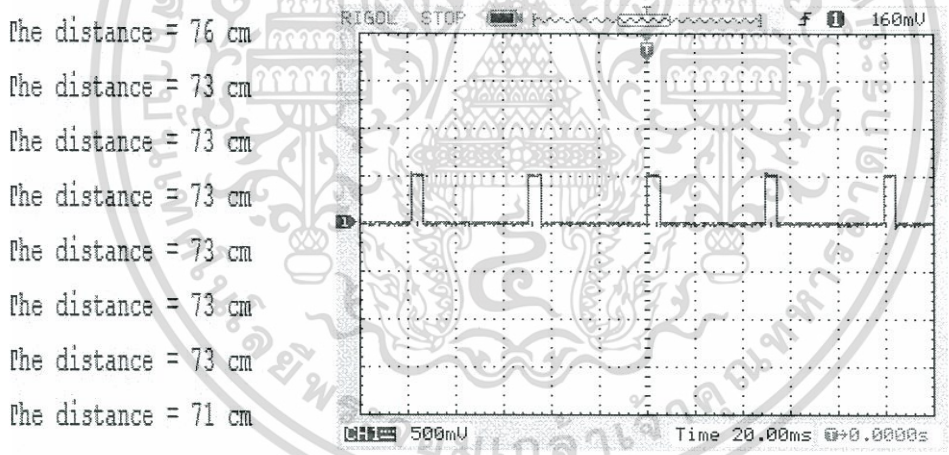
ผลการทดลอง

4.1 การทดสอบการตรวจจับสิ่งกีดขวาง

เมื่อ Ultrasonic Range Finder LV-Maxsonar EZ0 ตรวจพบสิ่งกีดขวางจะสามารถบอกระยะทางของวัตถุนั้นๆได้ว่าอยู่ห่างออกไปเป็นระยะเท่าใด ซึ่งสามารถดูได้จาก Serial Monitor เปรียบเทียบกับ Oscilloscope

-มุม 60 องศา

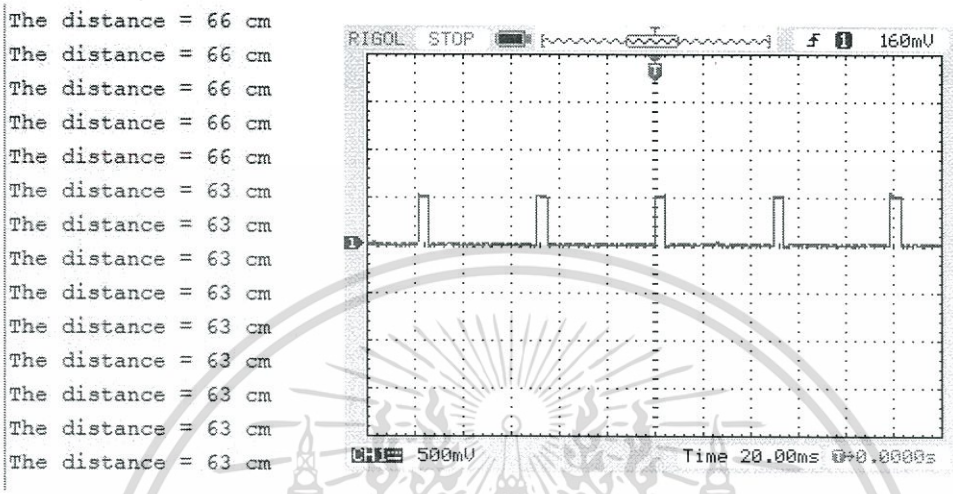
ในระยะไกล ที่ระยะ 60 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 73 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับต่ำ สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.1



รูปที่ 4.1 Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 60 องศา

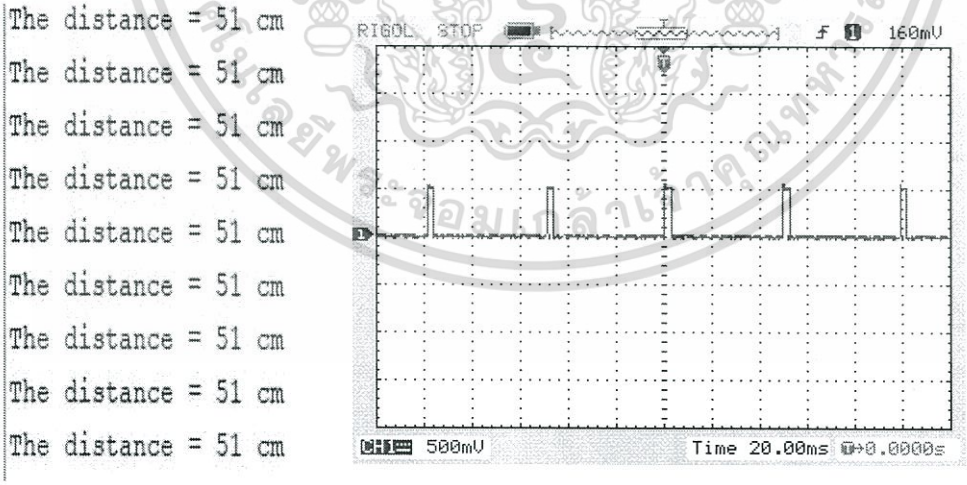
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระยะกลาง ที่ 60 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 63 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.2



รูปที่ 4.2 Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 60 องศา

ในระยะใกล้ที่ 60 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 51 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.3

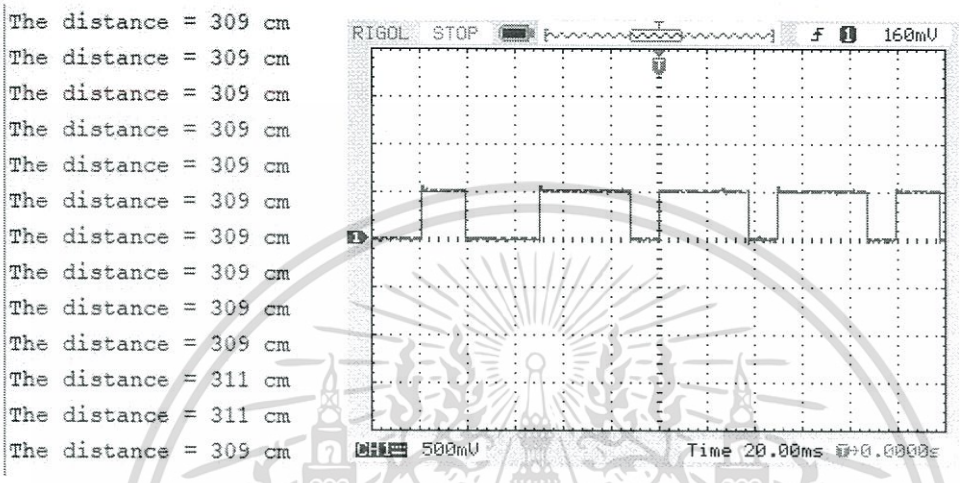


รูปที่ 4.3 Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 60 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

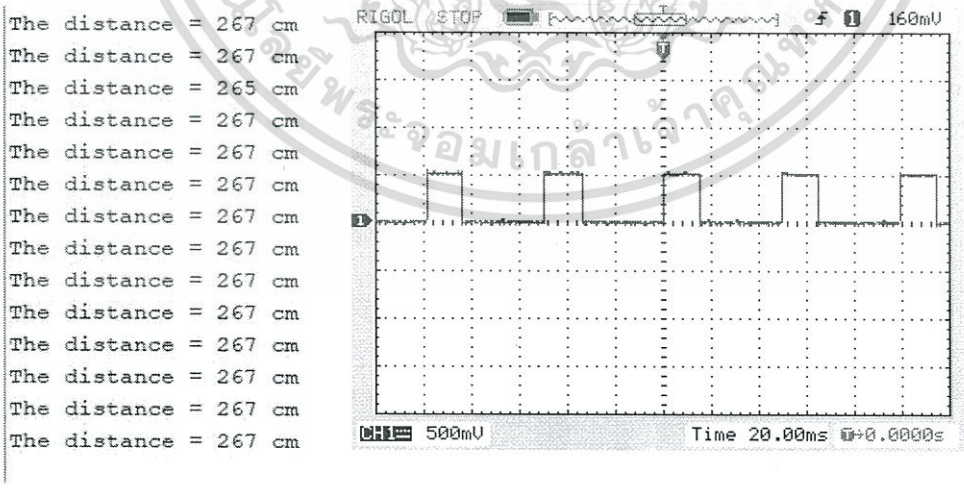
-มุม 80 องศา

ในระยะไกล ที่ระยะ 80 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 309 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับต่ำ สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.4



รูปที่ 4.4 Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 80 องศา

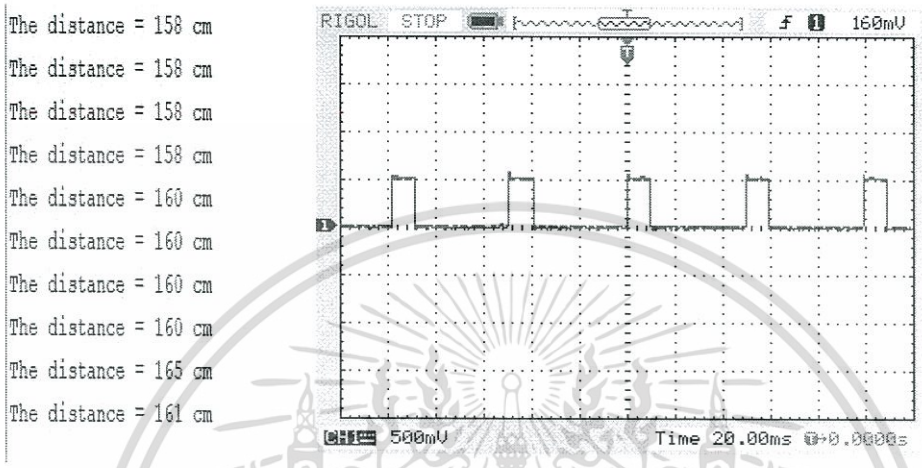
ในระยะกลางที่ 80 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 267 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.5



รูปที่ 4.5 Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 80 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

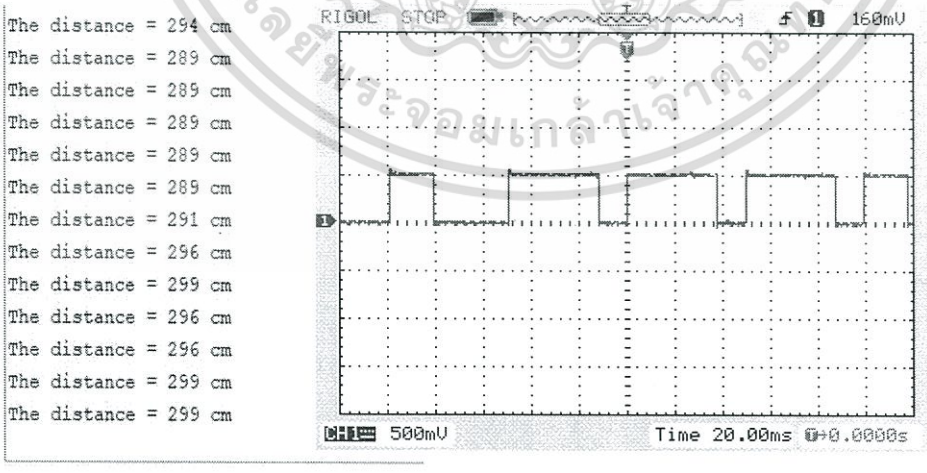
ในระยะใกล้ที่ 80 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 158 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้าวข้ามไปได้ แสดงดังรูปที่ 4.6



รูปที่ 4.6 Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 80 องศา

-มุม 100 องศา

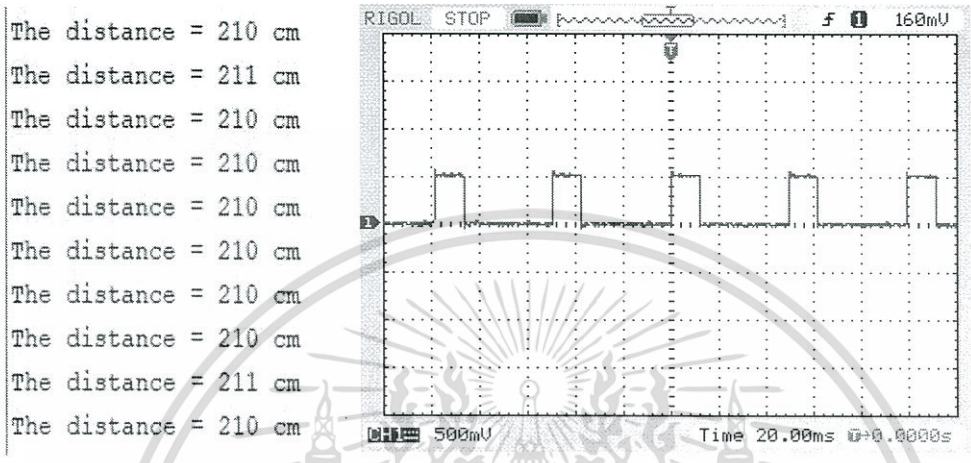
ในระยะไกลที่ 100 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 289 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับสูง สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.7



รูปที่ 4.7 Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 100 องศา

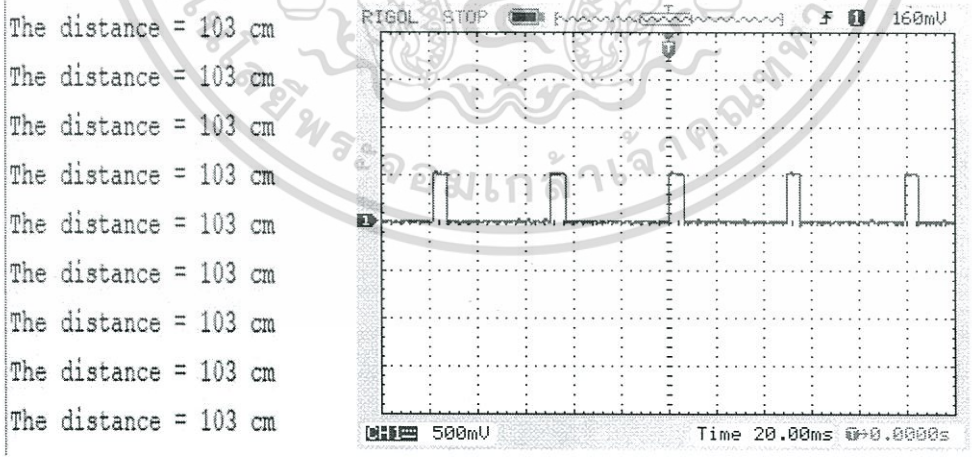
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระยะกลางที่ 100 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 210 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.8



รูปที่ 4.8 Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 100 องศา

ในระยะใกล้ที่ 100 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 105 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับกลาง ไม่สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.9

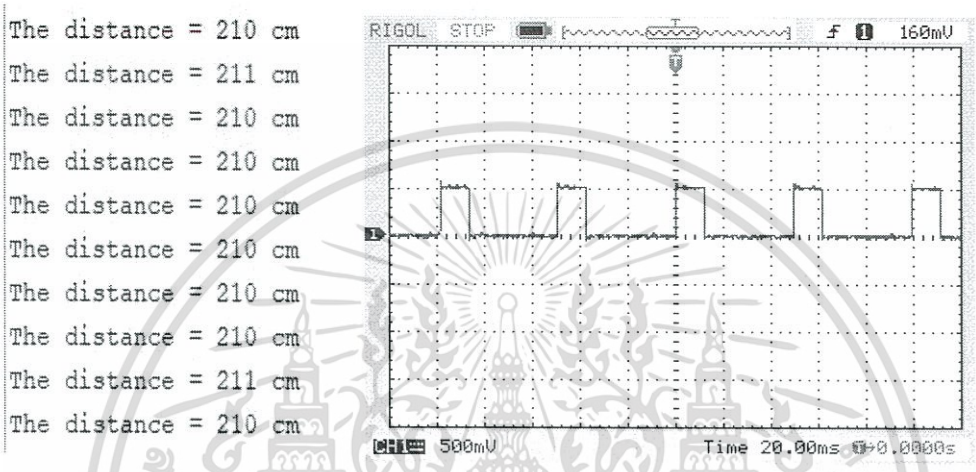


รูปที่ 4.9 Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 100 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

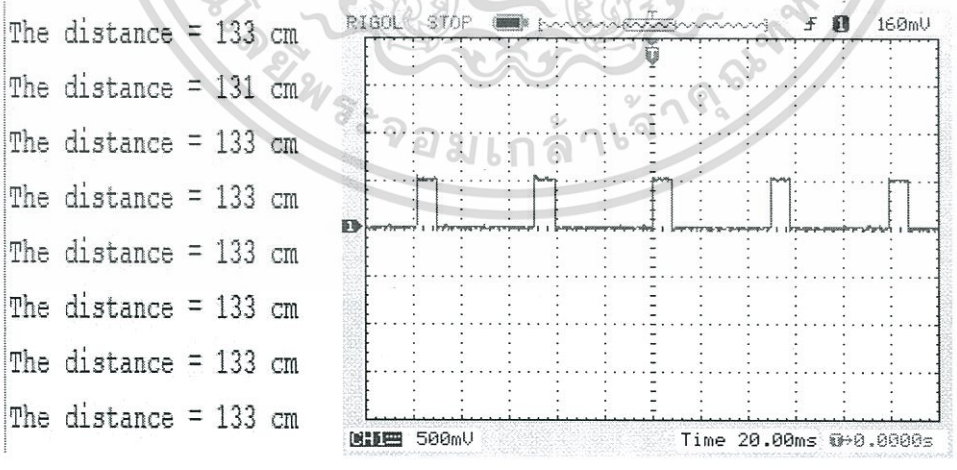
-มุม 120 องศา

ในระยะไกลที่ 120 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 210 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับสูง สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.10



รูปที่ 4.10 Serial monitor เทียบกับ Oscilloscope ที่ระยะไกลมุม 120 องศา

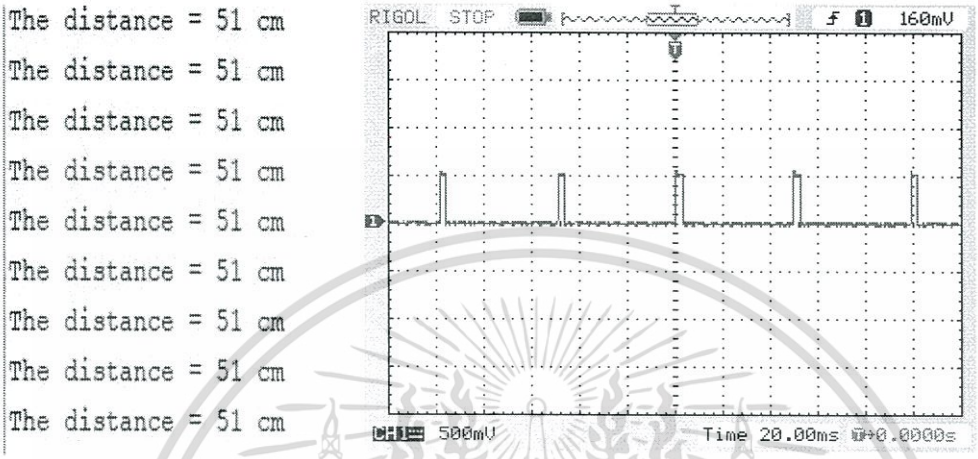
ในระยะกลางที่ 120 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 131 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับสูง ไม่สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.11



รูปที่ 4.11 Serial monitor เทียบกับ Oscilloscope ที่ระยะกลางมุม 120 องศา

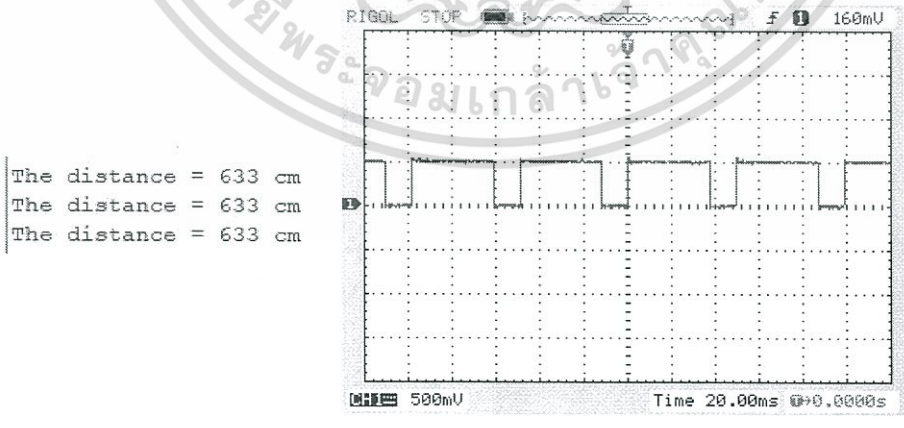
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระยะใกล้ที่ 120 องศาเมื่อสิ่งกีดขวางที่ตรวจพบมีระยะประมาณ 51 เซนติเมตร จะสามารถบอกได้ว่าสิ่งกีดขวางมีระดับสูง ไม่สามารถก้มผ่านไปได้ แสดงดังรูปที่ 4.12



รูปที่ 4.12 Serial monitor เทียบกับ Oscilloscope ที่ระยะใกล้มุม 120 องศา

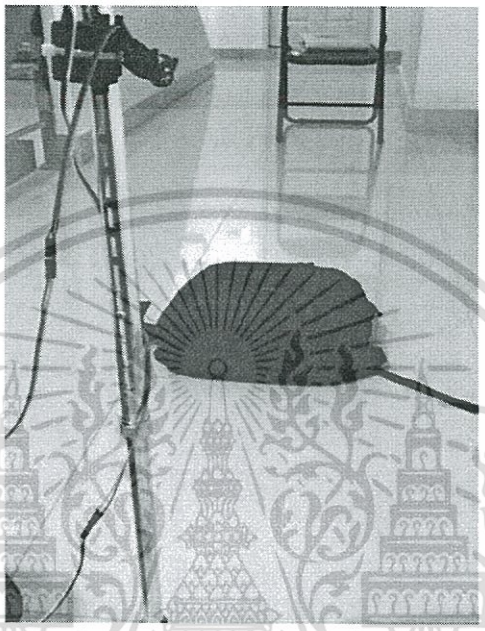
จากการตรวจพบที่องศาต่างๆทำให้เราสามารถเขียนความสัมพันธ์เพื่อแจ้งเตือนผ่าน Passive Buzzer โดยจะมีโทนเสียงที่แตกต่างกันตามระดับ เช่น สิ่งกีดขวางที่ตรวจพบอยู่ในระดับต่ำในระยะไกล จะส่งเสียงโทนต่ำที่ 1 ครั้งแต่หากสิ่งกีดขวางถูกตรวจพบในระดับต่ำในระยะใกล้ จะส่งโทนเสียงต่ำที่ 3 ครั้ง และหากสิ่งกีดขวางถูกตรวจจับในระดับสูงก้จะส่งเสียงอีกโทนหนึ่ง ซึ่งจำนวนครั้งจะดูจากระยะใกล้ไกลของวัตถุเช่นกัน โดยหากมีสิ่งกีดขวางไกลกว่าที่ตรวจจับได้จะแสดงดังรูปที่ 4.13



รูปที่ 4.13 Serial monitor เทียบกับ Oscilloscope ในระยะที่ไม่สามารถตรวจจับสิ่งกีดขวางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบจากการใช้งานจริงในสถานการณ์ต่างๆ เช่น สามารถก้าวผ่านสิ่งกีดขวางในระดับต่ำได้ แสดงดังรูปที่ 4.14 และสิ่งกีดขวางในระดับกลางที่ไม่สามารถผ่านได้แสดงดังรูปที่ 4.15

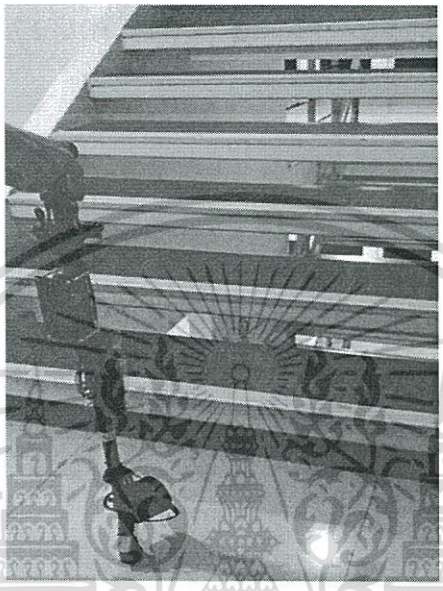


รูปที่ 4.14 การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่สามารถก้าวผ่านได้ในระยะใกล้

รูปที่ 4.15 การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่ไม่สามารถก้าวผ่านได้ในระยะกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งกีดขวางที่ไม่สามารถผ่านได้ แสดงดังรูปที่ 4.16 และเมื่อไม่เจอสิ่งกีดขวาง แสดงดังรูปที่ 4.17



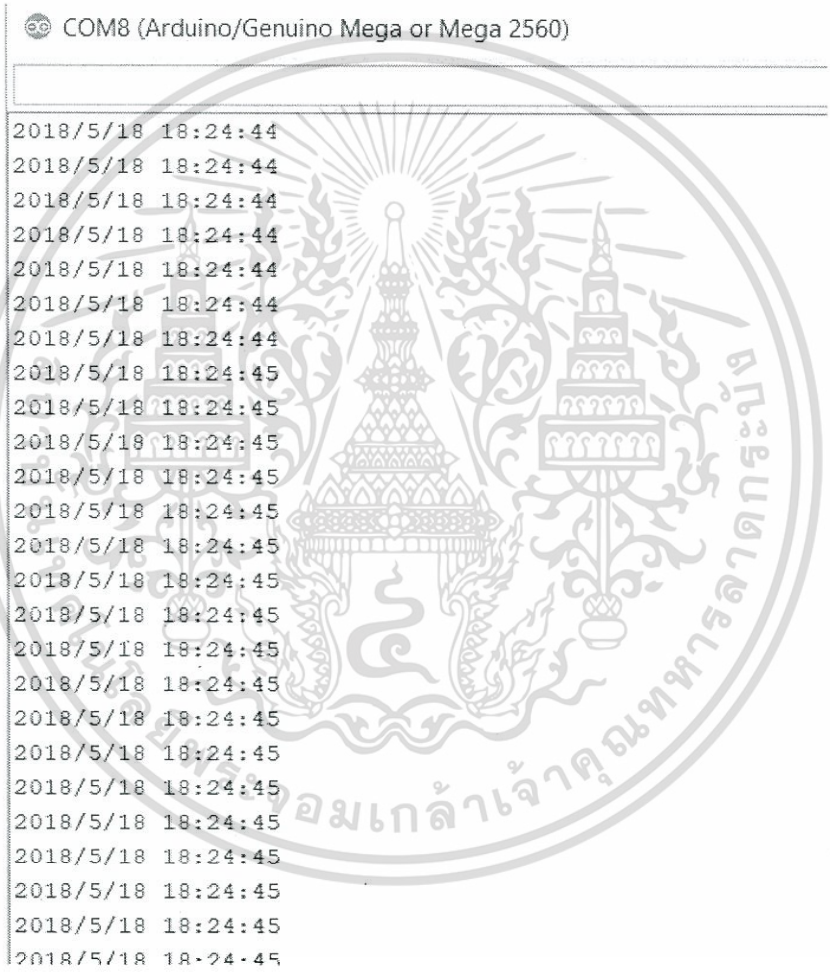
รูปที่ 4.16 การใช้งานจริงเมื่อตรวจจับสิ่งกีดขวางที่ไม่สามารถผ่านได้

รูปที่ 4.17 การใช้งานจริงเมื่อตรวจจับไม่เจอสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบการแสดงวัน เวลาและอุณหูมิ

การทดสอบการแสดงวัน เวลาและอุณหภูมิ โดยการตั้งวันและเวลาให้กับ Real time clock เนื่องจากมีแบตเตอรี่ 3 โวลต์ ทำให้เมื่อไม่มีการจ่ายไฟ Real time clock ยังสามารถจดจำค่าวันและเวลาไว้ได้ นอกจากนั้นยังสามารถบอกถึงอุณหภูมิ ณ เวลาปัจจุบันได้อีกด้วย ซึ่งแสดงผ่าน Serial monitor แสดงดังรูปที่ 4.18

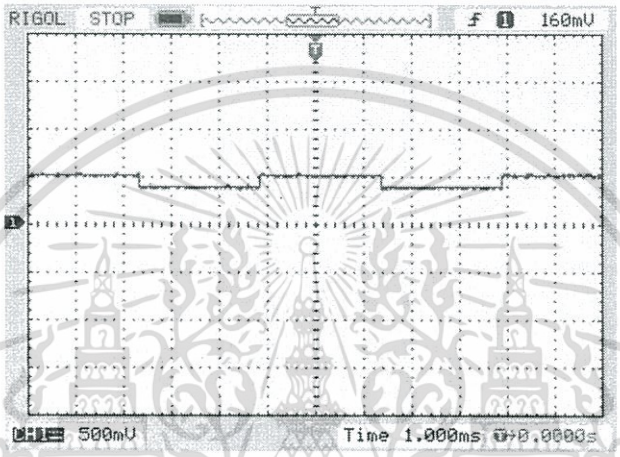


รูปที่ 4.18 วันและเวลาโดยแสดงผ่าน Serial monitor

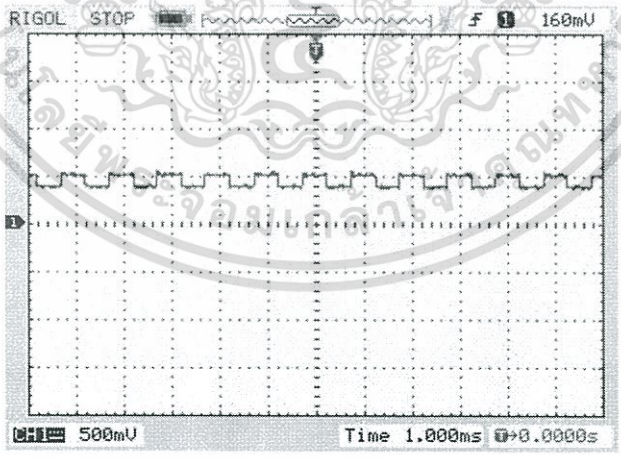
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบเสียงของ Passive Buzzer

อุปกรณ์ Passive Buzzer จะแสดงค่าสัญญาณเสียงออกมาในความถี่ต่างๆ โดยในที่นี้เราจะกำหนดให้มีความถี่ที่ 200Hz โดยมีกราฟสัญญาณแสดงดังรูปที่ 4.19 ความถี่ที่ 1000Hz แสดงดังรูปที่ 4.20 และความถี่ที่ 2000Hz แสดงดังรูปที่ 4.21

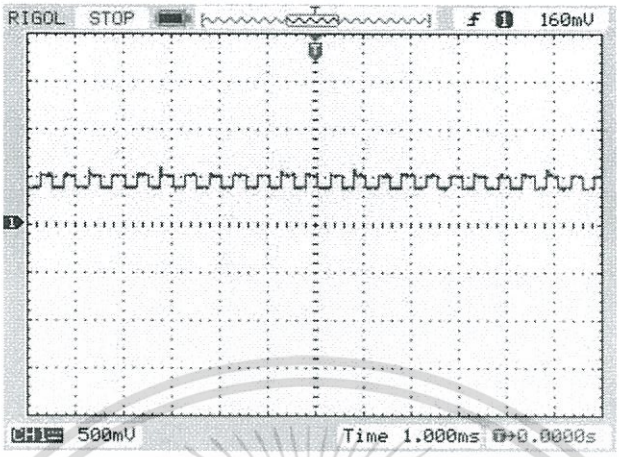


รูปที่ 4.19 สัญญาณของ Passive Buzzer ที่ความถี่ 200Hz



รูปที่ 4.20 สัญญาณของ Passive Buzzer ที่ความถี่ 1000Hz

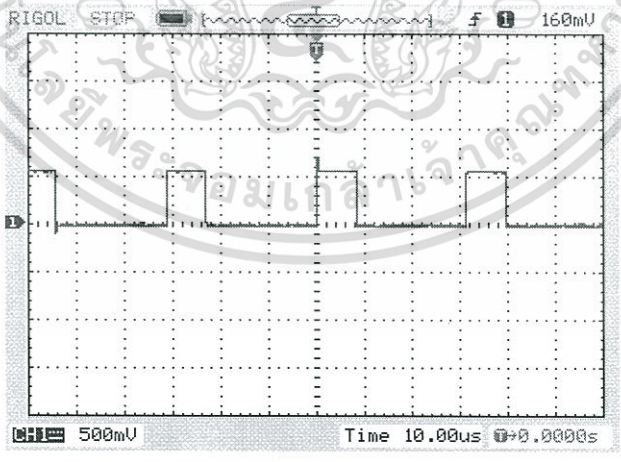
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 สัญญาณของ Passive Buzzer ที่ความถี่ 2000Hz

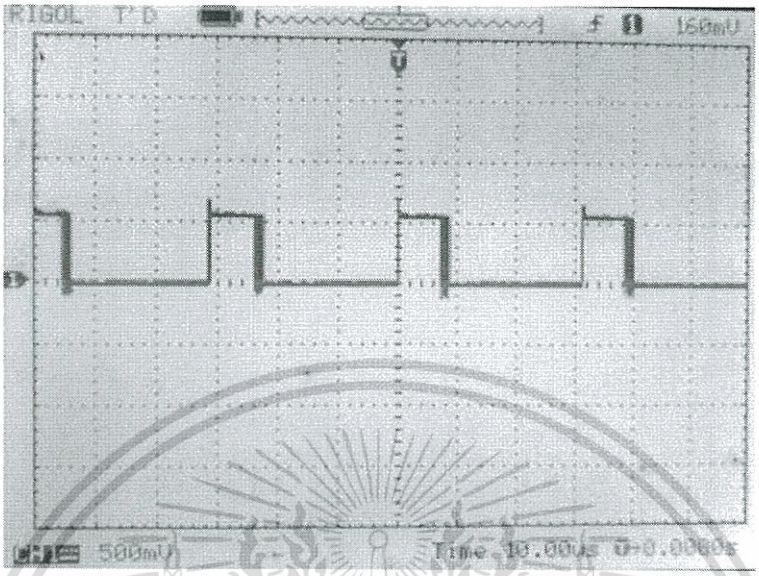
4.4 การทดสอบเสียงเวลาผ่านลำโพง

การทดสอบเสียงผ่านลำโพงสามารถทำได้โดยการวัดค่าระดับแรงดันของสัญญาณเมื่อมีเสียงแจ้งเตือนกับไม่มีเสียงแจ้งเตือน ซึ่งจากการทดลองจะพบว่าเมื่อไม่มีเสียงแจ้งเตือนผ่านลำโพงจะขึ้นกราฟที่มีลักษณะนิ่งแสดงดังรูปที่ 4.22 แต่หากมีเสียงแจ้งเตือนกราฟจะมีลักษณะเคลื่อนไหวแสดงดังรูปที่ 4.23



รูปที่ 4.22 สัญญาณเสียงเมื่อไม่มีเสียงแจ้งเตือนผ่านลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 สัญญาณเสียงเมื่อมีเสียงแจ้งเตือนผ่านลำโพง

4.5 ผลการทดลองโดยรวม

ผลการทดลองโดยรวมจะมีอุปกรณ์ในการต่อใช้งานร่วมกันได้แก่ Ultrasonic Range Finder LV-Maxsonar EZ0, SD Card Module, Real Time Clock, Passive Buzzer, ตัวต้านทาน, สวิตช์, Servo Motor และลำโพง แสดงดังรูปที่ 4.24

รูปที่ 4.24 การเชื่อมต่ออุปกรณ์บนไม้เท้าโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์การออกแบบไม้เท้าอัจฉริยะ ซึ่งใช้งานเซนเซอร์ต่างๆได้แก่ RTC-DS3231, Ultrasonic Range Finder LV-Maxsonar-EZ0 ทำงานร่วมกับ Servo Motor, Passive Buzzer, SD Card Module, Switch และลำโพง ซึ่งทำงานบนบอร์ด Arduino Mega2560 R3 ที่ได้ทำการป้อนโค้ดคำสั่งต่างๆจากโปรแกรม Arduino IDE จากการเก็บข้อมูล พบว่า RTC-DS3231 สามารถตั้งค่าวัน เวลาได้ และสามารถแจ้งเวลาเป็นเสียงให้กับผู้ใช้งานทราบผ่านลำโพงเมื่อผู้ใช้งานทำการกดสวิตช์ Ultrasonic Range Finder LV-Maxsonar-EZ0 สามารถตรวจวัดระยะทางของสิ่งกีดขวางและแจ้งเตือนโดยการสั่นผ่านมอเตอร์สั่น ซึ่งความเร็วในการสั่นของมอเตอร์จะแปรผกผันกับระยะทางที่ตรวจจับได้ และมีการแจ้งเตือนผ่านเสียงโดยหากเจอสิ่งกีดขวางในระดับต่ำ ระดับกลาง ระดับสูง ก็จะมีลักษณะเสียงเตือนที่แตกต่างกันออกไป และหากสิ่งกีดขวางอยู่ในระยะที่ใกล้ เสียงเตือนก็จะดังถี่ขึ้น

ไม้เท้าอัจฉริยะ ถ้าหากมีการพัฒนาและศึกษาเพื่อเพิ่มประสิทธิภาพของเซนเซอร์ในการตรวจวัดที่แม่นยำมากขึ้น อาจจะสามารถนำไปประยุกต์ใช้งานในชีวิตประจำวัน และอำนวยความสะดวกให้กับผู้ใช้งานมากขึ้น

5.2 ข้อเสนอแนะ

1) ควรมีองศาในการวัดที่ละเอียดมากขึ้น
2) ควรพัฒนาระบบให้มีเสถียรภาพมากยิ่งขึ้น
3) ควรออกแบบไม้เท้าให้ง่ายต่อการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Arduino MEGA 2560 R3

https://www.thaieasyelec.com/products/development-boards/arduino/official-boards-made-in-italy/arduino-mega-2560-detail.html

http://www.ett.co.th/prod2011/arduino/manET-MEGA2560.pdf

[2] Servo Motor

http://www.thaieasyelec.com/article-wiki/review-product-article/บทความตัวอย่างการควบคุม-rc-servo-motor-ด้วย-arduino.html.

https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html

[3] Coin vibration motor

https://www.precisionmicrodrives.com/vibration-motors/coin-vibration-motors

[4] SD Card Module

https://www.arduinoall.com/product/557

[5] Ultrasonic Range Finder LV-MaxSonar-EZ0

https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf

https://www.maxbotix.com/Arduino-Ultrasonic-Sensors-085/

https://playground.arduino.cc/Main/MaxSonar

[6] Passive Buzzer

https://github.com/fochica/fochica-wiki/wiki/Passive-buzzer-guide

[7] DS3231 RTC

http://cpre.kmutnb.ac.th/esl/learning/index.php?article=ds3231-i2c-rtc

https://www.arduitronics.com/article/35/real-time-clock-ds3231

[8] Arduino Program

https://www.arduitronics.com/article

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โค้ดคำสั่งควบคุมการทำงานของเซนเซอร์ต่างๆ บนไม้เท้าอัจฉริยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <SD.h>
#include <TMRpcm.h>
#define SD_ChipSelectPin 53
TMRpcm tmrpcm;
#include "Wire.h"
#include "SPI.h"
#include "RTClib.h"
int h,m,s;
RTC_DS3231 RTC;
const int buttonPin = 3;
const int buttonPin2 = 8;
const int ledPin = 2;
int buttonState = 0;
int buttonState2 = 0;
#include <Servo.h>
int pushPin = 13;
int LEDPin = 12;
int pw_pin=7;
int arraysize = 9;
int array[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0};
long inch;
int exact_cm_value,cm;
Servo myservo;
int jib40,jib60, jib80, jib100, jib120 , jib140;
int pos = 0;
int ASD ;
const int motorPin = 4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void setup() {
tmrpcm.speakerPin = 5;
pinMode(53, OUTPUT);
pinMode(ledPin, OUTPUT);
pinMode(motorPin, OUTPUT);
pinMode(buttonPin, INPUT);
Wire.begin();
RTC.begin();     //RTC.adjust(DateTime(__DATE__, __TIME__));
  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    //RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  DateTime now = RTC.now();
pinMode(pw_pin, INPUT);
 myservo.attach(6);
Serial.begin(9600);
}

void sensorRead(){
for(int i = 0; i < arraysize; i++)
{
inch = pulseIn(pw_pin, HIGH);
array[i] = inch/58;
delay(10);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void array_arrangment(int *a,int n){
for (int i = 1; i < n; ++i)
{
int j = a[i];
int k;
for (k = i - 1; (k >= 0) && (j < a[k]); k--)
{
a[k + 1] = a[k];
}
a[k + 1] = j;
}
}

int filter(int *a,int n){
int i = 0;
int count = 0;
int maxCount = 0;
int filter = 0;
int median;
int prevCount = 0;
while(i<(n-1)){
prevCount=count;
count=0;
while(a[i]==a[i+1]){
count++;
i++;
}

if(count>prevCount && count>maxCount){
filter=a[i];
maxCount=count;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
median=0;
}
if(count==0){
i++;
}
if(count==maxCount){//If the dataset has 2 or more modes.
median=1;
}
if(filter==0||median==1){      //Return the median if there is no mode.
filter=a[(n/2)];
}
return filter;
}
}

void loop() {
  motion1();
clocks();
}

void clocks(){
buttonState = digitalRead(buttonPin);
  DateTime now = RTC.now();
   h=now.hour(), DEC;
  m=now.minute(), DEC;
  s=now.second(), DEC;
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
      time1();
      return ;
}

void motion1(){
  buttonState2 = digitalRead(buttonPin2);
  if (buttonState2 == HIGH){
  for (pos = 40; pos <= 140; pos += 20) {  // goes from 40 degrees to 140 degrees
         sensorRead();
     myservo.write(pos);
array_arrangment(array,arraysize);
exact_cm_value= filter(array,arraysize);
cm=exact_cm_value;
Serial.print("The distance = ");
Serial.print(exact_cm_value);
Serial.println(" cm ");
Check();
      delay(1000);
        }
        does();
        motion2();
  }
  else{
  return ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  }

void motion2(){
  for (pos = 140; pos >= 40; pos -=20) {  // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
     delay(10);
  }
}
void Check(){
if(pos == 60 ){
 if(cm>81 && cm<=101){
jib60=1;
}
if(cm>=14 && cm<=81){
jib60=2;
}
else{
  return ;
  }
}

if(pos == 80 ){
 if(cm>300 && cm<=400){
jib80=1;
}
if(cm>200 && cm<=300){
jib80=2;
}
if(cm>=60 && cm<=200){
jib80=3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
else{
  return ;
}
}

if(pos == 100 ){
 if(cm>300 && cm<=400){
jib100=1;
}
if(cm>200 && cm<=300){
jib100=2;
}
if(cm>60 && cm<=200){
jib100=3;
}
else{
  return ;
}
}

if(pos == 120 ){
 if(cm>171 && cm<=230){
jib120=1;
}
if(cm>120 && cm<=171){
jib120=2;
}
else{
  return ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
}

void does(){
 if(jib60==1){
if(jib80==3 && jib100==3 && jib120==1){
for(int i=0 ; i<=1 ; i++){
    tone(12,1000,1000);
delay(1000);
                    }
}
 else if(jib80==1){
 for(int i=0 ; i<=1 ; i++){
    tone(12,200,1000);
delay(1000);
 }
 }
  else if(jib80==3 && jib100==3){
  for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
            }
 }
else if(jib80==3){
    for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
            }}
    else{
        tone(12,200,1000);
delay(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
        }

else if(jib60==2){
  if(jib80==3 && jib100==3 && jib120==2 ){
    for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
}   }
else if(jib80==3 && jib100==3){
  for(int i=0 ; i<=2 ; i++){
     tone(12,1000,1000);
delay(1000);
}   }

else if(jib80==3){
  for(int i=0 ; i<=2 ; i++){
    tone(12,400,1000);
delay(1000);
}}
else if(jib80==2){
    for(int i=0 ; i<=1 ; i++){
     tone(12,1000,1000);
delay(1000);
}}
else{
  for(int i=0 ; i<=2 ; i++){
      tone(12,1000,1000);
delay(1000);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

else if(jib80==1){
   if(jib100==1){
    tone(12,1000,1000);
delay(1000);
}
else{
      tone(12,200,1000);
delay(1000);
}
}

else if(jib80==2){
 if(jib100==2){
  for(int i=0 ; i<=1 ; i++){
    tone(12,1000,1000);
delay(1000);
}
 }
else{
  for(int i=0 ; i<=2 ; i++){
       tone(12,1000,1000);
    delay(1000);
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(jib80==3){
   if(jib100==3){
    for(int i=0 ; i<=2 ; i++){
   tone(12,1000,1000);
delay(1000);
}
   }
      else{
       for(int i=0 ; i<=2 ; i++){
      tone(12,1000,1000);
       delay(1000);
}
}
}

else if(jib100==1){
  if(jib120==1){
    tone(12,2000,1000);
 delay(1000);
}
 else if(jib120==2){
      tone(12,1000,1000);
delay(1000);
}
}

else if(jib100==2){
  if(jib120==1){
    for(int i=0 ; i<=1 ; i++){
        tone(12,1000,1000);
delay(10000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  }
 else if(jib120==2){
  for(int i=0 ; i<=1 ; i++){
     tone(12,1000,1000);
delay(1000);
}
}
}

else if(jib100==3){
  if(jib120==1){
for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
}
  }
 else if(jib120==2){
   for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
}}

 else{
   for(int i=0 ; i<=2 ; i++){
        tone(12,1000,1000);
delay(1000);
   }
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(jib120==1){
for(int i=0 ; i<=1 ; i++){
    tone(12,2000,1000);
  delay(1000);
}
}

else if(jib120==2){
for(int i=0 ; i<=2 ; i++){
    tone(12,1000,1000);
delay(1000);
}}
}

void time1(){
  if (buttonState == HIGH)
  {
    if(h==0){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("24hr.wav");
         delay(2000);
minute1();
     }
    if(h==1){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("1hr.wav");
         delay(1700);
minute1();
     }
    if(h==2){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("2hr.wav");
         delay(2000);
minute1();
     }

    if(h==3){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("3hr.wav");
         delay(2000);
minute1();
     }
    if(h==4){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("4hr.wav");
        delay(2000);
minute1();
    }
    if(h==5){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("5hr.wav");
        delay(2000);
minute1();
    }
    if(h==6){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("6hr.wav");
        delay(2000);
minute1();
    }
    if(h==7){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    tmrpcm.setVolume(4);
  tmrpcm.play("7hr.wav");
        delay(2000);
minute1();
     }
    if(h==8){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("8hr.wav");
        delay(2000);
minute1();
     }
    if(h==9){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("9hr.wav");
        delay(2000);
minute1();
     }
    if(h==10){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  tmrpcm.play("10hr.wav");
        delay(2000);
minute1();
    }
   if(h==11){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("11hr.wav");
        delay(2000);
minute1();
    }
   if(h==12){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("12hr.wav");
        delay(2000);
minute1();
    }
   if(h==13){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
     return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("14hr.wav");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(2000);
minute1();
    }
   if(h==14){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("14hr.wav");
        delay(2000);
minute1();
    }
   if(h==15){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("15hr.wav");
        delay(2000);
minute1();
    }
    if(h==16){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("16hr.wav");
        delay(2000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
minute1();
    }
    if(h==17){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("17hr.wav");
        delay(2000);
minute1();
    }
    if(h==18){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("18hr.wav");
        delay(2000);
minute1();
    }
    if(h==19){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("19hr.wav");
        delay(1700);
minute1();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    if(h==20){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("20hr.wav");
        delay(1700);
minute1();
    }
        if(h==21){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("21hr.wav");
        delay(1700);
minute1();
    }
    if(h==22){
  digitalWrite(ledPin, HIGH);
   if (!SD.begin(SD_ChipSelectPin)) {
       return;
  }
   tmrpcm.setVolume(4);
  tmrpcm.play("22hr.wav");
        delay(1700);
minute1();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(h==23){
  digitalWrite(ledPin, HIGH);
    if (!SD.begin(SD_ChipSelectPin)) {
        return;
  }
    tmrpcm.setVolume(4);
  tmrpcm.play("23hr.wav");
        delay(2200);
minute1();
    }
    }
    else{
  return ;
  }
}

void minute1(){
if(m==0){
        digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
  tmrpcm.play("1mi.wav");
    return;
      }
  if(m==1){
    digitalWrite(ledPin, LOW);
      tmrpcm.setVolume(4);
  tmrpcm.play("1mi.wav");
    return;
        }
   if(m==2){
    digitalWrite(ledPin, LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  tmrpcm.setVolume(4);
 tmrpcm.play("2mi.wav");
  return;
     }
 if(m==3){
   digitalWrite(ledPin, LOW);
        tmrpcm.setVolume(4);
 tmrpcm.play("3mi.wav");
   return;
       }
 if(m==4){
digitalWrite(ledPin, LOW);
   tmrpcm.setVolume(4);
 tmrpcm.play("4mi.wav");
   return;
     }
 if(m==5){
   digitalWrite(ledPin, LOW);
      tmrpcm.setVolume(4);
 tmrpcm.play("5mi.wav");
   return;
       }


       if(m==6){
        digitalWrite(ledPin, LOW);
   tmrpcm.setVolume(4);
 tmrpcm.play("6mi.wav");
  return;
     }
 if(m==7){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("7mi.wav");
return;
       }
       if(m==8){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("8mi.wav");
return;
     }
if(m==9){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("9mi.wav");
  return;
       }
  if(m==10){
    digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("10mi.wav");
  return;   }
if(m==11){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("11mi.wav");
  return;
       }
       if(m==12){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tmrpcm.play("12mi.wav");
  return;
    }
if(m==13){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("13mi.wav");
  return;
      }
      if(m==14){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("14mi.wav");
return;
    }
if(m==15){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("15mi.wav");
  return;   }
      if(m==16){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("16mi.wav");
  return;
    }
if(m==17){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("17mi.wav");
  return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }
      if(m==18){
  tmrpcm.setVolume(4);
tmrpcm.play("18mi.wav");
 return;
    }
if(m==19){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("19mi.wav");
  return;
      }
      if(m==20){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("20mi.wav");
  return;
    }
if(m==21){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("21mi.wav");
  return;
      }
      if(m==22){
       digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
 tmrpcm.play("22mi.wav");
  return;
      }
 if(m==23){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("23mi.wav");
 return;
    }
if(m==24){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("24mi.wav");
  return;
      }
if(m==25){
  digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("25mi.wav");
  return;   }
if(m==26){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("26mi.wav");
  return;
      }
      if(m==27){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("27mi.wav");
 return;
    }
if(m==28){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tmrpcm.play("28mi.wav");
return;
      }
      if(m==29){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("29mi.wav");
return;
    }
if(m==30){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("30mi.wav");
  return;  }
      if(m==31){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("31mi.wav");
delay(3000);
  return;
    }
if(m==32){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("32mi.wav");
 delay(3000);
  return;
      }
      if(m==33){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tmrpcm.play("33mi.wav");
 delay(3000);
  return;
    }
if(m==34){   digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("34mi.wav");
delay(3000);
  return;
      }


      if(m==35){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("35mi.wav");
delay(3000);
return;
    }
if(m==36){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("36mi.wav");
 delay(3000);
 return;
      }
      if(m==37){
    digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("37mi.wav");
 return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
if(m==38){
       digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("38mi.wav");
 return;
       }
       if(m==39){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("39mi.wav");
 return;
    }
if(m==40){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("40mi.wav");
  return;
       }
       if(m==41){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("41mi.wav");
  return;
    }
     if(m==42){
      digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("42mi.wav");
  return;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(m==43){
      digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("43mi.wav");
 return;
    }


if(m==44){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("44mi.wav");
  return;
       }
if(m==45){
  digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("45mi.wav");
  return;
    }
if(m==46){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("46mi.wav");
  return;
       }
       if(m==47){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("47mi.wav");
 return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
if(m==48){
  digitalWrite(ledPin, LOW);
     tmrpcm.setVolume(4);
tmrpcm.play("48mi.wav");
return;  }
       if(m==49){
        digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("49mi.wav");
return;
     }
if(m==50){
     tmrpcm.setVolume(4);
tmrpcm.play("50mi.wav");
  return;
       }
       if(m==51){
  tmrpcm.setVolume(4);
tmrpcm.play("51mi.wav");
  return;
     }
if(m==52){
     tmrpcm.setVolume(4);
 tmrpcm.play("52mi.wav");
  return;
       }
       if(m==53){
  tmrpcm.setVolume(4);
 tmrpcm.play("53mi.wav");
  return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
if(m==54){
    tmrpcm.setVolume(4);
tmrpcm.play("54mi.wav");
  return;
      }
      if(m==55){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
tmrpcm.play("55mi.wav");
return ;
    }
if(m==56){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
tmrpcm.play("56mi.wav");
  return;
      }
  if(m==57){
       digitalWrite(ledPin, LOW);
  tmrpcm.setVolume(4);
  tmrpcm.play("57mi.wav");
    return;
    }

if(m==58){
  digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
  tmrpcm.play("58mi.wav");
   delay(2500);
  return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
if(m==59){
          digitalWrite(ledPin, LOW);
    tmrpcm.setVolume(4);
  tmrpcm.play("59mi.wav");
     delay(2500);
   return;       }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้