ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

โดย

นรจ. จักรพงศ์ ชัยชาญ
นรจ. ธนสิษฐ์ ประทุมสุวรรณ
นรจ. จีระศักดิ์ มูลทองชุน
นรจ. วรพงศ์ อ่วมเอิบ
นรจ. ธนาสิน พุฒแตง

โครงงานนี้เป็นส่วนของการศึกษาตามหลักสูตรนักเรียนจ่าทหารชั้นปีที่ 2
พรรคพิเศษ เหล่าทหารช่างยุทธโยธา อิเล็กทรอนิกส์
ปีการศึกษา 2561

โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการกรมอิเล็กทรอนิกส์ทหารเรือ

หัวข้อโครงงาน ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

ผู้จัดทำ นรจ. จักรพงศ์ ชัยชาญ

นรจ. ธนสิษฐ์ ประทุมสุวรรณ

นรจ. จีระศักดิ์ มูลทองชุน

นรจ. วรพงศ์ อ่วมเอิบ

นรจ. ธนาสิน พุฒแตง

ครูที่ปรึกษา ร.ท.พฤทธิ์ กรณีย์

ว่าที่ ร.ต. ภูวนาท สอนประสาร

พ.จ.อ.ไกรสรณ์ รื่นเหลย

พ.จ.อ.จารึก แจ่มดี

สถานศึกษา โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือ

ปีการศึกษา 2561

บทคัดย่อ

โครงงานสิ่งประดิษฐ์นี้เป็นการศึกษาแบบจำลองถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ ในขณะเวลาที่ขยะเต็ม จะมีเสียงแจ้งเตือน ซึ่งต้องรู้จักหลักการทำงานของอุปกรณ์ การประกอบส่วนต่าง ๆ ของอุปกรณ์และการเขียนโปรแกรมควบคุมการทำงานตามวัตถุประสงค์ มีการวิเคราะห์และ หาข้อมูลในส่วน นั้น โดยควบคุมการทำงานจาก Arduino 2560 สั่งให้ Sensor รับสัญญาณ และส่งกลับไปยัง Arduino 2560 ให้สั่งให้servo ทำงานถังขยะจะเปิดปิดอัตโนมัติ เพื่อพัฒนาและขยายความสามารถให้มากขึ้นจาก แบบจำลองถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ต่อไป

ผลการทดลองพบว่าถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์การควบคุมด้วยArduino2560สามารถควบคุมการเปิด-ปิดถังขยะแบบอัตโนมัติด้วยการติดตัวSensorสามารถรับสัญญาณได้จากการที่ให้ผู้ใช้ไปยืนให้ตรงระยะของตัวSensorด้านหน้าถังขยะในระยะที่กำหนด จากนั้นตัวSensorส่งไปยังArduino2560 สั่งให้Servoทำงานจึงได้เพิ่มฐานรองพื้นถังขยะให้สูงขึ้นเพื่อป้องกันสัตว์ เดินผ่านตัวSensor ทำให้ทราบว่าSensorตรวจจับในระยะที่ผู้ใช้ไปยืนให้ตรงระยะของตัวSensorจึงทำให้ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์และมีSensorตรวจจับขยะอยู่บนฝาถังด้านในอีก 1 ตัวเพื่อวัดระดับปริมาณขยะในถังขยะโดยมีไฟ LED 3 ระดับ สีเขียวคือ ถังขยะอยู่ในระดับที่ยังทิ้งได้ สีเหลืองคือถังขยะอยู่ในระดับที่ใกล้เต็มแต่ยังสามารถทิ้ง และสีแดงถังขยะเต็มไม่สามารถทิ้งได้ทั้งหมดนี้ควบคุมด้วย Arduino2560จะสั่งเสียงเตือนเมื่อถังขยะอยู่ในระดับสีแดง จะส่งผลให้ทราบว่าขยะเต็ม

เอกสารโครงงานมีทั้งหมด 41 หน้า

กิตติกรรมประกาศ

โครงงานสิ่งประดิษฐ์นี้สำเร็จได้ด้วยความกรุณาของผู้อำนวยการโรงเรียนอิเล็กทรอนิกส์กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือและคณะครูที่ปรึกษาซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือต่าง ๆ ที่เป็นประโยชน์ จนกระทั่งโครงงานสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี่

ขอกราบขอบพระคุณ อาจารย์ที่ปรึกษา ที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของ โครงงาน และให้ความรู้ ให้คำแนะนำ ทั้งให้กำลังใจ

ท้ายสุดนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงงานนี้จะเป็นประโยชน์ต่อการศึกษาและเป็นที่น่าสนใจสำหรับผู้ที่สนใจต่อไป

คณะผู้จัดทำ

สารบัญ

เรื่อง	**หน้า**
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
บทที่ 1 บทนำ	1-2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3-13
บทที่ 3 ขั้นตอนการดำเนินการ	14-25
บทที่ 4 ผลการทดลอง	26-28
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	29
ภาคผนวก ก.	30-33
ภาคผนวก ข.	34-38
ภาคผนวก ค.	39
ภาคผนวก ง.	40
บรรณานุกรม	41

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

แต่เดิมขยะมูลฝอยหรือขยะทั่วไป เป็นปัญหาของคนส่วนใหญ่ ซึ่งสังคมในปัจจุบัน ยังไม่มี การจัดการที่ดี รวมถึงภาคครัวเรือนที่มีการทิ้งขยะหรือ สิ่งปฏิกูลทุกวันเช่น เศษอาหาร ถุงพลาสติก เศษกระดาษ อีกทั้งประเทศไทยเป็นเมืองการเกษตร ขยะอินทรีย์ หรือ สารเคมีจากปุ๋ยที่ส่งผลต่อสิ่งแวดล้อมรวมไปถึงจิตใต้สำนึกของคนในยุคปัจจุบันที่ยังไม่คำนึงถึงผลกระทบต่างๆที่ตามมา ปัจจุบันขยะมูลฝอยนั้นนับวันจะเพิ่มมากขึ้นตามจำนวนประชากร ถ้าหากไม่มีการจัดการที่ดี และเหมาะสมกับปัญหาความสกปรกต่างๆ ทำให้เกิดปัญหาต่อสุขภาพอนามัย เช่น ปัญหาเรื่องเชื้อโรค สารเคมี และกลิ่นเหม็นที่แพร่ไปตามสถานที่ต่างๆ

ปัจจุบันมีถังขยะอยู่จำนวนมาก แต่ถังขยะที่มีอยู่ในปัจจุบันส่วนใหญ่นั้นอยู่ในสภาพที่ไม่สมบูรณ์เนื่องจากผ่านการใช้งานเป็นระยะเวลานาน เช่น ไม่มีฝาปิด ฝาถังชำรุด ถังขยะมีรอยแตกราว มีการสะสมของขยะ และยังขาดการดูแลรักษาที่ดีซึ่งเป็นสถานที่สะสมของเชื้อโรคและแบคทีเรีย อาจจะทำให้เกิดการติดเชื้อแก่ผู้ที่ทิ้งขยะโดยการสัมผัสกับถังขยะ ความเป็นไปได้ของโครงงานเป็นไปได้ที่จะบรรลุตามเป้าหมาย เนื่องจากกลุ่มกระผมตระหนักถึงปัญหาเหล่านี้ดังที่กล่าวมาข้างต้น ฉะนั้นเราจึงร่วมกันคิดหาเทคนิคการทำต่อยอดจากถังขยะในรูปแบบเดิม มีการนำมาประยุกต์จากรายวิชาที่เรียนไมโครคอนโทรลเลอร์ พัฒนาคุณภาพชีวิตและเรดาร์โดยใช้เป็นระบบเซนเซอร์ของถังขยะโดย ร.ท.พฤทธิ์ กรณีย์ ว่าที่ ร.ต.ภูวนาท สอนประสาร พ.จ.อ.ไกรสรณ์ รื่นเหลย และ พ.จ.อ. จารึก แจ่มดี ครูที่ปรึกษาที่คอยให้การสนับสนุนและให้คำปรึกษาโดยมีงบประมาณสนับสนุนจากโรงเรียนอิเล็กทรอนิกส์ เมื่อโครงงานสำเร็จจึงส่งผลทำให้มีถังขยะที่มีสภาพพร้อมใช้งานและมีประสิทธิภาพ ทำให้เกิดสภาพแวดล้อมที่ดี สะอาด เป็นระเบียบเรียบร้อย รวมไปถึงการตอบสนองความต้องการของมนุษย์ในยุคปัจจุบัน

กลุ่มของกระผมจึงเล็งเห็นว่าโครงงาน ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์เป็นโครงงานที่เห็นถึงความสำคัญของการพัฒนาเทคโนโลยีประกอบเข้ากับปัญหาขยะในปัจจุบันเพื่อให้มนุษย์เห็นความสำคัญของการทิ้งขยะและปัญหาของสิ่งแวดล้อมสุขอนามัยของคนในยุคปัจจุบัน

1.2 วัตถุประสงค์

1.2.1 เพื่อสร้างถังขยะเปิด-ปิดอัตโนมัติควบคุมด้วยเซ็นเซอร์ตรวจจับความเคลื่อนไหว
1.2.2 เพื่อศึกษาเทคโนโลยีเครือข่ายเซ็นเซอร์และการควบคุมอัตโนมัติ
1.2.3 เพื่อประยุกต์และนำความรู้ที่ได้เรียนมาใช้ให้เกิดประโยชน์
1.2.4 เพื่อตอบสนองความต้องการของกลุ่มบุคคลหรือพนักงานออฟฟิศ
1.2.5 เพื่อป้องกันเชื้อโรคที่เกิดจากการสัมผัสถังขยะ

1.3 สมมุติฐานของการศึกษา

สามารถตอบสนองความสะดวกสบายในการทิ้งขยะ ป้องกันเชื้อโรคที่มาจากการสัมผัสกับถังขยะและสามารถประยุกต์นำความรู้ที่ได้เรียนมาใช้ในการทำโครงงาน

1.4 ขอบเขตของโครงงาน

1.4.1 ศึกษาการทำงานของไมโครคอนโทรเลอร์ เพื่อประยุกต์ใช้ในการควบคุมการทำงานระบบเปิด-ปิดอัตโนมัติของถังขยะ

1.4.2 ศึกษาการออกแบบระบบ การติดตั้งและการทำงานของระบบเซ็นเซอร์อัลตราโซนิค

1.4.3 ศึกษาการตอบสนองของระบบเซ็นเซอร์กับการเปิด-ปิดอัตโนมัติของถึงขยะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ระบบเปิด-ปิด ถังขยะอัตโนมัติด้วยเซ็นเซอร์ตรวจจับการเคลื่อนไหว

1.5.2 เพิ่มความสะดวกสบายต่อการใช้งานในแต่ละวัน

1.5.3 สามารถประยุกต์สิ่งของที่มีให้มีความทันสมัยขึ้น

1.5.4 ป้องกันเชื้อโรคจากถังขยะ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในการศึกษาโครงงานเรื่อง ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ สิ่งแรกที่จำเป็นในการจัดทำโครงงาน ต้องรู้จักหลักการทำงานของอุปกรณ์ การประกอบส่วนต่างๆของอุปกรณ์และการเขียนโปรแกรมควบคุมการทำงานตามวัตถุประสงค์ มีการวิเคราะห์และหาข้อมูลในส่วนนั้น เพื่อเป็นแหล่งข้อมูลในการนำไปใช้ในการทำโครงงานและพัฒนาต่อยอดการศึกษา คณะผู้จัดทำได้ทำการรวบรวมแนวคิดหลักการและทฤษฎีต่าง จากเอกสารที่เกี่ยวข้องดังต่อไปนี้

2.1 หลักการ Arduino MEGA 2560

2.2 หลักการ เซ็นเซอร์อัลตราโซนิค วัดระยะทาง

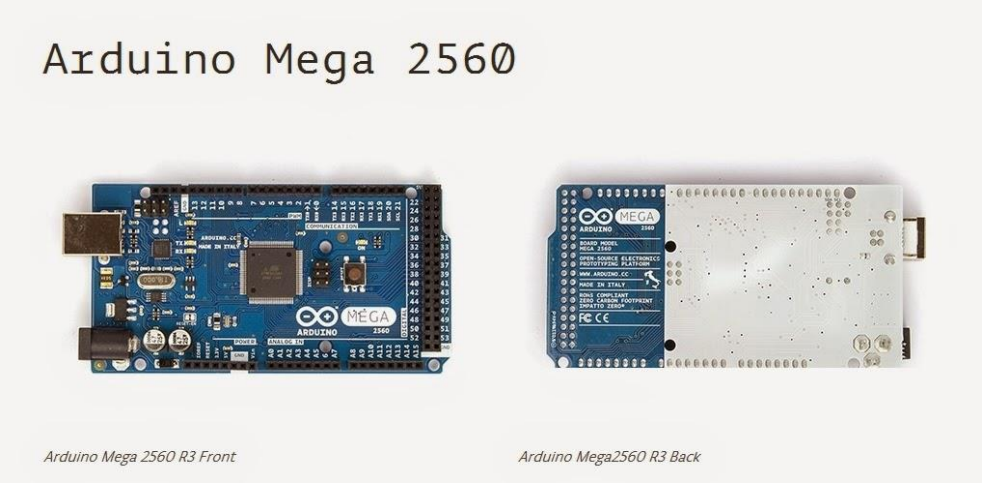
2.3 หลักการ LED

2.4 หลักการ Buzzer

2.5 หลักการ Servo Motor

2.6 หลักการ บอร์ดบันทึกเสียง

2.1 หลักการ Arduino MEGA 2560



รูปที่ 2.1.1 Arduino MEGA 2560

Arduino Mega 2560 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ทำงานบนพื้นฐานของ ATmega 2560 ซึ่งประกอบด้วย

- 54 digital input/output pins (15 pin สามารถใช้เป็น PWM (Pulse Width Modulation) output ได้)
- 16 analog inputs
- 4 UARTs

- 16 MHz crystal oscillator (ใช้สำหรับกรองความถี่ให้กับบอร์ดไมโครคอนโทรลเลอร์)
- USB connection
- ช่องเสียบแหล่งจ่าย
- ICSP header: In-Circuit Serial Programming (ส่วนที่เป็น AVR ขนาดเล็กสำหรับการโปรแกรม Arduino ซึ่งประกอบด้วย MOSI, MISO, SCK, RESET, VCC,GND)
- ปุ่มกด reset

โดยบอร์ด Arduino Leonardo นี้มีทุกสิ่งที่ไมโครคอนโทรลเลอร์จำเป็นต้องใช้ อย่างการต่อไฟเลี้ยงสามารถทำได้ทั้งการเชื่อมต่อเข้ากับ USB cable หรือ จ่ายไฟด้วย AC-DC adapter หรือ การใช้แบตเตอรี่ ซึ่ง Mega เป็นบอร์ดที่เข้ากันได้กับ shield ที่ออกแบบมาเพื่อ Arduino Duemilanove หรือ Diecimila

Mega 2560 นี้มีความแตกต่างจากบอร์ดก่อนหน้าตรงที่ไม่ใช้ FTDI USB-to-serial driver chip แต่จะมี ATmega16U2 เข้ามาเป็นโปรแกรมแปลง USB-to-serial

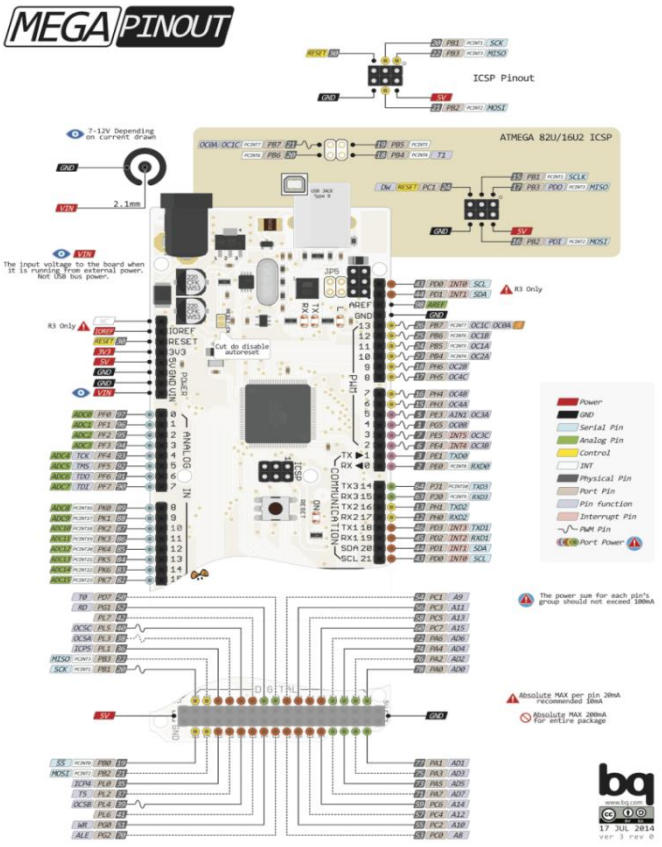
Arduino Mega2560 Revision 2 มี ATmega8U2 ทำให้อัพเดท firmware ผ่าน USB protocal ที่เรียกว่า DFU (Device Firmware Update) ได้ง่ายขึ้น

2.1.1 Arduino Mega Revision 3 มี featureใหม่ๆเพิ่มขึ้นมาดังนี้

2.1.1.1 1.0 pinout: เพิ่ม SDA และ SCL (อยู่ใกล้กับ AREF pin) และอีกสอง pins ใหม่คือ IOREF เป็น pin ที่ใช้ในการเชื่อมต่อกับ shields เพื่อแปลงเป็นแรงดันที่ได้จากบอร์ด ส่วนอีก 1 pin ที่เหลือมีไว้สำหรับใช้ร่วมกับ AVR ในอนาคต

2.1.1.2 วงจร Reset ที่ดีขึ้น

2.1.1.3 ใช้ ATmega 16U2 แทน 8U2



รูปที่ 2.1.2 อธิบายการใช้งานขาของ Arduino MEGA 2560 Power

Arduino Mega สามารถเชื่อมรับพลังงานโดยการเชื่อมต่อ micro USB connector หรือจาก power supply จากภายนอกได้ โดยแหล่งพลังงานจะถูกเลือกโดยอัตโนมัติ

แหล่งจ่ายจากภายนอกสามารถมาได้จาก AC-to-DC adapter หรือจากแบตเตอรี่ โดยต่อเข้ากับ 2.1mm center-positive plug ไปยังช่องเสียบแหล่งจ่าย และการต่อเข้ากับแบตเตอรี่สามารถ ทำได้โดยการต่อเข้ากับ GND และ Vin pin header ของ power connecter

บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 volts DC ถ้า แหล่งจ่ายมีค่าต่ำกว่า 7 Vdc อาจส่งผลให้ 5 Vdc pin มีแรงดันที่ต่ำกว่า 5Vdc และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่าสูงกว่า 12 Vdc อาจส่งผลให้บอร์ด Overheat และอาจทำให้บอร์ดเสียหายได้ ดังนั้นช่วงแรงดันที่เหมาะสมกับบอร์ดคือ 7 Vdc ถึง 12 Vdc

- VIN เป็น input voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก
- 5Vdc เป็น output pin ที่ควบคุม 5 Vdc จากบอร์ด
- 3V3 เป็น 3.3 volt supply ที่สร้างขึ้นจาก regulator บนบอร์ด และให้กระแสได้สูงสุด 50 mA
- GND เป็น ground pin
- IOREF เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

Memory

ATmega2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ bootloader) นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM

Input and Output

ในแต่ละ digital pins ทั้ง 54 pins บนบอร์ด Arduino Uno สามารถเป็นได้ทั้ง input และ output โดยจะทำงานที่แรงดัน 5 V และให้กระแสสูงสุด 40 mA

ฟังก์ชันอื่นๆเพิ่มเติม

Serial: 0 (Rx) และ 1(Tx); Serial 1: 19(Rx) และ 18 (Tx); Serial 2: 17 (Rx) และ 16(Tx); Serial 3:15 (Rx) และ 14 (Tx) ใช้สำหรับรับ (Rx) และส่ง(Tx) TTL serial data โดย pin 0 และ 1 จะถูกเชื่อมต่อไปยัง corresponding pins ของ ATmega16U2 USB-to-TTL serial chip

External Interrupts: 2 (interrupt 0) , 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2). pins เหล่านี้สามารถที่จะกำหนดค่าที่เรียก interrupt ในค่าต่ำๆ, ขอบขาขึ้นและลง หรือเปลี่ยนแปลงค่า

PWM: 2 ถึง 13 และ 44 ถึง 46 ให้ output PWM output 8-bits

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) ใช้สำหรับรองรับการสื่อสารแบบ SPI โดยที่ ไม่เกี่ยวข้องกันกับ ICSP header ซึ่งจะมีลักษณะคล้ายกับ Uno, Duemilanove และ Diecimila

LED 13 : เป็น build-in LED ที่เชื่อมต่อกับ digital pin 13 เมื่อ pin มีค่าเป็น HIGH LED จะติด , แต่เมื่อ pin เป็น LOW LED จะดับ

TWI : 20 (SDA) and 21 (SCL). รองรับการเชื่อมต่อแบบ TWI(I2C)

บอร์ด Mega2560 มี 16 analog inputs แต่ละ pins ให้ความละเอียด 10 bits

AREF. แรงดันอ้างอิง สำหรับ analog input

Reset ใช้ในการ reset ไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม reset ไว้บน sheild เพื่อป้องกันปุ่มที่อยู่บนบอร์ดCommunication

Arduino Uno สามารถสื่อสารกับคอมพิวเตอร์ Arduino ตัวอื่นๆ หรือ microcontroller ได้ โดยที่ไมโครคอนโทรลเลอร์บนบอร์ด คือ ATmega32U4 จะให้การสื่อสารแบบอนุกรม UART TTL (5 V) ซึ่งมีอยู่ใน pins 0 (Rx) และ 1 (Tx) นอกจากนี้ 32U4 สามารถใช้การสื่อสารแบบอนุกรมผ่าน USB และจะปรากฏเป็น COM port เสมือนไปยัง Software แต่อย่างไรก็ตามต้องใช้ ไฟล์ inf บนระบบปฏิบัติการ Windows แต่ OSX และ Linux สามารถ recognize ได้โดยอัตโนมัติ

Programming

Arduino Uno สามารถรองรับการโปรแกรมด้วย Arduino Software โดยสามารถใช้ได้ทั้งในระบบปฏิบัติการ Windows, Mac OS X และ Linux

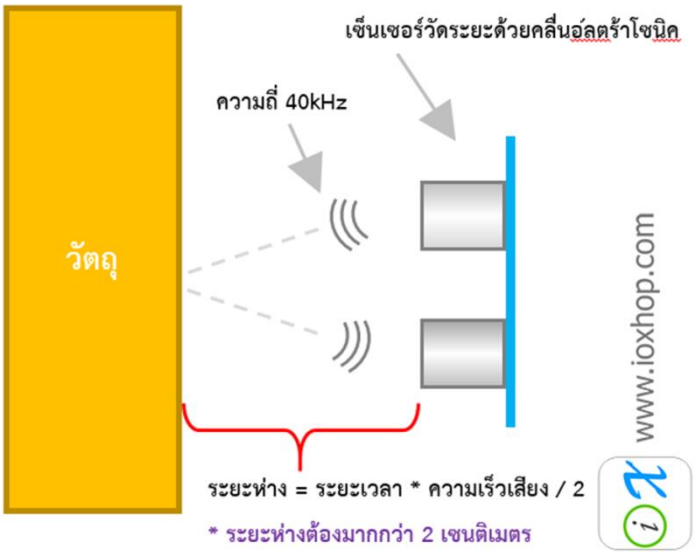
2.2 หลักการเซ็นเซอร์อัลตราโซนิค วัดระยะทาง

คลื่นอัลตราโซนิค เป็นคลื่นความถี่เหนือความถี่สัญญาณเสียง ปกติแล้วมนุษย์จะไม่สามารถได้ยินเสียง เพราะมนุษย์สามารถได้ยินเสียงได้ที่ความถี่ 20 Hz ถึง 20 kHz

ความถี่อัลตราโซนิคนั้น ที่นิยมใช้งานในเซ็นเซอร์วัดระยะรุ่นต่างๆ จะมีความถี่ที่ประมาณ 40kHz ข้อดีของการใช้ความถี่นี้ คือมีลักษณะของความยาวคลื่นที่สั้น ส่งผลให้คลื่นไม่แตกจายออกเป็นวงกว้าง และสามารถยิงคลื่นตรงไปชนวัตถุใดๆ ก็ได้ และนอกจากนี้ความถี่ 40kHz ยังเป็นความถี่ที่มีระยะเดินทางเพียงพอกับการใช้งาน หากใช้ความถี่สูงขึ้น จะทำให้คลื่นเดินทางได้ในระยะทางที่ลดลง ทำให้เมื่อนำมาใช้งานจริงจะวัดระยะได้ในระยะที่สั้น

หลักการวัดระยะด้วยคลื่นอัลตราโซนิค

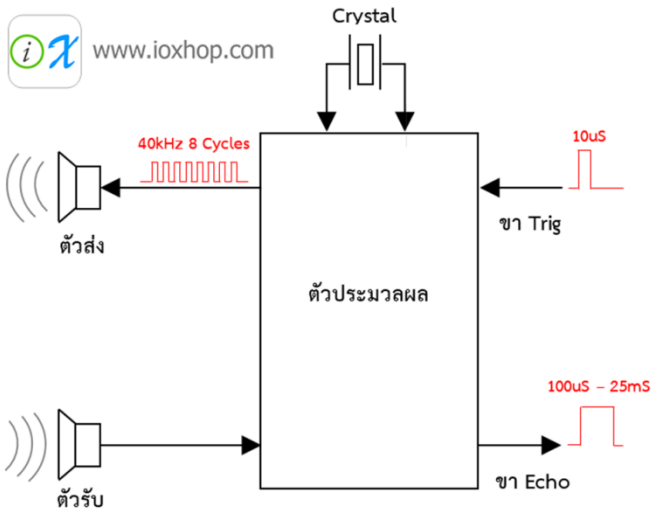
หลักการของการวัดระยะด้วยคลื่นอัลตราโซนิค คือ การส่งคลื่นอัลตราโซนิคออกไปจากตัวส่ง (Transmitter) เมื่อคลื่นที่ส่งออกไปวิ่งไปชนกับวัตถุ คลื่นจะมีการสะท้อนกลับมาแล้ววิ่งกลับไปชน ตัวรับ (Receiver) ด้วยการเริ่มนับเวลาที่ส่งคลื่นออกไปจนถึงได้รับคลื่นกลับมา ทำให้สามารถหาระยะห่างระหว่างวัตถุกับเซ็นเซอร์



รูปที่ 2.2.1 หลักการวัดระยะด้วยอัลตราโซนิค

หลักการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิค

ในโมดูลเซ็นเซอร์อัลตราโซนิค จะมีวงจรที่แตกต่างกัน แต่มีหลักการทำงานงานที่สำคัญที่เหมือนกัน



รูปที่ 2.2.2 ไดอะแกรมการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิค

เมื่อมีการส่งสัญญาณเข้าไปที่ Trigger วงจรภายในจะเริ่มสร้างความถี่ 40 kHz จำนวน 8 ลูกคลื่นออกไป โดยใช้ความถี่จากคริสตอลเป็นตัวอ้างอิง แล้วตัวส่งที่เปรียบเสมือนลำโพงจะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นที่ส่งออกไปวิ่งกลับมาที่ตัวรับ ที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้า
จะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo

หลักการใช้งานจะขึ้นอยู่กับบัสที่ใช้สื่อสาร สามารถแบ่งได้ดังนี้

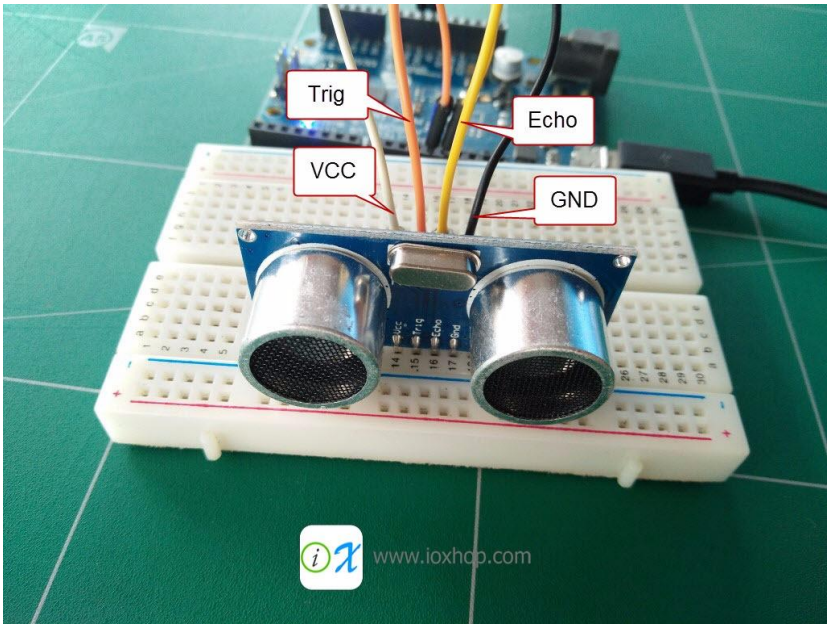
การสื่อสารแบบทริกสัญญาณ เริ่มต้นจะต้องให้สัญญาณขา Trigger มีสถานะทางลอจิกเป็น LOW จากนั้นจึงเริ่มทริกเกอร์สัญญาณ โดยให้ขา Trigger มีสถานะเป็น HIGH ค้างไว้อย่างน้อย 10uS แล้วจึงปรับสถานะเป็น LOW จากนั้น ที่ขา Echo ให้เตรียมรับสัญญาณทริกเกอร์ HIGH กลับมา เมื่อมีการส่งสัญญาณ HIGH กลับมา ให้เริ่มนับเวลาที่สัญญาณเป็น HIGH และเมื่อสัญญาณขา Echo กลับเป็น LOW ให้สิ้นสุด การนับเวลา แล้วจึงนำค่าเวลาที่นับได้ ไปคำนวณอีกที ซึ่งในการคำนวณนั้น จะขึ้นอยู่กับรุ่น ในบางรุ่นสามารถใช้ค่าอัตราเร็วเสียงมาคำนวณได้เลย แต่ในบางรุ่นต้องใช้สูตรคำนวณเฉพาะ

การสื่อสารแบบ UART นอกจากการสื่อสารผ่านการทริกสัญญาณแล้ว ยังสามารถใช้การสื่อสารแบบ UART ได้อีกด้วย ทำให้สามารถใช้งานง่ายมากขึ้น เพราะเมื่อส่งค่าบางอย่างไปให้กับตัวโมดูล
ก็จะตอบค่าที่วัดได้ออกมาให้เลย

การสื่อสารแบบ UART แบ่งออกเป็น 2 ส่วน คือส่วนส่งข้อมูลไป และส่วนรับข้อมูลกลับมา

การใช้งานเซ็นเซอร์วัดระยะด้วยคลื่นอัลตร้าโซนิค HC-SR04

รุ่น HC-SR04 เป็นรุ่นที่ถูกนิยมใช้งานมากที่สุด การสื่อสารกับไมโครคอนโทรลเลอร์จะใช้วิธีทริกสัญญาณ ขาใช้งานจะมี 4 ขา คือขา VCC Trig Echo และ GND ในรุ่นนี้รองรับแรงดันไฟเลี้ยงที่ 5V กรณีที่แรงดันไฟเลี้ยงน้อยกว่า 5V โมดูลจะไม่สามารถทำงานได้ตามปกติ

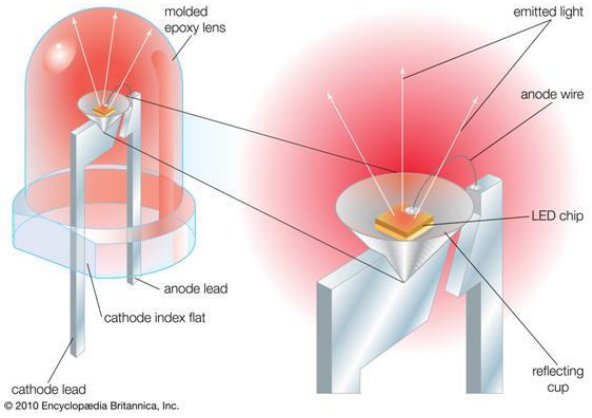


รูปที่ 2.2.3 การเชื่อมต่อสายกับโมดูล HC-SR04

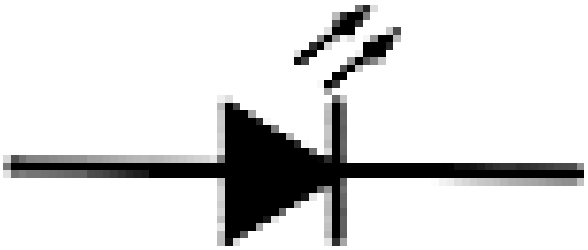
2.3 หลักการ LED

LED ย่อมาจากคำว่า Light Emitting Diode โดยการทำงานนั้นจะคล้ายๆกับการทำงานของ ไดโอด บางคนอาจจะเรียก LED ว่า ไดโอดเปล่งแสง ซึ่งประกอบด้วยสารกึ่งตัวนำชนิด P และ N ประกบกันมีผิวข้างหนึ่งเรียบเป็นมันคล้ายกระจก เมื่อ LED ถูกไบแอสตรง จะทำให้อิเล็คตรอนที่สารกึ่งตัวนำชนิด N มีพลังงานสูงขึ้นจนสามารถวิ่งข้ามรอยต่อไปรวมกับโฮลใน P การที่อิเล็กตรอนเคลื่อนที่ผ่านรอยต่อ PN ทำให้เกิดกระแสไหล เป็นผลให้ระดับพลังงานของอิเล็กตรอนเปลี่ยนไปและคายพลังงานออกมาในรูปคลื่นแสง

สีของแสงที่เกิดจากรอยต่อจะขึ้นอยู่กับชนิดของวัสดุที่นำมาใช้ในการสร้าง LED ทั้งชนิดที่เป็นของเหลวและก๊าซ เช่น ใช้แกลเลียมฟอสไฟด์ (GALLIUM PHOSPHIDE,GaP) ทำให้เกิดแสงสีแดง ใช้แกลเลียมอาซีไนด์ฟอสไฟด์ (GALLIUM ARSENIDE PHOSPHIDE,GaAsP) เกิดแสงสีเหลืองและเขียวการควบคุมปริมาณแสงสว่างจะควบคุมกระแสที่ไหลผ่านหลอด LED หากกระแสที่ไหลสูงมากไปจะทำให้หลอดมีความสว่างมาก แต่หากป้อนกระแสสูงมาไปจะทำให้บริเวณรอยต่อของสารกึ่งตัวนำเกิดความร้อนปริมาณมากจนทำให้โครงสร้างหลอดเสียหายไม่สามารถใช้งานได้

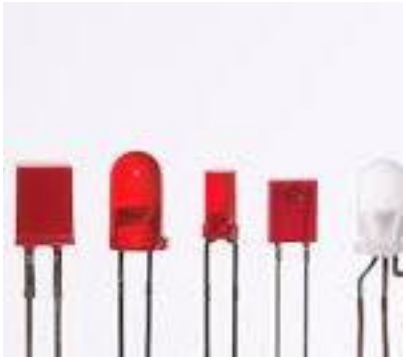


รูปที่ 2.3.1 ลักษณะของ LED



รูปที่ 2.3.2 สัญลักษณ์ของ LED

LED จะมีรูปร่างที่แตกต่างกันออกไปตามการนำไปใช้งาน



รูปที่ 2.3.3 รูปร่างและลักษณะของ LED แต่ละชนิด

2.4 หลักการ Buzzer

การเชื่อมต่อฮาร์ดแวร์ (Hardware Installation)

การเชื่อมต่อ จากตัวอย่างจะใช้การเชื่อมต่อโดยผ่าน i Module - Arduino Connector Base Shield หรือสามารถใช้สาย 4 Wire Cable(ELNK039) ในการเชื่อมต่อได้เช่นเดียวกัน โดยในการเชื่อมต่อ ควรเชื่อมต่อไปยังขาสัญญาณ Output ที่เป็น PWM เนื่องจากจะต้องสั่งงานโมดูลนี้ด้วยสัญญาณ PWM (หากใช้ สัญญาณที่ไม่มีความถี่ หรือความถี่ต่ำเกินไป ตัว Buzzer จะไม่ตอบสนอง)

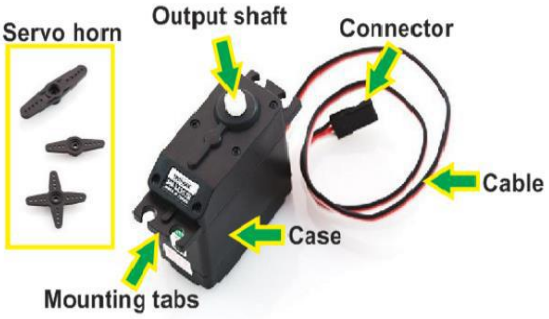
รูปที่ 2.4.1 รูปร่างของ Buzzer

2.5 หลักการ Servo Motor

Servo เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ มาจากภาษาละตินคำว่า Sevus หมายถึง “ทาส” (Slave) ในเชิงความหมายของ Servo Motor ก็คือ Motor ที่เราสามารถสั่งงานหรือตั้งค่า แล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบ
ป้อนกลับ (Feedback Control) ในหน่วยนี้จะกล่าวถึง RC Servo Motor ซึ่งนิยมนามาใช้ในเครื่องเล่นที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

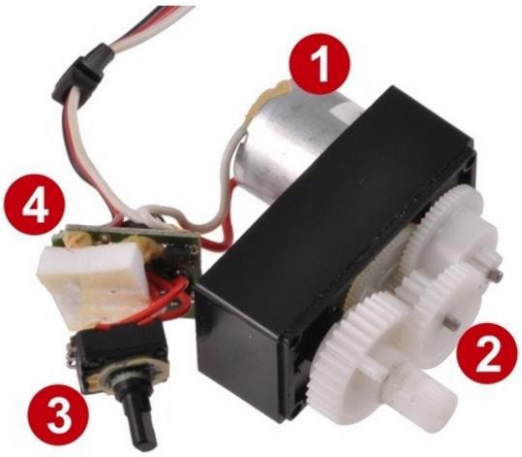
รูปที่ 2.5.1 ลักษณะโดยทั่วไปของเซอร์โวมอเตอร์

Feedback Control คือระบบควบคุมที่มีการนำค่าเอาต์พุตของระบบนำมาเปรียบเทียบ กับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต ส่วนประกอบภายนอก RC Servo Motor



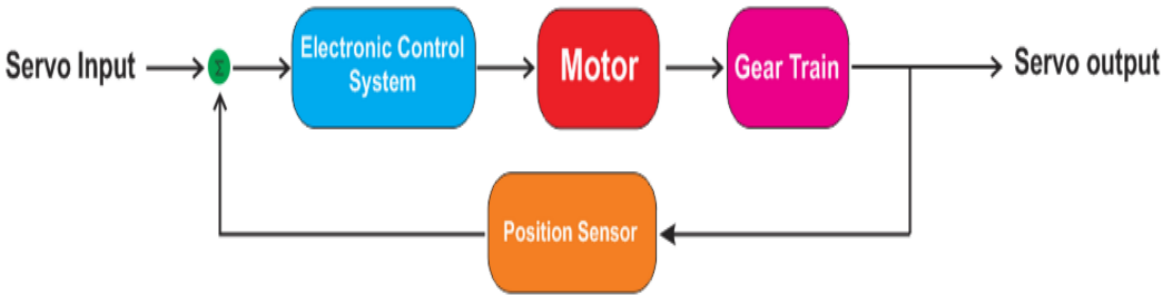
รูปที่ 2.5.2 ส่วนประกอบของเซอร์โวมอเตอร์

- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เพลาส่งกำลัง Servo Horns
- ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไกล Cable
- สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
- สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
- สายสีดำ หรือ น้ำตาล คือ กราวด์
- สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ



รูปที่ 2.5.3 ส่วนประกอบภายในของ Servo Motor

หมายเลข 1 คือ ดีซีมอเตอร์
หมายเลข 2 คือ ชุดเฟืองเกียร์ทดแรงมอเตอร์
หมายเลข 3 คือ ชุดควบคุมและปรับแต่งให้ค่าเอาต์พุตมีค่าเท่ากับ หรือใกล้เคียงกับค่าอินพุต
หมายเลข 4 คือ ส่วนที่ควบคุมและประมวลผล



รูปที่ 2.5.4 บล็อกไดอะแกรมของ Servo Motor

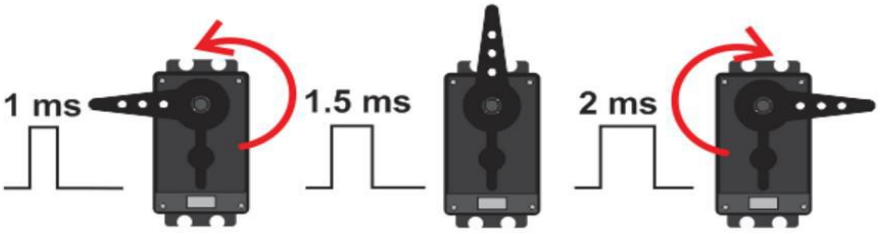
หลักการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายในServo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุม ที่ Motor กำลังหมุนเป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ สัญญาณ RC ในรูปแบบ PWM

ตัว RC Servo Motor ออกแบบมาใช้สำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่นด้วยสัญญาณวิทยุต่างๆ เช่นเครื่องบินบังคับ รถบังบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ที่ภาครับจะแปลงความถี่วิทยุออกมาในรูปแบบสัญญาณ PWM (Pulse Width Modulation)

มุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากความกว้างของพัลส์ที่ใช้ใน RC Servo Motor จะอยู่ในช่วง 1-2 ms หรือ 0.5-2.5 ms

ตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางด้ายซ้ายจนสุดในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี

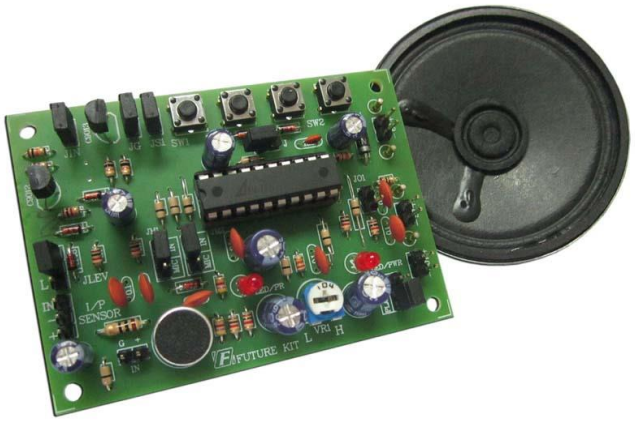


รูปที่ 2.5.5 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลส์

2.6 หลักการบอร์ดบันทึกเสียง

หัวใจของการทำงานของวงจรนี้อยู่ที่ไอซี ISD1720 ซึ่งเป็นไอซีที่สร้างขึ้น เพื่อบันทึกเสียงโดยเฉพาะการทำงานแบ่งออกได้เป็น 2 ส่วนคือ ส่วนบันทึกเสียงและเล่นกลับ การทำงานของส่วนบันทึกเสียง จะเริ่มจากเมื่อเราทำการกดสวิตซ์ SW1 ที่ไมโครโฟนก็จะทำการกับเสียงเข้ามา ยังตัว IC1 เพื่อทำการบันทึกสียง ในขณะ การทำการบันทึกเสียง ถ้าเราทำการปล่อยสวิตซ์ SW1 หรือ ไอซีทำการบันทึกเสียงจนเต็ม ตัวไอซีจะทำการหยุดบันทึกเสียงทันที พร้อมกับ LED1 ดับตามไปด้วย ในกรณี ที่ยังบันทึกเสียงไม่เต็ม เราก็สามารถบันทึก เสียงต่อไปได้ การทำงานของส่วนเล่นกลับ จะเมื่อจากเราทำการเล่นกดสวิตซ์ SW2 PLAY ตัวไอซีก็จะทำการเล่นกลับทันที โดยมี LED 1 กระพริบในขณะเล่นกลับด้วย พร้อมกับส่งเสียงที่เราบันทึกเอาไว้ออกมาทางลำโพง ในกรณีที่เราบันทึกเสียงเอาไว้หลายข้อความ เช่น บันทึกเอาไว้ 2 ข้อความ ตัวไอซีก็จะเล่นแต่ ข้อความที่ 1 ถ้าเราต้องการให้ตัวไอซีเล่นข้อความที่ 2 เราจะต้องทำการกดสวิตซ์ SW3 FWD เพื่อเป็นการเลื่อนการเล่นข้อความที่ 2 แต่ถ้าเรากดสวิตซ์ที่ 3 อีกครั้ง ตัวไอซีจะกลับไปเล่นข้อความที่ 1 เป็นต้น

สำหรับ VR1 จะเป็นตัวปรับระยะในการบันทึก ในส่วนสวิตซ์ SW4 ERASE จะใช้สำหรับลบข้อความที่เราบันทึกเอาไว้ โดยถ้ากดครั้งเดียว ก็จะลบข้อความข้อความตำแหน่งที่ข้อความเล่นอยู่ แต่ถ้ากดค้าง ก็จะเป็นการลบข้อความทั้งหมด



สรุป

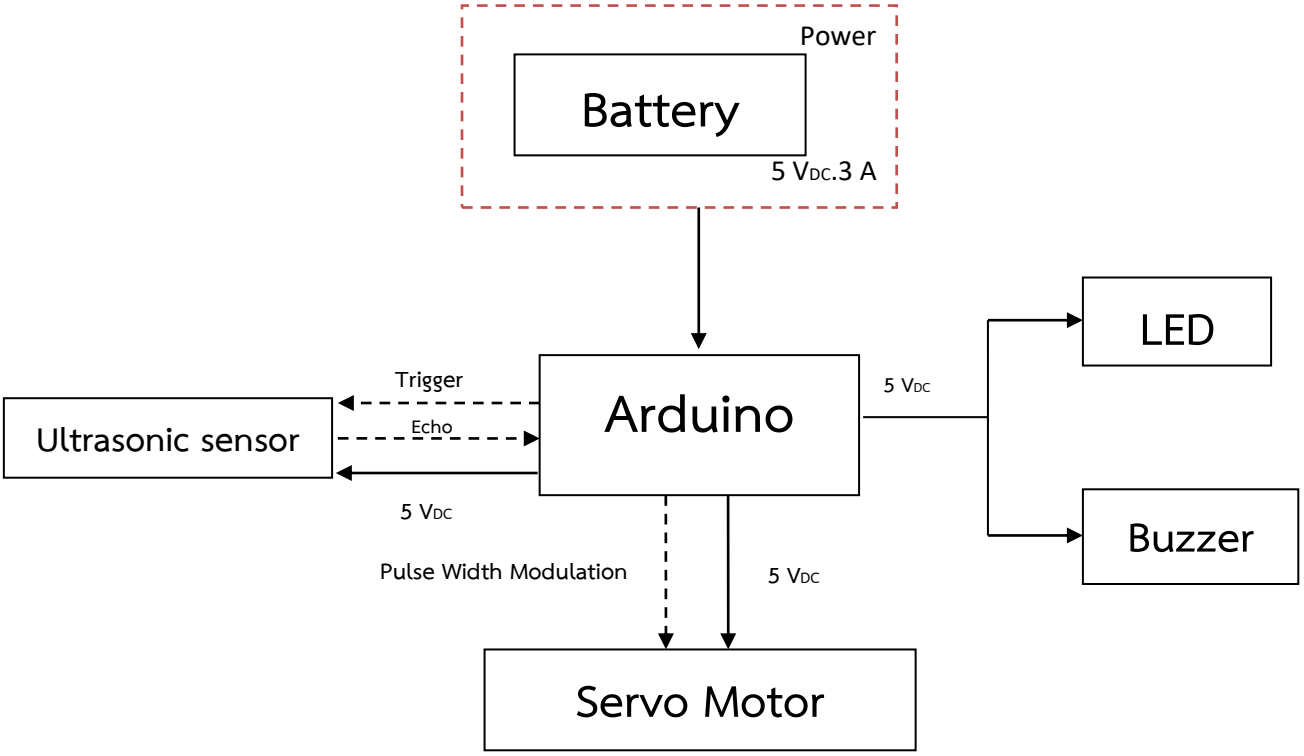
บทนี้นำเสนอหลักการที่เกี่ยวข้องใช้หลักการของการทำงานของบอร์ด Arduino โดยที่บอร์ด Arduino เป็นตัวควบคุมการทำงานของระบบต่างๆ กล่าวคือ คำสั่งที่เขียนในบอร์ด Arduino จะไปทำให้เซนเซอร์วัดระยะทำงานโดยหลักการทำงานของเซนเซอร์จะส่งสัญญาณเข้าไปที่ Trig ภายในวงจรจะเริ่มสร้างความถี่ 40 kHz ออกไป แล้วตัวส่งที่เปรียบเสมือนลำโพงจะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นที่ส่งสัญญาณออกไปวิ่งกลับมาที่ตัวรับ ที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้าจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo สัญญาณที่ออกจากขา Echo จะส่งค่าไปให้บอร์ด Arduino แล้วบอร์ด Arduino จะไปควบคุมการทำงานของ Servo Motor โดยสัญญาณที่สั่งให้ Servo ทำงานเป็นสัญญาณ PWM (Pulse Width Modulation) ภายใน Servo จะมีตัวอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น สัญญาณที่ออกจากค่า Echo จะส่งค่าไปยังบอร์ด Arduino แล้วบอร์ดจะไปควบคุมการทำงานของ LED , Buzzer ให้ทำงานพร้อมกับการที่ควบคุม Servo Motor โดยสัญญาณที่ส่งให้ Servo เป็นสัญญาณ

บทที่ 3

ขั้นตอนการดำเนินการ

ในบทนี้ได้อธิบายวิธีการออกแบบแต่ละส่วนที่ใช้ในโครงงานนี้ประกอบไปด้วย บล็อกไดอะแกรม วงจรรวม ชิ้นส่วนประกอบ การทำงานของเซนเซอร์ วัดระยะทาง การทำงานของ Servo Motor การทำงานของรีเลย์ การรับค่าสัญญาณจาก Sensor สำหรับการสั่งงานให้ Servo Motor ทำงาน การทำงานทั้งหมดควบคุมด้วย Arduino Mega 2560

3.1 บล็อกไดอะแกรมการควบคุมถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์



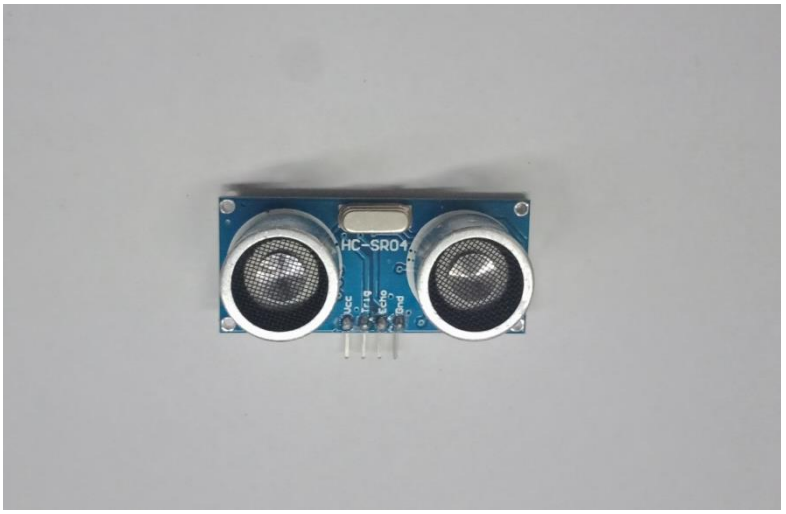
ภาพที่ 3.1 บล็อกไดอะแกรมการควบคุมถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

จากภาพที่ 3.1 เป็นบล็อกไดอะแกรมการควบคุมการทำงานของถังขยะเปิด-ปิดอัตโนมัติ ที่มีส่วนประกอบด้วย ส่วน INPUT, ส่วน PROCESSOR, ส่วน OUTPUT ควบคุมการทำงานของระบบ (Arduino Mega 2560) ตัวจับระยะทางด้วย Sensor ตัวเปิดปิดถังด้วย Servo Motor ควบคุมการทำงานด้วย Relay

สามารถอธิบายระบบโดยรวมได้ดังนี้ เมื่อตัววัดระยะจับระยะได้ Sensor จะส่งสัญญาณพัลส์ ซึ่งจะมีสัญญาณพัลส์แตกต่างกัน โดยมีการส่งสัญญาณ Trigger ออกไป จากนั้นเมื่อคลื่นที่ส่งออกไป วิ่งกลับมาที่ตัวรับสัญญาณที่ได้รับมาจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo ส่งให้กับตัวควบคุม (Arduino) ตัวควบคุมจะทำการถอดรหัสสัญญาณพัลส์ แล้วสัญญาณที่อ่านได้มาเปรียบเทียบ ให้ตรงกับเงื่อนไขของโปรแกรมที่ได้เขียนเอาไว้ ถ้าตรงกับเงื่อนไขจะส่งคำสั่งนั้นไปทำให้ Servo ทำงาน

3.2 ชิ้นส่วนประกอบของถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

3.2.1 Arduino Board

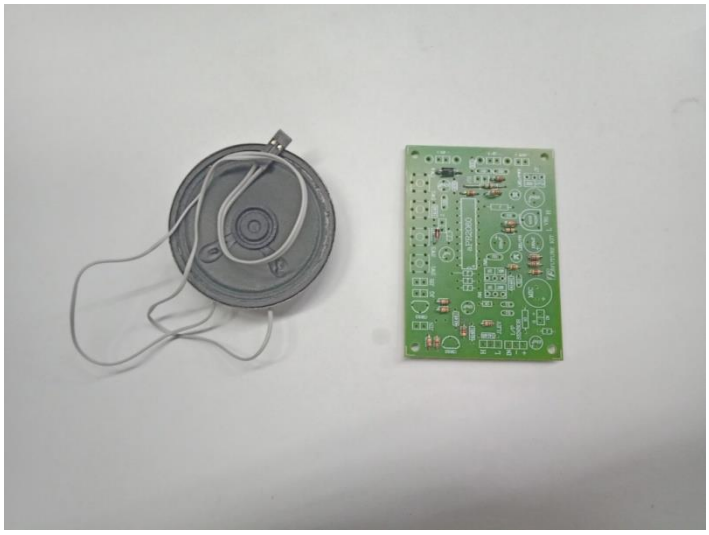


3.2.2 เซ็นเซอร์อัลตราโซนิค วัดระยะทาง

3.2.3 LED

3.2.4 Buzzer

3.2.5 Servo motor



3.2.6 บอร์ดบันทึกเสียง

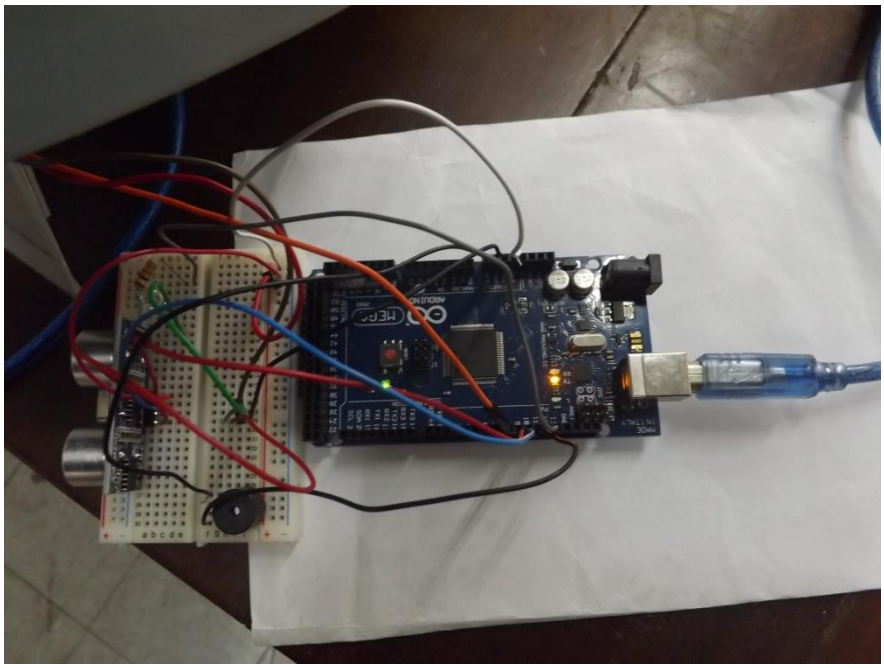
3.2.7 ถังขยะขนาดถัง 45 ลิตร

3.2.8 น็อต

3.2.9 กล่องเอนกประสงค์ใส่วงจร

3.3 การควบคุมการทำงานของระบบ

การควบคุมการทำงานของระบบจะใช้ Arduino เป็นตัวควบคุมการทำงานทั้งหมด ส่วนที่ Arduino ไปควบคุมได้แก่



ภาพที่ 3.3 ระบบการทำงาน

3.3.1 การควบคุมเซนเซอร์ วัดระยะทาง

ในการควบคุมเซนเซอร์ วัดระยะทาง ให้ Arduino สั่งให้ทางานเมื่อมีวัตถุมาใกล้ในระยะที่กำหนด โดย

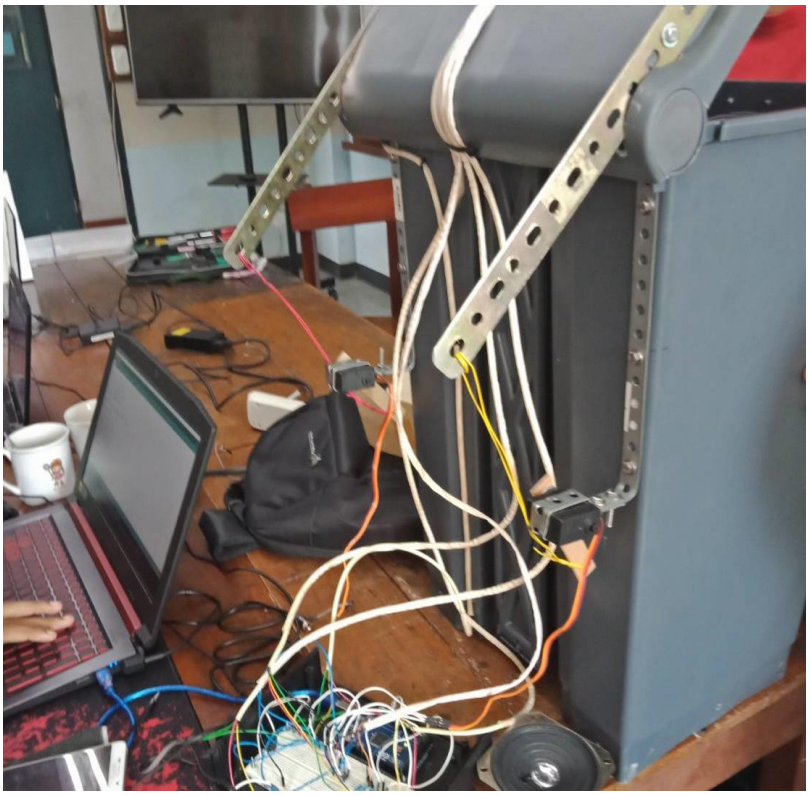
เซ็นเซอร์จะส่งสัญญาณออกไปในระยะที่กำหนด แล้วสะท้อนสัญญาณที่ส่งไปกลับมาประมวลผล

3.3.1.1 การติดตั้งเซนเซอร์ วัดระยะทางในตัวถังขยะ

3.3.2 การควบคุมการทำงานของ Servo motor

ในการควบคุม Servo ให้ Arduino เป็นตัวสั่งให้ทางานโดย Arduino ได้ประมวลผลที่ได้รับจาก

เซ็นเซอร์ ส่งสัญญาณมายัง Servo ในการเปิด-ปิดฝาถังขยะ

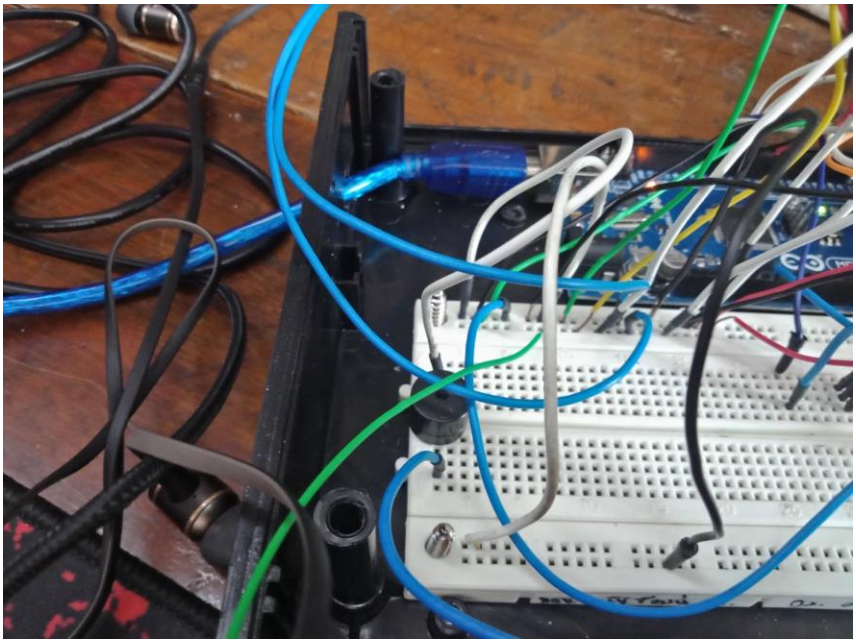


3.3.2.1 ระบบการทำงาน Servo

3.3.2.2 ระบบการทำงาน Servo ขณะรับสัญญาณ

3.3.3 การควบคุม Buzzer

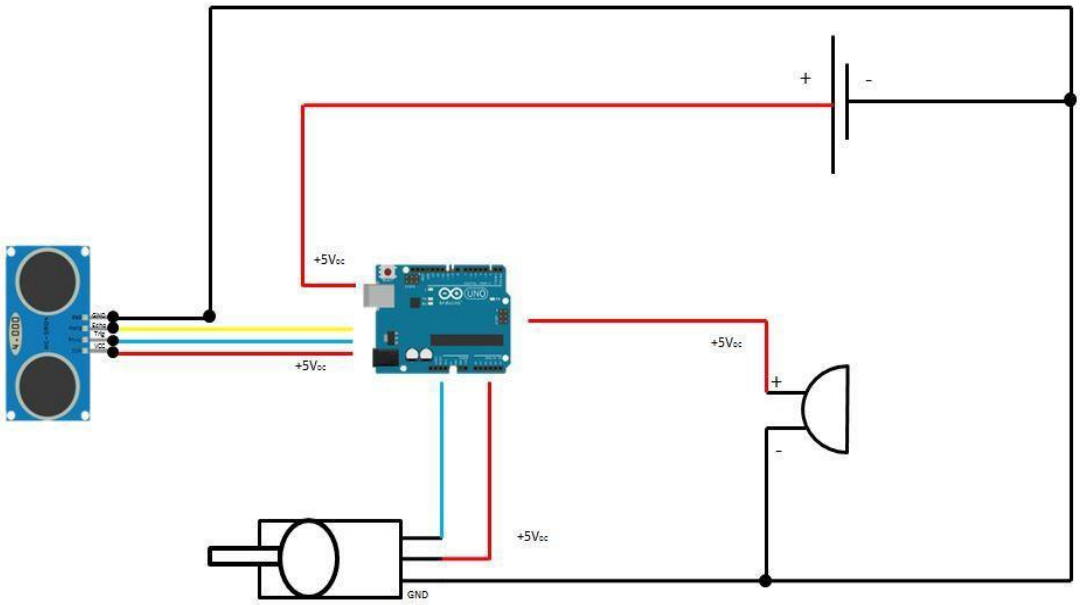
ในการควบคุม Buzzer ให้ Sensor เป็นเสมือน Switch ในการทำให้ Buzzer ทำงาน โดย Sensor
วัดระยะได้ตามที่ต้องการ จะส่งสัญญาณให้ Arduino ส่งคาสั่งให้ Buzzer ส่งเสียงออกมา



3.3.3.1 Buzzer

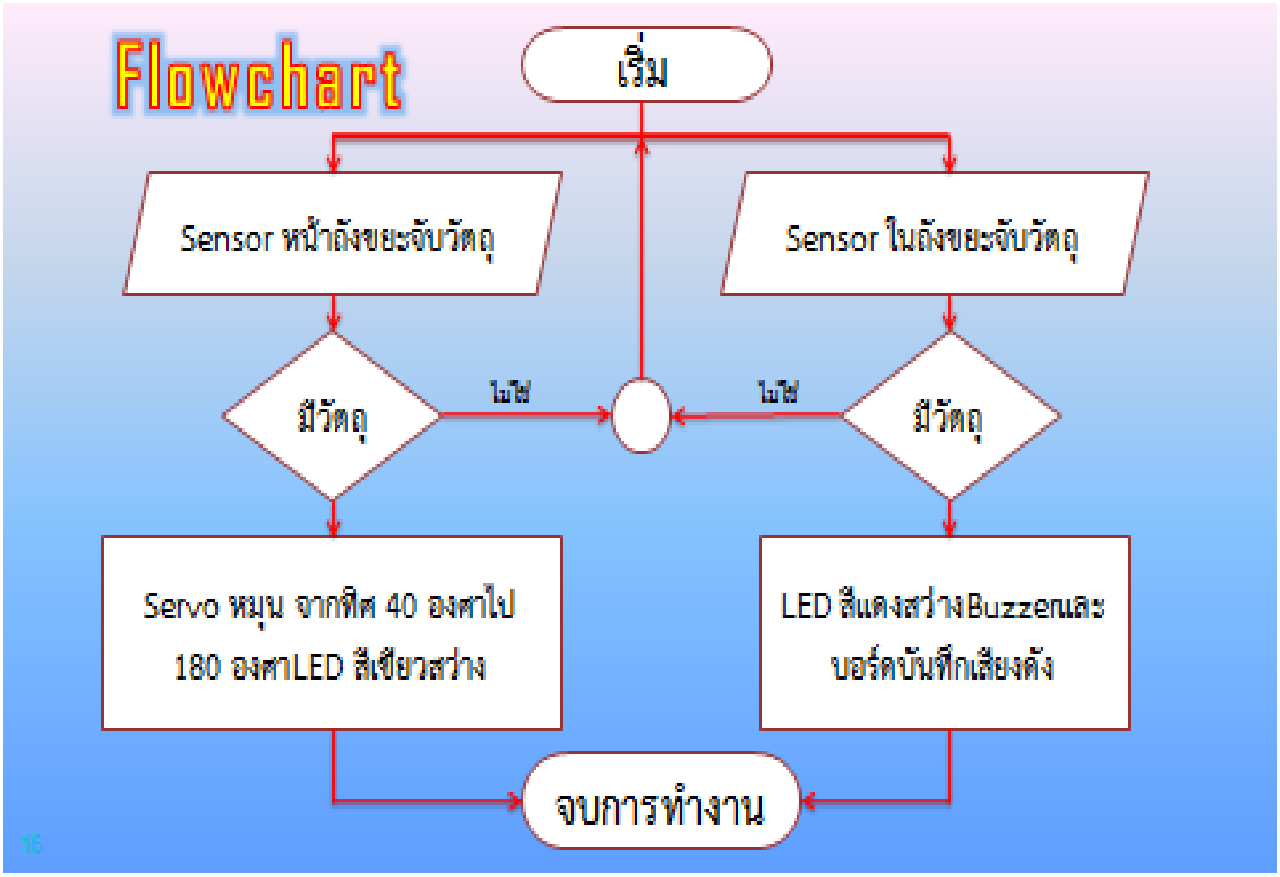
3.3.3.2 ระบบการทำงาน Sensor ขณะวัดระยะวัตถุ

3.4 วงจรรวมของระบบ



ภาพที่ 3.4 วงจรรวมควบคุมถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

3.5 Flowchart



ภาพที่ 3.5 ตัวอย่างผังระบบการทำงานของถังขยะ

3.6 โปรแกรมควบคุมการทำงาน

```
#include <Servo.h>
Servo myservo;
Servo myservo1;
const int musicPin = 7;
const int trigPin = 2;
const int echoPin = 3;
const int trigPin2 = 4;
const int echoPin2 = 5;
const int buzzer = 6;
const int ledGreen1 = 36;
const int ledYellow1 = 38;
const int ledYellow2 = 40;
const int ledRed1 = 42;
const int ledRed2 = 44;
long duration;
long duration2;
int distance;
int distance2;
int safetyDistance;
int safetyDistance2;
void setup()
{
{myservo.attach(12);}
{myservo1.attach(13);}
pinMode(musicPin, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(ledGreen1, OUTPUT);
pinMode(ledYellow1, OUTPUT);
pinMode(ledYellow2, OUTPUT);
```

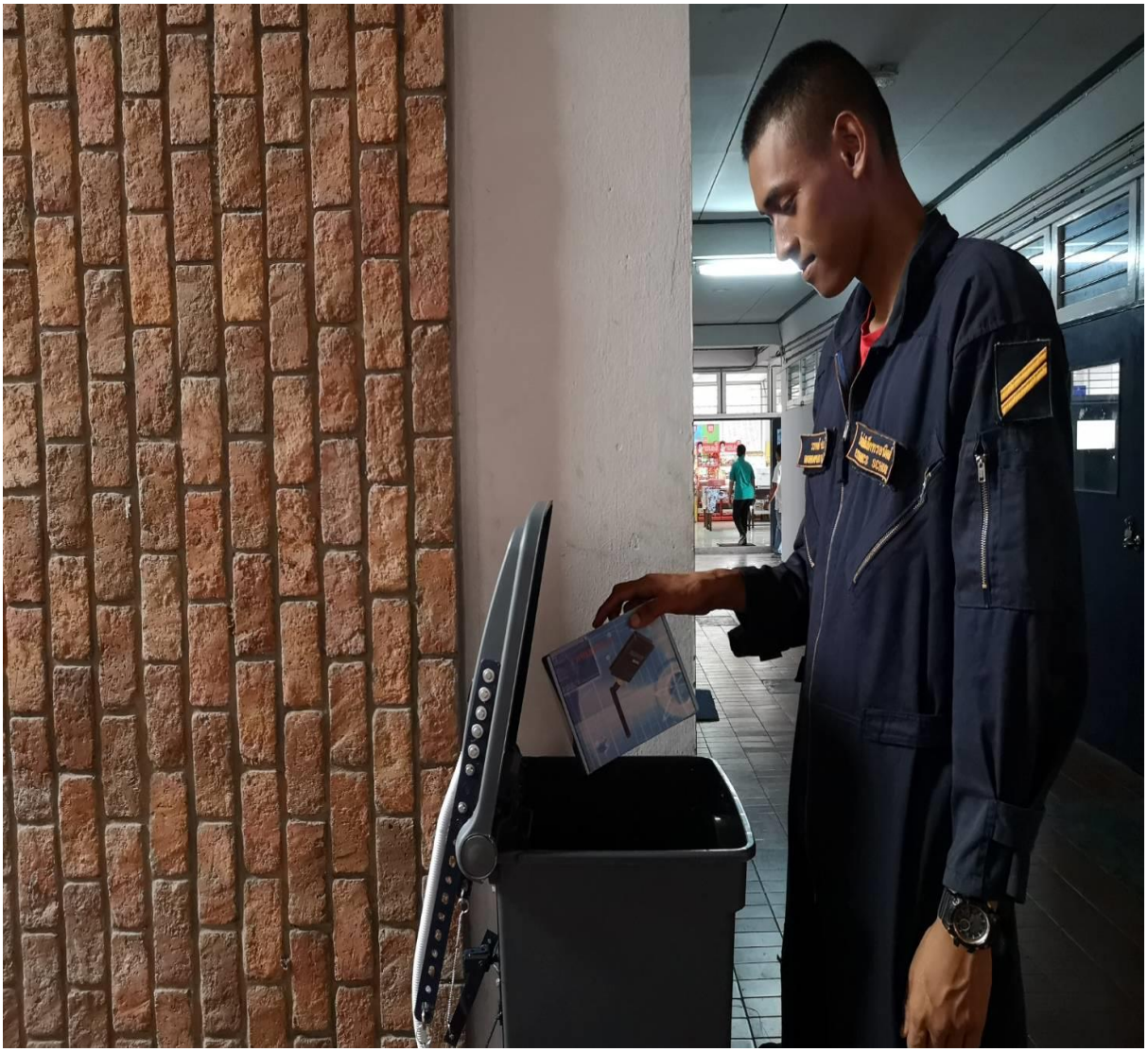
```
pinMode(ledRed1, OUTPUT);
pinMode(ledRed2, OUTPUT);
Serial.begin(9600); }
void loop()
{ digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2= duration2*0.034/2;
safetyDistance = distance;
safetyDistance2 = distance2;
if (safetyDistance <= 50)
{ digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW);
myservo.write(40); myservo1.write(180);
delay(5000); } if(safetyDistance >= 41)
{ myservo1.write(40);
myservo.write(180);
delay(500);
digitalWrite(musicPin, 1);
```

```
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if(safetyDistance2 >=35)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if(safetyDistance2 <=34 && safetyDistance2 >=11)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledYellow1, HIGH);
digitalWrite(ledYellow2, HIGH);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if (safetyDistance2 <= 10)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 0);
digitalWrite(buzzer, HIGH);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
```

```
digitalWrite(ledRed1, HIGH);
digitalWrite(ledRed2, HIGH); }
Serial.print("Distance: ");
Serial.println(distance);
Serial.print("Distance2: ");
Serial.println(distance2); }
```

บทที่ 4
ผลการทดลอง

ในบทนี้อธิบายเกี่ยวกับการทดสอบระบบและทดสอบการทำงานโดยแบ่งออกเป็น 3 การทดลอง คือ การทดลองตรวจจับของ Sensor, การทดลองสัญญาณเข้า Servo, การทดลองการทำงานของ Buzzer และ การทดลองตรวจจับวัตถุแล้วเปิด-ปิดอัตโนมัติ

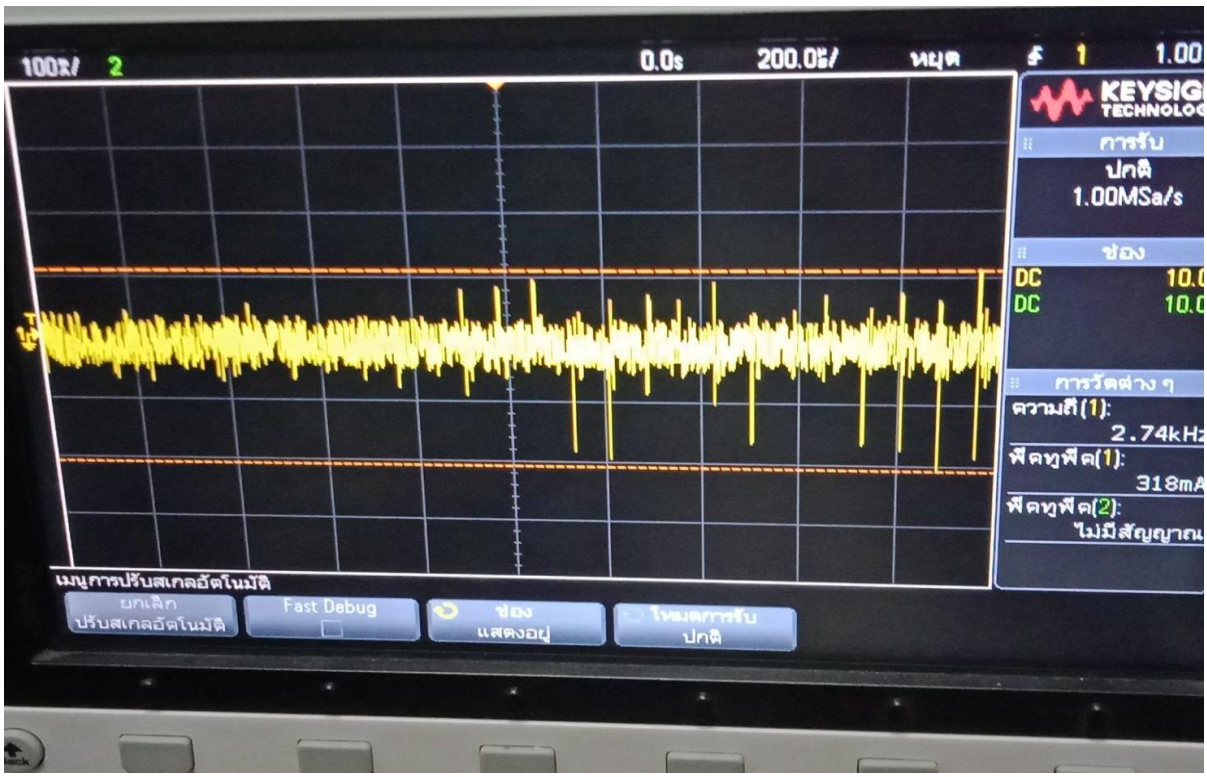


ภาพที่ 4 แสดงการทดสอบระบบ

4.1 การทดลองตรวจจับของ Sensor

ทดสอบ Sensor โดยการนำวัตถุเข้าใกล้เพื่อทำการตรวจจับระยะของSensor ต่อจากนั้นตัวควบคุม (Arduino Mega 2560 R3) จะทำการอ่านค่าตามที่เขียนโปรแกรมไว้

รูปที่ 4.1.1 การทดสอบ Sensor ขณะเปิดฝาถัง



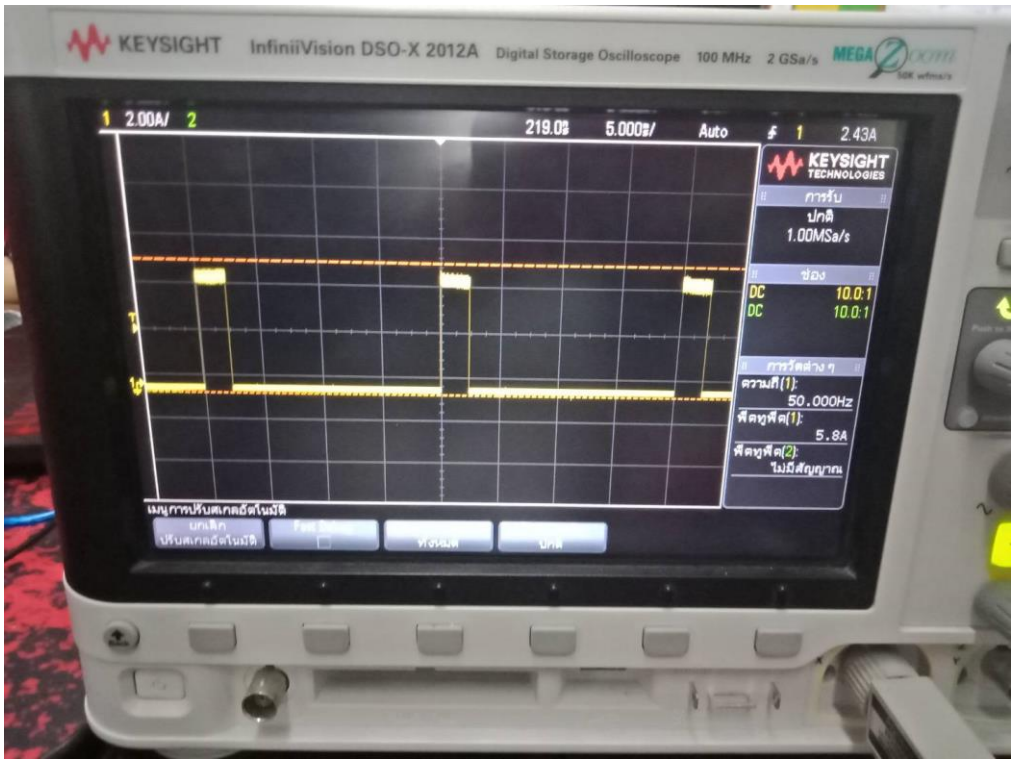
รูปที่ 4.1.2 การทดสอบ Sensor ขณะปิดฝาถัง

4.2 การทดลองสัญญาณเข้า

Servo

ในการทดลองใช้ Sensor เพื่อส่งสัญญาณออกไปยัง Arduino ทาการอ่านค่าและส่งสัญญาณ ไปให้ ควบคุม (Servo) จะทำการหมุนตัวอุปกรณ์ เพื่อให้ฝาถังขยะเปิด

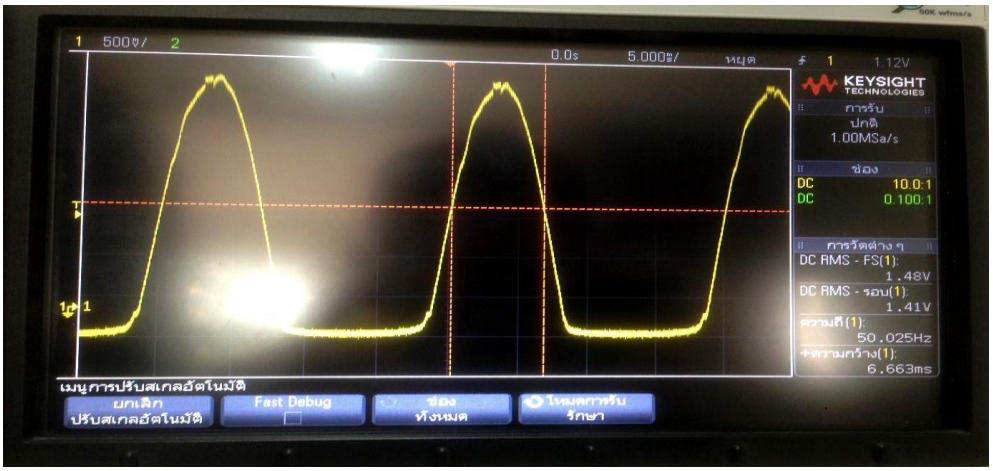
สัญญาณเข้า Servo



ภาพที่ 4.2 สัญญาณจาก Servo

4.3 การทดลองการทำงาน Buzzer

ในการทดลองใช้ Sensor จับวัตถุส่งสัญญาณให้ Arduino อ่านค่าและส่งสัญญาณให้ Buzzer ทำงาน เมื่อมีวัตถุอยู่ในระยะที่ตั้งไว้



ภาพที่ 4.3 สัญญาณจาก Buzzer

บทที่ 5
สรุปปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทำโครงงาน

โครงงานนี้เป็นระบบเปิด-ปิดของกลไกลให้ตรวจจับระยะที่กำหนดโดยทำการควบคุมสั่งการด้วยตัว ควบคุม Arduino Mega 2560 ควบคุมการทำงานด้วย Sensor และควบคุมการเปิด-ปิดด้วย Servo โดย จ่ายไฟ 5 Vdc ให้กับตัวควบคุมในการทำให้ทำงาน ตามลำดับ

Sensor สามารถทำงานได้จากการจับระยะที่ต้องการ โดยการสร้างความถี่ 40 Hz ด้วย Arduino mega 2560 โดยมีการเขียนโปรแกรมสั่งการ การทำงาน ของระบบตามความต้องการ

Servo สามารถทำงานได้โดยรับสัญญาณจาก Sensor ที่ส่งข้อมูลตามคาสั่งที่เขียนไว้ใน Arduino mega
2560 ให้เปิดปิดตามคำสั่งของระบบตามต้องการ

5.2 ปัญหาวิธีการแก้ไข

ปัญหาคือ

ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์จะต้องใช้เซนเซอร์ไว้ตรงไหน เพื่อไม่ให้พวกสุนัขหรือแมวเดินผ่านแล้วถังขยะเปิดขึ้นมา จึงเป็นสาเหตุที่ทาให้ถังขยะเปิด-ปิด

การแก้ไข

โดยการติดเซ็นเซอร์ไว้ให้สูงในระดับเอวโดยประมาณ เพื่อไม่ให้เซ็นเซอร์ตรวจจับสัตว์ที่เดินผ่าน

ปัญหาคือ

ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ เกิดเต็มถังขยะจะเปิดไหม เราจะรู้ยังไงว่ามันเต็ม

การแก้ไข

จะมีเซ็นเซอร์ตรวจจับขยะ ถ้าเกิดขยะเต็ม จะมีเสียงเตือน เราสามารถเปิดถังด้วยตัวเองเพื่อเอาถังขยะ
ไปทิ้งได้เลย

ปัญหาคือ

ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ เหตุใดถึงใช้ Servo motor 2 ตัว ทำไมไม่ใช้ 1 ตัว

การแก้ไข

เพื่อเพิ่มแรงยกฝาถังให้ง่ายขึ้น เพราะถังขยะมีขนาดที่ใหญ่ขึ้น ทำให้ต้องเพิ่ม Servo motor เป็น 2 ตัว

5.3 ข้อเสนอแนะ

ถ้าจะนำถังขยะไปใช้งานจริงต้องมีการพัฒนาให้มีความสามารถป้องกันน้ำและความชื้นได้เพื่ออุปกรณ์ที่ติดตั้งในตัวถังขยะจะได้ไม่เสียหาย

ภาคผนวก ก.

โปรแกรมระบบถังขยะเปิดปิดอัตโนมัติด้วยระบบเซ็นเซอร์

โปรแกรมควบคุมการทำงาน

```
#include <Servo.h>
Servo myservo;
Servo myservo1;
const int musicPin = 7;
const int trigPin = 2;
const int echoPin = 3;
const int trigPin2 = 4;
const int echoPin2 = 5;
const int buzzer = 6;
const int ledGreen1 = 36;
const int ledYellow1 = 38;
const int ledYellow2 = 40;
const int ledRed1 = 42;
const int ledRed2 = 44;
long duration;
long duration2;
int distance;
int distance2;
int safetyDistance;
int safetyDistance2;
void setup()
{
{myservo.attach(12);}
{myservo1.attach(13);}
pinMode(musicPin, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(buzzer, OUTPUT);
```

```
pinMode(ledGreen1, OUTPUT);
pinMode(ledYellow1, OUTPUT);
pinMode(ledYellow2, OUTPUT);
pinMode(ledRed1, OUTPUT);
pinMode(ledRed2, OUTPUT);
Serial.begin(9600); }
void loop()
{ digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2= duration2*0.034/2;
safetyDistance = distance;
safetyDistance2 = distance2;
if (safetyDistance <= 50)
{ digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW);
myservo.write(40); myservo1.write(180);
delay(5000); } if(safetyDistance >= 41)
{ myservo1.write(40);
```

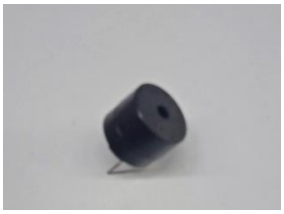
```
myservo.write(180);
delay(500);
digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if(safetyDistance2 >=35)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, HIGH);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if(safetyDistance2 <=34 && safetyDistance2 >=11)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 1);
digitalWrite(buzzer, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledYellow1, HIGH);
digitalWrite(ledYellow2, HIGH);
digitalWrite(ledRed1, LOW);
digitalWrite(ledRed2, LOW); }
if (safetyDistance2 <= 10)
{ myservo1.write(40);
myservo.write(180);
digitalWrite(musicPin, 0);
digitalWrite(buzzer, HIGH);
```

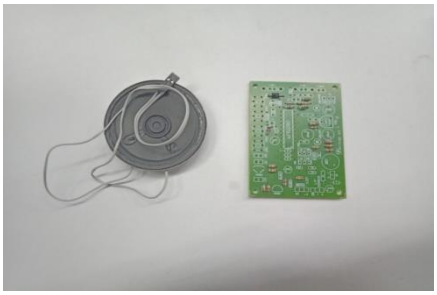
```
digitalWrite(ledGreen1, LOW);
digitalWrite(ledYellow1, LOW);
digitalWrite(ledYellow2, LOW);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledRed2, HIGH); }
Serial.print("Distance: ");
Serial.println(distance);
Serial.print("Distance2: ");
Serial.println(distance2); }
```

ภาคผนวก ข.

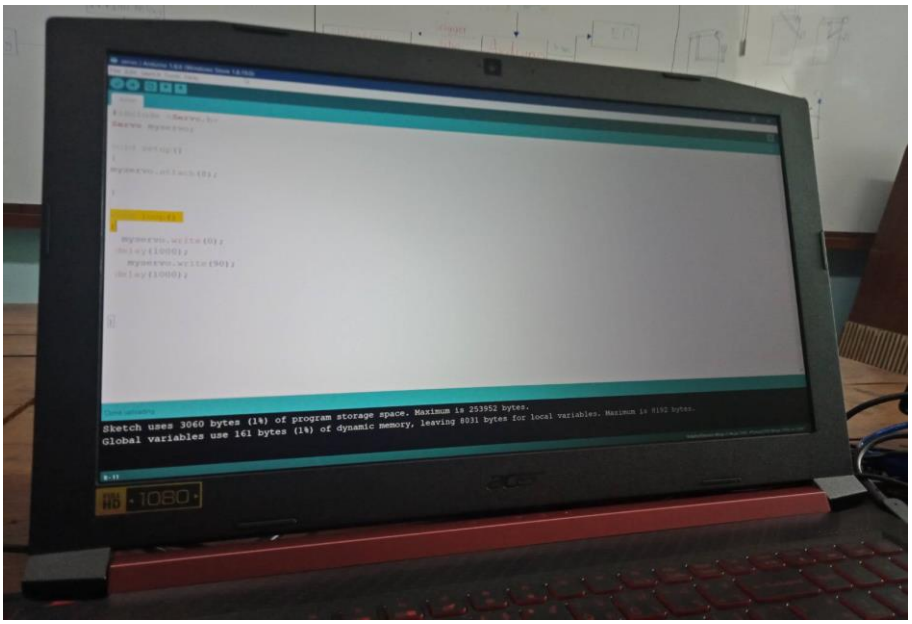
วิธีการประกอบถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

ชิ้นส่วนสำหรับการประกอบ

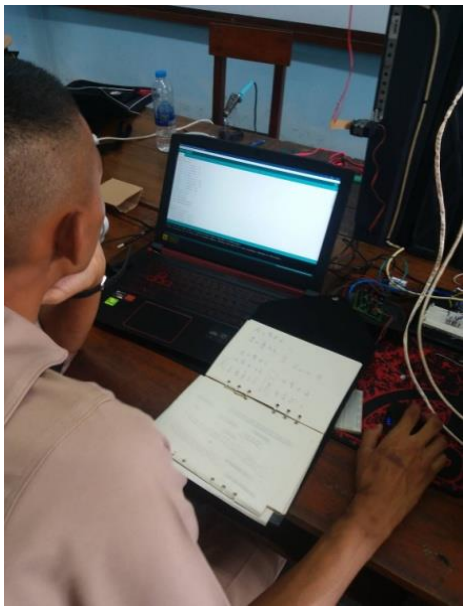


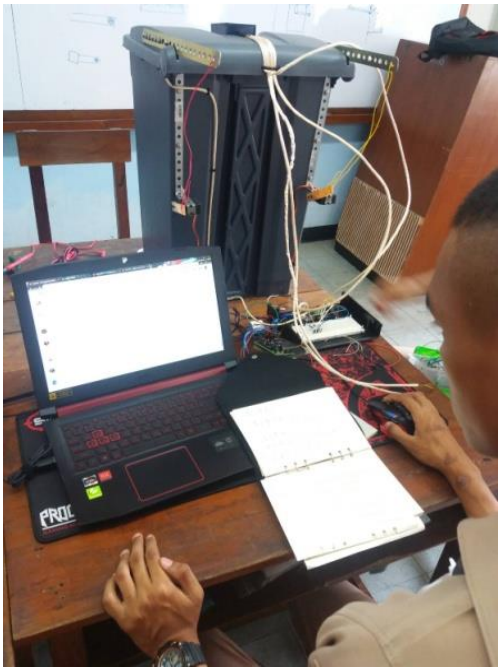


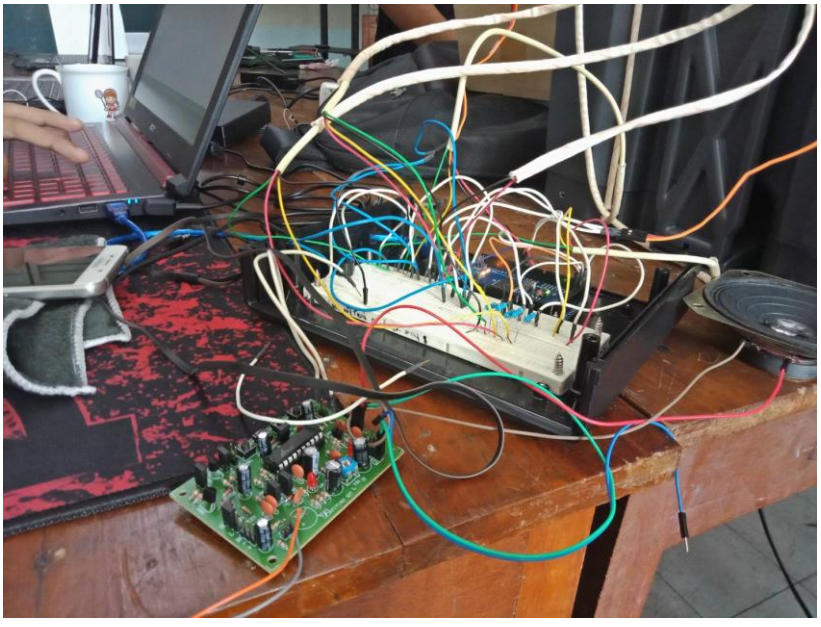
เขียนโปรแกรมลง Arduino



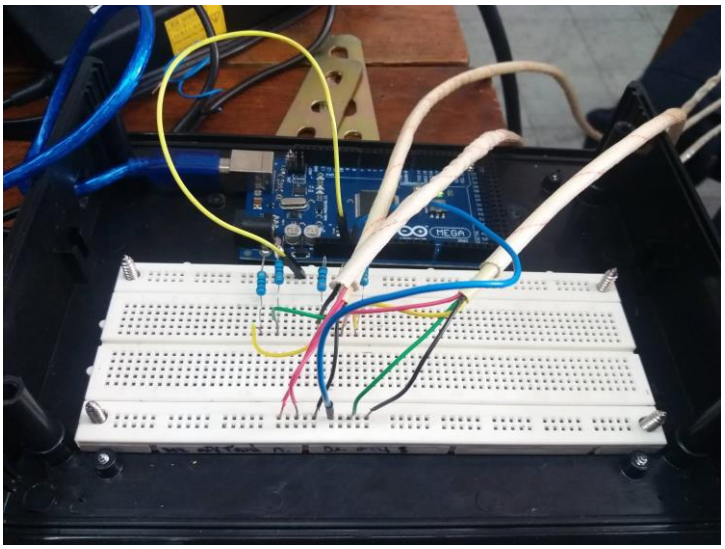
ประกอบเข้าที่ตามแผนก่อน

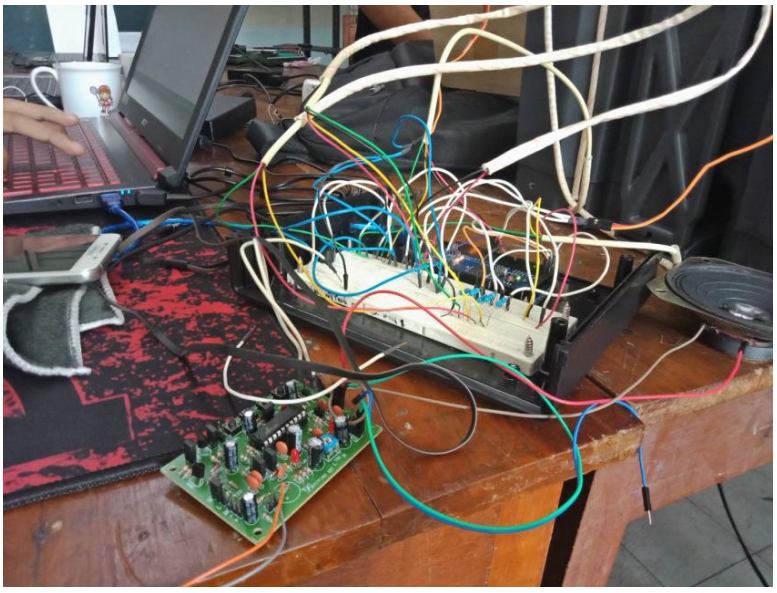






ประกอบสาย input/output ให้กับอุปกรณ์ควบคุม และแหล่งจ่ายไฟArduino





เมื่อทำเสร็จ

ภาคผนวก ค

คู่มือการใช้งาน

1. ใช้ไฟ 220 Vac 50 Hz 2.4 Amp. ขึ้นไปผ่านช่องรูเสียบ USB

ภาคผนวก ง.

ประวัติคณะผู้จัดทำโครงงาน

1. นรจ. จักรพงศ์ ชัยชาญ เหล่า อิเล็กทรอนิกส์
   ที่อยู่ 4 1 / 1 หมู่ 4 ตำบล ตาคง อำเภอ สังขะ จังหวัด สุรินทร์ 32150
   จบจาก โรงเรียนตาคงวิทยา จังหวัด สุรินทร์
2. นรจ. ธนสิษฐ์ ประทุมสุวรรณ เหล่า อิเล็กทรอนิกส์
   ที่อยู่ 2 5 9 / 6 0 บ้านปราณภิรมย์ ตำบล หนองตาแต้ม อำเภอ ปราณบุรี จังหวัด ประจวบคีรีขันธ์
   จบจาก โรงเรียนวันทามารีอา จังหวัด ประจวบคีรีขันธ์
3. นรจ. จีระศักดิ์ มูลทองชุน เหล่า อิเล็กทรอนิกส์
   ที่อยู่ 3 7 5 ตำบล บุคคโล อำเภอ ธนบุรี จังหวัด กรุงเทพมหานคร 10600
   จบจาก โรงเรียนทวีธาภิเศก จังหวัด กรุงเทพมหานคร
4. นรจ. ธนาสิน พุฒแตง เหล่า อิเล็กทรอนิกส์
   ที่อยู่ 3 5 / 1 หมู่ 11 ตำบล เบิกไพร อำเภอ บ้านโป่ง จังหวัด ราชบุรี 70110
   จบจาก โรงเรียนรัตนราษฎร์บำรุง จังหวัด ราชบุรี
5. นรจ. วรพงศ์ อ่วมเอิบ เหล่า อิเล็กทรอนิกส์
   ที่อยู่ 17/12 บ้าน หนองหญ้าม้า ตำบล พระธาตุบังพวน อำเภอ เมือง จังหวัด หนองคาย 43100
   จบจาก โรงเรียนทุ่มเทพวิทยาคาร จังหวัด หนองคาย

บรรณานุกรม

(1) https://www.arduinoall.com/
http://mbeddedweekly.blogspot.com/2014/08/arduino-mega2560.html

(2) https://www.arduinoall.com/product/688/hy-srf05-ultrasonic-sensor-module-ultrasonic-module-hy-srf05-distance-measuring-transducer-sensor

(3) https://www.arduinoall.com/product/16/arduino-uno-r3-%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%AA%E0%B8%B2%E0%B8%A2-usb-%E0%B8%84%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%AA%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99-arduino-starter-%E0%B8%AD%E0%B8%AD%E0%B8%99%E0%B9%84%E0%B8%A5%E0%B8%99%E0%B9%8C

(4) https://www.arduinoall.com/product/2591/servo-rds3120-robot-servo-20kg-metal-gear-digital-%E0%B9%81%E0%B8%81%E0%B8%99%E0%B9%82%E0%B8%A5%E0%B8%AB%E0%B8%B0%E0%B8%84%E0%B8%B9%E0%B9%88-0-90-%E0%B8%AD%E0%B8%87%E0%B8%A8%E0%B8%B2

(5) https://www.arduinoall.com/product/2265/ir-%E0%B8%95%E0%B8%B1%E0%B8%A7%E0%B8%AA%E0%B9%88%E0%B8%87%E0%B8%AA%E0%B8%B1%E0%B8%8D%E0%B8%8D%E0%B8%B2%E0%B8%93%E0%B8%AD%E0%B8%B4%E0%B8%99%E0%B8%9F%E0%B8%A3%E0%B8%B2%E0%B9%80%E0%B8%A3%E0%B8%94-940nm

(6) EADN028-i Module - Buzzer Kit V1.0-QuickStart Guide (PDF)