



ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

โดย

นรจ.พงศธร

นรจ.จิรกิตติ์

นรจ.สุรนาท

นรจ.อาชา

นรจ.ขันติ

ต่ายธานี

คณเมืองกุลศรี

สองนาม

ศรียโสภา

เผ่าภูรี

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรนักเรียนจำทหารชั้นปีที่ 2
พรรคพิเศษ เหล่าทหารช่างยุทธโยธา อีเล็คทรอนิกส์
ปีการศึกษา 2560

โรงเรียนอีเล็คทรอนิกส์ กองวิทยาการกรมอีเล็คทรอนิกส์ทหารเรือ

หัวข้อโครงการ ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

ผู้จัดทำ นรจ.พงศธร ต่ายธานี
 นรจ.จิรกิตติ์ คงเมืองกุลศรี
 นรจ.สุรนาท สองนาม
 นรจ.อาชา ศรีโสภา
 นรจ.ขันติ เผ่าภูรี

ครูที่ปรึกษา ร.ท.พฤทธิ์ กรณ์ย์
 พ.จ.อ.พงศกร เชื้อเถาว์
 พ.จ.อ.จารึก แจ่มดี

สถานศึกษา โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการกรมอิเล็กทรอนิกส์ทหารเรือ

ปีการศึกษา 2560

บทคัดย่อ

โครงการสิ่งประดิษฐ์นี้เป็นการศึกษาแบบจำลองถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ ในขณะที่เวลาที่ขยะเต็ม จะมีเสียงแจ้งเตือน ซึ่งต้องรู้จักหลักการทำงานของอุปกรณ์ การประกอบส่วนต่าง ๆ ของอุปกรณ์และการเขียนโปรแกรมควบคุมการทำงานตามวัตถุประสงค์ มีการวิเคราะห์และหาข้อมูลในส่วนนั้น โดยควบคุมการทำงานจาก Arduino 2560 สั่งให้ Sensor รับสัญญาณ และส่งกลับไปยัง Arduino 2560 ให้สั่งให้ servo ทำงาน ถังขยะจะเปิดปิดอัตโนมัติ เพื่อพัฒนาและขยายความสามารถให้มากขึ้นจากแบบจำลองถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ต่อไป

ผลการทดลองพบว่า ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ การควบคุมด้วย Arduino 2560 สามารถควบคุมการเปิด-ปิด ถังขยะ แบบอัตโนมัติด้วยการติดตั้ง Sensor สามารถรับสัญญาณได้ในทิศทางกับพื้น ส่งไปยัง Arduino 2560 สั่งให้ Servo ทำงาน จึงได้เพิ่มฐานรองรับถังขยะเพื่อให้ทราบว่า Sensor ตรวจจับในระยะนั้น จึงทำให้ถังขยะเปิด-ปิดอัตโนมัติด้วยและมี Sensor ตรวจจับขยะ อยู่บนฝาถัง ควบคุมด้วย Arduino 2560 จะส่งเสียงเตือน จะส่งผลให้ทราบว่าขยะเต็ม

กิตติกรรมประกาศ

โครงการสิ่งประดิษฐ์นี้สำเร็จได้ด้วยความกรุณาของผู้อำนวยการโรงเรียนอิเล็กทรอนิกส์กองวิทยาการ
กรมอิเล็กทรอนิกส์ทหารเรือและคณะครูที่ปรึกษาซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือต่าง ๆ ที่เป็น
ประโยชน์ จนกระทั่งโครงการสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอกราบขอบพระคุณ อาจารย์ที่ปรึกษา ที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของ โครงการ
และให้ความรู้ ให้คำแนะนำ ทั้งให้กำลังใจ

ท้ายสุดนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงการนี้จะเป็นประโยชน์ต่อการศึกษาและเป็นที่น่าสนใจ
สำหรับผู้สนใจต่อไป

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
บทที่ 1 บทนำ	1-2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3-12
บทที่ 3 ขั้นตอนการดำเนินการ	13-24
บทที่ 4 ผลการทดลอง	25-28
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	29-30
ภาคผนวก ก.	31-34
ภาคผนวก ข.	35-37
ภาคผนวก ค.	38
ภาคผนวก ง.	39
บรรณานุกรม	40

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ขยะมูลฝอยหรือขยะทั่วไป เป็นปัญหาของคนส่วนใหญ่ ซึ่งสังคมในปัจจุบัน ยังไม่มี การจัดการที่ดี รวมถึงภาคครัวเรือนที่มีการทิ้งขยะหรือ สิ่งปฏิกูลทุกวันเช่น เศษอาหาร ขยะพลาสติก เศษกระดาษ อีกทั้ง ประเทศไทยเป็นเมืองการเกษตร ขยะอินทรีย์ หรือ สารเคมีจากปุ๋ยที่ส่งผลต่อสิ่งแวดล้อมรวมถึงจิตใจสำนึกของคนในยุคปัจจุบันที่ยังไม่คำนึงถึงผลกระทบต่าง ๆ ที่ตามมา ปัจจุบันขยะมูลฝอยนั้นนับวันจะเพิ่มมากขึ้นตามจำนวนประชากร ถ้าหากไม่มีการจัดการที่ดี และเหมาะสมกับปัญหาความสกปรกต่าง ๆ ทำให้เกิดปัญหาต่อสุขภาพอนามัย เช่น ปัญหาเรื่องเชื้อโรค สารเคมี และกลิ่นเหม็นที่แพร่ไปตามสถานที่ต่าง ๆ

ปัจจุบันมีถังขยะอยู่จำนวนมาก แต่ถังขยะที่มีอยู่ในปัจจุบันส่วนใหญ่นั้นอยู่ในสภาพที่ไม่สมบูรณ์เนื่องจากผ่านการใช้งานเป็นระยะเวลานาน เช่น ไม่มีฝาปิด ฝาถังชำรุด ถังขยะมีรอยแตกร้าว มีการสะสมของขยะ และยังขาดการดูแลรักษาที่ดีซึ่งเป็นสถานที่สะสมของเชื้อโรคและแบคทีเรีย อาจจะทำให้เกิดการติดเชื้อแก่ผู้ที่ทิ้งขยะโดยการสัมผัสกับถังขยะ

ความเป็นไปได้ของโครงการเป็นไปได้ที่จะบรรลุตามเป้าหมาย เนื่องจากกลุ่มคณะกรรมการหนักถึงปัญหาเหล่านี้ดังที่กล่าวมาข้างต้น ฉะนั้นเราจึงร่วมกันคิดหาเทคนิคการทำต่อยอดจากถังขยะในรูปแบบเดิม มีการนำมาประยุกต์จากรายวิชาที่เรียนไมโครคอนโทรลเลอร์ พัฒนาคุณภาพชีวิต และเรดาร์โดยใช้เป็นระบบเซนเซอร์ของถังขยะโดย ร.ท.พฤทธิ กรณีย์ พ.จ.อ.พงศกร เชื้อเถาว์ และ พ.จ.อ.จารึก แจ่มดี ครูที่ปรึกษาที่คอยให้การสนับสนุนและให้คำปรึกษาโดยมีงบประมาณสนับสนุนจากโรงเรียนอัสสัมชัญธนบุรี เมื่อโครงการสำเร็จจึงส่งผลทำให้มีถังขยะที่มีสภาพพร้อมใช้งานและมีประสิทธิภาพ ทำให้เกิดสภาพแวดล้อมที่ดี สะอาด เป็นระเบียบเรียบร้อย รวมไปถึงการตอบสนองความต้องการของมนุษย์ในยุคปัจจุบัน

กลุ่มของคณะกรรมการจึงทำโครงการ ถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซนเซอร์เป็นโครงการที่เห็นถึงความสำคัญของการพัฒนาเทคโนโลยีประกอบเข้ากับปัญหาขยะในปัจจุบันเพื่อให้มนุษย์เห็นความสำคัญของการทิ้งขยะและปัญหาของสิ่งแวดล้อมสุขอนามัยของคนในยุคปัจจุบัน

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 ป้องกันเชื้อโรคที่มาจากการสัมผัสกับถังขยะและสามารถประยุกต์นำความรู้ที่ได้เรียนมาใช้ในการทำโครงการ

1.4 ขอบเขตของโครงการ

- 1.4.1 ศึกษาการทำงานของไมโครคอนโทรลเลอร์ เพื่อประยุกต์ใช้ในการควบคุมการทำงานของระบบเปิด-ปิดอัตโนมัติของถังขยะ
- 1.4.2 ศึกษาการออกแบบระบบ การติดตั้งและการทำงานของระบบเซ็นเซอร์อัลตราโซนิก
- 1.4.3 ศึกษาการตอบสนองของระบบเซ็นเซอร์กับการเปิด-ปิดอัตโนมัติของถังขยะ

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-ปิดอัตโนมัติด้วยเซ็นเซอร์อัลตราโซนิก สิ่งแรกที่เป็นในการจัดทำโครงงาน ต้องรู้จักหลักการทำงานของอุปกรณ์ การประกอบส่วนต่าง ๆ ของอุปกรณ์และการเขียนโปรแกรมควบคุมการทำงานตามวัตถุประสงค์ มีการวิเคราะห์และหาข้อมูลในส่วนนั้น เพื่อเป็นแหล่งข้อมูลในการนำไปใช้ในการทำโครงงานและพัฒนาต่อยอดการศึกษา คณะผู้จัดทำได้ทำการรวบรวมแนวคิดหลักการและทฤษฎีต่าง จากเอกสารที่เกี่ยวข้องดังต่อไปนี้

- 2.1 หลักการ Arduino MEGA 2560
- 2.2 หลักการ เซ็นเซอร์ วัดระยะทาง US-100
- 2.3 หลักการ Servo Motor
- 2.4 หลักการ Buzzer

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

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

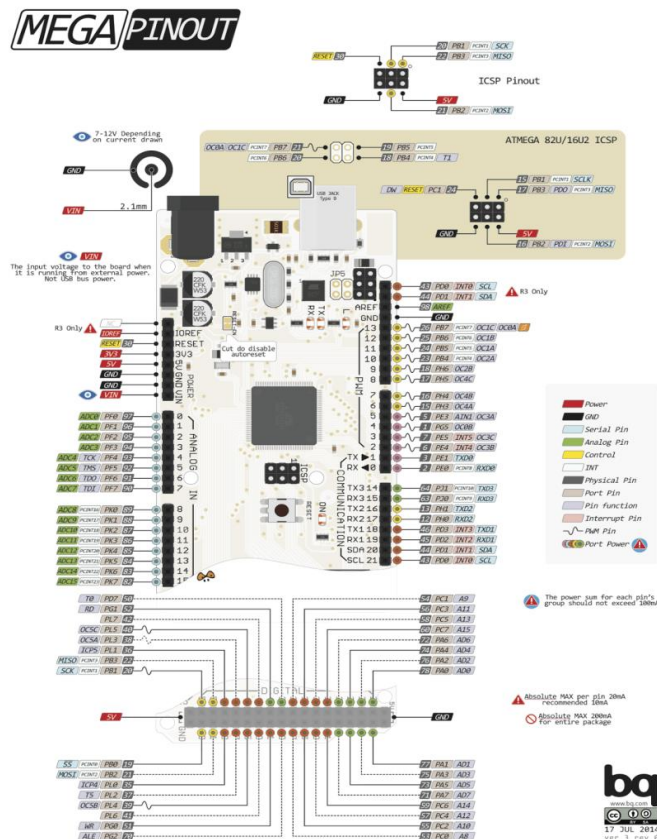
Arduino Mega2560 Revision 2 มี ATmega8U2 ทำให้อัพเดท firmware ผ่าน USB protocol ที่เรียกว่า DFU (Device Firmware Update) ได้ง่ายขึ้น

2.1.1 Arduino Mega Revision 3 มี feature ใหม่ ๆ เพิ่มขึ้นมาดังนี้

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 อธิบายการใช้งานของ 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 และการต่อเข้ากับแบตเตอรี่สามารถทำได้โดยการต่อเข้ากับ GND และ Vin pin header ของ power connector

บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 volts DC ถ้า แหล่งจ่ายมีค่าต่ำกว่า 7 Vdc อาจส่งผลให้ 5 Vdc pin มีแรงดันที่ต่ำกว่า 5Vdc และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่า สูงกว่า 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 ในค่าต่างๆ, 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 ไม่เกี่ยวข้องกับ 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). รองรับการทำงานเชื่อมต่อแบบ 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 shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

Communication

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

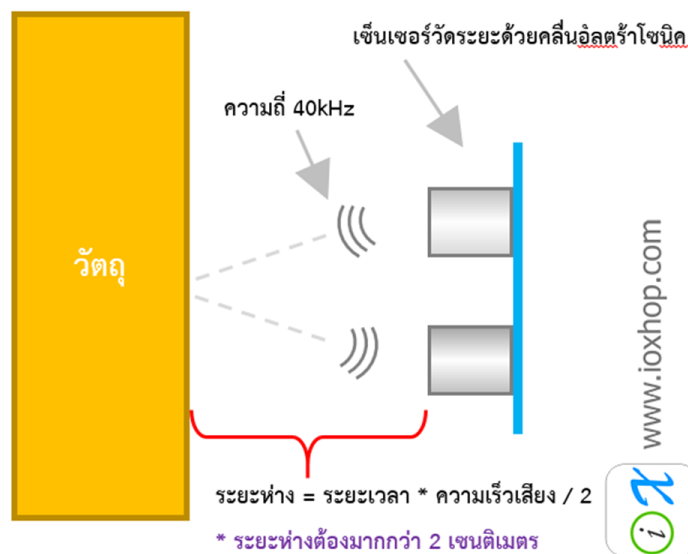
2.2 หลักการ เซ็นเซอร์ วัดระยะทาง US-100

คลื่นอัลตราโซนิก เป็นคลื่นความถี่เหนือความถี่สัญญาณเสียง ปกติแล้วมนุษย์จะไม่สามารถได้ยินเสียง เพราะมนุษย์สามารถได้ยินเสียงได้ที่ความถี่ 20 Hz ถึง 20 kHz

ความถี่อัลตราโซนิกนั้น ที่นิยมใช้งานในเซ็นเซอร์วัดระยะรุ่นต่าง ๆ จะมีความถี่ที่ประมาณ 40kHz ข้อดีของการใช้ความถี่นี้ คือมีลักษณะของความยาวคลื่นที่สั้น ส่งผลให้คลื่นไม่แตกกระจายออกเป็นวงกว้าง และสามารถยิงคลื่นตรงไปชนวัตถุใดๆ ก็ได้ และนอกจากนี้ความถี่ 40kHz ยังเป็นความถี่ที่มีระยะเดินทางเพียงพอกับการใช้งาน หากใช้ความถี่สูงขึ้น จะทำให้คลื่นเดินทางได้ในระยะทางที่ลดลง ทำให้เมื่อนำมาใช้งานจริงจะวัดระยะได้ในระยะที่สั้น

หลักการวัดระยะด้วยคลื่นอัลตราโซนิก

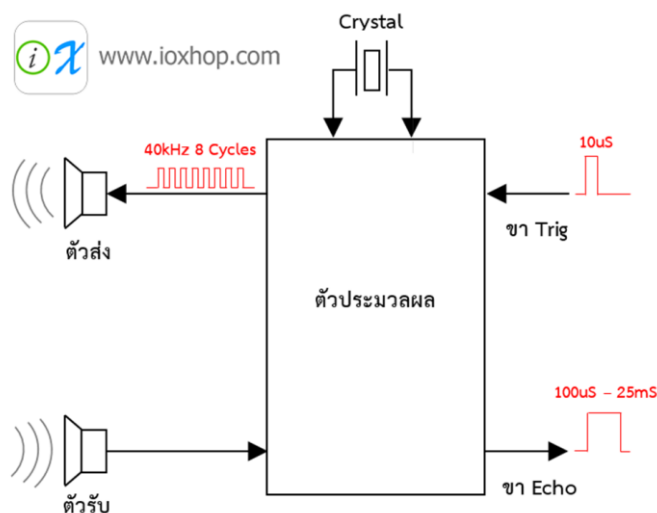
หลักการของการวัดระยะด้วยคลื่นอัลตราโซนิก คือ การส่งคลื่นอัลตราโซนิกออกไปจากตัวส่ง (Transmitter) เมื่อคลื่นที่ส่งออกไปวิ่งไปชนกับวัตถุ คลื่นจะมีการสะท้อนกลับมา แล้ววิ่งกลับไปยังตัวรับ (Receiver) ด้วยการเริ่มนับเวลาที่ส่งคลื่นออกไปจนถึงได้รับคลื่นกลับมา ทำให้สามารถหาระยะทางระหว่างวัตถุกับเซ็นเซอร์



รูปที่ 2.2.1 หลักการวัดระยะด้วยอัลตราโซนิก

หลักการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

ในโมดูลเซ็นเซอร์อัลตราโซนิก จะมีวงจรที่แตกต่างกัน แต่มีหลักการทำงานที่สำคัญที่เหมือนกัน



รูปที่ 2.2.2 ไดอะแกรมการทำงานของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก

เมื่อมีการส่งสัญญาณเข้าไปที่ Trigger วงจรภายในจะเริ่มสร้างคลื่นที่ 40 kHz จำนวน 8 ลูกคลื่นออกไป โดยใช้ความถี่จากคริสตอลเป็นตัวอ้างอิง แล้วตัวส่งที่เปรียบเสมือนลำโพงจะส่งสัญญาณออกไป จากนั้นเมื่อคลื่นที่ส่งออกไปวิ่งกลับมาที่ตัวรับ ที่เปรียบเสมือนเป็นไมโครโฟน สัญญาณไฟฟ้าจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo

หลักการใช้งานจะขึ้นอยู่กับสิ่งที่ใช้สื่อสาร สามารถแบ่งได้ดังนี้

การสื่อสารแบบทริกสัญญาณ เริ่มต้นจะต้องให้สัญญาณขา Trigger มีสถานะทางลอจิกเป็น LOW จากนั้นจึงเริ่มทริกเกอร์สัญญาณ โดยให้ขา Trigger มีสถานะเป็น HIGH ค้างไว้อย่างน้อย 10µs 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

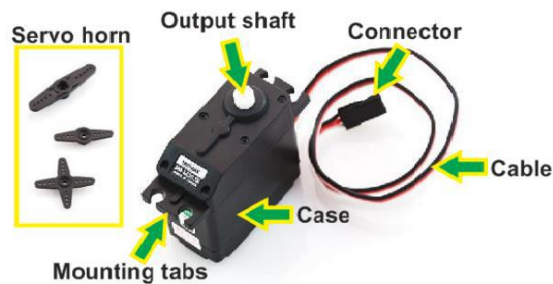
2.3 หลักการ Servo Motor

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



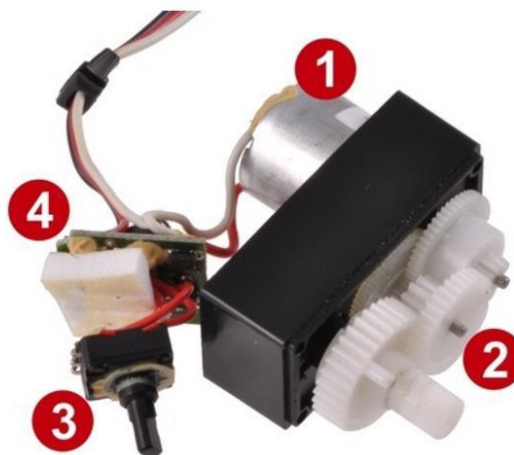
รูปที่ 2.3.1 ลักษณะโดยทั่วไปของเซอร์โวมอเตอร์

Feedback Control คือระบบควบคุมที่มีการนำค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่าเท่ากับหรือใกล้เคียง กับค่าอินพุต ส่วนประกอบภายนอก RC Servo Motor



รูปที่ 2.3.2 ส่วนประกอบของเซอร์โวมอเตอร์

- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เพลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- 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



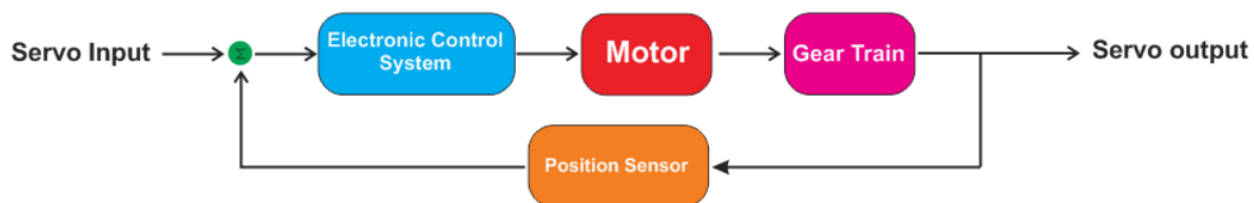
รูปที่ 2.3.3 ส่วนประกอบภายใน RC Servo Motor

หมายเลข 1 คือ ดิซีมอเตอร์

หมายเลข 2 คือ ชุดเฟืองเกียร์ทดแรงมอเตอร์

หมายเลข 3 คือ ชุดควบคุมและปรับแต่งให้ค่าเอาต์พุตมีค่าเท่ากับ หรือใกล้เคียงกับค่าอินพุต

หมายเลข 4 คือ ส่วนที่ควบคุมและประมวลผล



รูปที่ 2.3.4 บล็อกไดอะแกรมของ Servo Motor

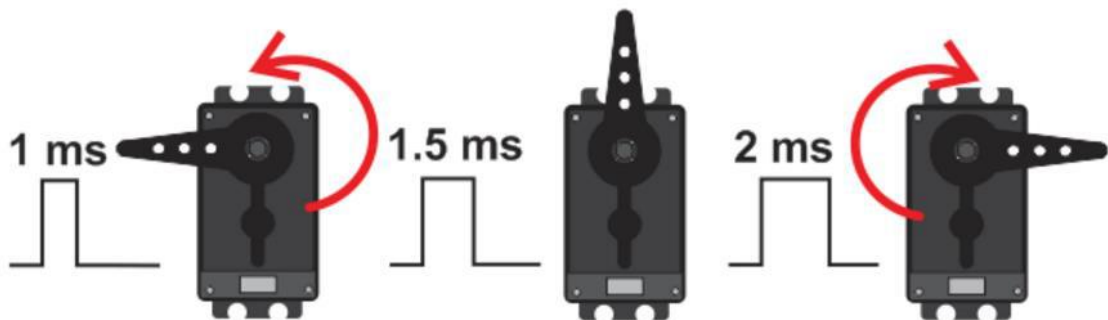
หลักการทำงานของ RC Servo Motor

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

ตัว RC Servo Motor ออกแบบมาสำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่นด้วยสัญญาณวิทยุต่าง ๆ เช่นเครื่องบินบังคับ รถบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ ที่ภาครับจะแปลงความถี่วิทยุออกมาในรูปแบบสัญญาณ 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 จะหมุนไปทางซ้ายจนสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะมีมุมมาอยู่ที่ตำแหน่งตรงกลางพอดี



รูปที่ 2.3.5 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลส์

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่น RC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ $1000\ \mu\text{s}$ ที่ 180 องศาความกว้างพัลส์เท่ากับ $2000\ \mu\text{s}$ เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน $(2000 - 1000) / 180$ เท่ากับ $5.55\ \mu\text{s}$

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาพัลส์ที่ต้องการได้จาก 5.55×45 เท่ากับ $249.75\ \mu\text{s}$ แต่ที่มุม 0 องศา เราเริ่มที่ความกว้างพัลส์ 1 ms หรือ $1000\ \mu\text{s}$ เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ $1000 + 249.75$ เท่ากับประมาณ $1250\ \mu\text{s}$

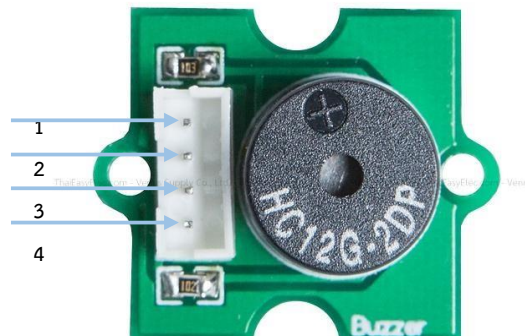
2.4 หลักการ Buzzer

การเชื่อมต่อฮาร์ดแวร์ (Hardware Installation)

การเชื่อมต่อ จากตัวอย่างจะการใช้การเชื่อมต่อโดยผ่าน i Module - Arduino Connector Base Shield หรือสามารถใช้สาย 4 Wire Cable(ELNK039) ในการเชื่อมต่อได้เช่นเดียวกัน โดยในการเชื่อมต่อ ควรเชื่อมต่อ

ต่อไปยังขาสัญญาณ Output ที่เป็น PWM เนื่องจากจะต้องสั่งงานโมดูลนี้ด้วยสัญญาณ PWM (หากใช้สัญญาณที่ไม่มีความถี่ หรือความถี่ต่ำเกินไป ตัว Buzzer จะไม่ตอบสนอง)

ขาสัญญาณและการเชื่อมต่อ (Pin Definition and Rating)



รูปที่ 2.6.1 ขา Buzzer

ขา 1 : Vcc

ขา 2 : GND

ขา 3 : PWM Signal IN

ขา 4 : -

สรุป

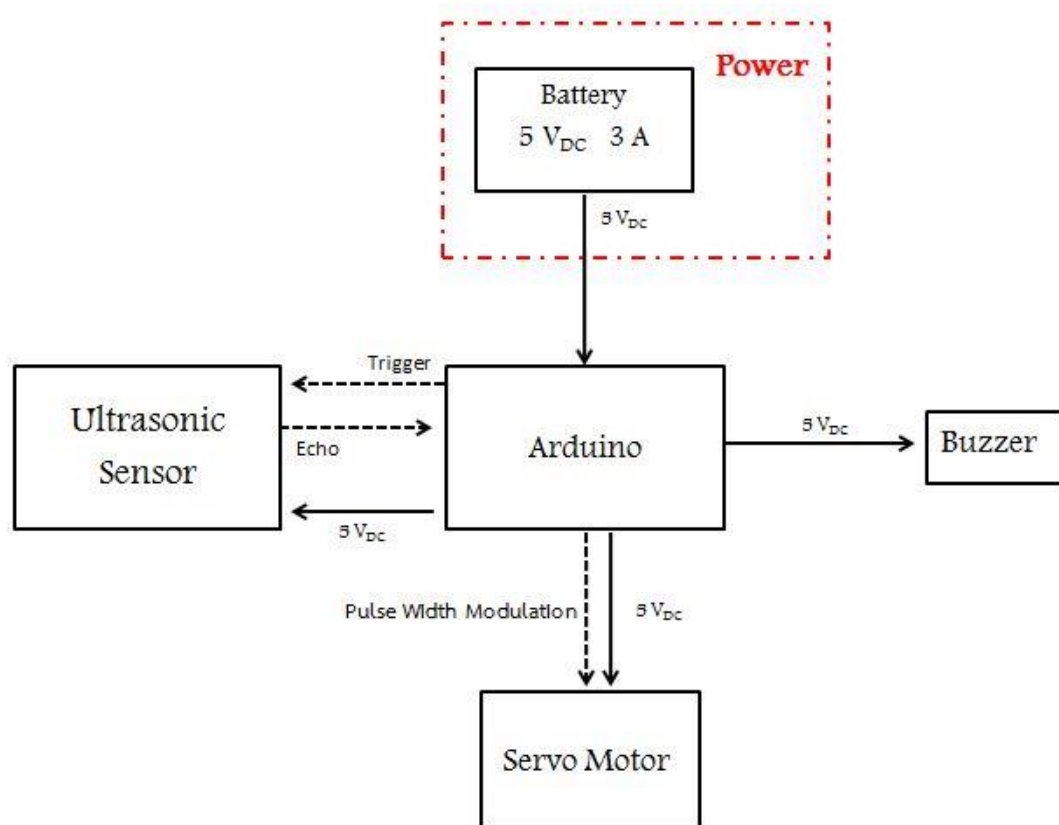
บทนี้แนะนำหลักการที่เกี่ยวข้องใช้หลักการของการทำงานของบอร์ด Arduino โดยที่บอร์ด Arduino เป็นตัวควบคุมการทำงานของระบบต่าง ๆ กล่าวคือ คำสั่งที่เขียนในบอร์ด Arduino จะไปทำให้เซนเซอร์วัดระยะทำงานโดยหลักการทำงานของเซนเซอร์จะส่งสัญญาณเข้าไปที่ Trig ภายในวงจรจะเริ่มสร้างความถี่ 40 kHz ออกไป แล้วตัวส่งที่เปรียบเสมือนลำโพงจะส่งสัญญาณออกไป 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

บทที่ 3

ขั้นตอนการดำเนินการ

ในบทนี้ได้อธิบายวิธีการออกแบบแต่ละส่วนที่ใช้ในโครงงานนี้ประกอบไปด้วย บล็อกไดอะแกรม วงจรรวม ชิ้นส่วนประกอบ การทำงานของเซนเซอร์ วัดระยะทาง การทำงานของ Servo Motor การทำงานของรีเลย์ การรับค่าสัญญาณจาก Sensor สำหรับการสั่งงานให้ Servo Motor ทำงาน การทำงานทั้งหมดควบคุมด้วย Arduino Mega 2560 R3 ให้ไปดูที่ภาคผนวก ก

3.1 บล็อกไดอะแกรมการควบคุมถึงขั้วเปิด-ปิดอัตโนมัติด้วยอัลตราโซนิก

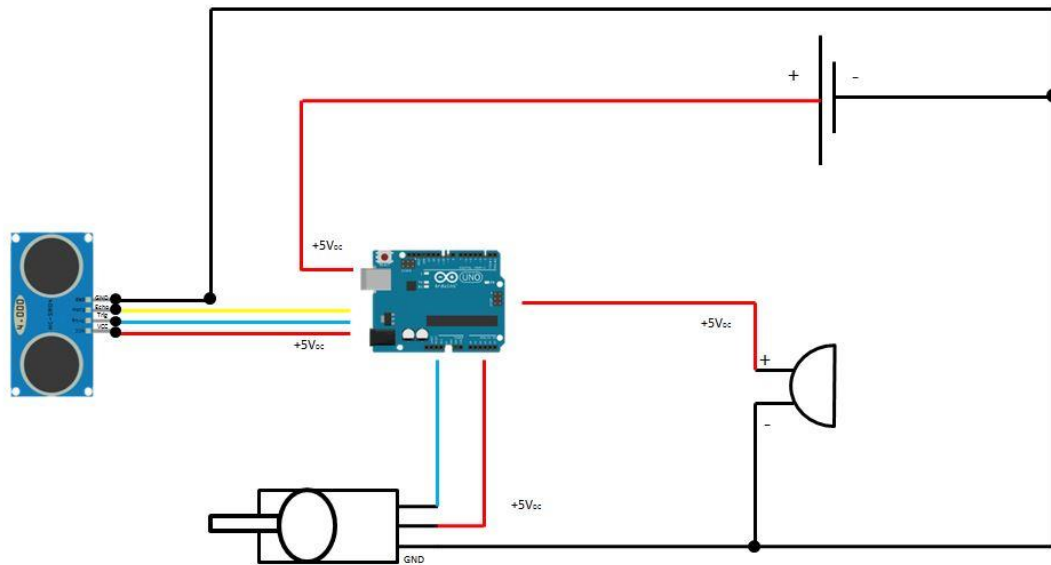


ภาพที่ 3.1 บล็อกไดอะแกรมการควบคุมถึงขั้วเปิด-ปิดอัตโนมัติด้วยอัลตราโซนิก

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

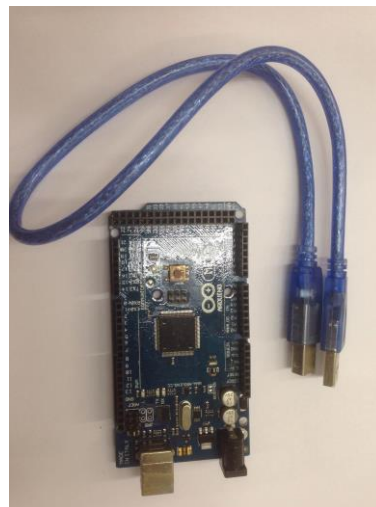
สามารถอธิบายระบบโดยรวมได้ดังนี้ เมื่อตัววัดระยะทางจับระยะได้ Sensor จะส่งสัญญาณพัลส์ ซึ่งจะมีสัญญาณพัลส์แตกต่างกัน โดยมีการส่งสัญญาณ Trigger ออกไป จากนั้นเมื่อคลื่นที่ส่งออกไป วิ่งกลับมาที่ตัวรับ สัญญาณที่ได้รับมาจะผ่านตัวประมวลผล แล้วให้ค่าเอาต์พุตออกมาทางขา Echo ส่งให้กับตัวควบคุม (Arduino) ตัวควบคุมจะทำการถอดรหัสสัญญาณพัลส์ แล้วสัญญาณที่อ่านได้มาเปรียบเทียบกับเงื่อนไขของโปรแกรมที่ได้เขียนเอาไว้ ถ้าตรงกับเงื่อนไขจะส่งคำสั่งนั้นไปทำให้ Servo ทำงาน

3.2 วงจรรวมของระบบ



ภาพที่ 3.2 วงจรรวมควบคุมถังขยะเปิด-ปิดอัตโนมัติด้วยอัลตราโซนิก

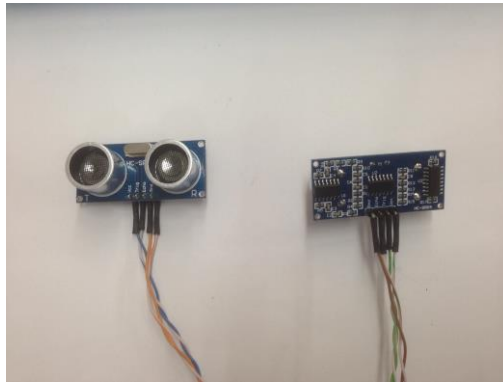
3.3 ชิ้นส่วนประกอบของถังขยะเปิด-ปิดอัตโนมัติด้วยอัลตราโซนิก



3.3.1 Arduino Board



3.3.2 Servo Motor



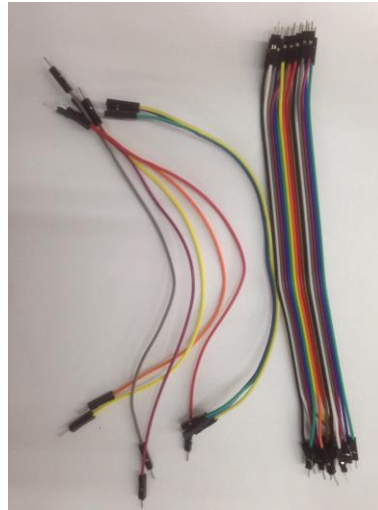
3.3.3 เซ็นเซอร์อัลตราโซนิก วัดระยะทาง US-100



3.3.4 Buzzer



3.3.5 เหล็กเส้นแบน กว้าง 1 นิ้ว ยาว 1 เมตร



3.3.6 สายไฟจัมเปอร์



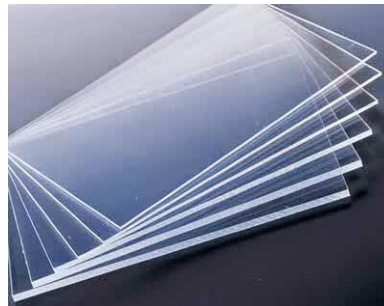
3.3.7 น็อต M3x6MM



3.3.8 กล่อง(สีดำ)เอนกประสงค์ใส่วงจร (Boxes)



3.3.9 แบตสำรอง (Powerback)



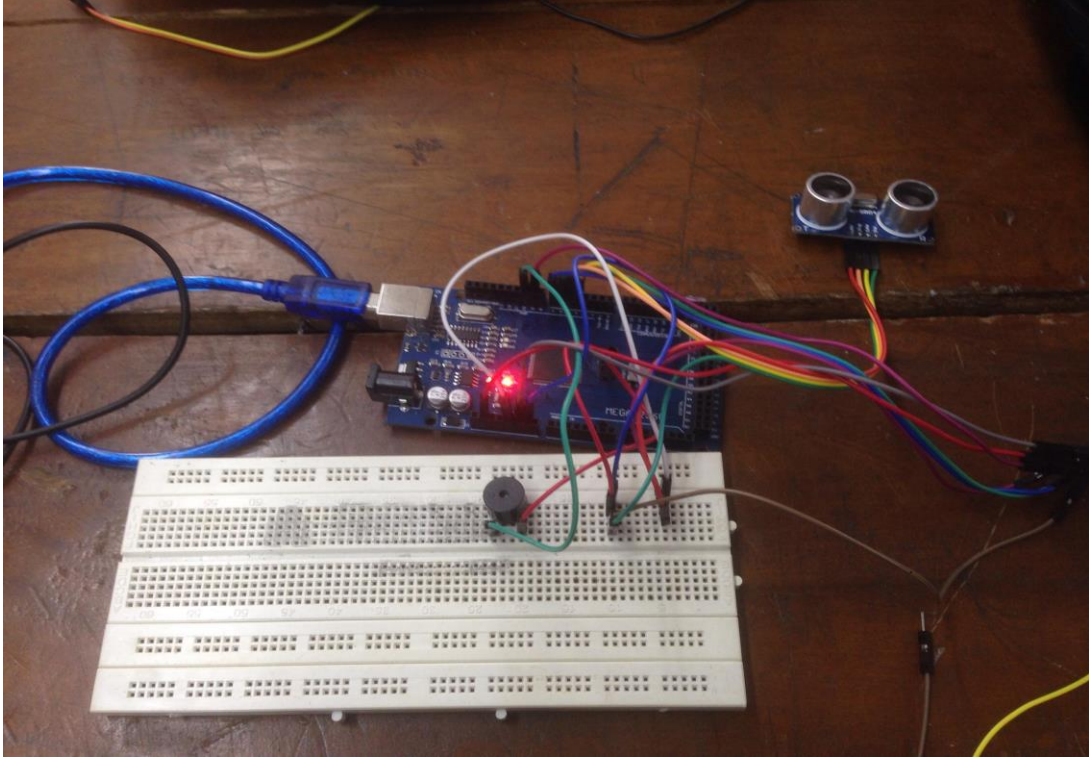
3.3.10 แผ่นอะคริลิกใส A4



3.3.11 ถังขยะขาเหยียบ ขนาด 27.5 × 35.5 ซม.

3.4 การควบคุมการทำงานของระบบ

การควบคุมการทำงานของระบบจะใช้ Arduino เป็นตัวควบคุมการทำงานทั้งหมด ส่วนที่ Arduino ไปควบคุมได้แก่



ภาพที่ 3.4 ระบบการทำงาน

3.4.1 การควบคุม เซ็นเซอร์ วัดระยะทาง

ในการควบคุมเซ็นเซอร์ วัดระยะทาง ให้ Arduino สั่งให้ทำงานเมื่อมีวัตถุมาใกล้ในระยะที่กำหนด โดยเซ็นเซอร์จะส่งสัญญาณออกไปในระยะที่กำหนด แล้วสะท้อนสัญญาณที่ส่งไปกลับมาประมวลผล



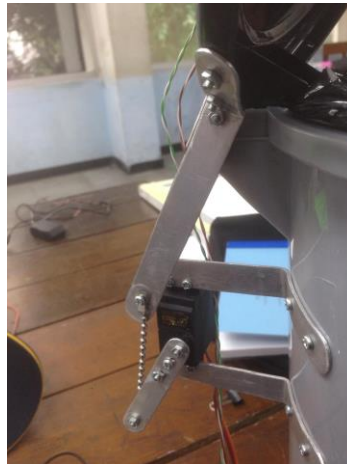
ภาพที่ 3.4.1 ระบบการทำงานเซ็นเซอร์

3.4.2 การควบคุม Servo Motor

ในการควบคุม Servo ให้ Arduino เป็นตัวสั่งให้ทำงานโดย Arduino ได้ประมวลผลที่ได้รับจากเซ็นเซอร์ ส่งสัญญาณมายัง Servo ในการเปิด-ปิดฝาถังขยะ



ภาพที่ 3.4.2 ระบบการทำงาน Servo ขณะยังไม่รับสัญญาณ



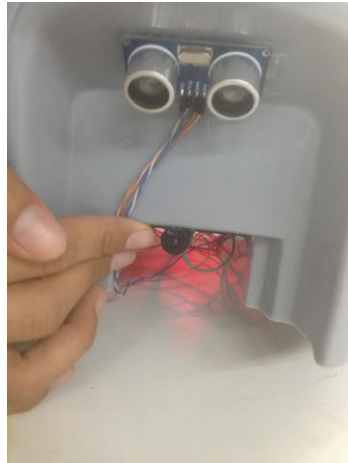
ภาพที่ 3.4.2 ระบบการทำงาน Servo ขณะรับสัญญาณ

3.4.3 การควบคุม Buzzer

ในการควบคุม Buzzer ให้ Sensor เป็นเหมือน Switch ในการทำให้ Buzzer ทำงาน โดย Sensor วัดระยะได้ตามที่ต้องการ จะส่งสัญญาณให้ Arduino ส่งคำสั่งให้ Buzzer ส่งเสียงออกมา

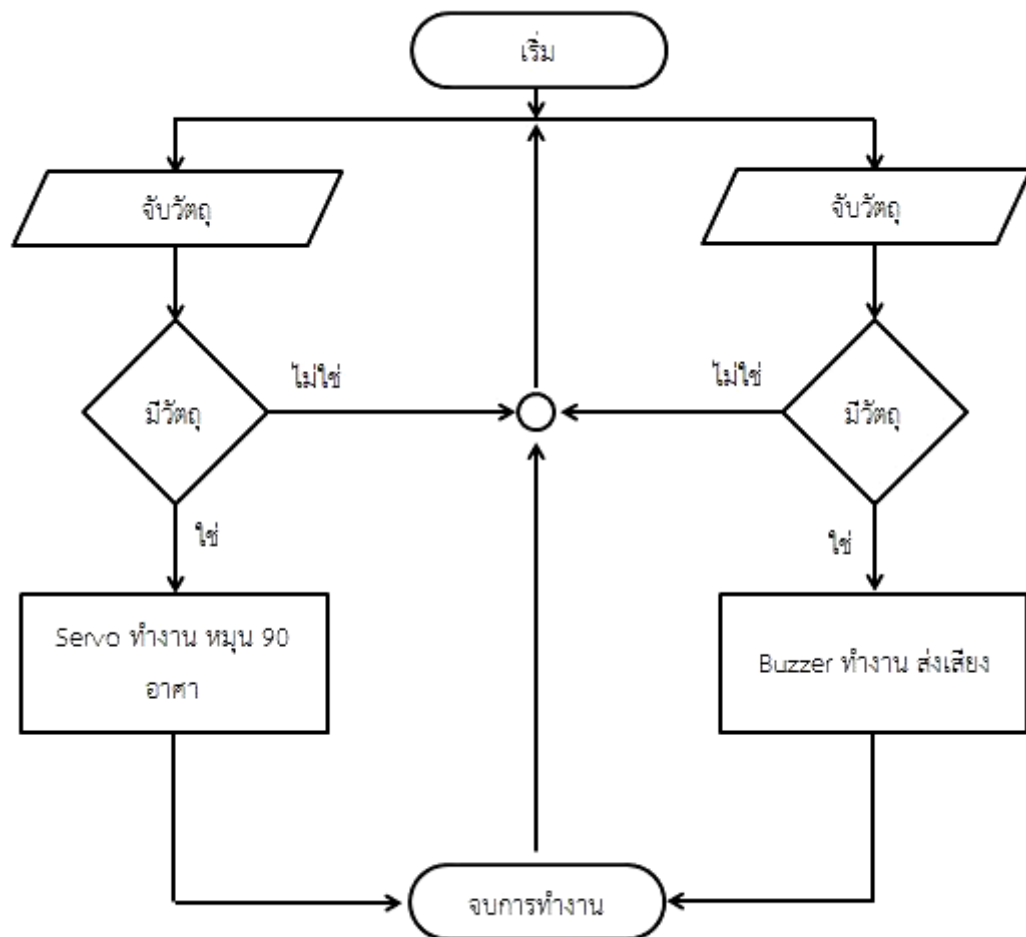


ภาพที่ 3.4.3 ระบบการทำงาน Sensor ขณะวัดระยะวัตถุ



ภาพที่ 3.4.3 ระบบการทำงาน Buzzer ขณะรับสัญญาณมาจากArduino

3.5 Flowchart



ภาพที่ 3.5 ตัวอย่างผังระบบการทำงานของถังขยะ

3.6 โปรแกรมควบคุมการทำงาน

```
#include <Servo.h>
#define trigPin 7
#define echoPin 6
#define trigPin1 5
#define echoPin1 4
int speakerPin = 10;
int length = 28; // the number of notes
char notes[] = "GGAGcB GGAGdc GGxecBA yyecdc";
int beats[] = { 2, 2, 8, 8, 8, 16, 1, 2, 2, 8, 8,8, 16, 1, 2,2,8,8,8,16, 1,2,2,8,8,8,16 };
int tempo = 150;
void playTone(int tone, int duration) {
for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
}
}
void playNote(char note, int duration) {
char names[] = {'C', 'D', 'E', 'F', 'G', 'A', 'B',
                'c', 'd', 'e', 'f', 'g', 'a', 'b',
                'x', 'y' };
int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
                956, 834, 765, 593, 468, 346, 224,
                655 , 715 };
int SPEE = 5;
for (int i = 0; i < 17; i++) {
    if (names[i] == note) {
        int newduration = duration/SPEE;
        playTone(tones[i], newduration);
    }
}
}
Servo servo;
void setup()
{
    Serial.begin (9600);
```

```

pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(speakerPin, OUTPUT);
servo.attach(9);
}
void loop()
{
  unsigned long showTime=millis();
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
  long duration1, distance1;
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1 / 2) / 29.1;
  if (distance < 6)
  {
    servo.write(90);
    delay(500);
  }
  else
  {
    servo.write(0);
  }
  if (distance >6)
  {
  }
}

```

```
else
{
    Serial.print(distance);
    Serial.println("cm");
}
if (distance1<=2)
{
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        }
        else {
            playNote(notes[i], beats[i] * tempo);
        }
        delay(tempo);
    }
}

}
else
{
    Serial.print(distance1);
    Serial.println("cm1");
}
delay(50);
}
```

บทที่ 4

ผลการทดลอง

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

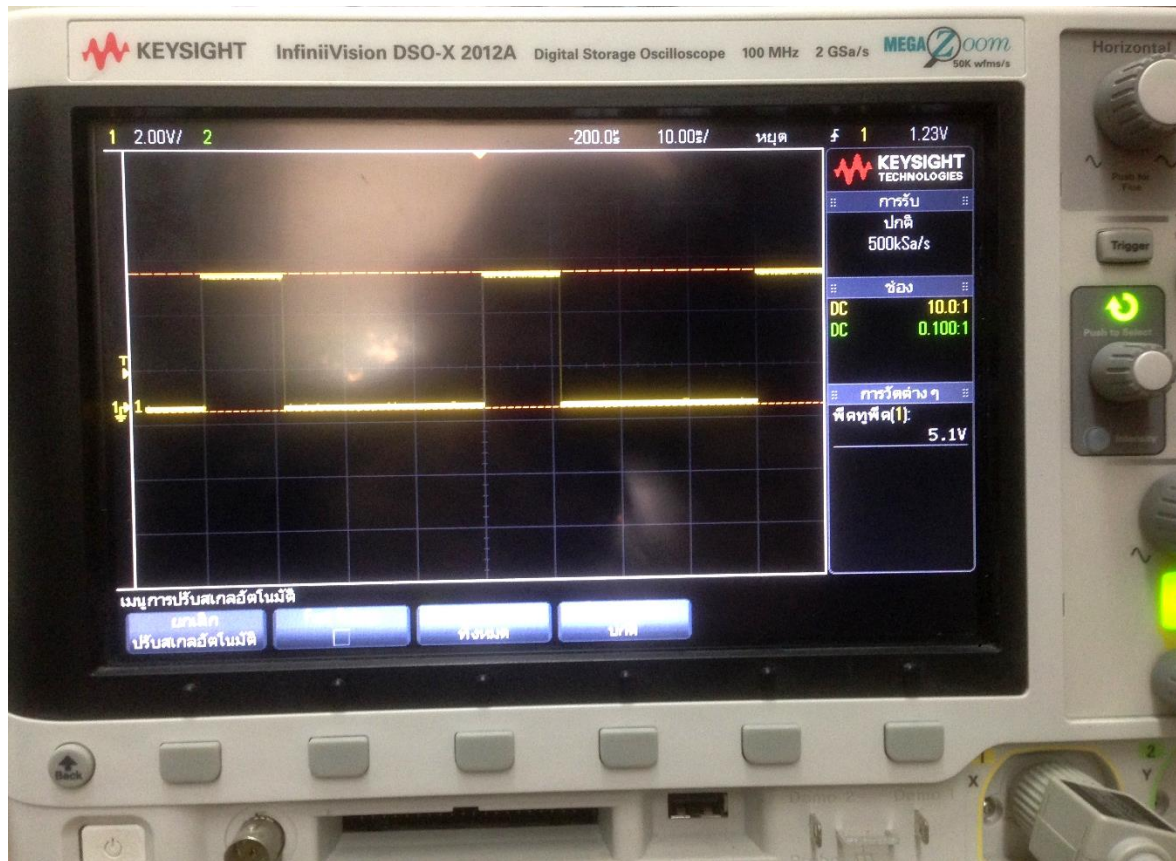


ภาพที่ 4 แสดงการทดสอบระบบ

4.1 การทดลองตรวจจ็ับของ Sensor

ทดสอบ Sensor โดยการนำวัตถุเข้าใกล้เพื่อทำการตรวจจ็ับระยะของSensor ต่อจากนั้นตัวควบคุม (Arduino Mega 2560 R3) จะทำการอ่านค่าตามที่เขียนโปรแกรมไว้

การทดสอบ Sensor

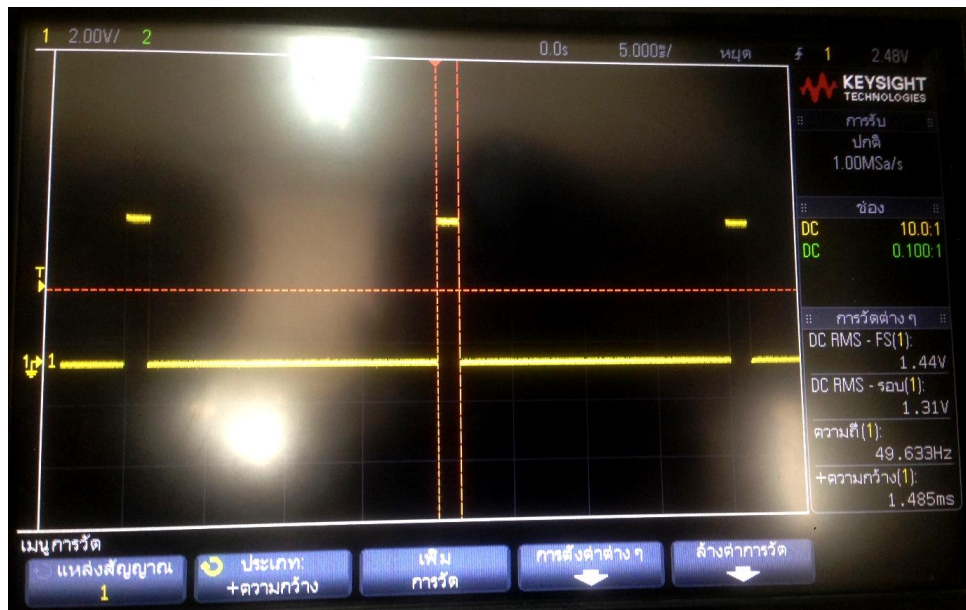


ภาพที่ 4.1 สัญญาณจาก Sensor

4.2 การทดลองสัญญาณเข้า Servo

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

สัญญาณที่เข้า Servo



ภาพที่ 4.2 สัญญาณจาก Servo

4.3 การทดลองการทำงานของ Buzzer

ในการทดลองใช้ Sensor จับวัตถุส่งสัญญาณให้ Arduino อ่านค่าและส่งสัญญาณให้ Buzzer ทำงานเมื่อมีวัตถุอยู่ในระยะที่ตั้งไว้



ภาพที่ 4.3 สัญญาณจาก Buzzer

4.4 การทดลองตรวจจับวัตถุแล้วเปิด-ปิดอัตโนมัติ

ในการทดลองใช้ Sensor ส่งสัญญาณออกไปยัง Arduino แล้วส่งค่าที่เขียนในโปรแกรมทำให้ Servo ทำงานโดยหมุนไปมุม 90 องศา ความกว้างของสัญญาณพัลส์ที่ 1.4 ms อีกการทดลองที่ใช้ Sensor ส่งสัญญาณออกไปยัง Arduino ทดลองการสั่งงานให้ Buzzer ดัง ระบบจะหยุดทำงานเมื่อไม่มีวัตถุอยู่ในระยะที่กำหนด จากการทดลองผลที่ได้สามารถจับระยะได้ Servo หมุนตามองศาที่ต้องการและมีเสียงเตือน

บทที่ 5

สรุปปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทำโครงการ

โครงการนี้เป็นระบบเปิด-ปิดของกลไกให้ตรวจจับระยะที่กำหนดโดยทำการควบคุมสั่งการด้วยตัวควบคุม Arduino Mega 2560 ควบคุมการทำงานด้วย Sensor และควบคุมการเปิด-ปิดด้วย Servo โดยจ่ายไฟ 5 Vdc ให้กับตัวควบคุมในการทำให้ทำงาน ตามลำดับ

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

Servo สามารถทำงานได้โดยรับสัญญาณจาก Sensor ที่ส่งข้อมูลตามคำสั่งที่เขียนไว้ใน Arduino mega 2560 ให้เปิดปิดตามคำสั่งของระบบตามต้องการ

5.2 ปัญหาวิธีการแก้ไข

จากการทดลอง

ปัญหาคือ

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์จะต้องใช้เซ็นเซอร์ไวตรงไหน เพื่อไม่ให้พวกสุนัขหรือแมว เดินผ่านแล้วถึงระยะเปิดขึ้นมา จึงเป็นสาเหตุที่ทำให้ถึงระยะเปิด-ปิด

การแก้ไขโดยการติดตั้งเซ็นเซอร์ไว้ที่ช่องด้านล่างทำมุมตั้งฉากลงมายังพื้น ใช้เซ็นเซอร์ ระยะห่าง 5 cm

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ การเหยยะเท้าจะส่งผลกระทบต่อวงจรการทำงานต่าง ๆ อย่างไร

การแก้ปัญหา ภายในถึงระยะได้ออกแบบมาใส่ระยะโดยมีถุงใส่ระยะด้านในไว้แล้ว เราสามารถนำระยะไปทิ้งได้เลย

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ เกิดเต็มถึงระยะจะเปิดใหม่ เราจะรู้ยังงามันเต็ม

การแก้ปัญหา จะมีเซ็นเซอร์ตรวจจับระยะ ถ้าเกิดระยะเต็ม จะมีเสียงเตือน เราสามารถเปิดถึงด้วยตัวเองเพื่อเอารยะไปทิ้งได้เลย

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ ถ้าถึงระยะมีขนาดที่ใหญ่ขึ้น

การแก้ปัญหา ใช้เซอร์โวที่ใหญ่ขึ้น แรงขึ้น

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ เซอร์โวที่ต้องการที่ใช้ทำโมเดลมีแรงเคลื่อนเท่าไร

การแก้ปัญหา Servo SG5010 4 kg/cm

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ เหตุใดที่ใช้เซอร์โวตัวเดียว จากเดิม 2 ตัว แล้วใช้เพียง 1

ตัว

การแก้ไขปัญหามาจากการทดลอง การใช้เซอร์โวเพียงตัวเดียวมีแรงมากพอ สามารถยกฝารยะได้

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ การเขียนโค้ดยังไม่สมบูรณ์

การแก้ไขปัญหามา ปรับปรุงแก้ไขเพิ่มเติมโดยการค้นคว้าหาความรู้เพิ่มเติม

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ วัตถุไม่ตั้งฉาก เซ็นเซอร์จะทำงาน

การแก้ไขปัญหามา ทำงาน แต่การทำงานจะไม่สมบูรณ์ เช่น ถึงระยะจะกึ่งเปิดๆปิดๆ

ถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ การสร้างเครื่องนี้ขึ้นมาใช้ จะคุ้มทุนไหม ที่เราจะนำมาใช้ ควรแก้ปัญหาน้อย่างไร

การแก้ไขปัญหามา เนื่องจากถึงระยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์ นี้ เป็นเพียงโมเดลที่

สร้างขึ้น การที่จะนำไปใช้จริง เช่น ใช้ในถังขยะที่ใหญ่ขึ้น เราจะใช้เซอร์โวลต์ที่ใหญ่ขึ้น แรงขึ้น ติดเซนเซอร์ไว้
ที่ๆเหมาะสม อีกที่

5.3 ข้อเสนอแนะ

ถ้าจะนำถึงขยะไปใช้งานจริงต้องมีการพัฒนาให้มีความสามารถป้องกันน้ำความชื้นได้เพื่ออุปกรณ์ของเราจะได้ไม่เสียหาย

ภาคผนวก ก.

โปรแกรมระบบถังขยะเปิดปิดอัตโนมัติด้วยระบบเซ็นเซอร์

โปรแกรมควบคุมการทำงาน

```
#include <Servo.h>
#define trigPin 7
#define echoPin 6
#define trigPin1 5
#define echoPin1 4
int speakerPin = 10;
int length = 28; // the number of notes
char notes[] = "GGAGcB GGAGdc GGxecBA yyecdc";
int beats[] = { 2, 2, 8, 8, 8, 16, 1, 2, 2, 8, 8,8, 16, 1, 2,2,8,8,8,16, 1,2,2,8,8,8,16 };
int tempo = 150;
void playTone(int tone, int duration) {
for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
}
}
void playNote(char note, int duration) {
char names[] = {'C', 'D', 'E', 'F', 'G', 'A', 'B',
                'c', 'd', 'e', 'f', 'g', 'a', 'b',
                'x', 'y' };
int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
                956, 834, 765, 593, 468, 346, 224,
                655 , 715 };
int SPEE = 5;
for (int i = 0; i < 17; i++) {
    if (names[i] == note) {
        int newduration = duration/SPEE;
        playTone(tones[i], newduration);
    }
}
}
```

```

Servo servo;
void setup()
{
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(speakerPin, OUTPUT);
  servo.attach(9);
}
void loop()
{
  unsigned long showTime=millis();
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
  long duration1, distance1;
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1 / 2) / 29.1;
  if (distance < 6)
  {
    servo.write(90);
    delay(500);
  }
  else
  {
    servo.write(0);
  }
}

```

```
}  
if (distance >6 )  
{  
}  
else  
{  
  Serial.print(distance);  
  Serial.println("cm");  
}  
if (distance1<=2)  
{  
  for (int i = 0; i < length; i++) {  
    if (notes[i] == ' ') {  
      delay(beats[i] * tempo); // rest  
    }  
    else {  
      playNote(notes[i], beats[i] * tempo);  
    }  
    delay(tempo);  
  }  
}  
else  
{  
  Serial.print(distance1);  
  Serial.println("cm1");  
}  
delay(50);  
}
```

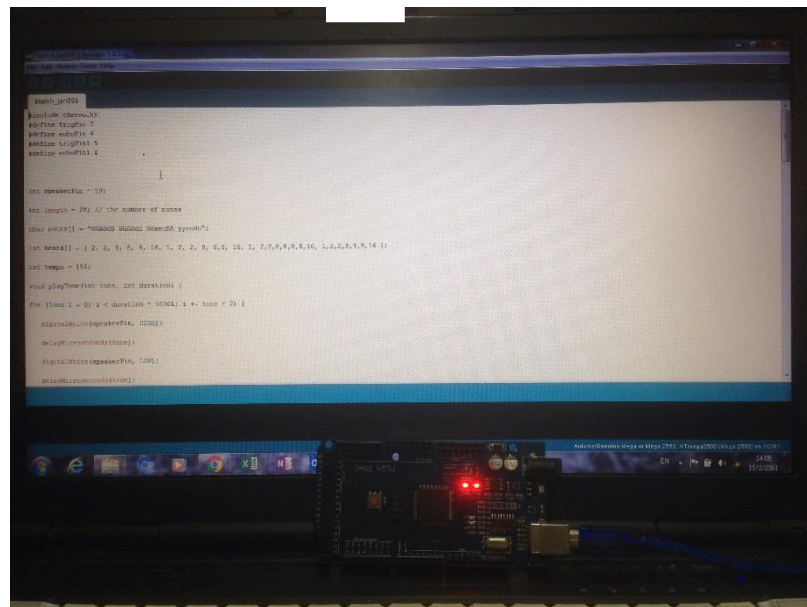

ภาคผนวก ข.

วิธีการประกอบแบบจำลองถังขยะเปิด-ปิดอัตโนมัติด้วยระบบเซ็นเซอร์

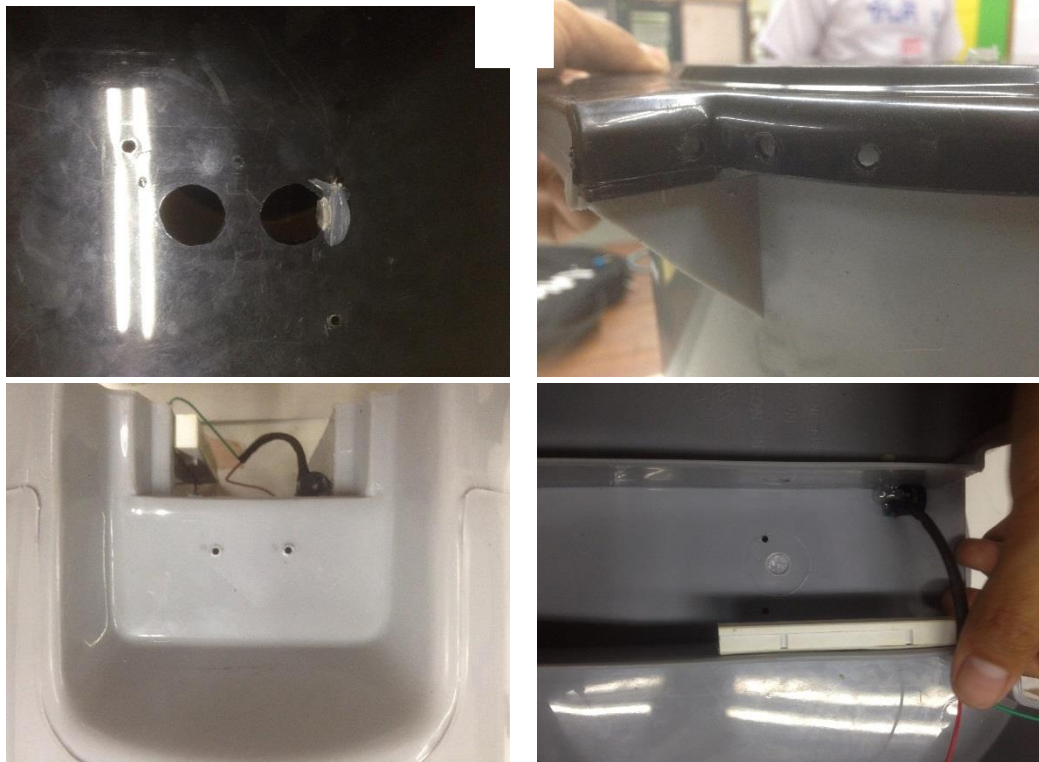
ชิ้นส่วนสำหรับการประกอบ



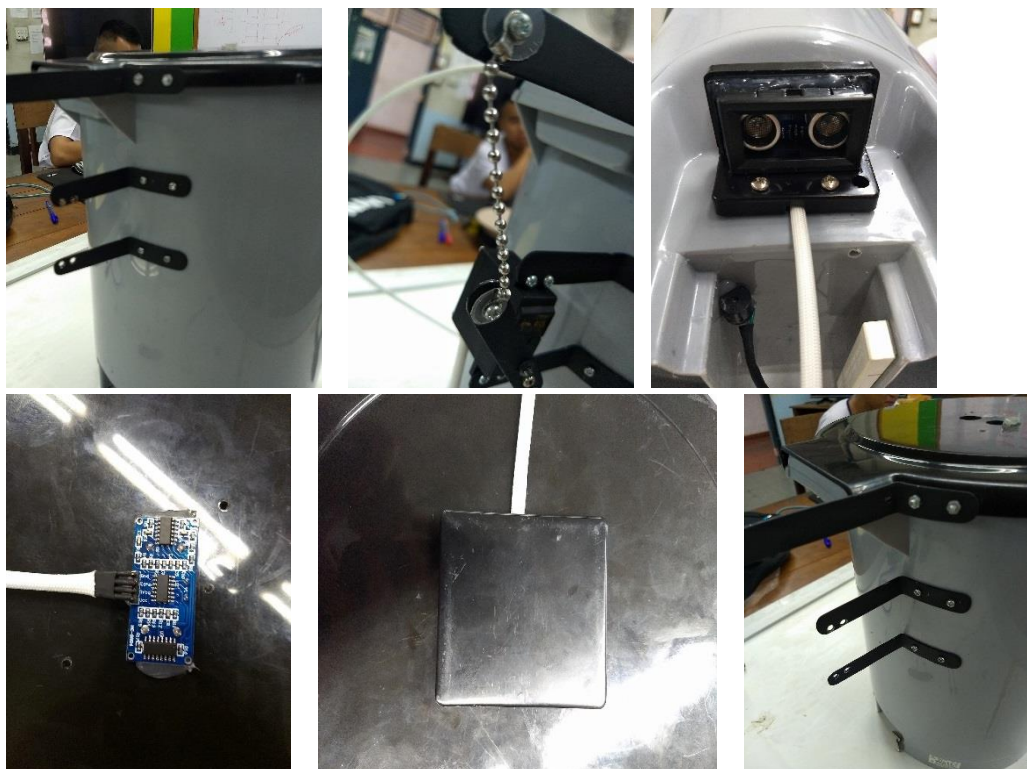
เขียนโปรแกรมลง Arduino



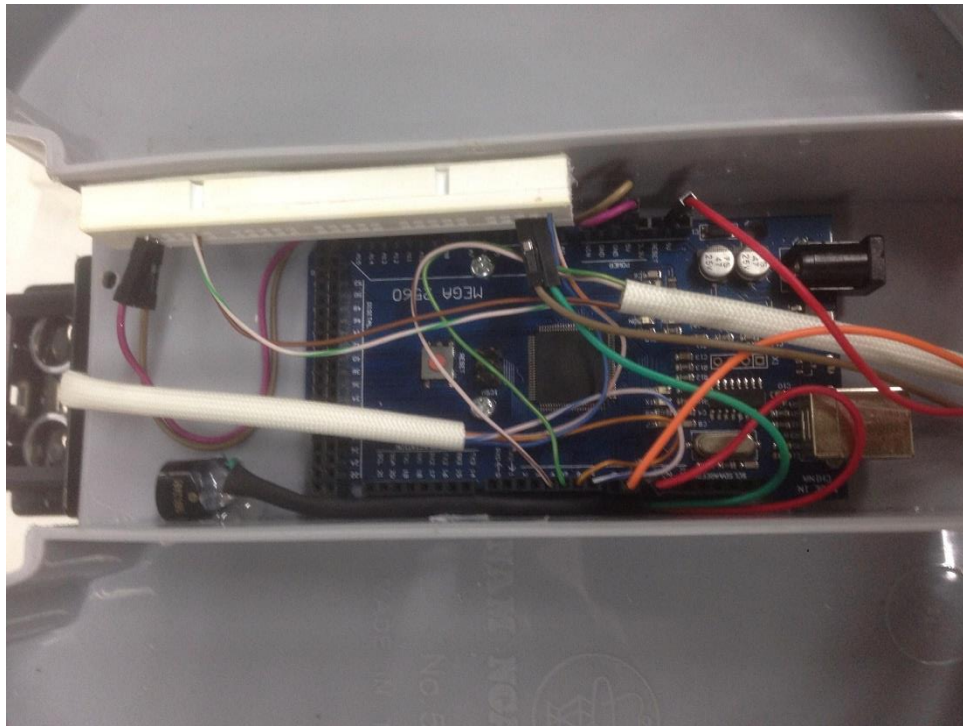
ประกอบเข้าที่ตามแผนก่อน



หลัง



ประกอบสาย input/output ให้กับอุปกรณ์ควบคุม และแหล่งจ่ายไฟ Arduino

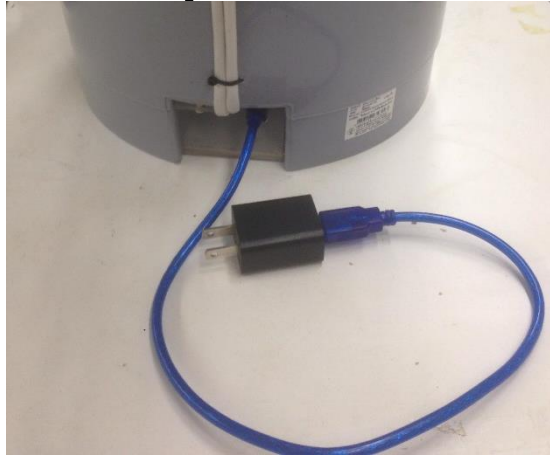


- พอร์ต 5V สาย Output จาก Arduino ส่งไปให้อุปกรณ์ทั้งหมด
- พอร์ต 10 สาย Output จาก Arduino ส่งไปให้กับ Buzzer เพื่อควบคุมการทำงานของ Buzzer
- พอร์ต 9 สาย Output จาก Arduino ส่งไปให้กับ Servo เพื่อควบคุมการทำงานของ Servo
- พอร์ต 7 สาย Output จาก Arduino ส่งไปให้กับ Sensor ตัวที่ 1 เพื่อส่งสัญญาณ Trig ความถี่ 40 Hz
- พอร์ต 6 สาย Input จาก Sensor ตัวที่ 1 ส่งไปให้กับ Arduino เพื่อรับสัญญาณ Echo
- พอร์ต 5 สาย Output จาก Arduino ส่งไปให้กับ Sensor ตัวที่ 2 เพื่อส่งสัญญาณ Trig ความถี่ 40 Hz
- พอร์ต 4 สาย Input จาก Sensor ตัวที่ 2 ส่งไปให้กับ Arduino เพื่อรับสัญญาณ Echo
- พอร์ต GND เป็นกราวด์ของอุปกรณ์ทั้งหมด

เมื่อทำเสร็จ



ภาคผนวก ค
คู่มือการใช้งาน



1. ใช้ไฟ 220 Vac 50 Hz ผ่านช่องรูเสียบ USB
2. ใช้แบตเตอรี่สำรอง (Power bank) ผ่านช่องเสียบ USB

ภาคผนวก ง.
ประวัติคณะผู้จัดทำโครงการ

1. นรจ.พงศธร ต่ายธานี เหล่า อีเล็กทรอนิกส์
ที่อยู่ 101/35 หมู่ 6 ตำบล พลุตาหลวง อำเภอ สัตหีบ จังหวัด ชลบุรี 20180
จบจาก โรงเรียนสิงห์สมุทร จังหวัด ชลบุรี
2. นรจ.จิรกิตติ คงเมืองกุลศรี เหล่า อีเล็กทรอนิกส์
ที่อยู่ 265 หมู่ 15 ตำบล หนองคอนไทย อำเภอ ภูเขียว จังหวัด ชัยภูมิ 36110
จบจาก โรงเรียนภูเขียว จังหวัด ชัยภูมิ
3. นรจ.สุรนาท สองนาม เหล่า อีเล็กทรอนิกส์
ที่อยู่ 24 หมู่ - ถนน บ้านพรุธานี ตำบล บ้านพรุ อำเภอ หาดใหญ่ จังหวัด สงขลา 90250
จบจาก โรงเรียนหาดใหญ่วิทยาลัย2 จังหวัด สงขลา
4. นรจ.อาชา ศรีโสภา เหล่า อีเล็กทรอนิกส์
ที่อยู่ 123/4 หมู่ 5 ตำบลพลั่ว อำเภอ แหม่มสิงห์ จังหวัดจันทบุรี 22190
จบจาก โรงเรียนแหลมสิงห์วิทยาคม จังหวัด จันทบุรี
5. นรจ.ขันติ เผ่าภูรี เหล่า อีเล็กทรอนิกส์
ที่อยู่ 58 หมู่ 2 ตำบล ดอนจิก อำเภอ พิบูลมังสาหาร จังหวัดอุบลราชธานี 34110
จบจาก โรงเรียนพิบูลมังสาหาร จังหวัด อุบลราชธานี

บรรณานุกรม

(1) <http://mbeddedweekly.blogspot.com/2014/08/arduino-mega2560.html>

(2) วันศุกร์ที่ 22 สิงหาคม พ.ศ. 2557

<http://www.myarduino.net/article/66/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B9%80%E0%B8%8B%E0%B8%99%E0%B9%80%E0%B8%8B%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B0%E0%B8%97%E0%B8%B2%E0%B8%87-hc-sr04-hc-sr05-us-100-us-016>

(3) <http://www.thaieasyelec.com/article-wiki/review-product-article/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%95%E0%B8%B1%E0%B8%A7%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%84%E0%B8%A7%E0%B8%9A%E0%B8%84%E0%B8%B8%E0%B8%A1-rc-servo-motor-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-arduino.html>

(4) <http://arduino.cc/en/reference/servo>

<http://eblcdc.com/?p=48>

<http://www.pololu.com/blog/13/gettin-all-up-in-your-servos>

<http://learn.parallax.com/KickStart/900-00008>

<https://www.sparkfun.com/tutorials/283>

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