หุ่นยนต์สำรวจ

OBSTACLE DETECTION FOR ROBOT'S APPLICATION

ณัฐนนท์ รัตนเรืองวัฒนา
วัศนัย ใจทน
ศุภณัฐ อินต๊ะผัด
สุรพศ เพ็ชรศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

หุ่นยนต์สำรวจ
OBSTACLE DETECTION FOR ROBOT'S APPLICATION

ณัฐนนท์ รัตนเรืองวัฒนา
วัศนัย ใจทน
ศุภณัฐ อินต๊ะผัด
สุรพศ เพ็ชรศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

OBSTACLE DETECTION FOR ROBOT'S APPLICATION

NATTANON RATTANARUANGWATTANA
VATSANAI JAITON
SUPANUT INTAPAD
SURAPOD PHETSRI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

ปริญญานิพนธ์ปีการศึกษา 2557
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	หุ่นยนต์สำรวจ OBSTACLE DETECTION FOR ROBOT'S APPLICATION		
นักศึกษาผู้จัดทำ	นายณัฐนนท์ รัตนเรืองวัฒนา	**รหัสนักศึกษา**	54010408
	นายวัศนัย ใจทน	**รหัสนักศึกษา**	54011184
	นายศุภณัฐ อินต๊ะผัด	**รหัสนักศึกษา**	54011290
	นายสุรพศ เพ็ชรศรี	**รหัสนักศึกษา**	54011425
ปริญญา	วิศวกรรมศาสตร์บัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2557		

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ ดร.พงษ์ชัย นิลาศ	29/05/58

หัวข้อปริญญานิพนธ์	หุ่นยนต์สำรวจ		
	OBSTACLE DETECTION FOR ROBOT'S APPLICATION		
นักศึกษาผู้จัดทำ	นายณัฐนนท์ รัตนเรืองวัฒนา	รหัสนักศึกษา	54010408
	นายวัศนัย ใจทน	รหัสนักศึกษา	54011184
	นายศุภณัฐ อินต๊ะผัด	รหัสนักศึกษา	54011290
	นายสุรพศ เพ็ชรศรี	รหัสนักศึกษา	54011425
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.พงษ์ชัย นิลาศ		
ปีการศึกษา	2557		

บทคัดย่อ

โครงงานนี้เป็นการศึกษาวิจัยและสร้างหุ่นยนต์หลบหลีกสิ่งกีดขวางเพื่อทำการสำรวจ โดยใช้เซนเซอร์ประเภทอัลตราโซนิคและอินฟราเรดเป็นตัวตรวจวัดและชี้นำทาง ขณะเดียวกันจะใช้ไมโครคอนโทลเลอร์ชนิด Arduino mega 2560 ในการประมวลผลและควบคุมการปฏิบัติการทั้งหมดของหุ่นยนต์

รูปแบบการตรวจจับสิ่งกีดขวางนั้นจะอ้างอิงจากเงื่อนไขที่ถูกสร้างขึ้นด้วยโปรแกรม ซึ่งหุ่นยนต์จะทำการกำหนดทิศทางเอง เมื่อมีสิ่งกีดขวางอยู่ตรงหน้า โดยที่เซนเซอร์อัลตราโซนิคและอินฟราเรดจะถูกใช้เป็นตัวตรวจจับและส่งคำสั่งในรูปสัญญาณไปยังไมโครคอนโทลเลอร์และเมื่อไมโครคอนโทลเลอร์ได้รับสัญญาณอินพุตก็จะทำการประมวลผลและสั่งการให้เกิดกลไกการเปลี่ยนทิศทางของหุ่นยนต์ผ่านมอเตอร์ นอกจากนั้นหุ่นยนต์ยังสามารถรับรู้และตอบสนองต่อมุมหรือทิศทางที่เปลี่ยนไปได้โดยใช้เซนเซอร์คอมพาส จากนั้นข้อมูลดังกล่าวจะถูกส่งไปยังคอมพิวเตอร์ผ่านระบบการส่งข้อมูลบลูทูธเพื่อทำการแสดงผลในรูปแบบแผนที่และใช้ในการแบ่งปันข้อมูล (knowledge sharing)

Thesis Title	OBSTACLE DETECTION FOR ROBOT'S APPLICATION
Authors	Mr.Nattanon Rattanaruangwattana
	Mr.Vatsanai Jaiton
	Mr.Supanut Intapad
	Mr.Surapod Phetsri
Thesis Advisor	Asst. Prof. Dr. Phongchai Nilas
Year	2014

ABSTRACT

This project is designed to research and build an obstacle avoidance robotic vehicle using ultrasonic and infrared sensors as its guidance while a microcontroller, Arduino mega 2560, is used to achieve the desired operation.

The obstacle detection scheme would be based on the intelligence that is built in it with programing language. It guides itself whenever an obstacle comes ahead of it. Ultrasonic and infrared sensors are used to detect any obstacle and send a command to the microcontroller through sensing signal. Depending on the input signal received, the microcontroller redirects the robot to move in an alternate direction by actuating the motors that interfaced to it through a motor driver circuit. Additionally, the robot can be perceived and responded to the changed angle by using a compass sensor then all the direction and angle data would be sent to computer via Bluetooth in order to map out the directions and used for knowledge sharing.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ผู้ควบคุมปริญญานิพนธ์ ผศ.ดร.พงษ์ชัย นิลาส เป็นอย่างสูง ที่ได้ให้คำปรึกษาและชี้นำแนวทางในการแก้ไขปัญหา ตลอดจนการเสริมสร้างและฝึกฝนให้คณะผู้จัดทำมีความรู้ความสามารถในการทำโครงงานชิ้นนี้ได้อย่างมีประสิทธิภาพ รวมถึงการดูแล เอาใจใส่และสอบถามถึงความก้าวหน้าอย่างสม่ำเสมอ ทำให้คณะผู้จัดทำสามารถทำงานได้อย่างเป็นระบบแบบแผนและมีประสิทธิภาพ

ขอขอบคุณภาควิชาวิศวกรรมการวัดคุมที่ได้เอื้อเฟื้ออุปกรณ์และสถานที่ให้คณะผู้จัดทำได้ใช้ในการทำปริญญานิพนธ์จนเสร็จสิ้น

ขอขอบคุณเพื่อนๆทุกๆคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ กระตุ้นเตือน รวมทั้งคอยสอบถามถึงความคืบหน้าของโครงงานอยู่เสมอ จนสามารถสำเร็จไปได้ด้วยดี

สุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว เป็นอย่างสูงสำหรับความรัก กำลังใจ คำปรึกษาและการสนับสนุนในด้านต่างๆ ที่มอบให้เสมอมา จนเกิดเป็นแรงผลักดันที่ทำให้โครงงานนี้ประสบความสำเร็จและผ่านลุล่วงมาได้อย่างสมบูรณ์

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญภาพ.....VIII

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1
1.3 ขอบเขตการศึกษา.....1
1.4 ขั้นตอนการศึกษา.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎี.....3

2.1 ความรู้พื้นฐานในการสร้างหุ่นยนต์.....3
2.1.1 ทฤษฎีเกี่ยวกับวิทยาการหุ่นยนต์.....3
2.1.2 ทฤษฎีการหลบหลีกสิ่งกีดขวางของหุ่นยนต์.....4
2.1.3 ประเภทของหุ่นยนต์.....5
2.2 ส่วนประกอบของหุ่นยนต์.....6
2.2.1 อุปกรณ์ทางกล แมคคานิค (mechanic).....7
2.2.2 อุปกรณ์ขับเร้า แอคชูเอเตอร์ (actuator).....7
2.2.3 อุปกรณ์ไฟฟ้า อิเลคทรอนิกส์ (electronic).....8
2.2.4 อุปกรณ์ควบคุม คอนโทรลเลอร์ (controller).....9
2.2.5 แหล่งจ่ายพลังงาน (Power Supply).....9
2.3 อุปกรณ์.....10
2.3.1 ไมโครคอนโทรเลอร์.....10
2.3.2 มอเตอร์ไฟฟ้ากระแสตรง.....11
2.3.3 อัลตราโซนิคเซนเซอร์.....15
2.3.4 อินฟราเรดเซนเซอร์.....18

2.3.5 เซนเซอร์บอกตำแหน่งและวัดมุมที่ทำการหมุน.....22
2.3.6 เซนเซอร์ตรวจวัดระยะการหมุนและการวัดความเร็วของมอเตอร์.....23
2.3.7 การสื่อสารผ่านบลูทูธ.....30

บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน.....36

3.1 ศึกษาข้อมูลเซนเซอร์ที่เลือกใช้.....39
3.1.1 อัลตราโซนิคเซนเซอร์.....39
3.1.2 อินฟราเรดเซนเซอร์.....40
3.1.3 Compass Module.....42
3.1.4 Encoder เพื่อทราบถึงระยะทางการเคลื่อนที่.....43
3.2 ทำ Sensing Profile.....44
3.3 ออกแบบโครงสร้างของหุ่นยนต์.....44
3.4 ศึกษาไมโครคอนโทรลเลอร์และโปรแกรมที่ใช้ควบคุม.....44
3.5 ออกแบบและสร้างลายวงจรควบคุม.....47
3.6 ศึกษาและเขียนโปรแกรม Visual Basic สำหรับ Mapping.....47
3.6.1 ศึกษาการสื่อสารระหว่าง Arduino กับ Visual Basic ผ่านพอร์ตอนุกรมและบลูทูธ.....47
3.6.2 ศึกษาการรับข้อมูลจากเซนเซอร์ผ่าน Arduino แล้วนำไปประมวลผลใน Visual Basic.....48
3.6.3 ศึกษาการเขียน Graphics เบื้องต้น ใน Visual basic เพื่อเขียน Mapping เส้นทางของหุ่นยนต์.....48
3.6.4 เขียนโปรแกรม Visual basic เพื่อสื่อสารและรับข้อมูลจาก Arduino จากนั้นนำไปประมวลผลและเขียน Mapping เส้นทางของหุ่นยนต์.....48
3.7 ประกอบชิ้นส่วนโครงสร้างหุ่นยนต์ พร้อมทำการทดลอง.....49
3.8 จัดทำวิทยานิพนธ์.....51

บทที่ 4 การทดลองและผลการทดลอง.....52

4.1 กล่าวนำ.....52
4.2 ขั้นตอนการทดลอง.....52
4.3 ผลการทดลอง.....58

บทที่ 5 บทสรุป.....64

5.1 สรุปผลการทดลอง.....64
5.2 สรุปผลโครงงาน.....68

บรรณานุกรม.....69

ภาคผนวก ก Source Code ที่ใช้ในการควบคุมหุ่นยนต์.....70
ภาคผนวก ข Source Code ที่ใช้ในการ Mapping.....94
ภาคผนวก ค Source Code Image Processing ใน MATLAB.....101
ภาคผนวก ง Sensing Profile.....102
ภาคผนวก จ วงจรลายปริ้น.....118

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงข้อมูลทางเทคนิคของ HMC5883L Module.....43

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างของหุ่นยนต์ เพลา และเฟือง....7

2.2 มอเตอร์ไฟฟ้า....7

2.3 อุปกรณ์ตรวจรู้อินฟราเรดและเอนโคดเดอร์....8

2.4 คอนโทรลเลอร์ควบคุมการทำงานของหุ่นยนต์....9

2.5 แบตเตอร์รี่อัลคาไลน์....9

2.6 Arduino mega 2560....10

2.7 มอเตอร์ไฟฟ้ากระแสตรง....11

2.8 วงจรภายในของมอเตอร์กระแสตรง....12

2.9 เฟรมของมอเตอร์....13

2.10 ขดลวดพันอยู่รอบขั้วแม่เหล็ก....13

2.11 โรเตอร์....14

2.12 อัลตราโซนิคเซนเซอร์....15

2.13 เปียโซอิเล็คทริค....16

2.14 แสดงตัวอย่างการเขียนสัญลักษณ์ของอุลตร้าโซนิคทรานสดิวเซอร์แบบต่างๆ กัน....17

2.15 ประเภทของคลื่นแม่เหล็กไฟฟ้า....18

2.16 แสดงการทำงานของ Infrared Sensor....20

2.17 แสดงวงจรอย่างง่ายของ Infrared Sensor....21

2.18 แสดงความสัมพันธ์ระหว่าง Output Voltage และ ระยะห่างของวัตถุ....21

2.19 แสดงตัวอย่างการต่อ IR Sensor กับบอร์ด Arduino....22

2.20 แสดงลักษณะของโพแทนทิออมิเตอร์แบบหมุนที่ใช้ในงานอุตสาหกรรม....23

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.21 แสดงลักษณะของโพแทนทิออมิเตอร์แบบหมุน....24

2.22 แสดงเอ็นโคเดอร์อย่างง่าย....25

2.23 แสดงจานBCD....25

2.24 แสดงโรตารีเอนโคเดอร์ชนิดแมกนีติก....26

2.25 แสดงวงจรการทำงานของโรตารีเอนโคเดอร์ชนิดแมกนีติก....27

2.26 แสดงส่วนประกอบของโรตารีเอนโคเดอร์ชนิดออฟติคคัล....27

2.27 แสดงโรตารีเอนโคเดอร์ชนิดออฟติคคัล....28

2.28 แสดงหลักการทำงานของแทคโคเจเนอเรเตอร์....28

2.29 แสดงการทำงานของแทคโคเจเนอเรเตอร์....29

2.30 แทคโคเจเนอเรเตอร์....29

2.31 แสดงการวัดโดยใช้แทคโคเจเนอเรเตอร์....30

2.32 แสดงส่วนประกอบชุดข้อมูลของบลูทูธ....31

2.33 แสดงรูปแบบการแลกเปลี่ยนข้อมูล....33

2.34 การส่งข้อมูลผ่านคลื่นวิทยุโดยเปลี่ยนความถี่ไปมา....34

2.35 แบบการส่งสัญญาณค้นหาอุปกรณ์....34

2.36 รูปแบบการสื่อสาร....35

3.1 Flow Chart แสดงการทำงาน....36

3.2 Flow Chart แสดงการทำงาน (ต่อ)....37

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.3 Flow Chart แสดงการทำงาน (ต่อ).....38

3.4 ภาพอุปกรณ์ HC-SR04.....39

3.5 แสดง GP2D120 Optoelectronic Device.....40

3.6 แสดงข้อมูลทางเทคนิคของ GP2D120 Optoelectronic Device.....41

3.7 อุปกรณ์ HMC5883L Module.....42

3.8 แสดงข้อมูลทางเทคนิกและคุณสมบัติของ HMC5883L Module.....42

3.9 DC Motor Encoder.....43

3.10 Flow Chart แสดงการทำงานของโปรแกรมควบคุมการเคลื่อนที่ใน Arduino IDE.....45

3.11 Flow Chart แสดงการทำงานของ Avoidance Control loop.....46

3.12 Flow Chart แสดงการทำงานของ Avoidance Control loop (ต่อ).....47

3.13 แสดง HC-05 Bluetooth Module.....48

3.14 แสดง User Interface ของโปรแกรม Visual Basic ที่ทำการออกแบบ.....49

3.15 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านบน).....50

3.16 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านหน้า).....50

3.17 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านข้าง).....50

3.18 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านหลัง).....51

4.1 แสดงสภาพแวดล้อมจำลอง แบบ ก.....52

4.2 แสดงการวางหุ่นยนต์.....53

4.3 แสดง User Interface ของโปรแกรม Mapping ขณะกด Connect.....53

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.4 แสดงสถานะของ Bluetooth ที่ยังไม่มีการเชื่อมต่อกับโปรแกรม....54

4.5 แสดงสถานะของ Bluetooth ที่เชื่อมต่อกับโปรแกรมแล้ว....54

4.6 แสดง User Interface ของโปรแกรม Mapping ขณะกด Start....54

4.7 แสดงการเคลื่อนที่สำรวจสภาพแวดล้อมจาลองของตัวหุ่นยนต์....55

4.8 แสดง Map ที่ได้หลังจากทำการสำรวจแล้ว....55

4.9 แสดงภาพ Map ที่ได้เป็นไฟล์ .png....56

4.10 แสดงการ Disconnect เพื่อตัดการเชื่อมต่อระหว่างหุ่นยนต์และโปรแกรม Mapping....56

4.11 แสดงสภาพแวดล้อมจำลอง แบบ ข....57

4.12 แสดงสภาพแวดล้อมจำลอง แบบ ค....57

4.13 แสดงสภาพแวดล้อมจำลอง แบบ ข....58

4.14 ผลการทดลองครั้งที่ 1....58

4.15 ผลการทดลองครั้งที่ 2....59

4.16 ผลการทดลองครั้งที่ 3....59

4.17 แสดงสภาพแวดล้อมจำลอง แบบ ข....60

4.18 ผลการทดลองครั้งที่ 1....60

4.19 ผลการทดลองครั้งที่ 2....61

4.20 ผลการทดลองครั้งที่ 3....61

4.21 แสดงสภาพแวดล้อมจำลอง แบบ ค....62

4.22 ผลการทดลองครั้งที่ 1....62

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.23 ผลการทดลองครั้งที่ 2.....63

4.24 ผลการทดลองครั้งที่ 3.....63

5.1 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 1 และผลการสำรวจ.....64

5.2 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 2 และผลการสำรวจ.....64

5.3 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 3 และผลการสำรวจ.....65

5.4 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 1.....65

5.5 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 2.....66

5.6 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 3.....66

5.7 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 1 และผลการสำรวจ.....67

5.8 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 2 และผลการสำรวจ.....67

5.9 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 3 และผลการสำรวจ.....68

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในโลกแห่งยุคโลกาภิวัตน์ได้ก่อให้เกิดการพัฒนาและเปลี่ยนแปลงในทุกๆด้านอย่างรวดเร็ว ซึ่งก็ส่งผลให้เกิดนวัตกรรมใหม่ๆและความก้าวหน้าขึ้นมากมาย ไม่ว่าจะเป็นระบบการสื่อสาร การคมนาคมขนส่ง รวมถึงเทคโนโลยีที่มีการพัฒนาแบบไม่เคยหยุดนิ่ง ซึ่งหนึ่งในเทคโนโลยีที่มีความก้าวหน้ามากที่สุดคือเทคโนโลยีหุ่นยนต์ ที่ถูกคิดค้นและสร้างขึ้นมาเพื่อแก้ไขปัญหาพื้นฐานในการดำรงชีวิตหรือเพื่ออำนวยความสะดวกสบายให้กับมนุษย์ นอกจากนั้นยังมีการประยุกต์ใช้เทคโนโลยีหุ่นยนต์ในงานด้านอุตสาหกรรมต่างๆมากมาย จึงเห็นได้ว่าเทคโนโลยีหุ่นยนต์นั้นได้เข้ามามีบทบาทต่อมนุษย์อย่างปฏิเสธไม่ได้

โดยปริญญานิพนธ์นี้จัดทำขึ้นเพื่อศึกษาการนำเอาเซนเซอร์ที่ใช้ในการวัดระยะด้วยอัลตราโซนิคและอินฟราเรด มาทำการสร้างหุ่นยนต์เพื่อตรวจจับสิ่งกีดขวาง และควบคุมการเคลื่อนที่เพื่อหลบหลีกสิ่งกีดขวาง ตลอดจนวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ ซึ่งการศึกษาคุณสมบัติของเซนเซอร์และการนำเอาเซนเซอร์หลายๆตัวหลายชนิดมาประมวลผลรวมกัน ถือเป็นพื้นฐานที่สำคัญของ Multisensor Fusion เพื่อเพิ่มความถูกต้องและลดขีดจำกัดในการตรวจวัด ซึ่งเป็นเทคโนโลยีที่มีการใช้งานกันอย่างกว้างขวางทั้งในภาคอุตสาหกรรมและในชีวิตประจำวัน จึงทำให้เกิดเป็นโครงงานการสร้างหุ่นยนต์สำรวจนี้ขึ้น

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการนำเอาเซนเซอร์ที่ใช้ในการวัดระยะด้วยอัลตราโซนิคและอินฟราเรด มาใช้งานรวมกัน
2. เพื่อนำเซนเซอร์ต่างๆมาประยุกต์ใช้เพื่อตรวจจับสิ่งกีดขวาง
3. เพื่อศึกษาการควบคุมการเคลื่อนที่ของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวางจากผลการตรวจจับของเซนเซอร์
4. เพื่อศึกษาการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาการทำงานของเซนเซอร์แต่ละตัวที่ใช้ทำหุ่นยนต์
2. สามารถนำเอาเซนเซอร์หลายๆตัว หลายๆชนิด มาประมวลผลรวมกันได้และสามารถประยุกต์ใช้ในการตรวจวัดและควบคุมการเคลื่อนที่ของหุ่นยนต์ได้

3. ศึกษาการทำงานของหุ่นยนต์ที่มีความสามารถในการหลบหลีกสิ่งกีดขวางและทำการสำรวจได้
4. สามารถสร้างและประกอบหุ่นยนต์อย่างง่ายได้
5. สามารถวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ได้

1.4 ขั้นตอนการศึกษา

1. ศึกษาค้นคว้าหลักการทำงาน ลักษณะการตรวจจับ และข้อจำกัดของเซนเซอร์แต่ละตัว เพื่อให้สามารถเลือกใช้เซนเซอร์ได้อย่างเหมาะสมและตรงกับวัตถุประสงค์
2. ศึกษาและออกแบบโครงสร้างของหุ่นยนต์สำรวจ โดยคำนึงถึงขนาด รูปร่างของเซนเซอร์ที่เลือก และรูปแบบการหลบหลีกสิ่งกีดขวางของหุ่นยนต์
3. ศึกษาภาษาที่จะใช้ในการเขียนโปรแกรมเพื่อควบคุมการเคลื่อนที่ และสร้างแผนที่การเดินทางของหุ่นยนต์จากข้อมูลที่ได้รับจากเซนเซอร์
4. จัดทำโครงสร้างและประกอบหุ่นยนต์ เพื่อทำการศึกษาถึงปัจจัยและตัวแปรต่างๆที่ส่งผลกระทบต่อกระบวนการหลบหลีกสิ่งกีดขวางของหุ่นยนต์ โดยนำเอาผลลัพธ์ที่ได้มาสร้างเป็นหุ่นยนต์สำรวจที่ตรงกับวัตถุประสงค์มากที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

จากโครงงานการสร้างหุ่นยนต์ถือเป็นการศึกษาเรียนรู้เกี่ยวกับอุปกรณ์การวัดและควบคุมอย่างง่ายที่นำเอาทั้งทฤษฎีและหลักการวัดต่างๆมาประยุกต์สร้างเป็นชิ้นงาน ซึ่งองค์ความรู้ที่ได้จากการศึกษานั้นก็จะนำไปสู่การวิเคราะห์เพื่อแก้ปัญหาต่างๆ เพื่อนำไปสู่ผลลัพธ์ที่ต้องการ ซึ่งผลลัพธ์จากการทำโครงงานนี้ สามารถใช้เป็นแนวทางในการวางแผนปรับปรุงและพัฒนาต่อยอดในการสร้างหุ่นยนต์ในขั้นที่สูงขึ้นต่อๆไปได้ นอกจากนั้นยังเป็นการเรียนรู้ถึงพื้นฐานของกระบวนการคิดอย่างเป็นระบบ และเรียนรู้การทำงานร่วมกับผู้อื่น การรับฟังความคิดเห็นและรับผิดชอบในงานที่ได้รับมอบหมายจนเกิดผลสำเร็จ อีกทั้งยังเป็นการสร้างเสริมประสบการณ์ในการเรียนรู้และสร้างทักษะในการลงมือปฏิบัติงาน ซึ่งเป็นประโยชน์อย่างยิ่งต่อการพัฒนาตัวเองให้มีความก้าวหน้ายิ่งๆขึ้นไป

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานในการสร้างหุ่นยนต์

หุ่นยนต์ หรือ โรบอต (robot) คือเครื่องจักรกลชนิดหนึ่ง มีลักษณะโครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่าง ๆ ตามการควบคุมโดยตรงของมนุษย์ การควบคุมระบบต่าง ๆ ในการสั่งงานระหว่างหุ่นยนต์และมนุษย์ สามารถทำได้โดยทางอ้อมและอัตโนมัติ โดยทั่วไปหุ่นยนต์ถูกสร้างขึ้นเพื่อสำหรับงานที่มีความยากลำบากเช่น งานสำรวจในพื้นที่บริเวณแคบหรืองานสำรวจดวงจันทร์ดาวเคราะห์ที่ไม่มีสิ่งมีชีวิต ปัจจุบันเทคโนโลยีของหุ่นยนต์เจริญก้าวหน้าอย่างรวดเร็ว เริ่มเข้ามามีบทบาทกับชีวิตของมนุษย์ในด้านต่าง ๆ เช่น ด้านอุตสาหกรรมการผลิต แตกต่างจากเมื่อก่อนที่หุ่นยนต์มักถูกนำไปใช้ ในงานอุตสาหกรรมเป็นส่วนใหญ่ ปัจจุบันมีการนำหุ่นยนต์มาใช้งานมากขึ้น เช่น หุ่นยนต์ที่ใช้ในทางการแพทย์ หุ่นยนต์สำหรับงานสำรวจ หุ่นยนต์ที่ใช้งานในอวกาศ หรือแม้แต่หุ่นยนต์ที่ถูกสร้างขึ้นเพื่อเป็นเครื่องเล่นของมนุษย์ จนกระทั่งในปัจจุบันนี้ได้มีการพัฒนาให้หุ่นยนต์นั้นมีลักษณะที่คล้ายมนุษย์ เพื่อให้อาศัยอยู่ร่วมกันกับมนุษย์ ให้ได้ในชีวิตประจำวัน

2.1.1 ทฤษฎีเกี่ยวกับวิทยาการหุ่นยนต์

2.1.1.1 วิทยาการหุ่นยนต์

วิทยาการหุ่นยนต์เป็นสาขาของวิศวกรรมเครื่องกล วิศวกรรมอิเล็กทรอนิกส์ และวิทยาศาสตร์คอมพิวเตอร์ ที่เกี่ยวข้องกับการออกแบบ โครงสร้าง การปฏิบัติการและการประยุกต์ใช้งานหุ่นยนต์ รวมถึงระบบคอมพิวเตอร์สำหรับการควบคุม การตอบสนองต่อการตรวจวัด และการประมวลผลข้อมูล

2.1.1.2 หุ่นยนต์

หุ่นยนต์เป็นเครื่องจักรกลที่ถูกออกแบบมาเพื่อจัดการหรือทำหน้าที่บางอย่างด้วยความต่อเนื่องและซ้ำเดิม ซึ่งจะต้องมีความเร็ว ความแม่นยำ และความเที่ยงตรง โดยทั่วไปหุ่นยนต์มีหลากหลายชนิดประเภทตามลักษณะหน้าที่ของมัน
หุ่นยนต์จะมีลักษณะเฉพาะที่สำคัญ ดังนี้

1. **การตรวจวัด** หุ่นยนต์จะต้องมีความสามารถในการรับรู้ถึงสิ่งต่างๆที่อยู่รอบๆตัวมันได้ ซึ่งจะรับรู้ในลักษณะที่แตกต่างกับมนุษย์ จะสามารถทำได้โดยการให้เซนเซอร์กับหุ่นยนต์ เช่น ติดตั้งเซนเซอร์แสงเปรียบเสมือนการให้ดวงตา, ติดตั้งเซนเซอร์สัมผัสและเซนเซอร์ความ

ดันเปรียบเสมือนการให้มือ, ติดตั้งเซนเซอร์ทางเคมีเปรียบเสมือนการให้จมูก, ติดตั้งเซนเซอร์โซน่าเปรียบเสมือนการให้หู, ติดตั้งเซนเซอร์รับรสเปรียบเสมือนการให้ลิ้น ซึ่งจะช่วยให้หุ่นยนต์สามารถที่จะตระหนักรู้ถึงสภาพแวดล้อมที่อยู่รอบๆตัวมัน

2. **การเคลื่อนที่** หุ่นยนต์ต้องมีความสามารถในการเคลื่อนที่ได้ ไม่ว่าจะเป็นการหมุนกลิ้งด้วยล้อหรือเดินด้วยขา
3. **พลังงานขับเคลื่อน** หุ่นยนต์ต้องมีความสามารถในการให้พลังงานกับตัวมันเอง ซึ่งอาจจะเป็นพลังงานแสงอาทิตย์ พลังงานไฟฟ้า หรือจากแบตเตอรี่
4. **ความฉลาด** หุ่นยนต์จะต้องมีความสามารถในการประมวลที่ชาญฉลาดตามโปรแกรมที่ผู้ใช้ได้เขียนลงไปในส่วนควบคุม

2.1.2 ทฤษฎีการหลบหลีกสิ่งกีดขวางของหุ่นยนต์

การหลบหลีกสิ่งกีดขวางของหุ่นยนต์จะกล่าวถึงความสามารถของหุ่นยนต์ในการปรับเปลี่ยนเส้นทางการเดินอย่างอัตโนมัติ เพื่อที่จะตอบสนองต่อวัตถุที่กีดขวางหรือสภาพแวดล้อมที่อยู่รอบๆตัวมัน ซึ่งสามารถทำได้โดยการควบคุมผ่านกระบวนการ 2 รูปแบบ คือ การควบคุมแบบพรอกซิมอล (Proximal Control or Reactive Control) และการควบคุมแบบดิสทอล (Distal Control or Rule-based Control) ซึ่งการควบคุมทั้งสองรูปแบบนั้นมีความแตกต่างกัน คือ การควบคุมแบบพรอกซิมอลนั้น เซนเซอร์ของตัวหุ่นยนต์จะถูกทำให้เชื่อมต่อกับการควบคุมมอเตอร์โดยตรง และความเร็วของมอเตอร์ก็จะแปรผันและตอบสนองต่อสัญญาณอินพุตของเซนเซอร์ ซึ่งทำได้โดยการสร้างเมตริกซ์ที่จะเปลี่ยนสัญญาณอินพุตจากเซนเซอร์ไปเป็นความเร็วมอเตอร์ หรือสามารถใช้การจัดเรียง (array) แบบสองมิติที่อยู่ในรูปแบบของแถวและคอลัมน์ โดยให้จำนวนแถวแทนจำนวนของเซนเซอร์ที่ใช้วัดระยะ และจำนวนคอลัมน์แทนจำนวนของมอเตอร์ ยกตัวอย่างเช่น ถ้าเราพิจารณาหุ่นยนต์ที่มีสองล้อและใช้เซนเซอร์แปดตัว ดังนั้นจะได้การจัดเรียงแบบแปดแถวสองคอลัมน์ ในกรณีนี้แต่ละแถวจะถูกคิดเป็นตัวกำหนดและควบคุมความเร็วมอเตอร์ทั้งข้างซ้ายและขวา และจะถูกใช้เป็นอัลกอริทึมในการหลบหลีกสิ่งกีดขวาง ซึ่งจะได้มาจากการคำนวณ เช่น การควบคุม Braitenberg จะทำตามอัลกอริทึมดังนี้

```
speed[i] += matrix[j][i] * (1.0 – (sensorsValue[j] / 512));
```

เมื่อ speed[] เป็นตัวกำหนดความเร็วของมอเตอร์ทั้งซ้ายและขวา, matrix[] เป็นค่าในเมตริกซ์, sensorValue[] เป็นค่าของเซนเซอร์, ค่า 512 เป็นค่าปกติมาตรฐานของเซนเซอร์ และ I,J เป็นค่าจากลูปที่ใช้แทนค่าของเซนเซอร์และเมตริกซ์

ส่วนการควบคุมแบบดิสทอลนั้นค่าของเซนเซอร์จะถูกส่งไปเข้ากฎหรือเงื่อนไขก่อนแล้วค่อยมีผลในการกำหนดความเร็วของมอเตอร์ ยกตัวอย่างเช่น

```
if (distance[6]+distance[7] > 1800 || distance[0]+distance[1] > 1800) {
```

ให้ทำอะไรสักอย่าง เช่น ตั้งให้เกิดความเร็วมอเตอร์ที่ทำให้เกิดการหมุนตัวของหุ่นยนต์ ในบางการควบคุมอาจจะมีเหตุการณ์ที่ทำให้หุ่นยนต์เกิดการสับสน เช่น การเคลื่อนที่เข้าไปในบริเวณที่แคบหุ่นยนต์มักจะหันซ้ายหันขวาในเวลาเดียวกัน หรือการเคลื่อนที่ไปในทางเดินยาวรูปตัวยู หุ่นยนต์อาจจะเสียเวลากับการวิ่งจากฝั่งหนึ่งไปยังอีกฝั่งหนึ่ง ซึ่งมีหนทางเดียวที่จะทำให้ตัวควบคุมมีความแข็งแกร่ง เที่ยงตรง แม่นยำ และผิดพลาดน้อยที่สุด คือ จะต้องเพิ่มสิ่งรบกวน (noise) ซึ่งอาจจะเป็นรูปแบบของการเบี่ยงเบนจากเงื่อนไข หรืออาจจะใช้สิ่งรบกวนที่เกิดขึ้นโดยธรรมชาติจากตัวเซนเซอร์ โดยทั่วไปแล้วเรามักจะคิดว่าสิ่งรบกวนจากเซนเซอร์นั้น จะต้องทำให้ลดน้อยลงมากที่สุดเท่าที่จะทำได้เพื่อให้เกิดความแม่นยำสูงสุด แต่ที่จริงแล้ว เราก็สามารถใช้สิ่งรบกวนเดียวกันนั้นในการปรับปรุงแก้ไขรูปแบบการหลบหลีกสิ่งกีดขวางหรืออัลกอริทึมให้ดีขึ้นได้เช่นเดียวกัน

2.1.3 ประเภทของหุ่นยนต์

ประเภทของหุ่นยนต์ สามารถแบ่งแยกได้หลากหลายรูปแบบตามลักษณะเฉพาะของการใช้งาน ได้แก่การแบ่งประเภทตามการเคลื่อนที่ นอกจากนี้อาจจำแนกตามรูปลักษณ์ภายนอกด้วยก็ได้

2.1.3.1 การแบ่งประเภทตามการเคลื่อนที่ได้

2.1.3.1.1 หุ่นยนต์ที่ติดตั้งอยู่กับที่ ไม่สามารถเคลื่อนที่ได้

หุ่นยนต์ที่ติดตั้งอยู่กับที่ สามารถเคลื่อนไหวไปมาแต่ไม่สามารถเคลื่อนที่ได้ หุ่นยนต์ในประเภทนี้ได้แก่ แขนกลของหุ่นยนต์ที่ใช้ในงานด้านอุตสาหกรรมต่าง ๆ เช่น งานด้านอุตสาหกรรมผลิตรถยนต์แขนกลของหุ่นยนต์ที่ใช้งานในด้านการแพทย์ เช่นแขนกลที่ใช้ในการผ่าตัด หุ่นยนต์ประเภทนี้จะมีลักษณะโครงสร้างที่ใหญ่โต เทอะทะและมีน้ำหนักมาก ใช้พลังงานให้สามารถเคลื่อนไหวได้จากแหล่งจ่ายพลังงานภายนอก และจะมีการกำหนดขอบเขตการเคลื่อนไหวของหุ่นยนต์เอาไว้ ทำให้หุ่นยนต์สามารถเคลื่อไหวไปมาได้ในเฉพาะที่ที่กำหนดเอาไว้เท่านั้น

2.1.3.1.2 หุ่นยนต์ที่สามารถเคลื่อนไหวและเคลื่อนที่ได้

หุ่นยนต์ที่สามารถเคลื่อนไหวร่างกายไปมาได้อย่างอิสระ หมายความถึงหุ่นยนต์ที่สามารถเคลื่อนย้ายตัวเองจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้อย่างอิสระ หรือมีการเคลื่อนที่ไปมาในสถานที่ต่าง ๆ เช่น หุ่นยนต์ที่ใช้ในการสำรวจดวงจันทร์ขององค์กรนาซ่า หุ่นยนต์สำรวจใต้ภิภพหรือหุ่นยนต์ที่ใช้ในการขนถ่ายสินค้า ซึ่งหุ่นยนต์ที่สามารถเคลื่อนไหวได้นี้

ถูกออกแบบลักษณะของโครงสร้างให้มีขนาดเล็กและมีระบบเคลื่อนที่ไปมา รวมทั้งมีแหล่งจ่ายพลังสำรองภายในร่างกายของตนเอง แตกต่างจากหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปมาได้ ซึ่งจะต้องมีแหล่งจ่ายพลังอยู่ภายนอกแหล่งจ่ายพลังสำรองภายในร่างกายของหุ่นยนต์ที่สามารถเคลื่อนไหวร่างกาย และสามารถเคลื่อนที่ไปมาได้นั้น โดยปกติแล้วจะถูกออกแบบลักษณะของโครงสร้างให้มีขนาดเล็กรวมทั้งมีปริมาณน้ำหนักไม่มาก เพื่อไม่ให้เป็นอุปกสรรคต่อการปฏิบัติงานของหุ่นยนต์หรืออุปสรรคในการเคลื่อนที่

2.1.3.2 การแบ่งประเภทตามลักษณะรูปร่างภายนอก

โดยทั่วไป หุ่นยนต์ยังถูกจำแนกตามลักษณะรูปลักษณ์ภายนอก และมีคำศัพท์เฉพาะเรียกต่างๆกันไป ได้แก่

- หุ่นยนต์ฮิวแมนนอยด์ (Humanoid Robot) เป็นลักษณหุ่นยนต์ที่เหมือนกับมนุษย์
- แอนดรอยด์ (Android) เป็นการเรียกหุ่นยนต์คล้ายมนุษย์ที่สามารถแสดงออกเหมือนมนุษย์ แม้ว่ารากศัพท์ภาษากรีกของคำนี้หมายถึงเพศชาย แต่การใช้ในบริบทภาษาอังกฤษมักไม่ได้มีความหมายเจาะจงว่าเป็นเพศใด
- จีนอยด์ (Gynoid) เป็นการเรียกหุ่นยนต์คล้ายมนุษย์เพศหญิง
- แอ็คทรอยด์ (Actriod) เป็นหุ่นยนต์ที่เลียนแบบพฤติกรรมมนุษย์ เช่น กระพริบตา หายใจ เริ่มพัฒนาโดย มหาวิทยาลัยโอซาก้าและบริษัทโคโคโระ
- ไซบอร์ก (Cyborg) เป็นหุ่นยนต์ที่เชื่อมต่อกับสิ่งมีชีวิต หรือ ครึ่งคนครึ่งหุ่น เริ่มปรากฏครั้งแรกในเรื่องแต่งปี 1960
- นาโนโรบอท (Nano robot) เป็นหุ่นยนต์ขนาดเล็กมาก ขนาดประมาณ 0.5-3 ไมครอน

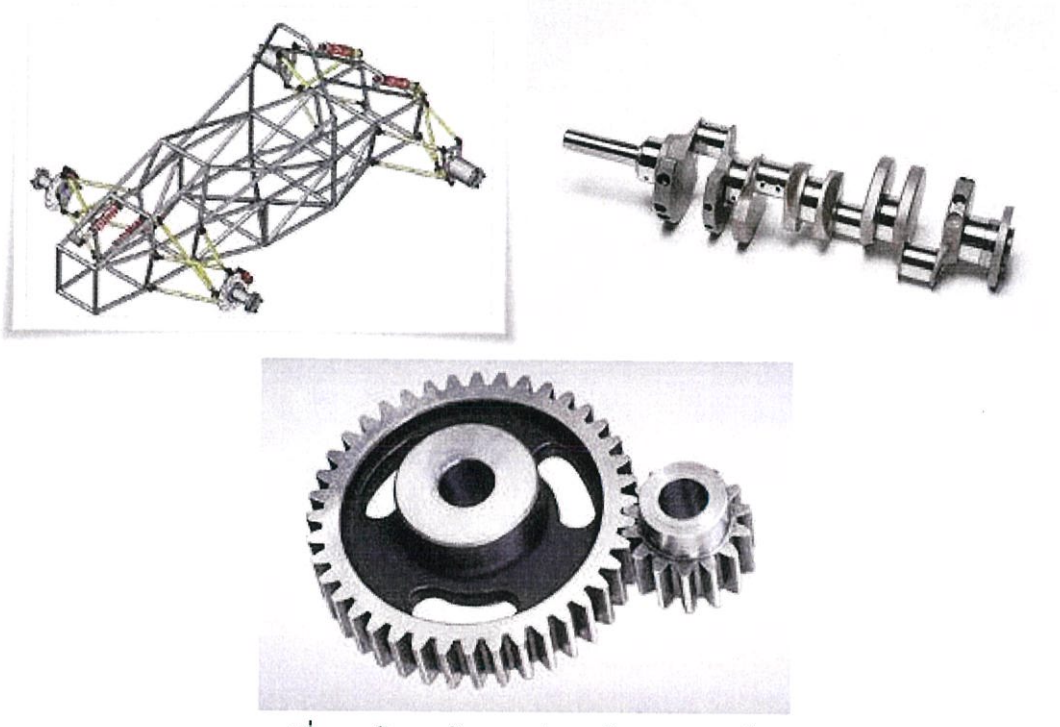
2.2 ส่วนประกอบของหุ่นยนต์

ในหุ่นยนต์หนึ่งตัวจะประกอบด้วยอุปกรณ์และชิ้นส่วนต่างๆมากมาย ซึ่งอุปกรณ์แต่ละชนิดนั้นจะมีหน้าที่แตกต่างกันไป ตามลักษณะและวัตถุประสงค์ของการใช้งาน การเลือกใช้จึงจำเป็นต้องอาศัยความรู้ความเข้าใจรวมถึงความเหมาะสม เพื่อให้หุ่นยนต์สามารถทำงานได้อย่างมีประสิทธิภาพ รวดเร็ว คงทน และประหยัดพลังงาน หุ่นยนต์จะแบ่งส่วนประกอบใหญ่ๆ เป็น 4 ส่วน ได้แก่

- อุปกรณ์ทางกล แมคคานิค (mechanic)
- อุปกรณ์ขับเร้า แอคชูเอเตอร์ (actuator)
- อุปกรณ์ไฟฟ้า อิเลคทรอนิกส์ (electronic)
- อุปกรณ์ควบคุม คอนโทรลเลอร์ (controller)
- แหล่งพลังงาน (Power Supply)

2.2.1 อุปกรณ์ทางกล แมคคานิค (mechanic)

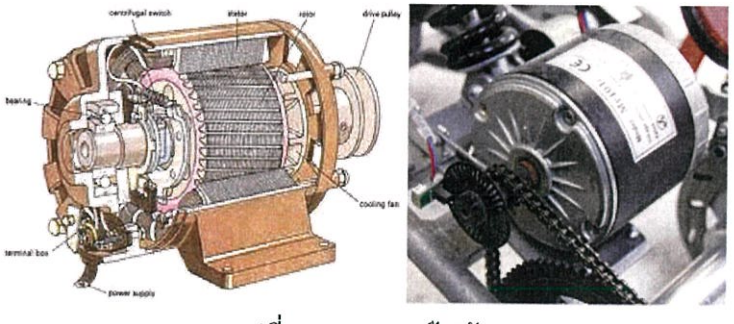
คือ ชิ้นส่วนกลไกต่างๆของหุ่นยนต์ เช่น โครงสร้าง เพลา เฟือง สกรูส่งกำลัง สายพาน โซ่ สปริง ข้อต่อสวมเพลา คลัตช์ เบรก ข้อต่อ ก้านต่อโยง ตลับลูกปืนและปลอกสวม



รูปที่ 2.1 โครงสร้างของหุ่นยนต์ เพลา และเฟือง

2.2.2 อุปกรณ์ขับเร้า แอคชูเอเตอร์ (actuator)

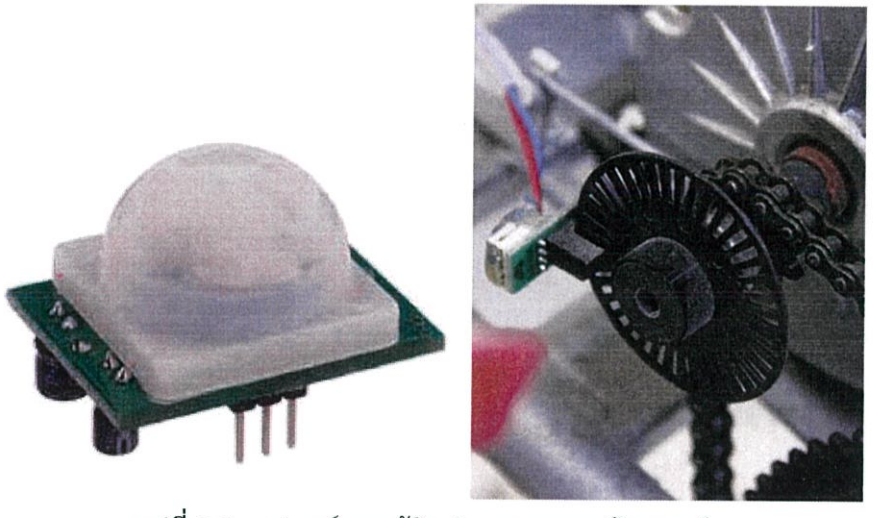
คือ อุปกรณ์ที่สามารถเปลี่ยนแปลงพลังงานไฟฟ้าที่ป้อนเข้าให้กลายเป็นการกระจัด การเคลื่อนที่ หรือแรง เช่น มอเตอร์ไฟฟ้า ระบบนิวแมติกส์ และระบบไฮโดรลิกส์



รูปที่ 2.2 มอเตอร์ไฟฟ้า

2.2.3 อุปกรณ์ไฟฟ้า อิเลคทรอนิกส์ (electronic)

คือ อุปกรณ์ที่ใช้สัญญาณทางระบบไฟฟ้า เช่น อุปกรณ์ตรวจรู้ วงจรขับต่างๆ และอุปกรณ์แสดงผลอุปกรณ์ตรวจรู้ใช้สำหรับตรวจวัดปริมาณของตัวแปรต่างๆ ใช้ในการรับค่า (input) ปริมาณทางฟิสิกส์ (physic) เช่น แสง สี อุณหภูมิ เสียง แรง ความดัน ความหนาแน่น ระยะทาง ความเร็ว อัตราเร่ง ระดับความสูง และอัตราการไหลเป็นต้น แล้วแปลงปริมาณทางฟิสิกส์ที่ได้เป็นสัญญาณทางไฟฟ้า หรือปริมาณการวัดในรูปแบบที่สามารถนำไปประมวลผลต่อได้

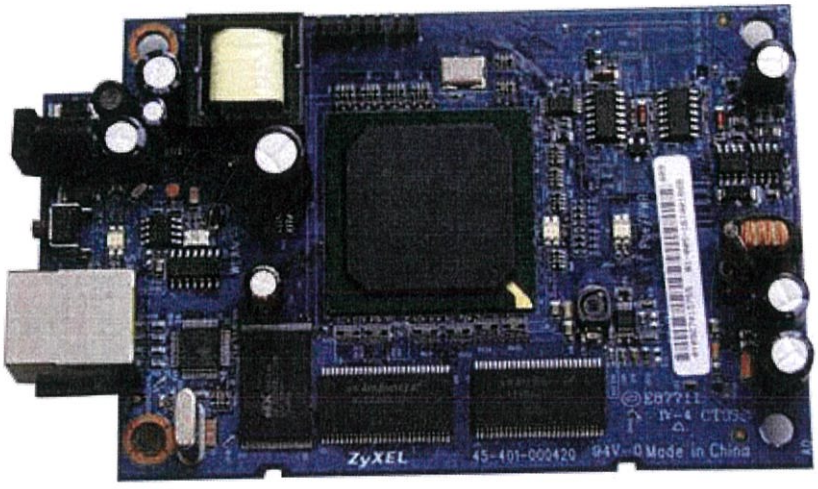


รูปที่ 2.3 อุปกรณ์ตรวจรู้อินฟราเรดและเอนโคดเดอร์

โดยเฉพาะอุปกรณ์ตรวจรู้ เป็นส่วนที่สำคัญในการทำงานของหุ่นยนต์ เปรียบเสมือนกับประสาทสัมผัสในการทำงานของมนุษย์ เช่น อุปกรณ์ตรวจรู้แสงที่ทำหน้าเหมือนตา โดยเปลี่ยน แสง สี ที่รับเข้ามาเป็นสัญญาณไฟฟ้า และส่งต่อให้ระบบประมวลผล อุปกรณ์ตรวจรู้มีมากมายหลายชนิด ตามสิ่งที่จะทำการตรวจวัด เช่น อุปกรณ์ตรวจรู้วัดตำแหน่ง (position sensor) อุปกรณ์ตรวจรู้วัดความเร็ว (velocity sensorอุปกรณ์ตรวจรู้วัดความเร่ง (acceleration sensor) อุปกรณ์ตรวจรู้วัดแรง (force sensor)อุปกรณ์ตรวจรู้วัดแรงบิด (torque sensor) อุปกรณ์ตรวจรู้อินฟราเรด (infrared sensor)ใช้บอกตำแหน่งโดยการสะท้อนของคลื่นแสงที่ความถี่ต่ำกว่าแสงสีแดง อุปกรณ์ตรวจรู้อัลตร้าโซนิค (ultrasonic sensor)ใช้บอกตำแหน่งโดยการสะท้อนของคลื่นเสียงที่มีความถี่สูง เลเซอร์เรนจ์ฟายเดอร์ (laser rangefinder sensor)ใช้ในการกะระยะนำทางโดยใช้แสงเลเซอร์และอุปกรณ์ตรวจรู้จีพีเอส(GPS: Global Position System) ใช้ในการระบุตำแหน่งโดยใช้การอ้างอิงจากดาวเทียมเป็นต้น

2.2.4 อุปกรณ์ควบคุม คอนโทรลเลอร์ (controller)

คือสมองกลที่ควบคุมการทำงานของหุ่นยนต์ เช่น สมองกลที่ประดิษฐ์จากอุปกรณ์อิเลคทรอนิกส์ เครื่องควบคุมขนาดเล็ก คอมพิวเตอร์ชนิดแผงวงจรสำเร็จรูป เครื่องควบคุมเชิงตรรกะที่สามารถโปรแกรมได้ และคอมพิวเตอร์ส่วนบุคคล



รูปที่ 2.4 คอนโทรลเลอร์ควบคุมการทำงานของหุ่นยนต์

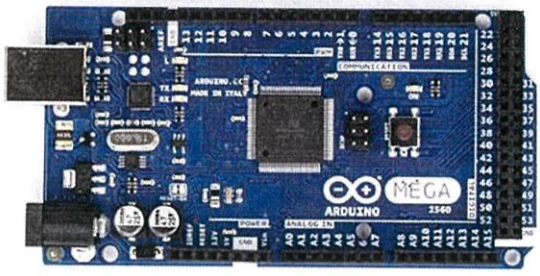
2.2.5 แหล่งพลังงาน (Power Supply)

เป็นอุปกรณ์ที่จ่ายพลังงานไฟฟ้าให้กับโหลดไฟฟ้า เป็นคำที่ใช้กันมากที่สุด ในการแปลงพลังงานไฟฟ้าจากรูปแบบหนึ่ง ไปเป็นอีกรูปแบบหนึ่ง แม้ว่ามันจะยังอาจหมายถึง อุปกรณ์ที่แปลงพลังงานรูปแบบหนึ่ง (เช่นพลังงานกล, พลังงานเคมี, พลังงานแสงอาทิตย์) ให้เป็นพลังงานไฟฟ้า ตัวอย่าง เช่น แบตเตอรี่อัลคาไลน์ แหล่งจ่ายกระแสไฟฟ้ากระแสตรง ฯลฯ

รูปที่ 2.5 แบตเตอร์รี่อัลคาไลน์

2.3 อุปกรณ์

2.3.1 ไมโครคอนโทรลเลอร์



รูปที่ 2.6 Arduino mega 2560

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้น ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

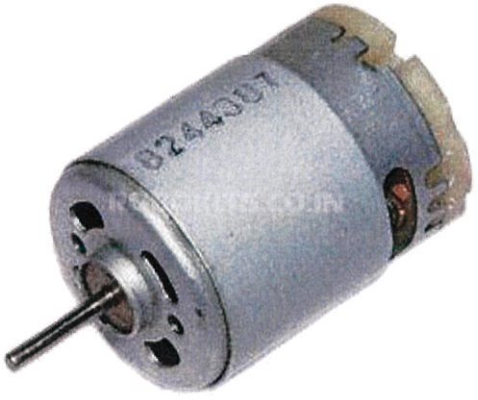
Arduino Mega 2560 บอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 ใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้การพัฒนาไมโครคอนโทรลเลอร์ที่ต้องการบอร์ด Arduino ที่มีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานมากขึ้น

<u>ข้อมูลทางเทคนิค</u>

ชนิด Microcontroller	ATmega2560
ทำงานที่แรงดัน	5V
แรงดัน Input (แนะนำ)	7-12V

แรงดัน Input (ห้ามเกิน)	6-20V
จำนวน Digital I/O Pins	54 (14 พอร์ตสำหรับ PWM และ 4 พอร์ตสำหรับ UART TTL)
จำนวน Analog Input Pins	16
กระแสไฟ DC ต่อ I/O Pin	40 mA
กระแสไฟ DC สำหรับพิน 3.3V	50 mA
หน่วยความจำ Flash	256 KB โดยที่ 8 KB ถูกใช้โดย boot loader
SRAM	8 KB
EEPROM	4 KB
ความเร็ว Clock	16 MHz

2.3.2 มอเตอร์ไฟฟ้ากระแสตรง



รูปที่ 2.7 มอเตอร์ไฟฟ้ากระแสตรง

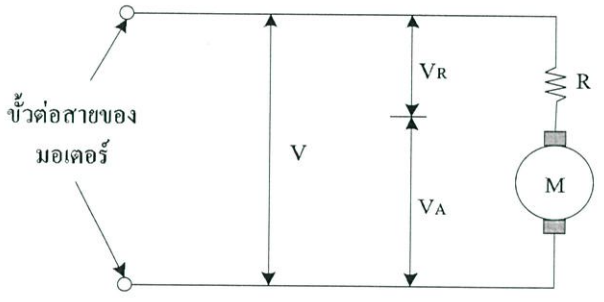
2.3.2.1 หลักการทำงานของมอเตอร์กระแสตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อเป็นแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะ แปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรง แม่เหล็ก จะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ซึ่งวางแกนเพลาและแกนเพลานี้ สวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความ

ว่าตัวหมุน การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกิริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์ หรือโรเตอร์หมุนไปนั้นเป็นไปตามกฎซ้ายของเฟลมมิ่ง (Fleming's left hand rule)

2.3.2.2 คุณสมบัติของมอเตอร์กระแสตรง

ในการอธิบายคุณสมบัติของมอเตอร์กระแสตรงให้ละเอียดนั้นต้องพิจารณาแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย วงจรภายในของมอเตอร์เขียนได้ดังรูปที่ 3



รูปที่ 2.8 วงจรภายในของมอเตอร์กระแสตรง

โดยสมมติให้ทุ่นโรเตอร์ไม่มีความต้านทานอยู่เลย อนุกรมกับความต้านทานซึ่งในที่นี้ก็คือความต้านทานของขดลวดนั่นเอง แรงดันที่ขั้วต่อสายของมอเตอร์ก็คือผลบวกระหว่างแรงดันที่ทุ่นโรเตอร์ (V_A) และ แรงดันตกคร่อมความต้านทานขดลวด (V_R) แรงดัน V_A ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำป้อนกลับ (BACK EMF) ซึ่งเกิดขึ้นในโรเตอร์ขณะที่หมุนแรงดันที่เกิดขึ้นนี้เป็นไปตากกฎของการเหนี่ยวนำแม่เหล็กไฟฟ้าจากการเคลื่อนที่ของตัวนำในสนามแม่เหล็ก สัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็ก และ ความเร็วในการเคลื่อนที่ของตัวนำ แรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์ และ แปรผันตรงกับความเร็วในการหมุน ผลบวกของแรงดันที่ทุ่นโรเตอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดันที่ป้อนให้กับมอเตอร์ (V)

$$V = V_A + V_R \qquad (V)$$

เมื่อพิจารณาตั้งแต่มอเตอร์หยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น $V_A = 0$, $V_R = V$ กระแสที่ไหลในมอเตอร์หาได้จาก

$$I = V_R / R \qquad (A)$$

เมื่อมอเตอร์เริ่มหมุนจะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว V_R ซึ่งมีค่าเท่ากับความแตกต่างระหว่าง V_A และ V จะเริ่มลดลงกระแส I ก็จะเริ่มลดลงเช่นกันขณะที่มอเตอร์ยังมีความเร่งอยู่

ความเร็วจะเพิ่มขึ้น แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์รับภาระโหลดได้สมดุลพอดี ขณะที่มอเตอร์ไม่มีโหลด และ หมุนอย่างอิสระจะมีเพียงค่าความฝืดของแบริ่ง และ แรงต้านอากาศทำให้ V_A เกือบเท่ากับค่า V

2.3.2.3 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงที่ส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

1. ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator) ประกอบด้วย

1.1 เฟรมหรือโยค (Frame Or Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้ว เหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาม้วนเป็นรูปทรงกระบอก

รูปที่ 2.9 เฟรมของมอเตอร์

ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวด ส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกอบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กจากขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์ มีกำลังหมุน (Torque)

รูปที่ 2.10 ขดลวดพันอยู่รอบขั้วแม่เหล็ก

ส่วนที่สอง ขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆแกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น

1.2 แปรงถ่าน (Brushes) ทำด้วยคาร์บอนมีรูปร่างเป็นแท่งสี่เหลี่ยมผืนผ้าอยู่ในซองแปรงมีสปิงกดอยู่ด้านบนเพื่อให้ถ่านนี้สัมผัสกับซี่คอมมิวเตเตอร์ตลอดเวลาเพื่อรับกระแส และส่งกระแสไฟฟ้าระหว่างขดลวดอาร์มาเจอร์ กับวงจรไฟฟ้าจากภายนอก คือถ้าเป็นมอเตอร์กระแสไฟฟ้าตรงจะทำหน้าที่รับกระแสจากภายนอกเข้าไปยังคอมมิวเตเตอร์ให้ลวดอาร์มาเจอร์เกดแรงบิดทำให้มอเตอร์หมุนได้

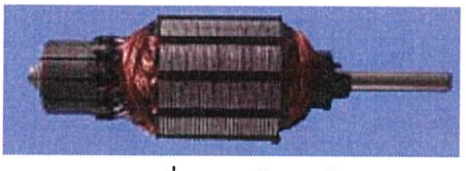
2. ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์ ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

2.1 แกนเพลา (Shaft)

2.2 แกนเหล็กอาร์มาเจอร์ (Armature Core)

2.3 คอมมิวเตอร์ (Commutator)

2.4 ขอลวดอาร์มาเจอร์ (Armature Widing)



รูปที่ 2.11 โรเตอร์

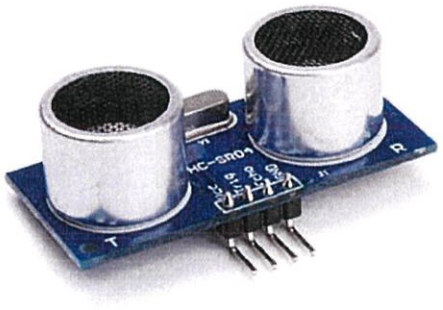
2.1 แกนเพลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลานี้จะวางอยู่บนแบริ่ง เพื่อบังคับให้หมุนอยู่ในแนวนิ่งไม่มีการสั่นสะเทือนได้

2.2 แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด

2.3 คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสาย ของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่าปฏิกิริยามอเตอร์ (Motor action)

2.4 ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ

2.3.3 อัลตราโซนิคเซนเซอร์



รูปที่ 2.12 อัลตราโซนิคเซนเซอร์

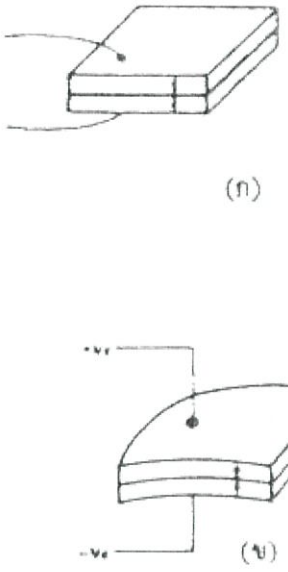
คือ อุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไปมา ซึ่งทำให้เกิดคลื่นเสียงย่านอุลตร้าโซนิคกระจายไปในอากาศได้หรือแปลงพลังงานทางกลให้มาเป็นพลังงานในรูปอื่นได้นั้น มีชื่อเรียกว่า อัลตร้าโซนิคทรานสดิวเซอร์ (Ultrasonic Transducer) ในปัจจุบันอุลตร้าโซนิคทรานสดิวเซอร์มีหลายแบบขึ้นอยู่กับหลักการที่ใช้ แบบที่นิยมใช้กันมากได้แก่

แบบเปียโซอิเลคทริก (Piezo-electric Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าและพลังงานทางกล โดยมีความถี่เรโซแนนซ์คงที่อยู่ค่าหนึ่ง

แบบแมกนีโตสตริคทีฟ (Magnetostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่

แบบอิเลคโตรสตริคทีฟ (Electrostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานทางกล

ทรานสดิวเซอร์แบบเปียโซอิเลคทริค ภายในตัวอุลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคทริค แบบที่มีใช้กันในปัจจุบันซึ่งได้รับการพัฒนากันมาในระดับหนึ่งแล้วจะประกอบด้วยชิ้นสารเซรามิคสี่เหลี่ยมซึ่งมีผิวโลหะเงินฉาบอยู่ทั้ง 2 หน้าเพื่อให้ต่อสายไฟออกมาเป็นขา 2 ขา ชิ้นสารเซรามิคนี้ประกอบขึ้นจากสารเซรามิค 2 ชิ้น ประกบกันอยู่โดยวางให้ขั้วไดโพลทางไฟฟ้าภายในอะตอมของมันมีทิศทางตรงข้ามกันดังรูป



รูปที่ 2.13 เปียโซอิเล็คทริค

(ก) โครงสร้างภายในตัวอุลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคทริคที่ใช้สารเซรามิค

(ข) เมื่อป้อนแรงดันให้แก่ตัวมันจะทำให้ชิ้นสารเซรามิคโก่งงอไปมาทำให้เกิดคลื่นเสียงอุลตร้าโซนิคกระจายไปในอากาศ

ชิ้นสารเซรามิคถูกยึดติดภายในตัวถังอย่างดีเพื่อไม่ให้การสั่นขณะที่มันทำงานอยู่ได้รับผลกระทบกระเทือนจากภายนอกตัวถังมักจะเป็นรูปทรงกระบอกที่มีเส้นผ่าศูนย์กลางและมีความสูงประมาณ 1 ถึง 2.5 ซม. ด้านหน้าทำเป็นช่องเปิดมีตะแกรงติดอยู่เพื่อให้คลื่นอุลตร้าโซนิคเข้ามาหรือออกจากช่องเปิดได้โดยสะดวก ถ้าตัวถังทำมาจากโลหะก็ควรต่อตัวถังลงกราวด์เพื่อทำหน้าที่ชีลด์ สำหรับบางยี่ห้อเขาจะต่อขาหนึ่งติดกับตัวถังมาให้เลย เมื่อพลิกดูขา 2 ขาที่โผล่ออกมาจากตัวถังจะเห็นมีขาหนึ่งติดกับตัวถัง เมื่อมีสัญญาณแรงดันมาตกคร่อมขั้วทั้งสองของชิ้นสารเซรามิคดังรูป (ข) จะทำให้ชิ้นสารโก่งงอมากหรือน้อยหรือในทิศทางใดตามขนาดและทิศทางการเปลี่ยนแปลงขนาดของสัญญาณนั้น ๆ ทำให้เกิดการกดอัดอากาศโดยรอบเกิดเป็นคลื่นเสียงที่มีความถี่เดียวกับสัญญาณนั้นออกไป โดยทั่ว ๆ ไป กำลังเอาท์พุทที่ออกมาจะตกประมาณ 10% ของกำลังไฟฟ้าที่ป้อนเข้าไป แต่กำลังเอาท์พุท์จะสูงสุดที่ค่าประมาณนี้ต่อเมื่อความถี่ของสัญญาณตรงกับความถี่เรโซแนนซ์ซึ่งเป็นความถี่ทางกลตามธรรมชาติของชิ้นสารเซรามิคนั้น ๆ ส่วนที่ความถี่อื่น ๆ กำลังเอาท์พุทจะลดลงกว่านี้มาก

ในทำนองกลับกันเมื่อมีคลื่นเสียงที่มีความถี่ตรงกับความถี่เรโซแนนซ์ของชิ้นสารเซรามิคเข้ามาจะทำให้ชิ้นสารโก่งงอไปมาและเกิดสัญญาณแรงดันซึ่งมีขนาดเล็กขึ้นมาคร่อมขั้วทั้งสองของตัวมันเองได้ คุณสมบัติโดยทั่วไปของอุลตร้าโซนิคทรานสดิวเซอร์แบบเปียโซอิเลคตริคก็คือมีค่าความต้านทานไฟตรงสูงมากอาจสูงถึง 100 MW เรียกว่าถ้าเอามัลติมิเตอร์ธรรมดามาตั้งสเกลวัดค่าความต้านทานสูง ๆ เข็มจะไม่กระดิกเลย แต่ในขณะที่มันทำงานความต้านทานทางด้านไฟสลับจะลดลง

ตัวส่งและตัวรับ

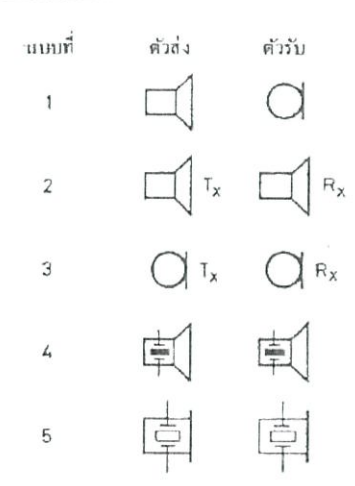
ทรานสดิวเซอร์แบบเปียโซอิเลคทริคที่ใช้สารเซรามิค (หรือที่ผู้ผลิตบางรายเรียกว่าอุลตร้าโซนิคทรานสดิวเซอร์แบบเซรามิค) จะมีอยู่ 2 อย่าง คือ ตัวส่งหรือ Transmitter และ ตัวรับ (เสียง) หรือ Receiver

ตัวส่ง

ก็คืออุลตร้าโซนิคทรานสดิวเซอร์ที่ถูกออกแบบเจาะจงมาให้แปลงสัญญาณไฟฟ้าที่ให้แก่ตัวมัน ให้ออกมาเป็นคลื่นเสียงย่านอุลตร้าโซนิค หน้าที่ของตัวส่งจึงคล้าย ๆ กับเป็นลำโพง

ตัวรับ

ก็คืออุลตร้าโซนิคทรานสดิวเซอร์ที่ถูกออกแบบเจาะจงมาให้แปลงคลื่นเสียงย่านอุลตร้าโซนิคที่มาตกกระทบตัวมันให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจึงคล้าย ๆ กับเป็นไมโครโฟน ด้วยเหตุนี้เวลาเขียนสัญลักษณ์ของอุลตร้าโซนิคทรานสดิวเซอร์จึงนิยมเขียนตามหน้าที่ของมันคือถ้าเป็นตัวส่งก็เขียนสัญลัษณ์เป็นลำโพง ถ้าเป็นตัวรับก็เขียนสัญลักษณ์เป็นไมโครโฟน ดังรูป



รูปที่ 2.14 แสดงตัวอย่างการเขียนสัญลักษณ์ของอุลตร้าโซนิคทรานสดิวเซอร์แบบต่างๆ กัน

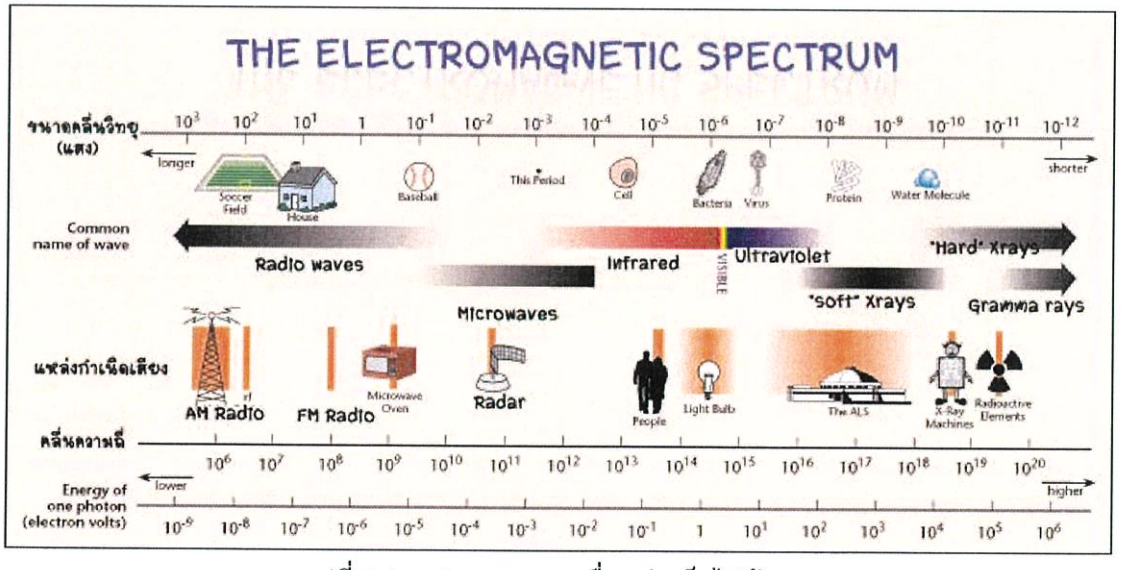
อุลตร้าโซนิคทรานสดิวเซอร์แบบเซรามิคที่มีจำหน่ายกันจะมีค่าความถี่เรโซแนนซ์ให้เลือกตั้งแต่ 23 KHz ขึ้นไปจนถึง 40 KHz แต่ที่พบเห็นกันบ่อยก็มี 23 KHz, 25 KHz, และ 40 KHz โดยความถี่ 40 KHz เป็นรุ่นที่นิยมใช้กันมากที่สุดเพราะมีทิศทางดีกว่า

2.3.4 อินฟราเรดเซนเซอร์

คลื่นแม่เหล็กไฟฟ้า (Electromagnetic Radiation (EM radiation หรือ EMR)) เป็นคลื่นชนิดหนึ่งที่ไม่ต้องใช้ตัวกลางในการเคลื่อนที่ เช่น คลื่นวิทยุ (Radio waves) คลื่นไมโครเวฟ (Microwaves) โดยคุณสมบัติของคลื่นแม่เหล็กไฟฟ้าคือเป็นคลื่นที่เกิดจากคลื่นไฟฟ้าและคลื่นแม่เหล็กตั้งฉากกันและเคลื่อนที่ไปยังทิศทางเดียวกัน คลื่นแม่เหล็กไฟฟ้าสามารถเดินทางได้ด้วยความเร็ว 299,792,458 เมตร/วินาที หรือเทียบเท่ากับความเร็วแสง

คลื่นแม่เหล็กไฟฟ้าเป็นคลื่นตามขวาง ประกอบด้วยสนามไฟฟ้าและสนามแม่เหล็กที่มีการสั่นในแนวตั้งฉากกัน และอยู่บนระนาบตั้งฉากกับทิศการเคลื่อนที่ของคลื่น คลื่นแม่เหล็กไฟฟ้าเป็นคลื่นที่เคลื่อนที่โดยไม่อาศัยตัวกลาง จึงสามารถเคลื่อนที่ในสุญญากาศได้

ปัจจุบันมีการใช้คลื่นแม่เหล็กไฟฟ้าในหลาย ๆ ด้าน เช่น การติดต่อสื่อสาร (มือถือ โทรทัศน์ วิทยุ เรดาร์ ใยแก้วนำแสง) ทางการแพทย์ (รังสีเอกซ์) การทำอาหาร (คลื่นไมโครเวฟ) การควบคุมรีโมท (รังสีอินฟราเรด)



รูปที่ 2.15 ประเภทของคลื่นแม่เหล็กไฟฟ้า

แสงที่ตามองเห็น (Visible light) เป็นเพียงส่วนหนึ่งของคลื่นแม่เหล็กไฟฟ้า ในช่วงซึ่งประสาทตาของมนุษย์สามารถสัมผัสได้ ซึ่งมีความยาวคลื่นอยู่ระหว่าง 400 – 700 นาโนเมตร (1 เมตร = 1,000,000,000 นาโนเมตร) หากนำแท่งแก้วปริซึม (Prism) มาหักเหแสงอาทิตย์ เราจะเห็นว่าแสงสีขาวถูกหักเหออกเป็นสีม่วง คราม น้ำเงิน เขียว เหลือง แสด แดง คล้ายกับสีของรุ้งกินน้ำ

เรียกว่า “สเปคตรัม” (Spectrum) แสงแต่ละสีมีความยาวคลื่นแตกต่างกัน สีม่วงมีความยาวคลื่นน้อยที่สุด สีแดงมีความยาวคลื่นมากที่สุด

นอกจากแสงที่ตามองเห็นแล้วยังมีคลื่นแม่เหล็กไฟฟ้าชนิดอื่นๆ ได้แก่ รังสีที่มีความยาวคลื่นถัดจากสีแดงออกไป เรียกว่า เราเรียกว่า “รังสีอินฟราเรด” หรือ “รังสีความร้อน” เรามองไม่เห็นรังสีอินฟราเรด แต่เราก็รู้สึกถึงความร้อนได้ สัตว์บางชนิด เช่น งู มีประสาทสัมผัสรังสีอินฟราเรด มันสามารถทราบตำแหน่งของเหยื่อได้ โดยการสัมผัสรังสีอินฟราเรดซึ่งแผ่ออกมาจากร่างกายของเหยื่อ รังสีที่มีความยาวคลื่นน้อยกว่าแสงสีม่วงเรียกว่า “รังสีอุลตราไวโอเล็ต” แม้ว่าเราจะมองไม่เห็น แต่เมื่อเราตากแดดนานๆ ผิวหนังจะไหม้ด้วยรังสีชนิดนี้ นอกจากรังสีอุลตราไวโอเล็ตและรังสีอินฟราเรดแล้ว ยังมีคลื่นแม่เหล็กไฟฟ้าประเภทอื่นๆ ซึ่งเรียงลำดับตามความยาวคลื่นได้ดังนี้

รังสีแกมมา (Gamma ray) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นน้อยกว่า 0.01 นาโนเมตร โฟตอนของรังสีแกมมามีพลังงานสูงมาก กำเนิดจากแหล่งพลังงานนิวเคลียร์ เช่น ดาวระเบิด หรือ ระเบิดปรมาณู เป็นอันตรายมากต่อสิ่งมีชีวิต

รังสีเอ็กซ์ (X-ray) มีความยาวคลื่น 0.01 - 1 นาโนเมตร มีแหล่งกำเนิดในธรรมชาติมาจากดวงอาทิตย์ เราใช้รังสีเอ็กซ์ในทางการแพทย์ เพื่อส่องผ่านเซลล์เนื้อเยื่อ แต่ถ้าได้ร่างกายได้รับรังสีนี้มากๆ ก็จะเป็นอันตราย

รังสีอุลตราไวโอเล็ต (Ultraviolet radiation) มีความยาวคลื่น 1 - 400 นาโนเมตร รังสีอุลตราไวโอเล็ตมีอยู่ในแสงอาทิตย์ เป็นประโยชน์ต่อร่างกาย แต่หากได้รับมากเกินไปก็จะทำให้ผิวไหม้ และอาจทำให้เกิดมะเร็งผิวหนัง

แสงที่ตามองเห็น (Visible light) มีความยาวคลื่น 400 – 700 นาโนเมตร พลังงานที่แผ่ออกมาจากดวงอาทิตย์ ส่วนมากเป็นรังสีในช่วงนี้ แสงแดดเป็นแหล่งพลังงานที่สำคัญของโลก และยังช่วยในการสังเคราะห์แสงของพืช

รังสีอินฟราเรด (Infrared radiation) มีความยาวคลื่น 700 นาโนเมตร – 1 มิลลิเมตร โลกและสิ่งชีวิตแผ่รังสีอินฟราเรดออกมา ก๊าซเรือนกระจก เช่น คาร์บอนไดออกไซด์ และไอน้ำ ในบรรยากาศดูดซับรังสีนี้ไว้ ทำให้โลกมีความอบอุ่น เหมาะกับการดำรงชีวิต

คลื่นไมโครเวฟ (Microwave) มีความยาวคลื่น 1 มิลลิเมตร – 10 เซนติเมตร ใช้ประโยชน์ในด้านโทรคมนาคมระยะไกล นอกจากนั้นยังนำมาประยุกต์สร้างพลังงานในเตาอบอาหาร

คลื่นวิทยุ (Radio wave) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นมากที่สุด คลื่นวิทยุสามารถเดินทางผ่านชั้นบรรยากาศได้ จึงถูกนำมาใช้ประโยชน์ในด้านการสื่อสาร โทรคมนาคม

รังสีอินฟราเรด

รังสีอินฟราเรด (infrared) คือ แสงที่มนุษย์เราไม่สามารถมองเห็นได้ เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำกว่าแสงสีแดง แหล่งกำเนิดของรังสีอินฟราเรด คือ ความร้อน (Heat) จะเกิดการแผ่รังสีความร้อนกับวัตถุใดก็ตามที่มีอุณหภูมิสูงกว่าค่า Absolute Zero หรืออุณหภูมิศูนย์สัมบูรณ์ (-273.15 องศาเซลเซียล หรือ 0 องศาเคลวิน) เท่านั้น โดยจะมีการแผ่รังสีในย่านอินฟราเรด

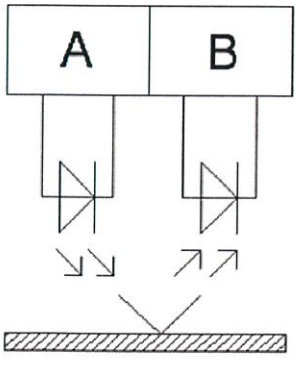
หรือรังสีความร้อนที่แผ่ออกมาจากพื้นผิวของวัตถุ แม้วัตถุนั้นจะเป็นไนโตรเจนเหลวที่มีอุณหภูมิอุณหภูมิ -196 องศาเซลเซียล ก็ยังมีการแผ่รังสีอินฟราเรดออกมาได้ ถึงแม้จะมีค่าน้อยก็ตาม ซึ่งในความเป็นจริงและในชีวิตประจำวัน เราจะสัมผัสและเกี่ยวข้องกับรังสีอินฟราเรดอยู่ตลอดเวลา กล่าวคือ เรารู้สึกร้อนเมื่ออยู่กลางแสงอาทิตย์หรือใกล้ไฟ ถึงแม้ว่าตามนุษย์จะไม่สามารถมองเห็นรังสีอินฟราเรดได้ แต่ผิวเราก็สามารถรู้สึกสัมผัสถึงความร้อนได้ นั่นแสดงให้เห็นว่าวัตถุที่ร้อนย่อมแผ่พลังงานรังสีความร้อนหรือรังสีอินฟราเรดออกมาได้มากกว่าวัตถุที่เย็น

การประยุกต์ใช้อินฟราเรดในชีวิตประจำวัน

- กล้องถ่ายรูปใช้กลางคืน และกล้องส่องทางไกลที่ใช้ในเวลากลางคืน แสดงภาพความร้อน เพิ่มความปลอดภัยเวลาขับรถในเวลากลางคืน
- รีโมทคอนโทรลในเครื่องใช้ไฟฟ้าก็เป็นอินฟราเรดอีกชนิดหนึ่ง
- การไล่ล่าทางทหาร มิดไซ ที่ใช้ไล่ล่าเครื่องบินก็เป็นอินฟราเรดอีกชนิดหนึ่ง
- เครื่องกำเนิดความร้อนทั่วไป เช่นเตาแก๊สอินฟราเรดในครัวเรือน เครื่องกำเนิดความร้อนในห้องซาวด์น่า
- แผ่นกายภาพบำบัด มีเป็นประคบร้อนอินฟราเรด ปัจจุบันเป็นวิธีการกายภาพบำบัดที่ปลอดภัยชนิดหนึ่ง เช่น ความร้อนอุณหภูมิต่ำมาจากอินฟราเรดสามารถซึมเข้าลึกถึงผิวหนัง 1-1.5นิ้ว ลดอาการปวดหัวเข่า หรือทำให้แผลเรื้อรังโลหิตหมุนเวียนดีขึ้นจึงทำให้แผลหายเร็ว

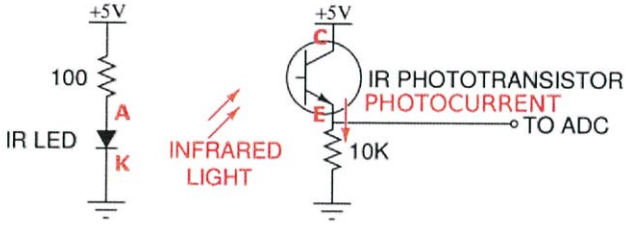
การประยุกต์ใช้อินฟราเรดในการวัดระยะ

การประยุกต์ใช้อินฟราเรดในการวัดระยะสามารถทำได้โดยใช้หลักการสะท้อนของแสงแล้วทำการวัดค่าความเข้มของแสง(Light Intensity) ผ่านตัวเซนเซอร์ที่เป็นทางแสงเช่น Photodiode Phototransistor ซึ่งต่ออยู่กับวงจรอิเล็กทรอนิคส์เพื่อทำการแปลงค่าความแสงให้ออกมาเป็นแรงดันไฟฟ้า



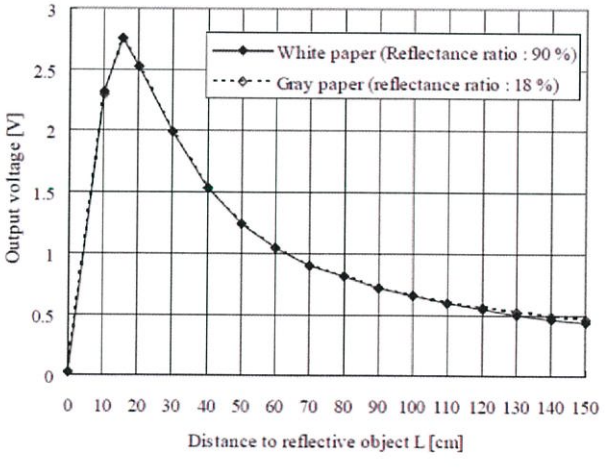
รูปที่ 2.16 แสดงการทำงานของ Infrared Sensor

อินฟราเรดเซนเซอร์จะประกอบด้วย 2 ส่วนหลักๆคือ ส่วน Emitter และส่วน Receiver โดยส่วน Emitter จะมีหน้าที่ในการให้กำเนิดแสง แล้วส่งออกไปยังภายนอก และส่วน Receiver จะทำหน้าที่รับแสงที่สะท้อนกลับมาและแปลงให้ออกมาเป็นแรงดันไฟฟ้า ซึ่งค่าแรงดันไฟฟ้าที่จะจะเป็นแปรผันโดยตรงกับค่าความเข้าแสงที่สะท้อนกลับมา และค่าความเข้าแสงที่สะท้อนกลับมาจะแปรผกผันกับระยะทางของวัตถุที่กั้นขวางแสงอยู่ หรืออาจกล่าวโดยสรุปได้ว่าค่าแรงดันไฟฟ้าจะแปรผกผันระยะห่างของวัตถุ

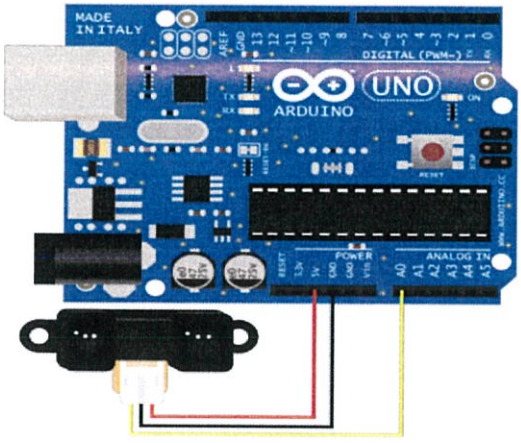


รูปที่ 2.17 แสดงวงจรอย่างง่ายของ Infrared Sensor

โดยเซนเซอร์ที่ทำการนำมาใช้ในการศึกษาครั้งนี้คือ SHARP GP2Y0A02YK0F Distance Measuring Sensor ซึ่งมีย่านการวัดอยู่ที่ 20 – 150 cm และให้สัญญาณเอาต์พุตออกมาเป็นแรงดันไฟฟ้า และใช้งานกับแรงดันไฟฟ้า 4.5 – 5 V ที่ Current consumption 33 mA โดยค่าเอาต์พุตที่ได้มีความสัมพันธ์กับระยะห่างของวัตถุดังรูป



รูปที่ 2.18 แสดงความสัมพันธ์ระหว่าง Output Voltage และ ระยะห่างของวัตถุ



รูปที่ 2.19 แสดงตัวอย่างการต่อ IR Sensor กับบอร์ด Arduino

2.3.5 เซนเซอร์บอกตำแหน่งและวัดมุมที่ทำการหมุน

เทคโนโลยีที่ใช้ในการตรวจจับหาทิศทางและตำแหน่งของอุปกรณ์ต่างๆนั้นมีอยู่หลากหลายวิธีการและหลากหลายอุปกรณ์ เช่น

1. แมกนิโทมิเตอร์ (magnetometer) คือ เครื่องมือวัดค่าความเข้มสนามแม่เหล็ก ในการสำรวจทางภาคพื้นดินใช้วัดค่าความเข้มสนามแม่เหล็กแนวตั้ง บางครั้งใช้วัดความเข้มสนามแม่เหล็กแนวราบหรือแนวรวม แต่ในการสำรวจทางอากาศส่วนใหญ่ใช้วัดสนามแม่เหล็กแนวรวม
2. Accelerometer คือ เครื่องตัววัดความเร่ง ของการเคลื่อนที่ของวัตถุ คุ้นเคยดีในมือถือสมาร์ทโฟนทั่วไป เช่น iPhone ตัวอย่างการใช้งานเช่น การเขย่าเพื่อเปลี่ยนเพลง หรือการเขย่าตัวเครื่องเพื่อใช้ในการควบคุมการเล่นเกม ล้วนเป็นคุณสมบัติของ accelerometer ที่ติดมาในเครื่อง โครงสร้างของ accelerometer จะประกอบด้วยสปริงและลูกตุ้มน้ำหนัก เมื่อมีการเคลื่อนที่ด้วยความเร่งลูกตุ้มน้ำหนักจะถูกกดไปอีกฝั่งตรงข้ามกับการเคลื่อนที่ สปริงก็ทำหน้าที่ดึงกลับเข้าที่อีกครั้งเมื่อหยุดการเคลื่อนที่ การเคลื่อนที่ด้วยความเร็วคงที่คือความเร่งเท่ากับศูนย์ ค่าที่วัดได้ก็จะไม่เปลี่ยนแปลง
3. ไจโรสโคป เป็นอุปกรณ์ที่อาศัยแรงเฉื่อยของล้อหมุน เพื่อช่วยรักษาระดับทิศทางของแกนหมุนที่ดี ประกอบด้วยล้อหมุนเร็วบรรจุอยู่ในกรอบอีกทีหนึ่ง ทำให้เอียงในทิศทางต่างๆ ได้โดยอิสระ นั่นคือ หมุนในแกนใดๆ ก็ได้ โมเมนตัมเชิงมุมของล้อดังกล่าวทำให้มันคงรักษาตำแหน่งของมันไว้แม้กรอบล้อจะเอียง จากคุณสมบัติดังกล่าวทำให้สามารถนำหลักการนี้ไปประยุกต์ใช้เพื่อประโยชน์ต่างๆ มากมาย เช่น เข็มทิศ และนักบินอัตโนมัติของเครื่องบิน เรือ กลไกบังคับหางเสือของตอร์ปิโด อุปกรณ์ป้องกันการกลิ้งบนเรือใหญ่ และระบบนำร่องเฉื่อย (inertial guidance) รวมถึงระบบในยานอวกาศ และสถานีอวกาศ

2.3.6 เซนเซอร์ตรวจวัดระยะการหมุนและการวัดความเร็วของมอเตอร์

หลักการทำงาน: เครื่องมือแสดงระยะของการเคลื่อนไหวโดยแสดงผลเป็นตัวเลขนั้นเป็นแหล่งกำเนิดแสงวาบที่มีความไวซึ่งจะถูกนำไปใช้ในการวัดความเร็วของวัตถุที่เคลื่อนที่เร็ว หรือใช้เพื่อทำให้เห็นภาพของการหยุด หรือลดความเร็วของการเคลื่อนที่ที่มีความเร็วสูงเพื่อการสังเกตการณ์การวิเคราะห์หรือการถ่ายภาพความเร็วสูง เครื่องมือแสดงระยะของการเคลื่อนไหวนี้จะปล่อยแสงวาบที่มีความเข้มสูงเป็นระยะเวลาสั้นๆเครื่องมือนี้จะมีชุดกำเนิดแรงกระตุ้นอีเล็กทรอนิกส์ซึ่งจะควบคุมอัตราการวาบของแสง สายไฟส่งพลังงานและไดโอดเปล่งแสง (LED) ซึ่งอ่านค่าได้เป็นจำนวนครั้งของการวาบต่อหนึ่งนาที และอาจเล็งแสงไปยังวัตถุเคลื่อนที่ส่วนใหญ่ได้ซึ่งรวมถึงวัตถุที่อยู่ในพื้นที่ที่เข้าไม่ถึงด้วย เมื่อทำการวัดความเร็วในการหมุนของวัตถุ ให้ตั้งค่าอัตราการวาบแสงเริ่มต้นให้สูงกว่าค่าความเร็วโดยประมาณของวัตถุ จากนั้นให้ลดอัตราเร็วของการวาบแสงลงช้าๆ จนกระทั่งเห็นภาพเดี่ยวเป็นครั้งแรก ณจุดนี้อัตราการวาบแสงรูปกรวยจะมีค่าเท่ากับความเร็วในการหมุนของวัตถุ และสามารถอ่านค่าความเร็วได้โดยตรงจากจอแสดงผลเป็นตัวเลข

<u>ทฤษฎีที่เกี่ยวข้อง</u>

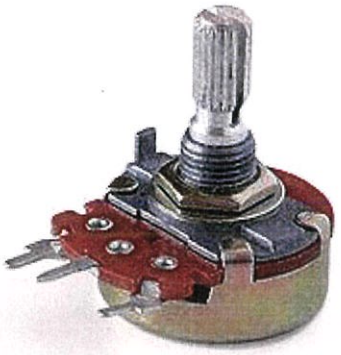
การกระจัดเชิงมุม (Angular Displacement) คือการเคลื่อนที่รอบตัวเองเป็นวงกลมไม่มีจุดเริ่มต้นและจุดสิ้นสุด จะเคลื่อนที่ซ้ำรอบเดิมอยู่อย่างนั้น

มอเตอร์ เป็นเครื่องใช้ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล ประกอบด้วยขดลวดที่พันรอบแกนโลหะที่วางอยู่ระหว่างขั้วแม่เหล็ก โดยเมื่อผ่านกระแสไฟฟ้าเข้าไปยังขดลวดที่อยู่ระหว่างขั้วแม่เหล็ก จะท่าให้ขดลวดหมุนไปรอบแกน และเมื่อสลับขั้วไฟฟ้า การหมุนของขดลวดจะหมุนกลับทิศทางเดิม

<u>ประเภท</u>

- **โพแทนทิออมิเตอร์แบบหมุน** โพแทนทิออมิเตอร์เป็นตัวต้านทานไฟฟ้าที่สามารถปรับค่าได้

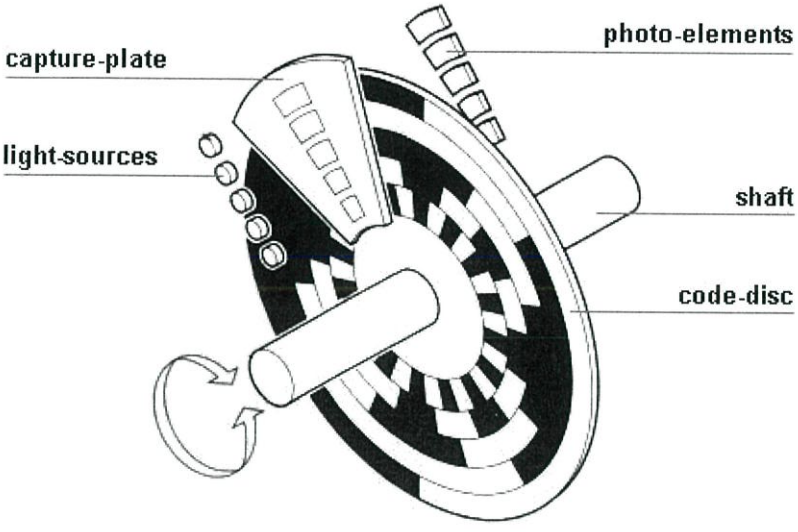
รูปที่ 2.20 แสดงลักษณะของโพแทนทิออมิเตอร์แบบหมุนที่ใช้ในงานอุตสาหกรรม



รูปที่ 2.21 แสดงลักษณะของโพแทนทิออมิเตอร์แบบหมุน

แต่พื้นฐานที่เราพบทั่วไปแสดงดังรูปที่ 3 มีขาใช้งาน 3 ขาคือ 1, 2 และ 3 สามารถหาซื้อได้ง่ายตามร้านจำหน่ายอุปกรณ์อิเล็กทรอนิกส์ทั่วไป โพแทนทิออมิเตอร์ส่วนใหญ่ที่ใช้แบ่งได้ 2 แบบคือ แบบ A (Taper A) และแบบ B (Taper B) ต่างกันที่การเพิ่มค่าความต้านทานของแบบ A จะเป็นแบบลอการิทึม (Logarithm) นิยมใช้เป็นตัวปรับแต่งเสียงในงานระบบเสียง ส่วนแบบ B จะเพิ่มขึ้นคงที่เป็นแบบเชิงเส้น (Linear) โดยแบบของโพแทนทิออมิเตอร์แต่ละตัวจะระบุไว้ที่ตัวอุปกรณ์ เช่น B100K หมายถึง โพแทนทิออมิเตอร์ตัวนั้นมีค่าความต้านทาน 100 กิโลโอห์ม และเป็นแบบ B

- **เอนโคเดอร์ (Encoder)** เอนโคเดอร์เป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนสัญญาณเชิงกลในรูปแบบของการหมุนหรือการกระจัดมาเป็นสัญญาณทางไฟฟ้า โดยใช้หลักการทางแสง (Optical) เข้ามาช่วย การใช้งานส่วนใหญ่จะถูกนำมาใช้เป็นอุปกรณ์นับรอบการหมุนของมอเตอร์หรืออุปกรณ์วัดองศาการหมุนของมอเตอร์ก็ได้ สัญญาณทางเอาท์พุตของเอนโคเดอร์มีทั้งแบบที่เป็นสัญญาณดิจิตอลและแบบที่เป็นสัญญาณอนาล็อก แต่ที่พบเห็นส่วนใหญ่จะเป็นแบบดิจิตอล



รูปที่ 2.22 แสดงเอ็นโคเดอร์อย่างง่าย

จากรูปจะเห็นว่าส่วนประกอบที่สำคัญของเอนโคเดอร์ประกอบด้วย ตัวส่งแสง ตัวรับแสง และจานรหัส โดยส่วนที่เป็นตัวส่งและตัวรับแสงจะติดตั้งคนละด้านโดยมีจานรหัสเป็นตัวคั่นกลาง จานรหัสจะยึดกับมอเตอร์ที่เราต้องการวัดรอบ ลักษณะของจานรหัสจะมีส่วนที่เป็นช่องที่แสงผ่านได้และส่วนที่ทึบแสง เมื่อมอเตอร์หมุนจะทำให้จานรหัสหมุนตามไปด้วย ถ้าจานรหัสหมุนส่วนทึบแสงจะไปกั้นลำแสงทำให้เอาท์พุตที่ส่วนรับแสงเป็นลอจิก 0 แต่ถ้าจานรหัสหมุนส่วนที่แสงผ่านได้ตรงกับลำแสง ทำให้ตัวรับสามารถรับแสงได้ เอาท์พุตที่ตัวรับแสงจะเป็นลอจิก 1 สลับกันเป็นเช่นนี้จนครบรอบ เอนโคเดอร์ขนาด 1 บิต มีรหัสเพียงหลักเดียวในการใช้งานจริง ตัวรับและตัวส่งแสงอาจมีมากกว่า 1 บิต เพื่อให้สามารถระบุตำแหน่งของจานได้ละเอียดขึ้น และตัวจานรหัสเองก็จะมีหลายชั้นขึ้นสอดคล้องกับจำนวนบิต รหัสที่ได้ออกมาจะมีหลายรูปแบบแล้วแต่การออกแบบ โดยทั่วไปรหัสที่ใช้ได้แก่ Binary code, BCD code เป็นต้น

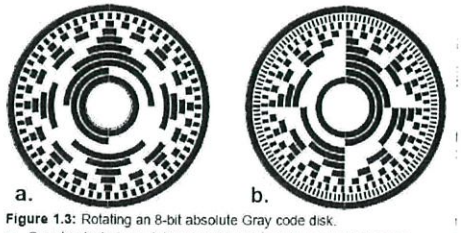


Figure 1.3: Rotating an 8-bit absolute Gray code disk.
a. Counterclockwise rotation by one position increment will cause only one bit to change.
b. The same rotation of a binary-coded disk will cause all bits to change in the particular case (255 to 0) illustrated by the reference line at 12 o'clock.
[Everett, 1995].

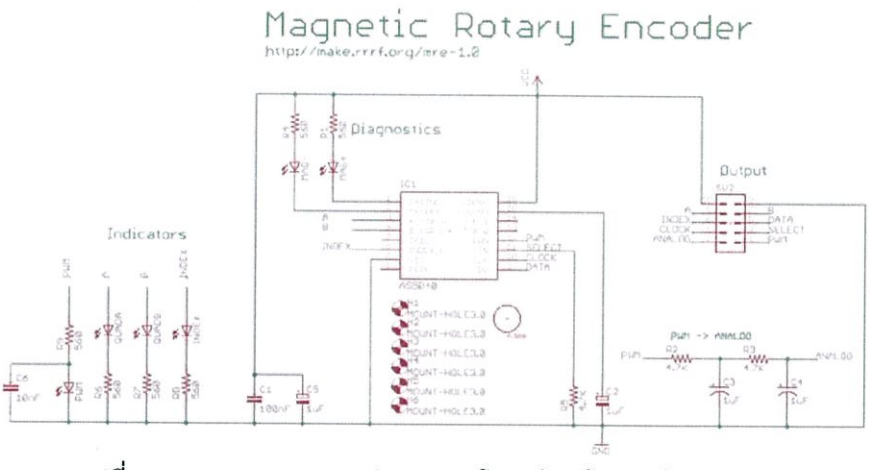
รูปที่ 2.23 แสดงจานBCD

ชนิดของเอนโคเดอร์มีดังนี้

1. โรตารีเอนโคเดอร์ชนิดแมกนีติก (Magnetic Type Rotary Encoder) ฮอลล์เซนเซอร์ (Hall Sensor) และทรงกลมหมุนที่เป็นแม่เหล็ก, ทรงกลมหมุนที่เป็นแม่เหล็กนี้ จะมีแม่เหล็กเล็ก ๆ วางอยู่ในที่ตำแหน่งถาวร, ฮอลล์เซนเซอร์ จะถูกติดตั้งกับทรงกลมหมุน ในขณะที่ทรงกลมหมุนสนามแม่เหล็กจากขั้วเหนือจะวิ่งไปสู่ขั้วใต้จะผ่านฮอลล์เซนเซอร์ ทำให้ความต้านทานของฮอลล์เซนเซอร์นี้เปลี่ยนแปลงไป และการเปลี่ยนแปลงที่ถูกตรวจจับได้นี้ จะเป็นสัญญาณให้รู้ว่า ขณะนี้ทรงกลมกำลังหมุน ดังนั้น บางเวลาระบบอาจจะถูกออกแบบให้สัญญาณเฟสเป็นตัวบอกทิศทางการหมุนด้วย ฮอลล์เซนเซอร์ (Hall Sensor) มีความไวต่อการเปลี่ยนแปลงของอุณหภูมิ ดังนั้นจึงต้องการวงจรขยายสัญญาณ หรือวงจรบริดจ์เข้ามาช่วยด้วย ข้อได้เปรียบของ Encoder ชนิดนี้คือจะมีความทนทานต่อการสั่นสะเทือนทางกล รวมทั้งมีการปฏิบัติงานที่ดีในที่ความเร็วสูง ดังนั้นมันจึงเหมาะสมที่จะใช้ในวงจรควบคุมเซอโวมอเตอร์ (Servo motor)

การใช้งาน: ตรวจจับความเร็วใน AC/DC Servo Motor, วัดความเร็วรอบ (RPM) ของมอเตอร์

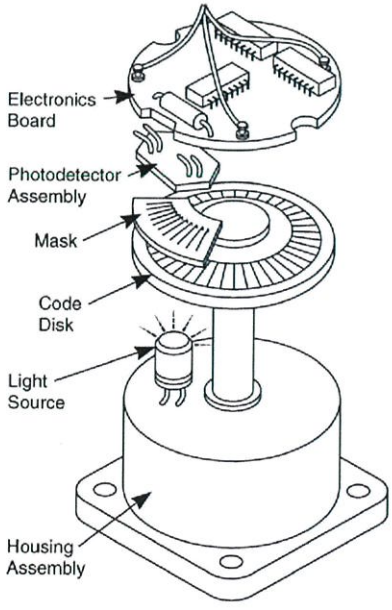
รูปที่ 2.24 แสดงโรตารีเอนโคเดอร์ชนิดแมกนีติก



รูปที่ 2.25 แสดงวงจรการทำงานของโรตารีเอนโคเดอร์ชนิดแมกนีติก

2. โรตารีเอนโคเดอร์ชนิดออฟติคคัล (Optical Type Rotary Encoder) ออฟติคัลโรตารีเอนโคเดอร์ (Optical Rotary Encoder) จะไม่มีผลตอบสนองต่ออุณหภูมิและการสั่นสะเทือนทางกลมากนัก แผ่นดิสก์นี้จะมีน้ำหนักเบาและมันสามารถที่จะหมุนและหยุดหมุนได้โดยไม่มีการหน่วงจึงทำให้ค่าผิดพลาดที่เกิดขึ้นในข้อมูลจะน้อยมาก ดังนั้นมันจึงง่ายมากที่จะขับแผ่นดิสก์

การใช้งาน: ตรวจจับความเร็วและตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรงเล็ก ๆ และเครื่องจักรกลหมุนต่าง ๆ

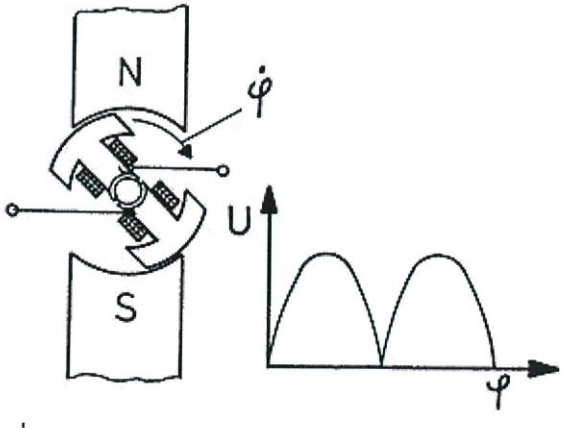


รูปที่ 2.26 แสดงส่วนประกอบของโรตารีเอนโคเดอร์ชนิดออฟติคคัล

รูปที่ 2.27 แสดงโรตารีเอนโคเดอร์ชนิดออฟติคคัล

- **แทคโคเจเนอเรเตอร์** (Tacho Generator)

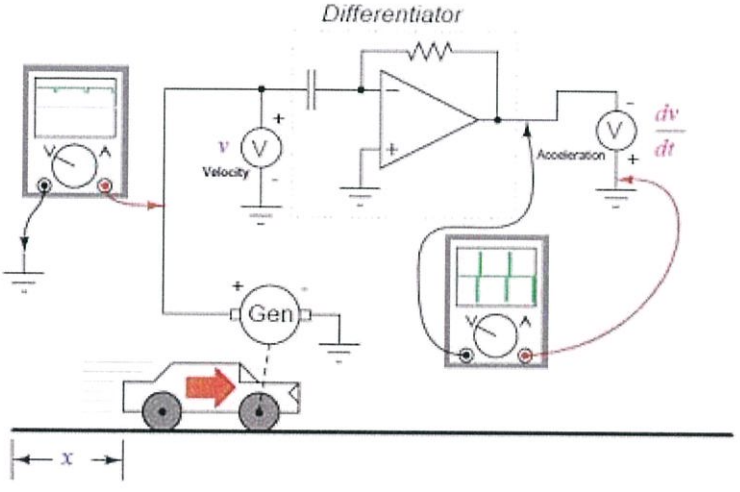
แทคโคเจเนอเรเตอร์ เป็นอุปกรณ์เชิงกลที่เปลี่ยนการหมุนเป็นสัญญาณทางไฟฟ้าอีกชนิดหนึ่งที่นิยมใช้งาน เป็นอุปกรณ์เซนเซอร์แบบแอคทีฟใช้หลักการของแรงเคลื่อนไฟฟ้าเหนี่ยวนำในการทำงาน มีทั้งแบบที่ให้สัญญาณเอาท์พุตเป็นสัญญาณไฟฟ้ากระแสตรงและแบบที่เป็นสัญญาณไฟฟ้ากระแสสลับ



รูปที่ 2.28 แสดงหลักการทำงานของแทคโคเจเนอเรเตอร์

หลักการกำเนิดแรงเคลื่อนไฟฟ้าเหนี่ยวนำมีอุปกรณ์สำคัญ 2 ตัวคือ ขดลวด และแม่เหล็กถาวร ถ้าขดลวดหมุนในสนามแม่เหล็กของแม่เหล็กถาวรครบ 1 รอบ จะได้สัญญาณไฟฟ้ารูปคลื่นไซน์ที่สมบูรณ์ 1 ลูก โดยความแรงของสัญญาณขึ้นอยู่กับความเร็วในการหมุนของขดลวด ยิ่งหมุนเร็วสัญญาณที่ออกมาก็จะมีความแรงมาก ถ้า

เราต่อแกนของมอเตอร์ที่เราต้องการวัดรอบเข้ากับขดลวด เราก็จะได้สัญญาณที่แทโคเจเนอเรเตอร์เปลี่ยนแปลงตามความเร็วของการหมุนของมอเตอร์ตัวนั้นได้ไม่ยาก ส่วนแทโคเจเนอเรเตอร์แบบไฟฟ้ากระแสตรง ก็มีโครงสร้างพื้นฐานเหมือนมอเตอร์ไฟฟ้ากระแสตรง สามารถทดลองโดยนำมอเตอร์ในรถของเล่นมาปั่น LED ให้ติดสว่างได้ โดยความแรงของสัญญาณจะเปลี่ยนแปลงตามความเร็วของการหมุน เช่นเดียวกับแทคโคเจเนอเรเตอร์แบบกระแสสลับ



รูปที่ 2.29 แสดงการทำงานของแทคโคเจเนอเรเตอร์

รูปที่ 2.30 แทคโคเจเนอเรเตอร์

รูปที่ 2.31 แสดงการวัดโดยใช้แทคโคเจเนอเรเตอร์

2.3.7 การสื่อสารผ่านบลูทูธ

บลูทูธ คือ เทคโนโลยีการสื่อสารไร้สายระยะใกล้แบบเครือข่ายไร้สายส่วนบุคคล (wireless personal area networks: WPAN) เป็นมาตรฐานที่ถูกออกแบบมาเพื่อใช้ในการเชื่อมต่ออุปกรณ์ไร้สายขนาดเล็ก เช่น เครื่องพีดีเอ (personal digital assistant:PDA) อุปกรณ์สื่อสารแบบพกพาหรือเคลื่อนที่รวมไปถึงการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตผ่านอุปกรณ์ปลายทางที่ให้บริการ

ประวัติของบลูทูธ

บลูทูธถูกระบุว่ามาจากชื่อของกษัตริย์ชาวเดนมาร์กที่พระนามว่า ฮาราล์ด บลาทานด์ (Herald Blatand) หรือ ฮาราล์ด บลูทูธ (Harald Bluetooth) ซึ่งครองราชย์ในปี ค.ศ. ๙๔๐ – ค.ศ. ๙๘๕ (พ.ศ. ๑๔๘๓ – พ.ศ.๑๕๒๘) โดยพระองค์ทรงเป็นผู้ที่รวบรวมอาณาจักรเดนมาร์กและนอร์เวย์เข้าด้วยกันให้เป็นปึกแผ่น ซึ่งมีลักษณะคล้ายจุดประสงค์ของเทคโนโลยีการสื่อสารไร้สายชนิดนี้ ที่ต้องการรวบรวมการสื่อสารระหว่างอุปกรณ์คอมพิวเตอร์และอุปกรณ์การสื่อสารชนิดต่างๆ ให้เป็นเครือข่ายหนึ่งเดียว ดังนั้นเทคโนโลยีการสื่อสารชนิดนี้จึงได้ชื่อว่า บลูทูธ

หลักการพื้นฐานของบลูทูธ

เทคโนโลยีการสื่อสารไร้สายบลูทูธ ถูกนำมาประยุกต์ใช้งานในการเชื่อมต่อกับอุปกรณ์ต่างๆ ทำให้สะดวกต่อการใช้งาน เนื่องจากไม่จำกัดพื้นที่ มิต้องใช้อุปกรณ์ที่เป็นสายสัญญาณ สามารถเชื่อมต่อได้ไกล เช่น การส่งข้อมูลจากโทรศัพท์เคลื่อนที่เครื่องหนึ่งไปยังโทรศัพท์เคลื่อนที่อีกเครื่องหนึ่ง หากส่งผ่านสายสัญญาณ จำเป็นต้องใช้อุปกรณ์เสริมเพื่อทำให้อุปกรณ์ทั้งสองเชื่อมต่อกันได้ แต่เทคโนโลยีบลูทูธ ช่วยให้การส่งข้อมูลของอุปกรณ์ทั้งสองสะดวกขึ้นโดยการส่งผ่านคลื่นวิทยุ

ระบบเครือข่ายไร้สายส่วนบุคคลที่ใช้เชื่อมต่อ โดยตรงระหว่างอุปกรณ์ที่อยู่ใกล้กันชนิดนี้ ในแต่ละเครือข่าย จะมีอุปกรณ์ตัวหนึ่ง เรียกว่า มาสเตอร์ (Master) หรือตัวแม่ข่าย ทำหน้าที่ควบคุมการทำงานและประสานงานให้กับอุปกรณ์ตัวอื่นๆในเครือข่ายเดียวกัน ส่วนอุปกรณ์ที่เชื่อมต่อตัวอื่นๆ เรียกว่า สลาฟ(Slave) หรือตัวลูกข่าย ซึ่งโครงสร้างการทำงานของบลูทูธนี้คล้ายกับระบบบัสอนุกรมแบบใช้ร่วมร่วม (universal serial bus : USB) ที่ใช้กับเครื่องคอมพิวเตอร์หรืออุปกรณ์อิเล็กทรอนิกส์ทั่วๆไป แต่ต่างกันในส่วนของการเชื่อมต่อ โดยอุปกรณ์บลูทูธส่วนใหญ่สามารถทำหน้าที่ได้ทั้งมาสเตอร์หรือสลาฟ ตามความเหมาะสม ซึ่งภายในเครือข่ายจะมีการจัดการกันเองโดยอัตโนมัติด้วย โพรโทคอลมาตรฐาน

อุปกรณ์บลูทูธแต่ละตัวจะมีแอดเดรส (Address) หรือการระบุตำแหน่ง ซึ่งเป็นรหัสประจำตัวที่ไม่ซ้ำกับอุปกรณ์ตัวอื่น มีความยาวขนาด ๔๘ บิต เรียกว่า บีดี แอดเดอ (BD_ADDR) ใช้ในการจำแนกอุปกรณ์แต่ละตัวและใช้ในการระบุความถี่ที่ใช้สำหรับอุปกรณ์ตัวนั้นๆ ด้วย

ความถี่คลื่นวิทยุ

ความถี่มาตรฐานสำหรับเทคโนโลยีบลูทูธประมาณ ๒.๔ – ๒.๔๘๓ กิกะเฮิรตซ์ (GHz) ซึ่งช่วงความถี่ที่ใช้งานอาจแตกต่างกันบ้างในบางประเทศ เนื่องจากความถี่ที่ใช้สำหรับบลูทูธ เป็นความถี่สาธารณะ (Unlicensed frequency) ไม่ต้องขออนุญาตการใช้งานความถี่ดังกล่าวจากหน่วยงานกำหนดหรือจัดสรรความถี่ของประเทศนั้นๆ ทำให้การใช้งานความถี่นี้แออัด อาจถูกรบกวนจากสิ่งต่าง ๆ เช่น คลื่นสัญญาณรบกวนจากเครือข่าย ที่อยู่ใกล้กันได้ง่าย ดังนั้นประสิทธิภาพของการใช้งานบลูทูธจึงขึ้นอยู่กับคุณภาพของอุปกรณ์ จำนวนหรือความหนาแน่นของการใช้งานด้วย

ระยะเชื่อมต่อของบลูทูธ

อุปกรณ์บลูทูธถูกแบ่งออกเป็นสามระดับ ตามความสามารถในการส่งข้อมูล ดังนี้

ระดับหนึ่ง (Class 1) สามารถรับส่งข้อมูลในรัศมี ๑๐๐ เมตร ใช้พลังงานประมาณ ๑๐๐ มิลลิวัตต์

ระดับสอง (Class 2) สามารถรับส่งข้อมูลในรัศมี ๑๐ เมตร ใช้พลังงานประมาณ ๒.๕ มิลลิวัตต์

ระดับสาม (Class 3) สามารถรับส่งข้อมูลในรัศมี ๑ เมตร ใช้พลังงานประมาณ ๑ มิลลิวัตต์

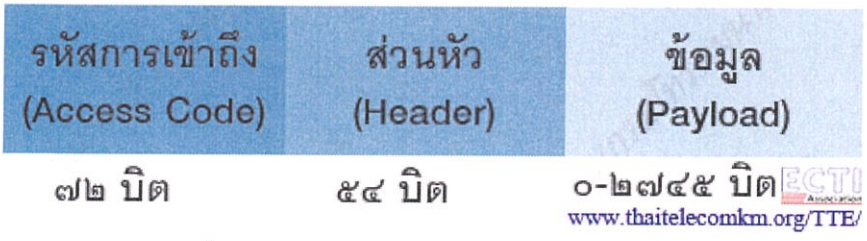
<u>ส่วนประกอบของชุดข้อมูล</u>

ข้อมูลที่รับส่งอยู่ในเครือข่ายบลูทูธ ถูกแบ่งออกเป็นหน่วยย่อยๆ เรียกว่า พีดียู (packet data unit: PDU) ซึ่งประกอบไปด้วย

ก) รหัสการเข้าถึง (Access Code) เป็นส่วนที่เก็บข้อมูลหมายเลขเครือข่ายและแอดเดรสหรือตำแหน่งของอุปกรณ์ต้นและปลายทาง มีขนาดยาว ๗๒ บิต

ข) ส่วนหัว (Header) เป็นส่วนที่เก็บข้อมูลเส้นทางที่เหมาะสมในการส่งข้อมูล มีขนาดยาว ๕๔ บิต

ค) ข้อมูล (Payload) คือข้อมูลที่ต้องการส่งไปยังปลายทาง มีขนาดระหว่าง ๐ – ๒,๗๔๕ บิต ขึ้นอยู่กับการใช้งาน ดังรูปที่ 2.32



รูปที่ 2.32 แสดงส่วนประกอบชุดข้อมูลของบลูทูธ

โครงสร้างชุดข้อมูลของบลูทูธ

เครือข่ายขนาดย่อม (Piconet) เครือข่ายขนาดย่อมหรือเรียกว่า พิโคเน็ต (Piconet) เป็นเครือข่ายที่เกิดขึ้นระหว่างการเชื่อมต่อกันของอุปกรณ์บลูทูธ ตั้งแต่ ๒ ตัวขึ้นไป แต่ไม่เกิน ๘ ตัว ซึ่งจะแบ่งช่องสัญญาณออกเป็น ๗๙ ช่องสัญญาณ และส่งข้อมูลสลับช่องไปมา ๑,๖๐๐ ครั้งต่อวินาที ทำให้แต่ละพิโคเน็ตสามารถทำงานในพื้นที่เดียวกันได้ โดยโอกาสในการถูกรบกวนจากเครือข่ายอื่นที่อยู่ใกล้มีเพียงร้อยละ ๑.๕ ทั้งนี้เครือข่ายบลูทูธได้ออกแบบให้เครื่องที่เป็นตัวแม่ข่ายมีหน้าที่ในการจัดการควบคุมลำดับการส่งข้อมูลของอุปกรณ์แต่ละตัว เพื่อป้องกันการเกิดการชนกันของข้อมูลอันเนื่องมาจากการส่งข้อมูลพร้อมกันบนช่องสัญญาณเดียวกัน

นอกจากนี้ในแต่ละพิโคเน็ตสามารถเชื่อมต่อข้ามเครือข่ายกัน เกิดเป็นเครือข่ายที่ใหญ่ขึ้นเรียกว่า สแคทเทอร์เน็ต (Scatternet) แต่การเชื่อมต่อแบบนี้จะต้องมีการจัดลำดับการทำงานบนเครือข่ายที่ยุ่งยากขึ้นและต้องแบ่งความสามารถในการส่งข้อมูลกัน ส่งผลให้ประสิทธิภาพในการทำงานของระบบลดลง

อัตราเร็วในการแลกเปลี่ยนข้อมูล

ความเร็วสูงสุดพื้นฐานในการส่งข้อมูลของแต่ละช่องสัญญาณประมาณ ๑ เมกกะบิตต่อวินาที (Mbps) ทั้งนี้ความเร็วที่ส่งได้จริงอาจน้อยกว่า เนื่องจากความเร็วบางส่วนจะเสียไปจากการควบคุมและจัดการการส่งข้อมูล นอกจากนี้ผู้ใช้ในแต่ละพิโคเน็ต คงต้องแบ่งความสามารถในการส่งข้อมูลกันด้วย ทำให้ความสามารถในการส่งข้อมูลลดลง

(ก)

(ข)

(ค)

รูปที่ 2.33 แสดงรูปแบบการแลกเปลี่ยนข้อมูล

(ก) แบบไม่สมมาตร

(ข) แบบสมมาตร

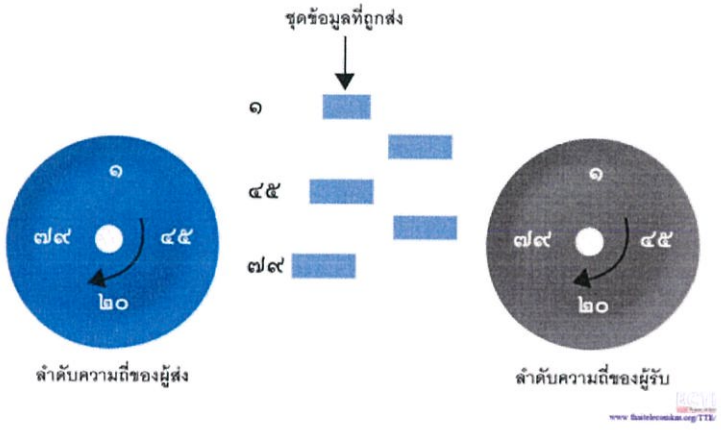
(ค) แบบทำงานร่วมกัน

รูปแบบการแลกเปลี่ยนข้อมูล รูปที่ 2.33 แสดงความแตกต่างในการส่งข้อมูล ซึ่งความสามารถในการส่งข้อมูลจะต่างกันไปตามรูปแบบของการส่ง ดังรูป ก. การส่งข้อมูลแบบไม่สมมาตร ความสามารถในการส่งข้อมูลไปและกลับจะไม่เท่ากัน เหมาะกับการส่งข้อมูลขนาดใหญ่ รูป ข. การส่งข้อมูลแบบสมมาตร ความสามารถในการส่งข้อมูลไปและกลับเท่ากัน และรูป ค. การส่งข้อมูลแบบทำงานร่วมกันหลายอุปกรณ์ แต่ละอุปกรณ์จะแบ่งความสามารถในการส่งข้อมูลกัน ทำให้ความสามารถในการส่งข้อมูลลดลง

พื้นฐานเทคโนโลยีที่ใช้ในบลูทูธ

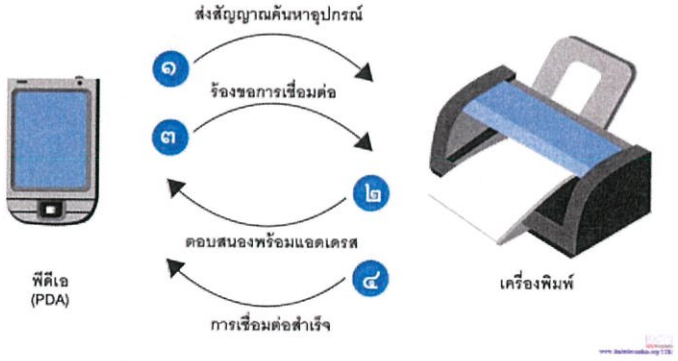
เทคโนโลยีที่สำคัญ ซึ่งใช้ในระบบบลูทูธประกอบไปด้วยสามส่วน คือ

1. การส่งข้อมูลผ่านคลื่นวิทยุด้วยการแผ่สเปกตรัม (Frequency Hopping Spread Spectrum: FHSS) กระบวนการส่งข้อมูลผ่านคลื่นวิทยุ ผู้ส่งจะส่งข้อมูลบนความถี่ซึ่งถูกแบ่งออกเป็น ๗๙ ช่องสัญญาณ โดยส่งข้อมูลไปบนช่องสัญญาณที่เปลี่ยนไปมาอย่างต่อเนื่องตามลำดับที่ได้กำหนดไว้ ผู้รับก็ต้องรับข้อมูลด้วยลำดับความถี่ที่เปลี่ยนแปลงตามความถี่ของช่องสัญญาณที่ผู้ส่งส่งมาในเวลาเดียวกัน ดังรูป 2.34 ทั้งนี้เพื่อป้องกันการดักจับข้อมูลจากผู้ใช้อื่น เนื่องจากลำดับความถี่ที่ส่งจะเข้าใจกันระหว่างผู้รับและผู้ส่งเท่านั้น



รูปที่ 2.34 การส่งข้อมูลผ่านคลื่นวิทยุโดยเปลี่ยนความถี่ไปมา

2. รูปแบบสัญญาณค้นหาอุปกรณ์ (Service discovery protocol) รูปแบบของสัญญาณที่ใช้ในการค้นหาอุปกรณ์ บลูทูธที่อยู่ใกล้เคียงกัน ซึ่งกระบวนการนี้เริ่มจากอุปกรณ์บลูทูธส่งสัญญาณซึ่งมีรูปแบบที่เข้ากันใจระหว่างอุปกรณ์บลูทูธเมื่ออุปกรณ์บลูทูธอื่น พบคลื่นสัญญาณนี้ ก็จะตอบสนองกลับมา โดยส่งแอดเดรสประจำเครื่อง กลับมาด้วยทำให้อุปกรณ์ทั้งสองสามารถเชื่อมต่อกันได้ในขั้นตอนต่อไป



รูปที่ 2.35 แบบการส่งสัญญาณค้นหาอุปกรณ์

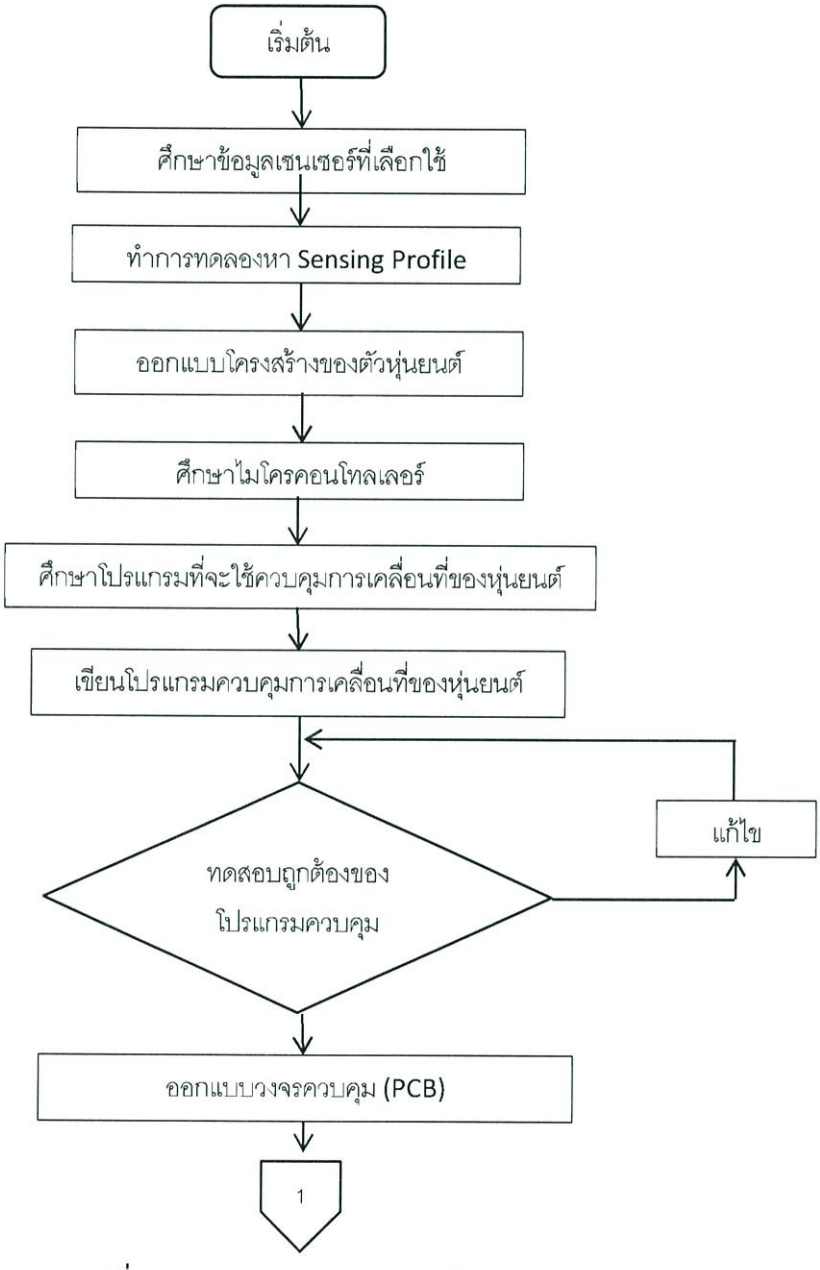
รูปที่ 2.35 แสดงตัวอย่างกระบวนการในการค้นหาอุปกรณ์ของเครื่องพีดีเอ เริ่มจากเครื่องพีดีเอส่งสัญญาณเพื่อตรวจหาอุปกรณ์ใกล้เคียง เมื่อเครื่องพิมพ์ตรวจพบก็จะตอบสนองการทำงานโดยระบุเลขแอดเดรสของตัวเองกลับมายังเครื่องพีดีเอ หลังจากนั้นเครื่องพีดีเอร้องขอการเชื่อมต่อไปยังเครื่องพิมพ์ เมื่อเครื่องพิมพ์ตอบรับการเชื่อมต่อ เครื่องพีดีเอก็จะสามารถใช้งานเครื่องพิมพ์ผ่าน

เครือข่ายบลูทูธได้ ทั้งนี้ ขึ้นอยู่กับซอฟต์แวร์และเงื่อนไขเฉพาะอื่นๆ ของอุปกรณ์ทั้งสองด้วย

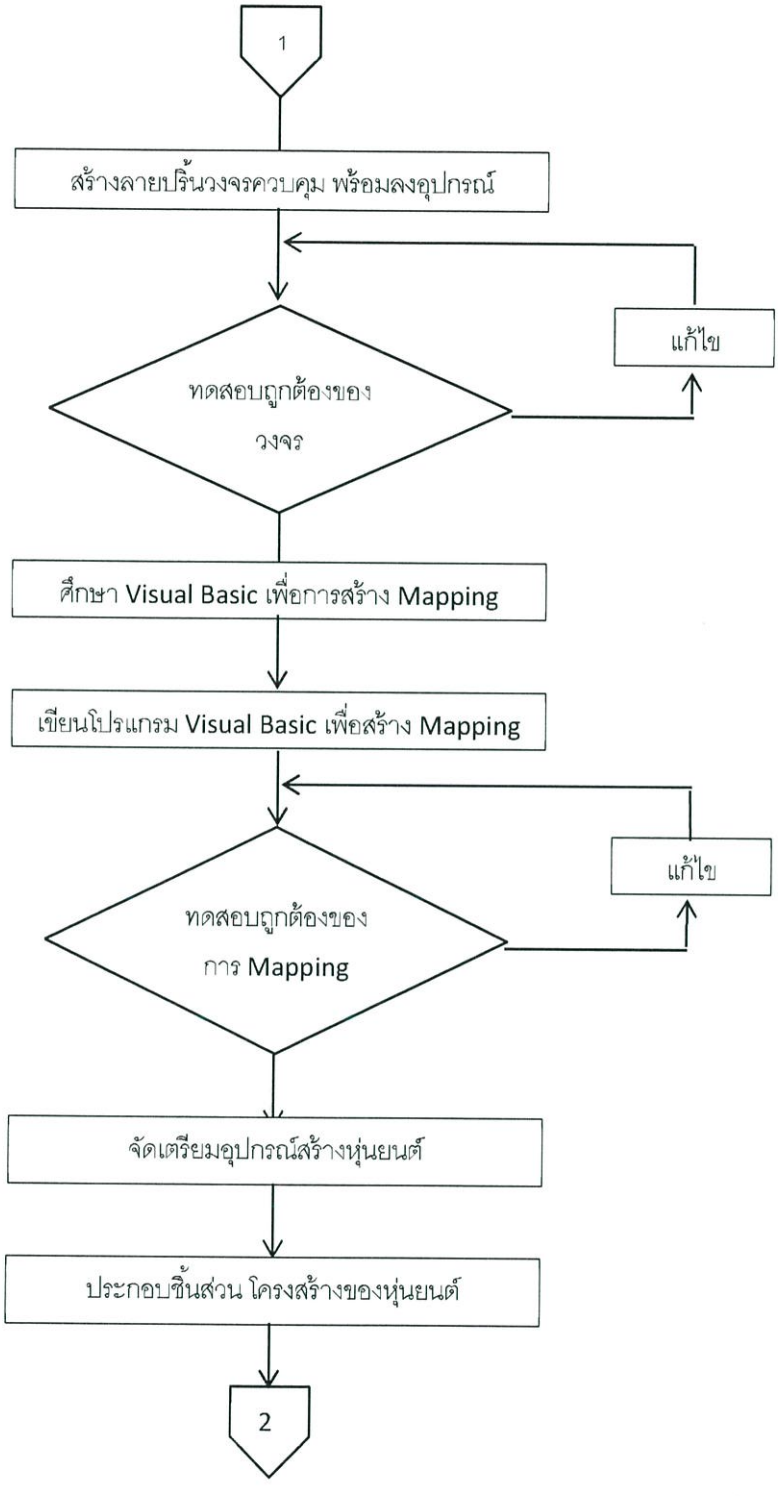
3. รูปแบบการสื่อสาร (Application protocol) คำสั่งหรือกระบวนการที่ถูกใช้โดยโปรแกรมประยุกต์ เพื่อกำหนดขั้นตอน กระบวนการในการรับและส่งข้อมูล ถูกพัฒนาขึ้นมาให้มีความเป็นอิสระ โดยเทคโนโลยีหรือโปรแกรมประยุกต์ทั่วไปสามารถนำไปใช้งานได้ ซึ่งระบบบลูทูธได้นำเอาโพรโทคอลเหล่านี้มาใช้งานในการเชื่อมต่อระหว่างอุปกรณ์การสื่อสารชนิดต่างๆ เนื่องจากรูปแบบการสื่อสารจะแตกต่างกันเช่น การเชื่อมต่อผ่านเครือข่ายบลูทูธระหว่างคอมพิวเตอร์พกพาและเมาส์ (mouse) ใช้โพรโทคอลอาร์เอส - 232 (RS-232) การแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์พกพาและเครื่องพีดีเอใช้โพรโทคอลแลกเปลี่ยนอ็อบเจกต์ (Object exchange : OBEX) และการเชื่อมต่อระหว่างคอมพิวเตอร์พกพาผ่านจุดเชื่อมต่อไปยังอุปกรณ์จัดเส้นทาง (router) ใช้โพรโทคอลจุดต่อจุด (Point to Point protocol : PPP) ดังรูปล่าง

รูปที่ 2.36 รูปแบบการสื่อสาร

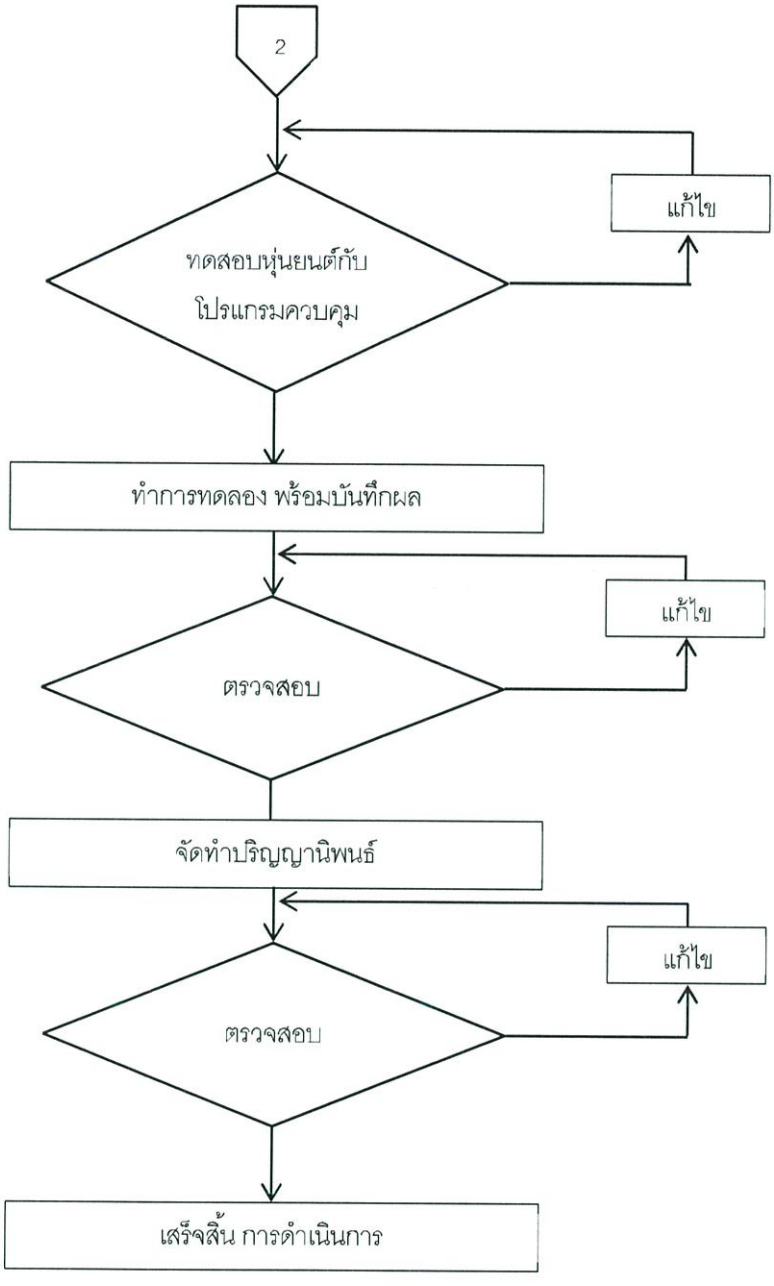
บทที่ 3
ขั้นตอนและวิธีการดำเนินงาน



รูปที่ 3.1 Flow Chart แสดงการทำงาน



รูปที่ 3.2 Flow Chart แสดงการทำงาน (ต่อ)



รูปที่ 3.3 Flow Chart แสดงการทำงาน (ต่อ)

ในการดำเนินงานของโครงงาน ได้มีการวางแผนและจัดให้มีลำดับขั้นตอนการทำงาน เพื่อทำให้สามารถควบคุมและจัดแบ่งเวลาในการทำงานแต่ละช่วงได้อย่างมีประสิทธิภาพ โดยมีขั้นตอนต่างๆ ดังนี้

3.1 ศึกษาข้อมูลเซนเซอร์ที่เลือกใช้

3.2 ทำ Sensing Profile

3.3 ออกแบบโครงสร้างของตัวหุ่นยนต์

3.4 ศึกษาไมโครคอนโทรลเลอร์และโปรแกรมที่ใช้ควบคุม

3.5 ออกแบบและสร้างลายวงจรควบคุม

3.6 ศึกษาและเขียนโปรแกรม Visual Basic สำหรับ Mapping

3.7 ประกอบชิ้นส่วนโครงสร้างหุ่นยนต์ พร้อมทำการทดลอง

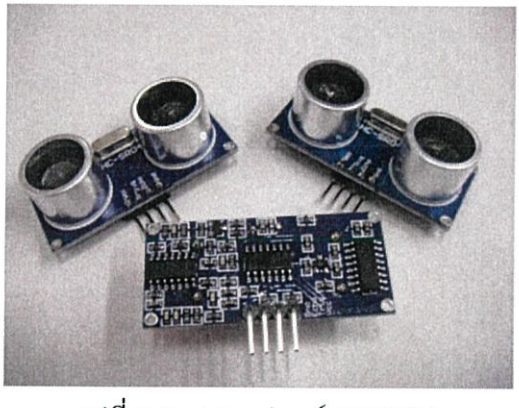
3.8 จัดทำปริญญานิพนธ์

3.1 ศึกษาข้อมูลเซนเซอร์ที่เลือกใช้

ผู้จัดทำได้ทำการศึกษาถึงรายละเอียดในทางเทคนิคต่างๆของอุปกรณ์เซนเซอร์ที่ได้เลือกมาใช้เป็นส่วนประกอบของตัวหุ่นยนต์ โดยสามารถจำแนกประเภทของเซนเซอร์ที่ใช้ได้เป็น 4 ชนิด ดังนี้

3.1.1 อัลตราโซนิคเซนเซอร์

อัลตราโซนิคเซนเซอร์ที่เลือกใช้ เป็นโมดูล HC-SR04 ซึ่งเป็นอุปกรณ์อิเล็กทรอนิกส์ราคาถูก สำหรับวัดระยะห่างด้วยคลื่นอัลตราโซนิค (ใช้คลื่นเสียงความถี่ ประมาณ 40kHz) มีสองส่วนหลักคือ ตัวส่งคลื่นที่ทำหน้าที่สร้างคลื่นเสียงออกไปในการวัดระยะแต่ละครั้ง ("Ping") แล้วเมื่อไปกระทบวัตถุหรือสิ่งกีดขวาง คลื่นเสียงถูกสะท้อนกลับมายังตัวรับแล้วประมวลผลด้วยวงจรอิเล็กทรอนิกส์ภายในโมดูล ถ้าจับเวลาในการเดินทางของคลื่นเสียงในทิศทางไปและกลับ และถ้าทราบความเร็วเสียงในอากาศ ก็จะสามารถคำนวณระยะห่างจากวัตถุกีดขวางได้



รูปที่ 3.4 ภาพอุปกรณ์ HC-SR04

โมดูล HC-SR04 ทำงานที่แรงดันประมาณ +5V (4.5V ถึง +5.5V) โดยป้อนให้ขา VCC และ GND โมดูลนี้ มีขาสัญญาณดิจิทัล TRIG (อินพุต) และ ECHO (เอาต์พุต) ที่นำไปเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้ อย่างเช่น Arduino ในการวัดระยะห่างแต่ละครั้ง จะต้องสร้างสัญญาณแบบ Pulse ที่มีความกว้าง (Pulse Width) อย่างน้อย 10 usec ป้อนให้ขา TRIG และหลังจากนั้นให้วัดความกว้างของสัญญาณช่วง HIGH จากขา ECHO ถ้าวัตถุอยู่ใกล้ ความกว้างของสัญญาณ Pulse ที่ได้ก็จะน้อย แต่ถ้าวัตถุอยู่ไกลออกไป ก็จะได้ค่าความกว้างของสัญญาณ Pulse ที่มากขึ้น การเลือกใช้งานโมดูลประเภทนี้ มีประเด็นที่สำคัญ เช่น ช่วงระยะห่างของการวัด ความกว้างของมุมเมื่อคลื่นเสียงเดินทางออกไปจากตัวส่ง (เรียกว่า Beam Angle) นอกจากนั้น การสะท้อนกลับของคลื่นเสียงที่วัตถุกีดขวาง ขนาดและรูปทรงของวัตถุ และการสะท้อนกลับของเสียงจากหลายทิศทาง หรือต่างระยะกันก็มีผลต่อความถูกต้องหรือความผิดพลาดในการวัดค่าระยะห่างได้เช่นกัน

ข้อมูลเชิงเทคนิคของโมดูล HC-SR04

ใช้แรงดันประมาณ +5V

กินกระแสประมาณ 15mA

ช่วงการวัดระยะทาง (measurement range): ประมาณ 4cm ถึง 4m

ความกว้างเชิงมุมในการวัด (measuring angle): 15 องศา

ความกว้างของสัญญาณ Pulse สำหรับ Trigger: 10 usec

ระดับแรงดันลอจิกสำหรับขา TRIG และ ECHO: 5V TTL

3.1.2 อินฟราเรดเซนเซอร์

อินฟราเรดเซนเซอร์ที่ผู้จัดทำได้เลือกใช้คือ GP2D120 Optoelectronic Device

รูปที่ 3.5 แสดง GP2D120 Optoelectronic Device

คำอธิบาย : เป็นเซนเซอร์วัดระยะทางที่ใช้หลักการ integrated signal processing

<u>คุณสมบัติ</u>

- อนาลอค เอ้าท์พุต
- ระยะวัด 4 – 30 cm
- ระยะเวลาตอบสนอง 39 ms
- กินกระแส 33 mA
- ดีเลย์เริ่มต้น 44 ms

ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

Ta = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V_{CC}	-0.3 to +7	V
Output Terminal Voltage	V_O	-0.3 to (V_{CC} +0.3)	V
Operating Temperature	Topr	-10 to +60	°C
Storage Temperature	Tstg	-40 to +70	°C

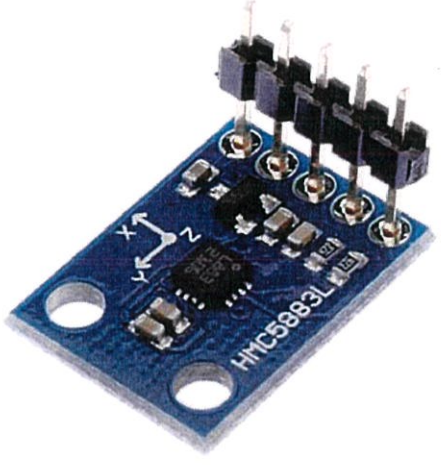
Operating Supply Voltage

PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	V_{CC}	4.5 to 5.5	V

รูปที่ 3.6 แสดงข้อมูลทางเทคนิคของ GP2D120 Optoelectronic Device

3.1.3 Compass Module

เซนเซอร์ที่ใช้ในการตรวจวัดการหมุนและบอกตำแหน่งของตัวหุ่นยนต์ ผู้จัดทำได้เลือกใช้ 3-Axis Digital Compass IC HMC5883L Module ซึ่งเป็นอิเล็คโทรนิกส์โมดูลที่สามารถทำการวัดค่าของสนามแม่เหล็กโลกได้ โดยอาศัยหลักการของ Magneto resistive จากนั้นจึงเปลี่ยนค่าของสนามแม่เหล็กนั้นให้กลายเป็นสัญญาณทางไฟฟ้า ทำให้เราสามารถทราบตำแหน่งของตัวหุ่นยนต์ที่ได้มีการหมุนไป โดยอ้างอิงกับตำแหน่งของสนามแม่เหล็กโลกได้



รูปที่ 3.7 อุปกรณ์ HMC5883L Module

FEATURES	BENEFITS
3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package	Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 5 milli-gauss Resolution in ±8 Gauss Fields	Enables 1° to 2° Degree Compass Heading Accuracy
Built-In Self Test	Enables Low-Cost Functionality Test after Assembly in Production
Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA)	Compatible for Battery Powered Applications
Built-In Strap Drive Circuits	Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
I^2C Digital Interface	Popular Two-Wire Serial Data Interface for Consumer Electronics
Lead Free Package Construction	RoHS Compliance
Wide Magnetic Field Range (+/-8 Oe)	Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
Software and Algorithm Support Available	Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
Fast 160 Hz Maximum Output Rate	Enables Pedestrian Navigation and LBS Applications

รูปที่ 3.8 แสดงข้อมูลทางเทคนิกและคุณสมบัติของ HMC5883L Module

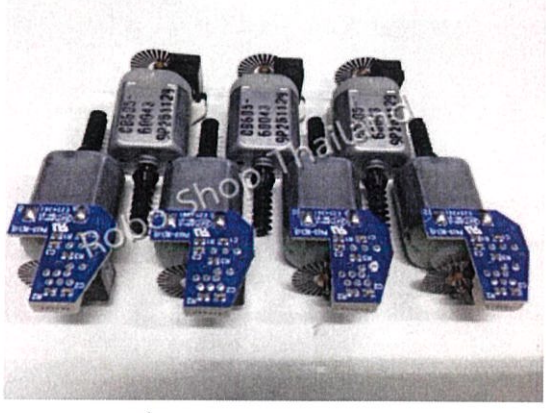
ตารางที่ 3.1 แสดงข้อมูลทางเทคนิคของ HMC5883L Module

HMC5883L

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND	2.16		3.6	Volts
	VDDIO Referenced to DGND	1.71	1.8	VDD+0.1	Volts
Average Current Draw	Idle Mode	-	2	-	µA
	Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V	-	100	-	µA
Performance					
Field Range	Full scale (FS) – total applied field (Typical)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Resolution	VDD=3.0V, GN=2		5		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurment Mode	0.75		75	Hz
	Single Measurement Mode			160	Hz
Measurement Period	From receiving command to data ready		6		msec
Turn-on Time	Ready for I2C commands		200		µs
Gain Tolerance	All gain/dynamic range settings		±5		%
I^2C Address	7-bit address		0x1E		hex
	8-bit read address		0x3D		hex
	8-bit write address		0x3C		hex
I^2C Rate	Controlled by I^2C Master			400	kHz
I^2C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V)		0.2*VDDIO		Volts
	Rise (VDDIO=1.8V)		0.8*VDDIO		Volts
Self Test	X & Y Axes		±1.16		gauss
	Z Axis		±1.08		
	X & Y Axes (GN=100)		510		LSb
	Z Axis (GN=100)				

3.1.4 Encoder เพื่อทราบถึงระยะทางการเคลื่อนที่



รูปที่ 3.9 DC Motor Encoder

เราได้เลือกใช้ DC Motor Encoder เพื่อที่จะทราบระยะทางที่มีการเคลื่อนที่ไป โดยตัว Encoder มีคุณสมบัติดังนี้

Vcc Encoder: 5 V.
Vmotor : min 3V.
ขาต่อใช้งาน 6 ขาดังนี้
Motor. GND. Encoder.
Motor. +5V. Encoder.

3.2 ทำ Sensing Profile

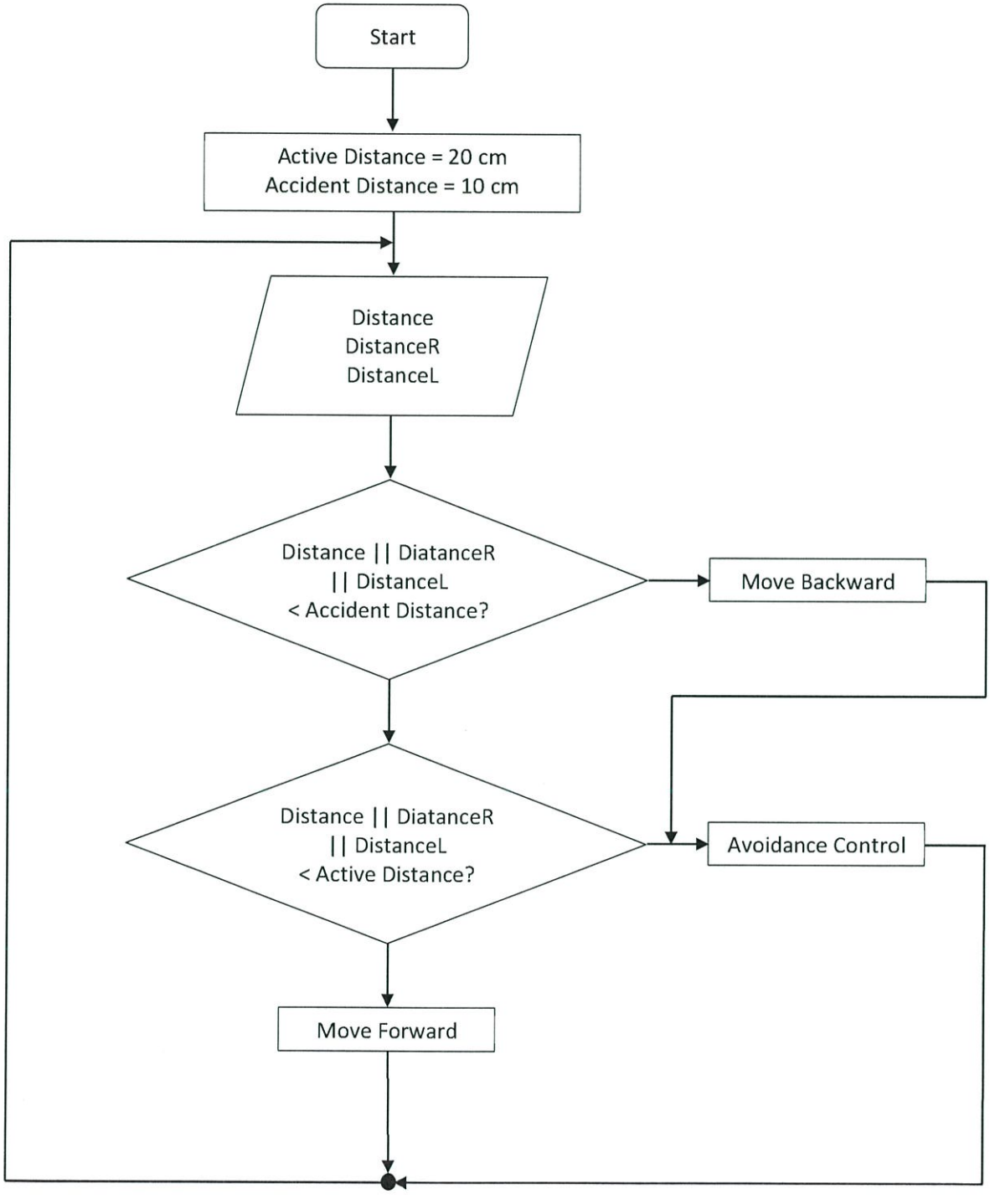
หลังจากที่ได้มีการเลือกเซนเซอร์มาใช้งานแล้ว ขั้นตอนต่อไปคือการทดสอบการทำงานของเซนเซอร์อัลตราโซนิกและอินฟราเรดเซนเซอร์ที่เงื่อนไขต่างๆ เพื่อศึกษาถึงค่าที่ได้จากการ Sensing ว่าเป็นอย่างไร ซึ่งจะได้นำข้อมูลที่ได้จากการทดลองนี้ไปใช้ประโยชน์ต่อ ทั้งในแง่ของการออกแบบการติดตั้งเซนเซอร์บนหุ่นยนต์และการวิเคราะห์ค่าความผิดพลาดที่เกิดขึ้นในการวัดขณะที่หุ่นยนต์ทำงาน จึงเป็นเหตุผลว่าทำไมจึงต้องมีการทำ Sensing Profile ทั้งนี้ในรายละเอียดของการทดลองจะถูกแสดงไว้ในภาคผนวก ง

3.3 ออกแบบโครงสร้างของตัวหุ่นยนต์

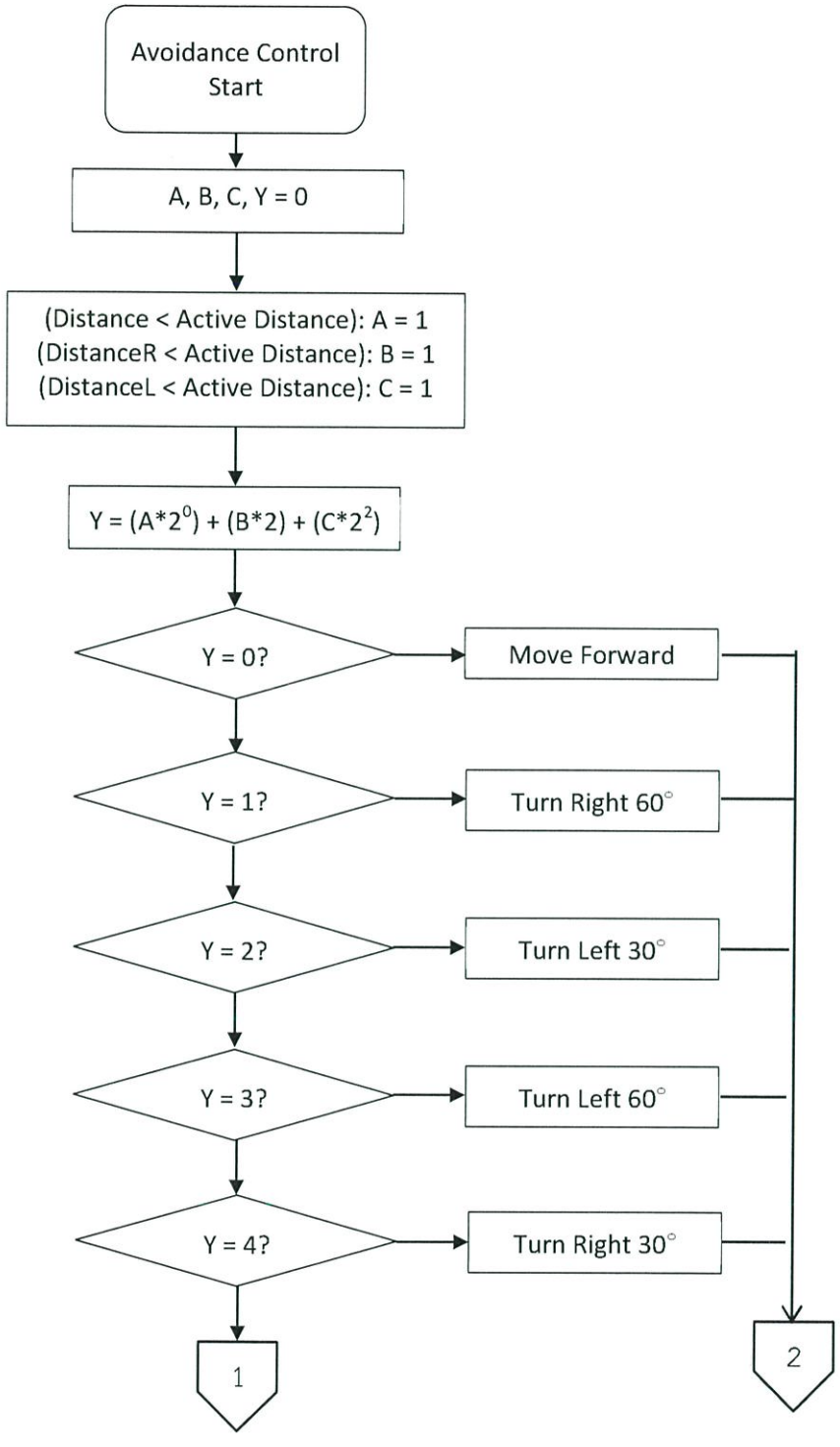
ในการออกแบบหุ่นยนต์นั้นสิ่งที่คำนึงถึงเป็นสำคัญคือ ลักษณะและจำนวนการติดตั้ง Sensor บนตัวหุ่นยนต์ ซึ่งจากการทดลองทำให้เราได้ทราบว่าต้องใช้ เซนเซอร์อัลตราโซนิกทั้งหมด 5 ตัว, อินฟราเรด 3 ตัว, Compass Module 1 ตัว และ DC Motor Encoder 2 ตัว และในส่วนของโครงสร้างของตัวหุ่น วัสดุที่เลือกใช้จะใช้เป็นอะครีลิค เนื่องจากมีความแข็งแรงทนทานและมีน้ำหนักเบา จะทำให้มอเตอร์สามารถลดการรับภาระโหลดอันเนื่องจากน้ำหนักที่ไม่จำเป็นออกไปได้

3.4 ศึกษาไมโครคอนโทรลเลอร์และโปรแกรมที่ใช้ควบคุม

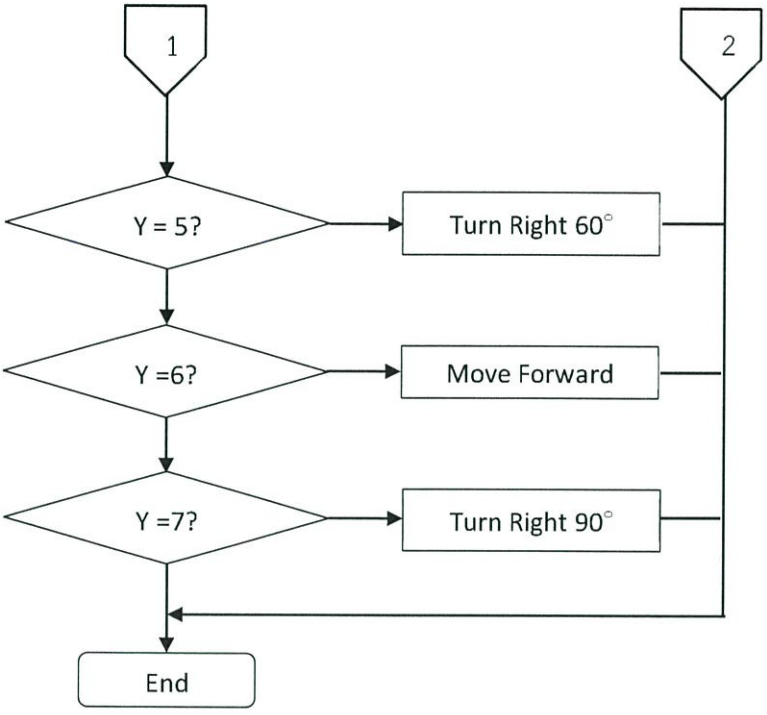
ไมโครคอนโทรลเลอร์ที่ใช้เป็น Arduino Mega 2560 ซึ่งได้กล่าวถึงรายละเอียดไปแล้วในบทที่ 2 ในส่วนของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์นั้น ผู้จัดทำได้ใช้โปรแกรม Arduino IDE สั่งการควบคุมไปยัง Arduino Mega 2560 ซึ่งได้มีการออกแบบอัลกอริทึมการทำงานไว้ดังนี้ (รายละเอียดโค้ดโปรแกรมอยู่ในภาคผนวก ก)



รูปที่ 3.10 Flow Chart แสดงการทำงานของโปรแกรมควบคุมการเคลื่อนที่ใน Arduino IDE



รูปที่ 3.11 Flow Chart แสดงการทำงานของ Avoidance Control loop



รูปที่ 3.12 Flow Chart แสดงการทำงานของ Avoidance Control loop (ต่อ)

3.5 ออกแบบและสร้างลายวงจรควบคุม

ออกแบบวงจรและสร้างลายปริ้นวงจรจากโปรแกรม Altium ซึ่งภายในวงจรจะประกอบไปด้วยส่วนของวงจรไดร์ฟมอเตอร์ วงจรกำจัดสัญญาณรบกวนของอินฟราเรดเซนเซอร์ วงจรบลูทูธ และซอคเก็ตต่างๆ รวมถึงอุปกรณ์อิเล็คทรอนิคส์ทั้งหมดที่ใช้งานในหุ่นยนต์จะถูกรวมไว้ใน PCB เดียว ซึ่งสามารถดูภาพการออกแบบจากโปรแกรม Altium ได้จากภาคผนวก จ

3.6 ศึกษาและเขียนโปรแกรม Visual Basic สำหรับ Mapping

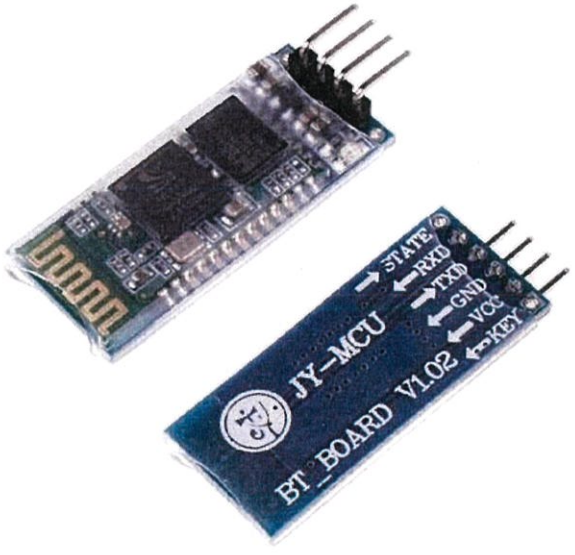
3.6.1 ศึกษาการสื่อสารระหว่าง Arduino กับ Visual basic ผ่านพอร์ตอนุกรมและบลูทูธ ซึ่งอุปกรณ์บลูทูธที่ได้เลือกมาใช้งานเพื่อการสื่อสารและส่งข้อมูลนั้น คือ HC-05 Bluetooth Module HC-05 Bluetooth Module เป็นโมดูลไร้สายที่ใช้สื่อสารกันด้วย Bluetooth SPP (Serial Port Protocol) โดยSerial Port เป็น Bluetooth V2.0+EDR (Enhance Data Rate) 3Mbps Modulation พร้อมกับความถี่ใช้งาน 2.4 GHz สามารถเชื่อมต่อกับอุปกรณ์บลูทูธอื่นๆได้ รวมทั้งเชื่อมต่อด้วย Serial Interface ระหว่างคอมพิวเตอร์ด้วยวงจร RS232 เพื่อเข้าสู่ AT Command การปรับแต่งค่าต่างๆของโมดูล โมดูลรุ่นนี้สามารถตั้งค่าให้เป็นได้ทั้ง Master และ Slave Mode ซึ่งรายละเอียดจะอยู่ในส่วนของ AT Command

คุณสมบัติต่างๆ

- ความตอบสนองสัญญาณ -80dBm
- กำลังส่งคลื่นวิทยุเพิ่มได้ถึง +4dbm
- แรงดันทำงาน 1.8-3.6v
- PIO Control
- UART Interface พร้อมกับสามารถปรับแต่ง baud rate ได้
- เสาอากาศในตัว

รายละเอียดการทำงานต่างๆ

- Baud Rate ปกติ : 38400, Data bits:8, Stop Bits:1, Parity: No
- รองรับ Baud Rate : 9600, 19200, 38400, 57600, 115200, 230400, 460800

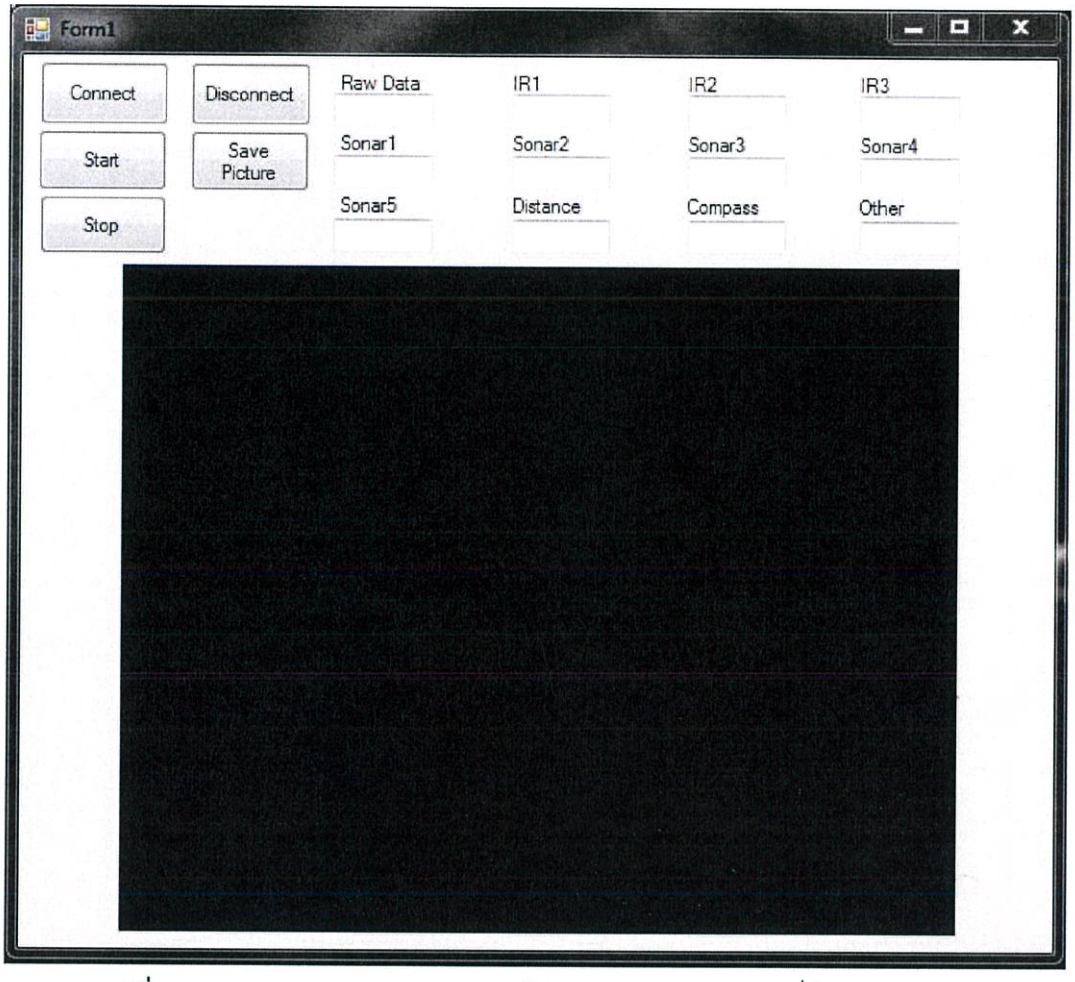


รูปที่ 3.13 แสดง HC-05 Bluetooth Module

3.6.2 ศึกษาการรับข้อมูลจากเซนเซอร์ผ่าน Arduino แล้วนำไปประมวลผลใน Visual basic

3.6.3 ศึกษาการเขียน Graphics เบื้องต้น ใน Visual basic เพื่อเขียน Mapping เส้นทางของหุ่นยนต์

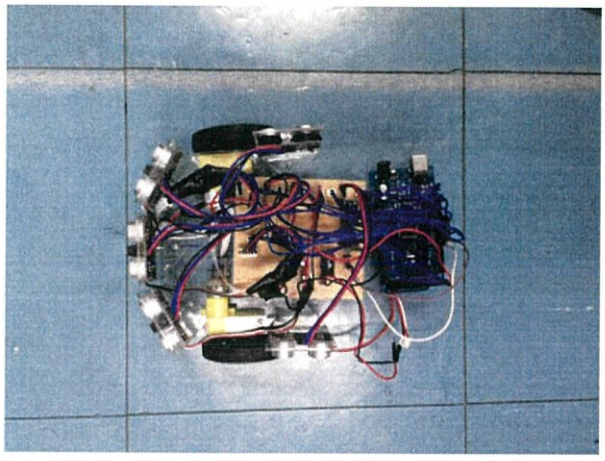
3.6.4 เขียนโปรแกรม Visual basic เพื่อสื่อสารและรับข้อมูลจาก Arduino จากนั้นนำไปประมวลผลและเขียน Mapping เส้นทางของหุ่นยนต์(รายละเอียดโค้ดอยู่ในภาคผนวก ข)



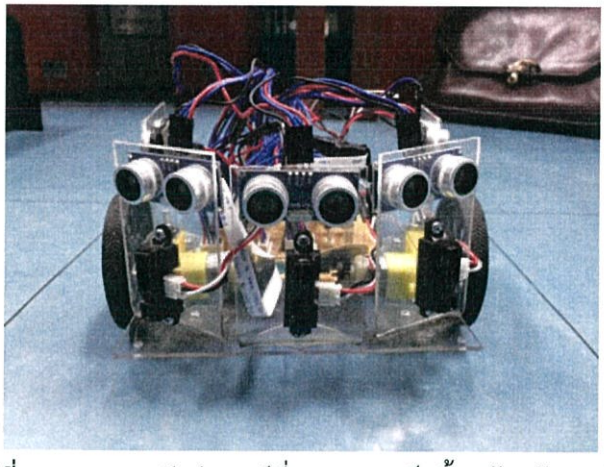
รูปที่ 3.14 แสดง User Interface ของโปรแกรม Visual Basic ที่ทำการออกแบบ

3.7 ประกอบชิ้นส่วนโครงสร้างหุ่นยนต์ พร้อมทำการทดลอง

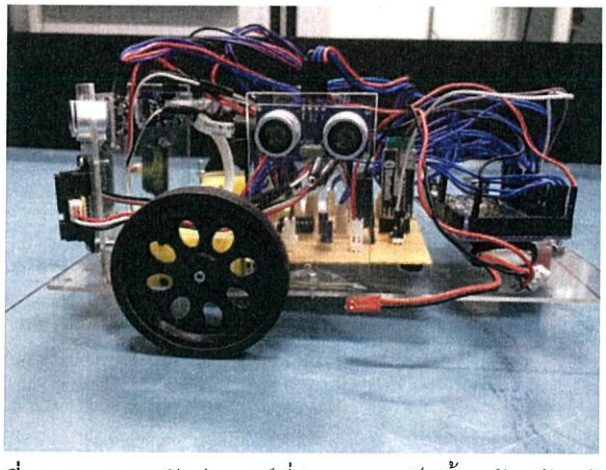
นำส่วนประกอบต่างๆ ทั้งส่วนของเซนเซอร์ โครงรถ มอเตอร์ แผงวงจรที่ทำการกัดปริ้นแล้ว ฯ มาประกอบเข้าด้วยกัน จากนั้นรวมการทำงานในส่วนของโปรแกรม ทั้งโปรแกรมควบคุม และโปรแกรมในการ Mapping ให้สามารถทำงานร่วมกันได้ สุดท้ายจึงทำการทดสอบการทำงานทุกส่วนร่วมกัน ทั้งฮาร์ดแวร์และซอฟต์แวร์ ภายใต้เงื่อนไขการทดลองที่ได้กำหนดขึ้น จากนั้นทำการสังเกตุและ บันทึกผล ซึ่งจะได้กล่าวถึงรายละเอียดของการทดลองและผลการทดลองต่อไป ในบทที่ 4



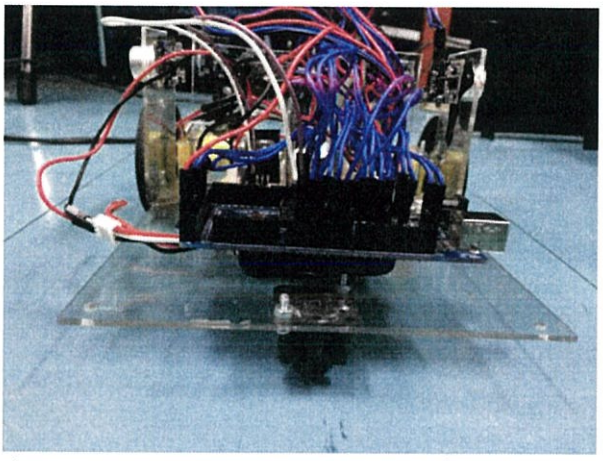
รูปที่ 3.15 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านบน)



รูปที่ 3.16 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านหน้า)



รูปที่ 3.17 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านข้าง)



รูปที่ 3.18 แสดงตัวหุ่นยนต์ที่ประกอบเสร็จสิ้นแล้ว (ด้านหลัง)

3.8 จัดทำปริญญานิพนธ์

รวบรวมเนื้อหาและจัดพิมพ์รูปเล่ม เพื่อเป็นพื้นฐานในการศึกษาและพัฒนาต่อไป

บทที่ 4
การทดลองและผลการทดลอง

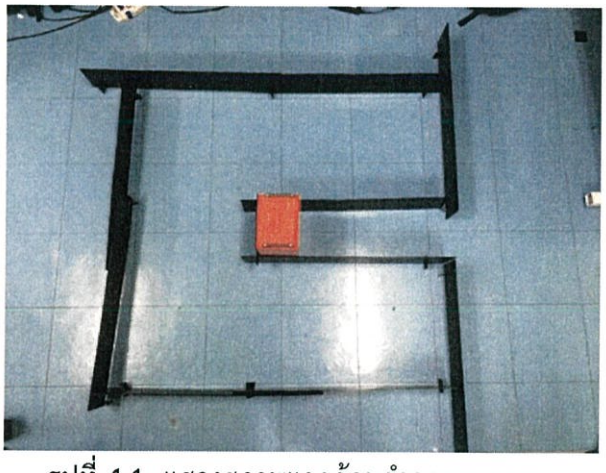
4.1 กล่าวนำ

การศึกษาและพัฒนาหุ่นยนต์หลบหลีกสิ่งกีดขวาง เพื่อใช้ในการหลบหลีกสิ่งกีดขวางและสำรวจเส้นทางที่หุ่นยนต์สามารถเดินทางไปได้ โดยทำการปล่อยหุ่นยนต์ให้วิ่งไปตามเส้นทางต่างๆ และส่งข้อมูลเส้นทาง ตำแหน่ง ตลอดจนสิ่งกีดขวางที่มีอยู่ในระหว่างการเดินทางของหุ่นยนต์ กลับมายังคอมพิวเตอร์เพื่อบันทึกและสร้างเป็นแผนที่ (Map) สำหรับใช้เป็นความรู้และแผนที่ต้นแบบให้กับผู้ใช้งานหรือหุ่นยนต์ตัวอื่นที่ต้องการเดินทางในภายหลัง

ทั้งนี้เพื่อเป็นการศึกษาและวิเคราะห์การทำงานหุ่นยนต์ รวมทั้งความถูกต้องของข้อมูลที่ได้จากการเดินทางและการสร้างแผนที่ต้นแบบ จึงได้ออกแบบการทดลองเพื่อสำรวจเส้นทางที่กำหนดเปรียบเทียบกับเส้นทางหรือแผนที่ที่ได้จากการสำรวจของหุ่นยนต์ เพื่อใช้เป็นข้อมูลในการพัฒนาและเพิ่มศักยภาพของหุ่นยนต์ในอนาคต

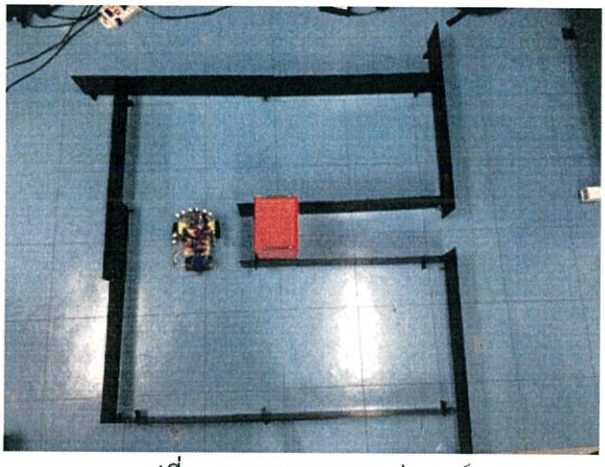
4.2 ขั้นตอนการทดลอง

4.2.1 สร้างสภาพแวดล้อมจำลองแบบ ก. ดังภาพ



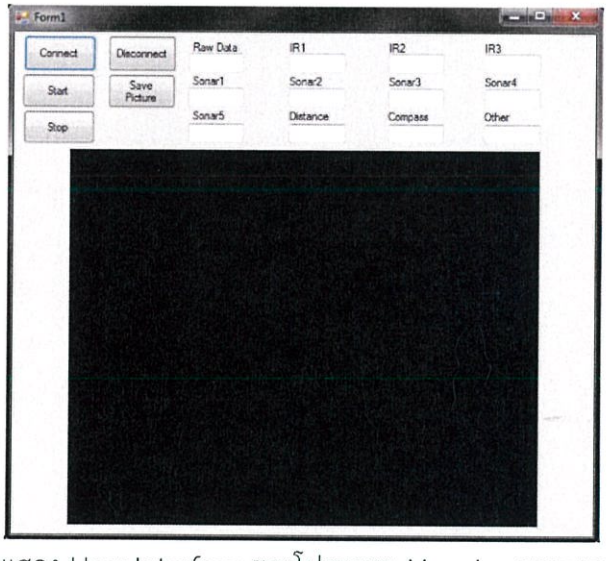
รูปที่ 4.1 แสดงสภาพแวดล้อมจำลอง แบบ ก.

4.2.2 วางหุ่นยนต์ไว้ ณ จุดเริ่มต้นที่ต้องการ

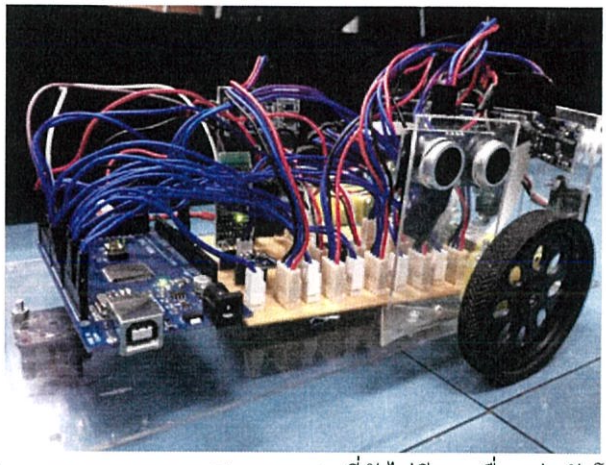


รูปที่ 4.2 แสดงการวางหุ่นยนต์

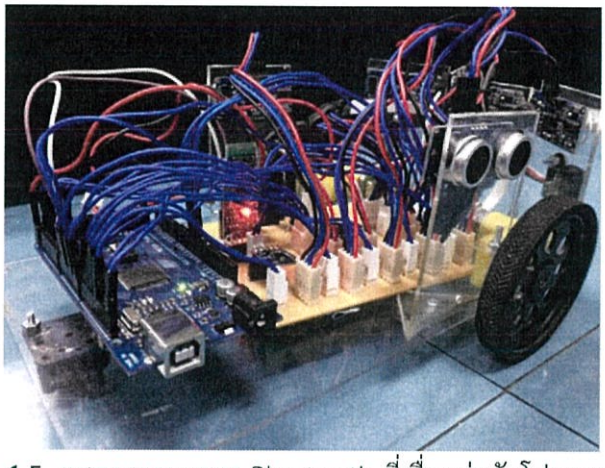
4.2.3 เริ่มการทำงานโปรแกรม Mapping จาก Visual Basic เพื่อสั่งให้หุ่นยนต์ทำงาน โดยเริ่มที่การกด Connect เพื่อเชื่อมต่ออุปกรณ์สื่อสารข้อมูล (Bluetooth) ของตัวหุ่นยนต์กับโปรแกรม Mapping จากนั้นรอจนกว่าไฟกระพริบสีเขียวบนโมดูล Bluetooth จะกลายเป็นไฟสีแดง



รูปที่ 4.3 แสดง User Interface ของโปรแกรม Mapping ขณะกด Connect

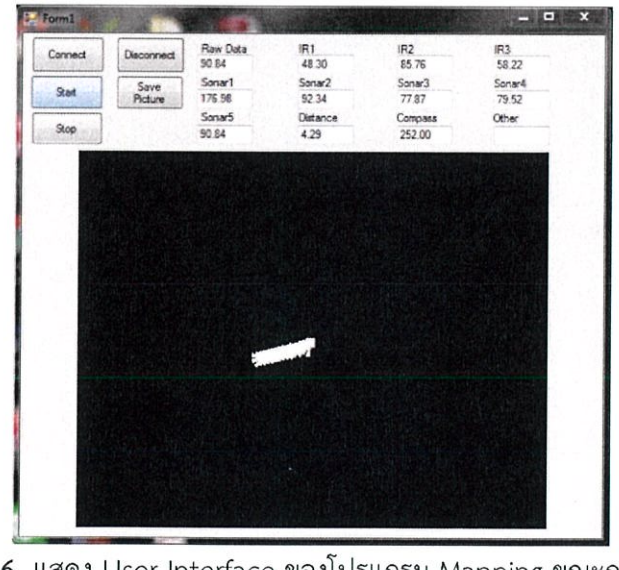


รูปที่ 4.4 แสดงสถานะของ Bluetooth ที่ยังไม่มีการเชื่อมต่อกับโปรแกรม

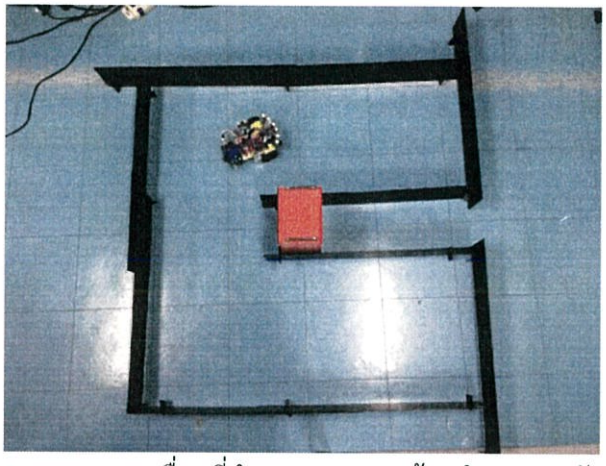


รูปที่ 4.5 แสดงสถานะของ Bluetooth ที่เชื่อมต่อกับโปแกรมแล้ว

4.2.4 กด Start เพื่อเริ่มต้นการสำรวจของหุ่นยนต์

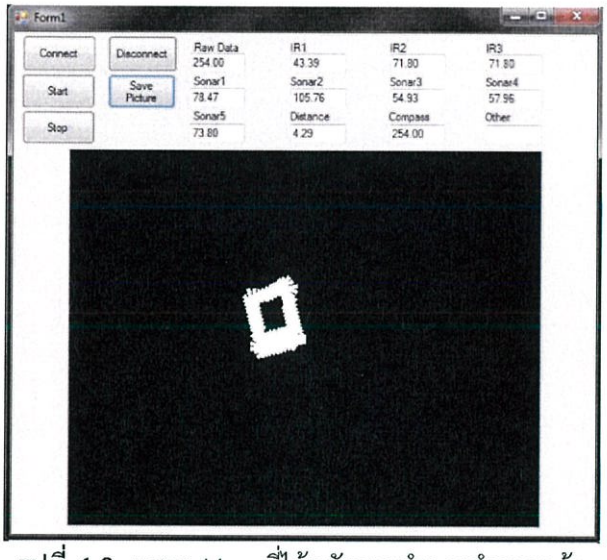


รูปที่ 4.6 แสดง User Interface ของโปรแกรม Mapping ขณะกด Start

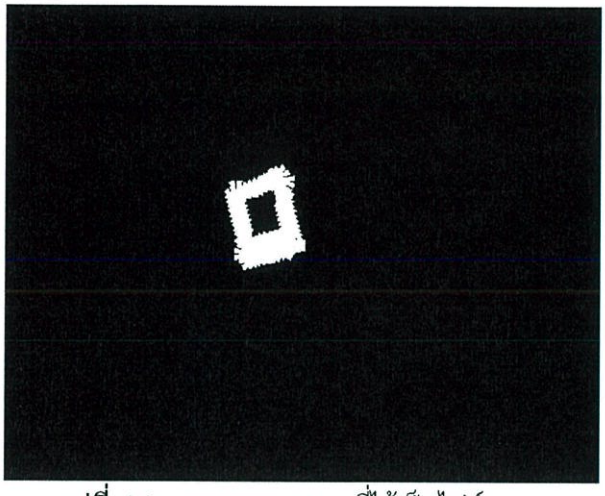


รูปที่ 4.7 แสดงการเคลื่อนที่สำรวจสภาพแวดล้อมจำลองของตัวหุ่นยนต์

4.2.5 หลังจากปล่อยให้หุ่นยนต์ทำการสำรวจจนถึงระยะเวลาที่ผู้สังเกตุการณ์พึงพอใจแล้ว ให้กด Save Picture เพื่อเก็บภาพ Map ที่ได้จากการสำรวจ

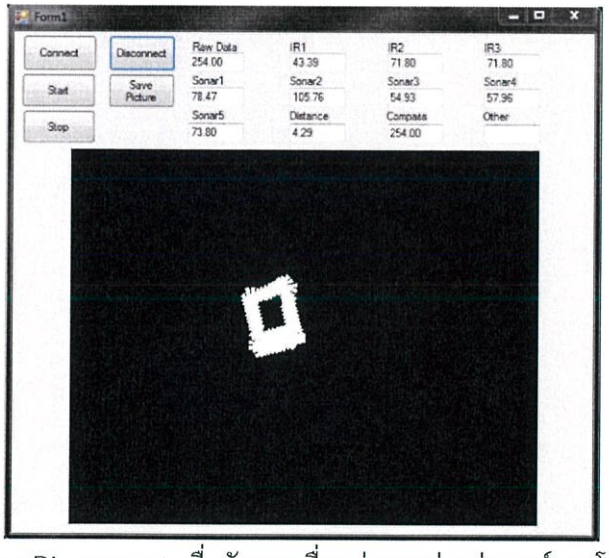


รูปที่ 4.8 แสดง Map ที่ได้หลังจากทำการสำรวจแล้ว



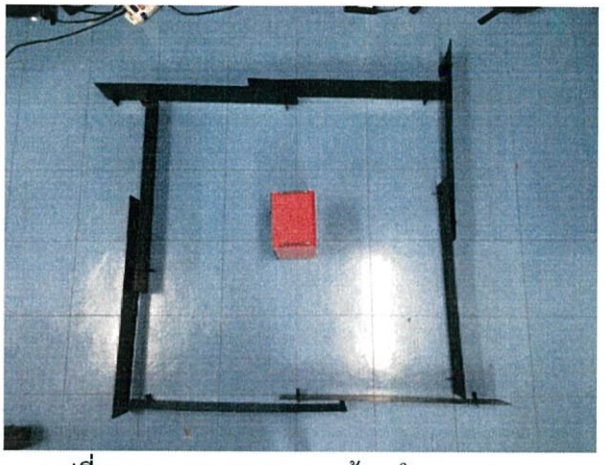
รูปที่ 4.9 แสดงภาพ Map ที่ได้เป็นไฟล์ .png

4.2.6 จากนั้นกด Disconnect เพื่อตัดการเชื่อมต่อระหว่างหุ่นยนต์กับโปรแกรม Mapping เป็นอันเสร็จสิ้นการสำรวจในสภาพแวดล้อมจำลองแบบ ก.



รูปที่ 4.10 แสดงการ Disconnect เพื่อตัดการเชื่อมต่อระหว่างหุ่นยนต์และโปรแกรม Mapping

4.2.7 ให้ดำเนินการทดลองทั้งหมด 3 ครั้ง ต่อหนึ่งสภาพแวดล้อมจำลอง จากนั้นให้ทำการทดลองตามขั้นตอนเดียวกันกับที่กล่าวไปข้างต้นทั้งหมดกับสภาพแวดล้อมจำลองแบบ ข. และแบบ ค. ตามลำดับ

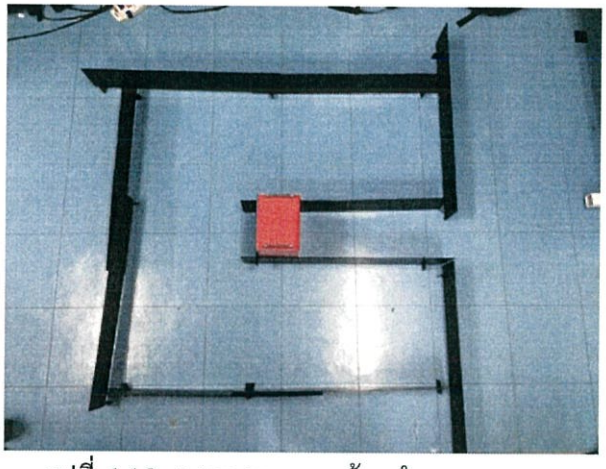


รูปที่ 4.11 แสดงสภาพแวดล้อมจำลอง แบบ ข.

รูปที่ 4.12 แสดงสภาพแวดล้อมจำลอง แบบ ค.

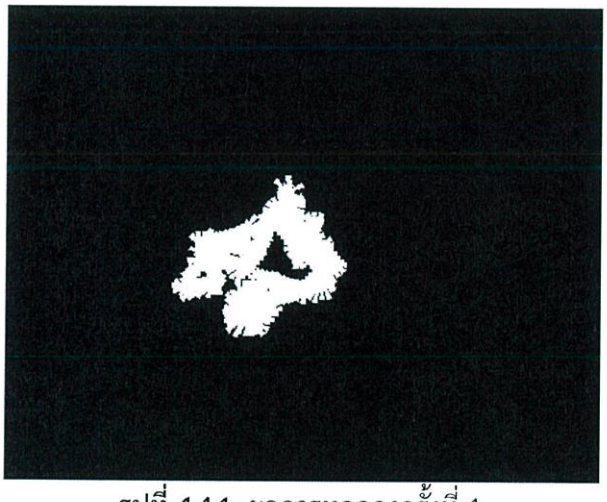
4.3 ผลการทดลอง

4.3.1 ผลการทดลองกับสภาพแวดล้อมแบบ ก.



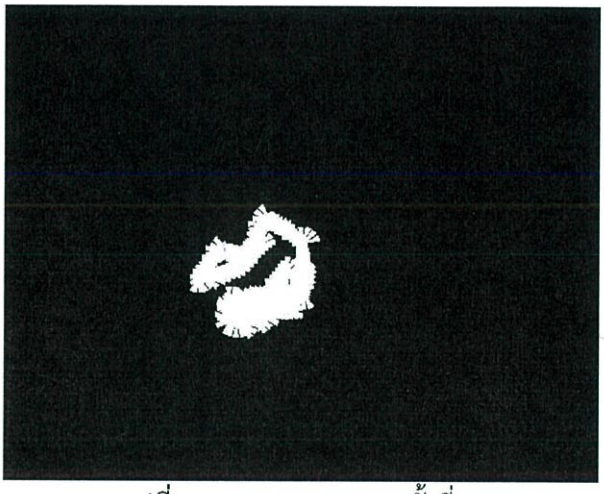
รูปที่ 4.13 แสดงสภาพแวดล้อมจำลอง แบบ ก.

4.3.1.1 ผลการทดลองครั้งที่ 1



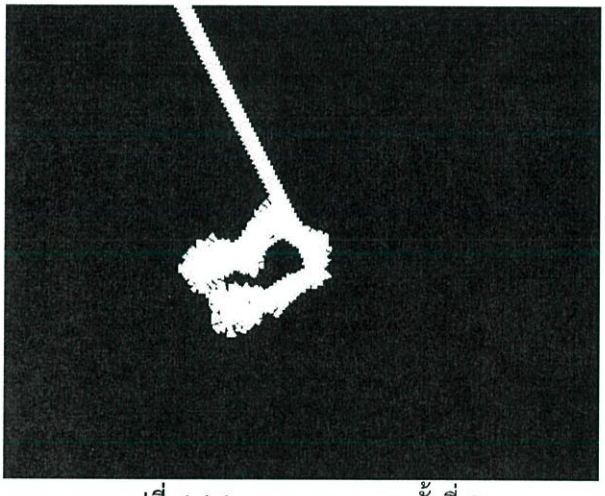
รูปที่ 4.14 ผลการทดลองครั้งที่ 1

4.3.1.2 ผลการทดลองครั้งที่ 2



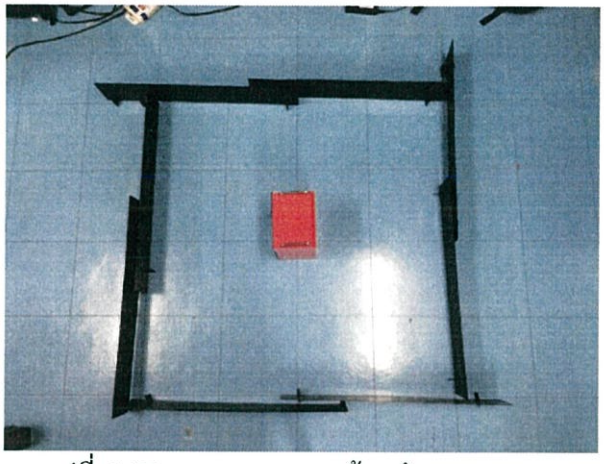
รูปที่ 4.15 ผลการทดลองครั้งที่ 2

4.3.1.3 ผลการทดลองครั้งที่ 3



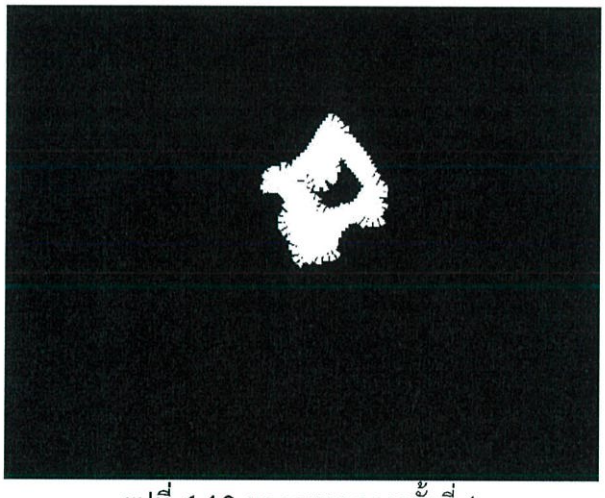
รูปที่ 4.16 ผลการทดลองครั้งที่ 3

4.3.2 ผลการทดลองกับสภาพแวดล้อมแบบ ข.



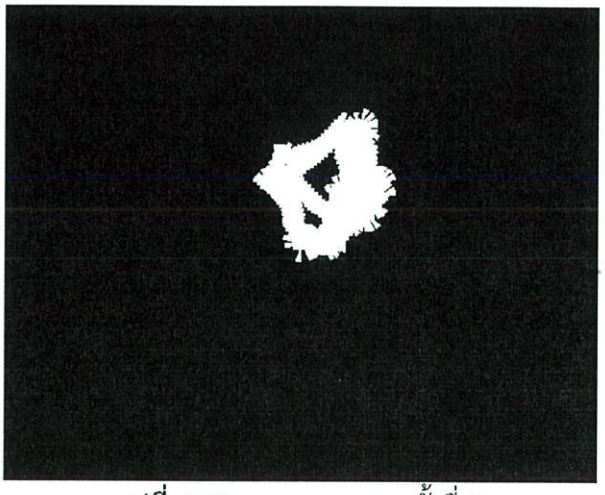
รูปที่ 4.17 แสดงสภาพแวดล้อมจำลอง แบบ ข.

4.3.2.1 ผลการทดลองครั้งที่ 1



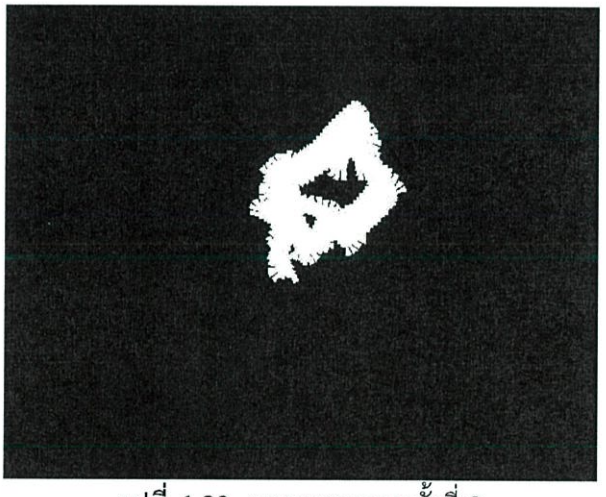
รูปที่ 4.18 ผลการทดลองครั้งที่ 1

4.3.2.2 ผลการทดลองครั้งที่ 2



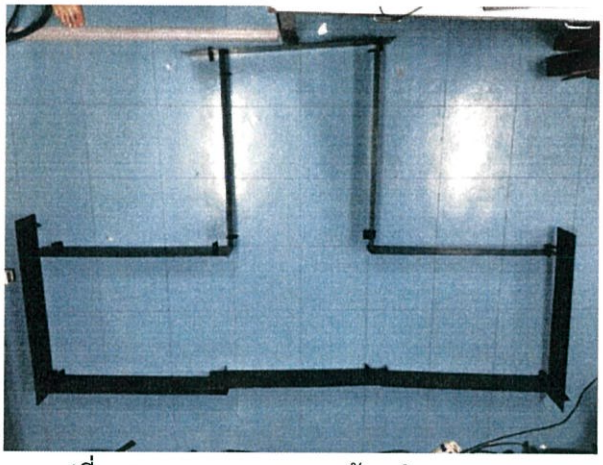
รูปที่ 4.19 ผลการทดลองครั้งที่ 2

4.3.2.3 ผลการทดลองครั้งที่ 3



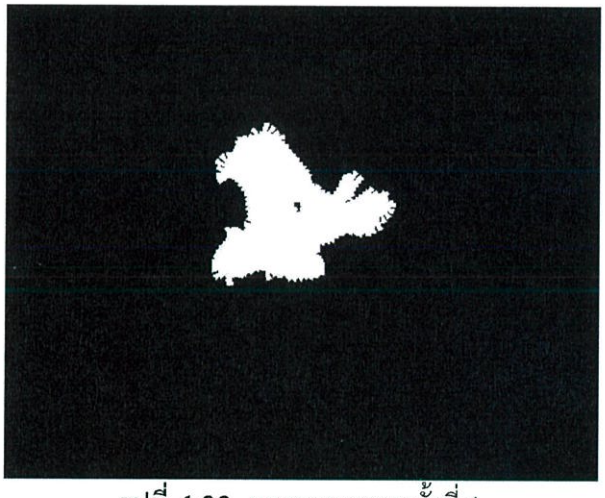
รูปที่ 4.20 ผลการทดลองครั้งที่ 3

4.3.3 ผลการทดลองกับสภาพแวดล้อมแบบ ค.



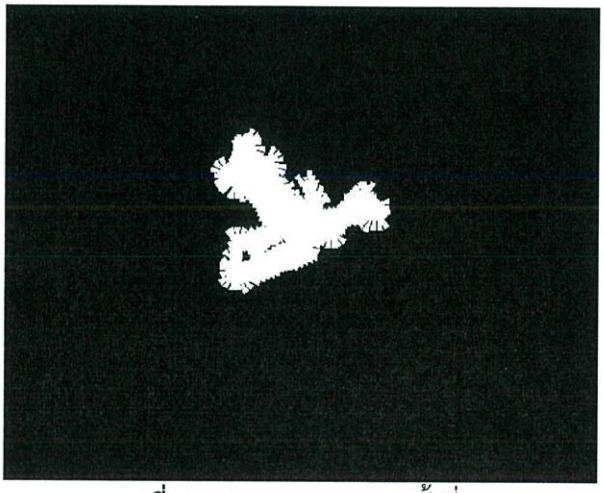
รูปที่ 4.21 แสดงสภาพแวดล้อมจำลอง แบบ ค.

4.3.3.1 ผลการทดลองครั้งที่ 1



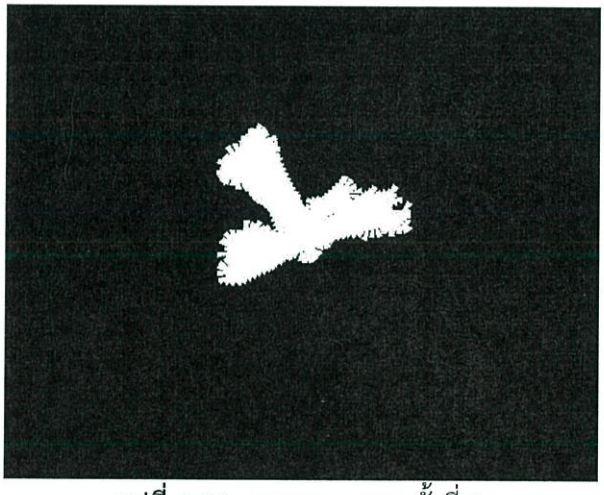
รูปที่ 4.22 ผลการทดลองครั้งที่ 1

4.3.3.2 ผลการทดลองครั้งที่ 2



รูปที่ 4.23 ผลการทดลองครั้งที่ 2

4.3.3.3 ผลการทดลองครั้งที่ 3



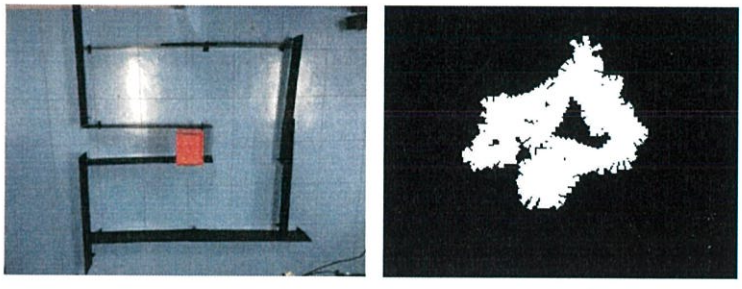
รูปที่ 4.24 ผลการทดลองครั้งที่ 3

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากการทดลองเพื่อศึกษาและพัฒนาหุ่นยนต์หลบหลีกสิ่งกีดขวางและสำรวจเส้นทางที่หุ่นยนต์สามารถเดินทางไปได้ โดยการปล่อยหุ่นยนต์ให้สำรวจพื้นที่หรือเส้นทางที่กำหนด แล้วนำแผนที่ต้นแบบที่ได้จากการสำรวจของหุ่นยนต์ไปเปรียบเทียบกับพื้นที่จริง ซึ่งได้ทดลองในพื้นที่ที่แตกต่างกัน 3 แบบดังนี้

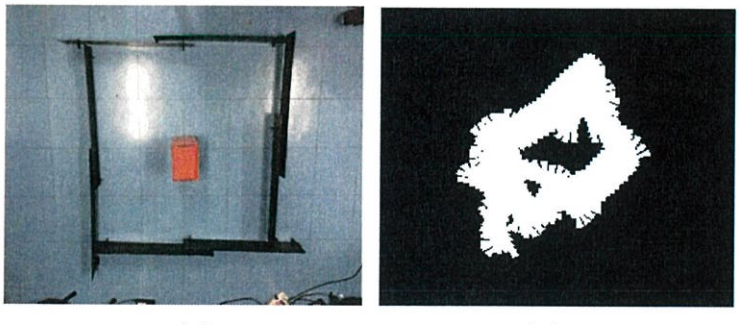
<u>แบบที่ 1</u>



(ก) (ข)

รูปที่ 5.1 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 1 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้

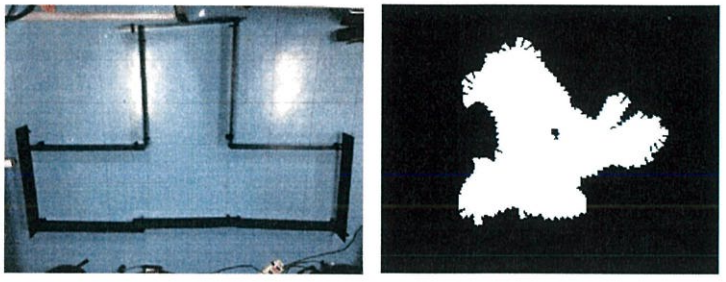
<u>แบบที่ 2</u>



(ก) (ข)

รูปที่ 5.2 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 2 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้

แบบที่ 3

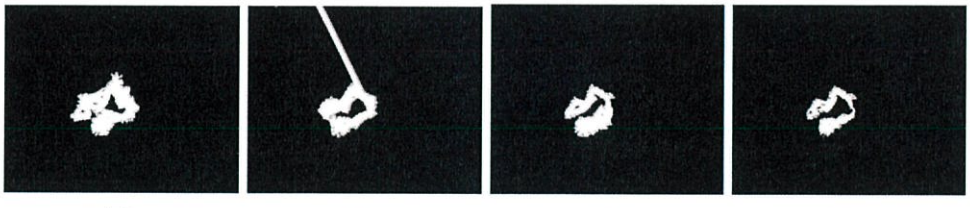


(ก) (ข)

รูปที่ 5.3 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 3 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้

ซึ่งจากการทดลอง จะเห็นได้ว่ารูปหรือแผนที่ที่ได้จากการสำรวจของหุ่นยนต์ มีความคล้ายคลึงกับพื้นที่จริงแต่ยังมีความคลาดเคลื่อนอยู่ ทั้งนี้อาจเป็นผลมาจากความผิดพลาดและความไม่แน่นอนในการวัดของเซนเซอร์ รวมทั้งความเสถียรของฮาร์ดแวร์และอุปกรณ์ที่ใช้ในการทำหุ่นยนต์ เช่น มอเตอร์ ล้อ เกียร์ ฯลฯ ทั้งนี้ทางทีมงานจึงได้ใช้เทคนิคทางสถิติ ผ่านการประมวลผลด้วยภาพ โดยการทำการสำรวจซ้ำหลายๆครั้ง แล้วนำเอารูปหรือแผนที่ที่ได้มาทำการประมวลผลกัน ในลักษะของสีที่เหมือนหรือแตกต่างกัน ซึ่งจะทำให้ได้ภาพหรือแผนที่ที่มีความถูกต้องใกล้เคียงกับพื้นที่จริงมากยิ่งขึ้น

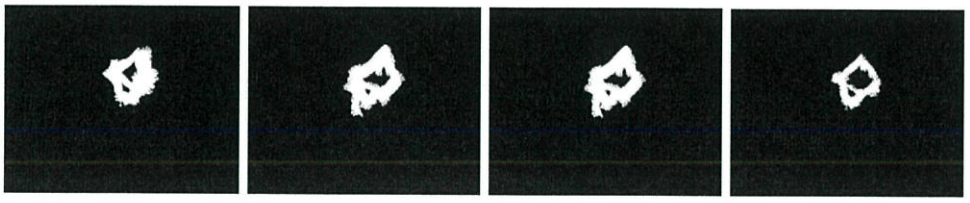
แบบที่ 1



(ก) (ข) (ค) (ง)

รูปที่ 5.4 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 1
(ก) สำรวจครั้งที่ 1
(ข) สำรวจครั้งที่ 2
(ค) สำรวจครั้งที่ 3
(ง) รูปที่ผ่านการประมวลผล

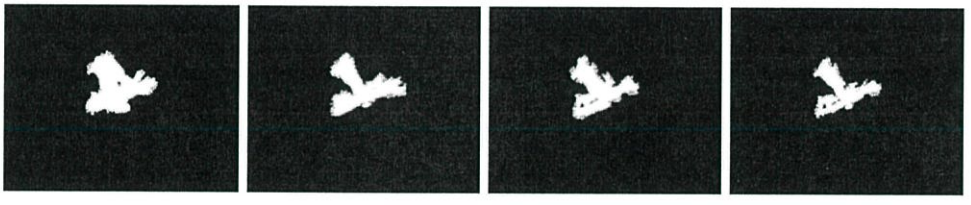
แบบที่ 2



(ก) (ข) (ค) (ง)

รูปที่ 5.5 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 2
(ก) สำรวจครั้งที่ 1
(ข) สำรวจครั้งที่ 2
(ค) สำรวจครั้งที่ 3
(ง) รูปที่ผ่านการประมวลผล

แบบที่ 3

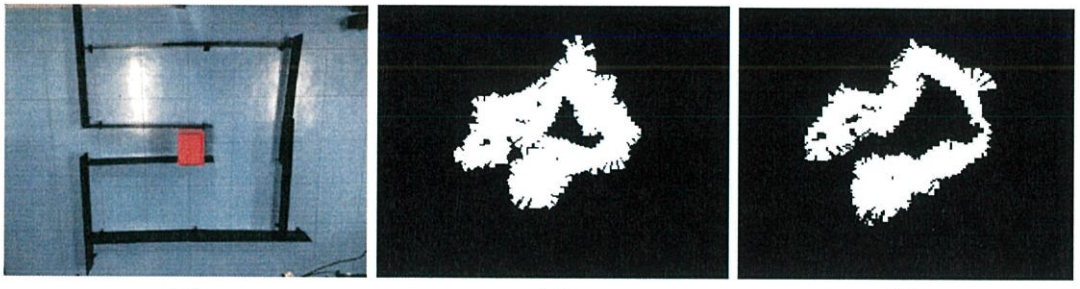


(ก) (ข) (ค) (ง)

รูปที่ 5.6 แสดงการใช้การประมวลผลด้วยภาพเพื่อเพิ่มความถูกต้องในการสำรวจพื้นที่แบบที่ 3
(ก) สำรวจครั้งที่ 1
(ข) สำรวจครั้งที่ 2
(ค) สำรวจครั้งที่ 3
(ง) รูปที่ผ่านการประมวลผล

ซึ่งเมื่อนำเอารูปที่ได้จากการประมวลผลมาทำเปรียบเทียบกับรูปเดิมก่อนการประมวลผลจะได้ผลดังนี้

<u>แบบที่ 1</u>



(ก) (ข) (ค)

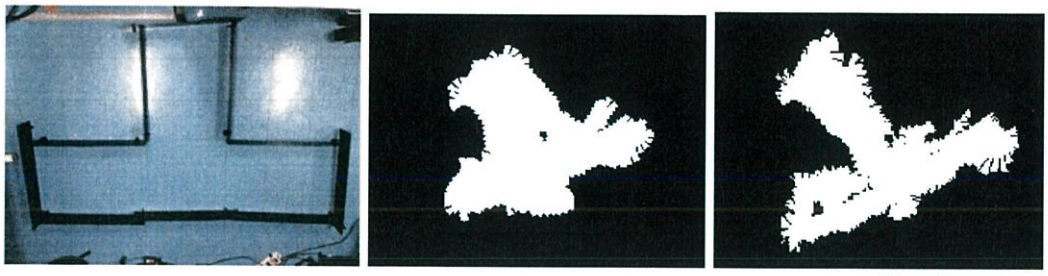
รูปที่ 5.7 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 1 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้
(ค) รูปที่ผ่านการประมวลผล

<u>แบบที่ 2</u>

(ก) (ข) (ค)

รูปที่ 5.8 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 2 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้
(ค) รูปที่ผ่านการประมวลผล

แบบที่ 3



(ก) (ข) (ค)

รูปที่ 5.9 แสดงพื้นที่ที่ใช้ในการสำรวจแบบที่ 3 และผลการสำรวจ
(ก) รูปพื้นที่จริง
(ข) รูปพื้นที่ที่สำรวจได้
(ค) รูปที่ผ่านการประมวลผล

5.2 สรุปผลโครงงาน

การศึกษาโครงการ " หุ่นยนต์สำรวจ (Obstacle Detection for Robot's Application) " ได้ศึกษาและพัฒนาหุ่นยนต์เพื่อใช้ในการหลบหลีกสิ่งกีดขวาง สำรวจเส้นทางที่หุ่นยนต์สามารถเดินทางไปได้ และสร้างแผนที่ต้นแบบ ได้เลือกอุปกรณ์ในการสร้างหุ่นยนต์ ซึ่งประกอบด้วยตัวตรวจรู้หรือเซนเซอร์ในการวัดระยะแบบอัลตราโซนิคและแบบอินฟราเรด อุปกรณ์ในการวัดมุมแบบเข็มทิศ (Compass HMC5883L) อุปกรณ์วัดระยะทางการเคลื่อนที่แบบเอ็นโคดเดอร์ (Motor Encoder) ไมโครคอนโทรเลอร์ (Arduino Mega 2560) และส่งข้อมูลเส้นทางการเคลื่อนที่เพื่อสร้างแผนที่ใน Microsoft Visual Basic ผ่านทางบลูทูธ (HC-05) ซึ่งในการพิจารณาการทำงานของหุ่นยนต์จะพิจารณาอยู่ 2 ส่วน คือการเคลื่อนที่หลบสิ่งกีดขวางเพื่อสำรวจเส้นทางและการสร้างแผนที่ต้นแบบ โดยตัวแปรที่มีผลต่อการเคลื่อนที่ของหุ่นยนต์คือระยะทางที่ได้จากการวัดของเซนเซอร์แบบอัลตราโซนิคและแบบอินฟราเรด และมุมในการหมุนของรถ ส่วนตัวแปรที่มีผลต่อการสร้างแผนที่ต้นแบบคือระยะทางการเคลื่อนที่ของรถและมุมในการหมุนของรถ ซึ่งเซนเซอร์ทุกตัวจะต้องมีความถูกต้องและแม่นยำในการวัด เพื่อที่จะควบคุมการเคลื่อนที่ของหุ่นยนต์ได้อย่างถูกต้องและสามารถสร้างแผนที่ต้นแบบได้อย่างแม่นยำ

ทั้งนี้การสร้างแผนที่ต้นแบบยังสามารถทำให้ถูกต้องและแม่นยำยิ่งขึ้นด้วยเทคนิคทางสถิติผ่านการประมวลผลด้วยภาพ อย่างไรก็ตามการศึกษาและพัฒนาหุ่นยนต์สำรวจ (Obstacle Detection for Robot's Application) ในครั้งนี้ ยังมีความผิดพลาดอยู่ อันเนื่องมาจากความผิดพลาดและความไม่แน่นอนในการวัดของเซนเซอร์ รวมทั้งความเสถียรของฮาร์ดแวร์และอุปกรณ์ที่ใช้ในการทำหุ่นยนต์ เช่น มอเตอร์ ล้อ เกียร์ ฯลฯ ดังนั้นในการศึกษาและพัฒนาหุ่นยนต์ต่อไปในอนาคตจำเป็นที่จะต้องมีหุ่นยนต์ต้นแบบที่มีเสถียรภาพ และมีเซนเซอร์ที่มีความถูกต้องและแม่นยำสูง เพื่อให้ได้หุ่นยนต์ที่มีความสามารถในการหลบหลีกและสำรวจพื้นที่เป้าหมายที่มีประสิทธิภาพสูงสุด

บรรณานุกรม

[1] T. Ratanajaratrod, N. Promjaree, Nuttama Maksri. "Aquatic Robot". Instrumentation Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang. 2013

[2] The Budding Electronics Engineers' Knowledge Space [Internet]. Obstacle Avoidance Robotics Vehicle Using Ultrasonic Sensor For Obstacle Detection. Available from: https://www.elprocus.com/obstacle-avoidance-robotic-vehicle

[3] University of Colorado [Internet]. Correll Lab CU Computer Science. Obstacle Avoidance [update Sep. 9, 2011]. Available from: http://correll.cs.colorado.edu/?p=974

[4] Arduino [Internet]. Arduino Mega 2560. Available from: http://www.arduino.cc/en/Main/arduinoBoardMega2560#

[5] Arduino [Internet]. Tutorial. Available from: http://www.arduino.cc/en/Tutorial/HomePage

[6] Microsoft Developer Network [Internet]. Visual Basic. Available from: https://msdn.microsoft.com/en-us/library/2x7h1hfk.aspx

[7] Ekkachai Makarn. "เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino". 1st ed. Bangkok: ETT Co., Ltd.; 2009

[8] Buncha Paseelatesang. "พัฒนาแอปพลิเคชั่นด้วย Visual Basic 2010". Bangkok: SE-EDUCATION; 2011

ภาคผนวก ก
Source Code ที่ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์

Robot Controlling Program Source code
SOFTWARE: ARDUINO IDE

```
//distance
float pathdistanceL=0;
float pathdistanceR=0;
int encoderL=2;
int encoderR=3;
float checkpulseL;
float checkpulseR;
float PathDistance;

//Compass
#include <Wire.h>
#include <HMC5883L.h>
int t = 0;
float Angle;
float TempAngle;
float MoveAngle;
float heading;
float rawheading;
float headingDegrees;

HMC5883L compass;

//IR
int Sleft_Infrared_Pin = A0;
int left_Infrared_Pin = A1;
int front_Infrared_Pin = A2;
int right_Infrared_Pin = A3;
int Sright_Infrared_Pin = A4;
float front_Infrared_Value = 0.00;
float left_Infrared_Value = 0.00;
float right_Infrared_Value = 0.00;
float front_Infrared_Output = 0.00;
```

```
float left_Infrared_Output = 0.00;
float right_Infrared_Output = 0.00;
float front_Infrared_Output_V = 0.00;
float left_Infrared_Output_V = 0.00;
float right_Infrared_Output_V = 0.00;


//ultrasonic
int MotorL=28;
int MotorLB=26;
int MotorR=24;
int MotorRB=22;
int D_active = 20;
int D_accident = 10;
int A,B,C,D,S,E,F,i,val;

#define trigPinSL 13
#define echoPinSL 12
#define trigPinL 11
#define echoPinL 10
#define trigPin 9
#define echoPin 8
#define trigPinR 7
#define echoPinR 6
#define trigPinSR 5
#define echoPinSR 4
void setup() {
  Serial.begin (9600);

  //sensor setting
  pinMode(trigPinSL, OUTPUT);
  pinMode(echoPinSL, INPUT);
  pinMode(trigPinL, OUTPUT);
  pinMode(echoPinL, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trigPinR, OUTPUT);
  pinMode(echoPinR, INPUT);
```

```
  pinMode(trigPinSR, OUTPUT);
  pinMode(echoPinSR, INPUT);

  //motor setting
  pinMode(MotorL,OUTPUT);
  digitalWrite(MotorL,LOW);
  pinMode(MotorLB,OUTPUT);
  digitalWrite(MotorLB,LOW);
  pinMode(MotorR,OUTPUT);
  digitalWrite(MotorR,LOW);
  pinMode(MotorRB,OUTPUT);
  digitalWrite(MotorRB,LOW);

  //Compass
  Wire.begin();
 compass = HMC5883L(); //new instance of HMC5883L library
 setupHMC5883L(); //setup the HMC5883L

  //Distance
  pinMode(encoderL,INPUT);
  pinMode(encoderR,INPUT);
  }

void loop() {

// ********************** Sensing Part ******************** //
 if (PathDistance < 30)
  {
  //Sonar sensor side left
  float durationSL, distanceSL;
  digitalWrite(trigPinSL, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinSL, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinSL, LOW);
  durationSL = pulseIn(echoPinSL, HIGH);
  distanceSL = (durationSL/2) / 29.1;
```

```
  if (distanceSL >= 400 || distanceSL <= 2){
  //Serial.println("SensorSL = Out of range");
  //Serial.println(" ");
  }
 else {
  //Serial.print("SensorSL = ");
  //Serial.print(distanceSL);
  //Serial.println(" cm");
      }
//delay(30);

//Sonar sensor left
float durationL, distanceL;
digitalWrite(trigPinL, LOW);
delayMicroseconds(2);
digitalWrite(trigPinL, HIGH);
delayMicroseconds(10);
digitalWrite(trigPinL, LOW);
durationL = pulseIn(echoPinL, HIGH);
distanceL = (durationL/2) / 29.1;

  if (distanceL >= 400 || distanceL <= 2){
  //Serial.println("Sensor2 = Out of range");
  //Serial.println(" ");
  }
 else {
 //Serial.print("SensorL = ");
  //Serial.print(distanceL);
  //Serial.println(" cm");
      }
//delay(30);

//Sonar sensor front

float duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

    if (distance >= 400 || distance <= 2){
    //Serial.println("SensorF = Out of range");
  }
  else {
    //Serial.print("SensorF = ");
    //Serial.print(distance);
    //Serial.println(" cm");
  }
  //delay(30);


  //Sonar sensor right
  float durationR, distanceR;
  digitalWrite(trigPinR, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinR, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinR, LOW);
  durationR = pulseIn(echoPinR, HIGH);
  distanceR = (durationR/2) / 29.1;

   if (distanceR >= 400 || distanceR <= 2){
    //Serial.println("SensorR = Out of range.....");
  }
  else {
    //Serial.print("SensorR = ");
    //Serial.print(distanceR);
    //Serial.println(" cm");
  }
  //delay(30);

//Sonar sensor side right
  float durationSR, distanceSR;
```

```
  digitalWrite(trigPinSR, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPinSR, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPinSR, LOW);
  durationSR = pulseIn(echoPinSR, HIGH);
  distanceSR = (durationSR/2) / 29.1;

   if (distanceSR >= 400 || distanceSR <= 2){
    //Serial.println("SensorSR = Out of range");
  }
  else {
    //Serial.print("SensorSR = ");
    //Serial.print(distanceSR);
    //Serial.println(" cm");
  }
  //delay(30);


  //IR Sensor
  //IRF
   front_Infrared_Value = analogRead( front_Infrared_Pin);
   front_Infrared_Output_V = (5.00/1023.00)*( front_Infrared_Value);
   front_Infrared_Output                =                ((69.33)*(pow(2.71828,(-3.477*
front_Infrared_Output_V))))+((18.85)*(pow(2.71828,(-0.5725*
front_Infrared_Output_V))));
   //Serial.print("IRSensorF = ");
   //Serial.print(front_Infrared_Output);
   //Serial.println(" cm");
  //IRF

  //IRL
   left_Infrared_Value = analogRead( left_Infrared_Pin);
   left_Infrared_Output_V = (5.00/1023.00)*( left_Infrared_Value);
   left_Infrared_Output                =                ((69.33)*(pow(2.71828,(-3.477*
left_Infrared_Output_V))))+((18.85)*(pow(2.71828,(-0.5725* left_Infrared_Output_V))));
   //Serial.print("IRSensorL = ");
   //Serial.print(left_Infrared_Output);
```

```
  //Serial.println(" cm");
  //IRL

 //IRR
   right_Infrared_Value = analogRead(right_Infrared_Pin);
   right_Infrared_Output_V = (5.00/1023.00)*(right_Infrared_Value);
   right_Infrared_Output = ((69.33)*(pow(2.71828,(-
3.477*right_Infrared_Output_V))))+((18.85)*(pow(2.71828,(-
0.5725*right_Infrared_Output_V))));
  //Serial.print("IRSensorR = ");
  //Serial.print(right_Infrared_Output);
  //Serial.println(" cm");
 //IRR


//sensor data

//pathdistance
checkpulseL=pulseIn(encoderL,HIGH);
if(checkpulseL!=0)
{
  PulseL();
}
checkpulseR=pulseIn(encoderR,HIGH);
if(checkpulseR!=0)
{
  PulseR();
}
PathDistance = ((pathdistanceL + pathdistanceR)/2);
PathDistance = PathDistance * 300;
//Compass
  float heading = getHeading();
  //Serial.print("Angle = ");
  ///Serial.println(heading);
  //Serial.println("  ");
  //delay(100); //only here to slow down the serial print
```

```
// ********************** Communication Part ********************** //

  for (int x = 0;x < 50 ; x++)
  //{
  //do
  {
  while (Serial.available() == 0); // do nothing if nothing sent
  int val = Serial.read() - '0'; // deduct ascii value of '0' to find numeric value of sent
number

  if (val == 0)
  {
  Serial.println(front_Infrared_Output);
  }
  else if (val == 1)
  {
  Serial.println(left_Infrared_Output);
  }
  else if (val == 2)
  {
  Serial.println(right_Infrared_Output);
  }
  else if (val == 3)
  {
  Serial.println(distanceSL);
  }
  else if (val == 4)
  {
  Serial.println(distanceL);
  }
  else if (val == 5)
  {
  Serial.println(distance);
  }
  else if (val == 6)
  {
```

```
  Serial.println(distanceR);
  }
  else if (val == 7)
  {
  Serial.println(distanceSR);
  }
  else if (val == 8)
  {
  Serial.println(PathDistance);

  pathdistanceR = 0;
  pathdistanceL = 0;
  }
  else if (val == 9)
  {
  Serial.println(heading);
  x = 100;
  }
  }
  //while (val <= 8);

  Serial.flush(); // clear serial port
  //}

// ********************************************************** //


// ***************** Control Part ************************** //

//motor control

if
((distance<D_accident)||(distanceR<D_accident)||(distanceL<D_accident)||(left_Infrared_
Output<D_accident)||(right_Infrared_Output<D_accident)||(front_Infrared_Output<D_ac
cident))
    {
      MoveBackward();
      if (distanceSR<distanceSL)
```

```
      {
        TurnLeft60Degree ();
      }
      else
      {
      TurnRight60Degree ();
      }
    }
 else if ((distance<D_active)||(distanceR<D_active)||(distanceL<D_active)) //if path is
blocked
 {
   if ((distance<D_active))
    {A = 1;}
   else {A = 0;}
   if ((distanceR<D_active))
    {B = 1;}
   else {B = 0;}
   if ((distanceL<D_active))
    {C = 1;}
   else {C = 0;}
   //accident
   /*
   if
((distance<D_accident)||(distanceR<D_accident)||(distanceL<D_accident)||(left_Infrared_
Output<D_accident)||(right_Infrared_Output<D_accident)||(front_Infrared_Output<D_ac
cident))
    {
      MoveBackward();
      if (distanceSR<distanceSL)
      {
        TurnLeft90Degree ();
      }
      else
      {
      TurnRight90Degree ();
      }
    }
   */
```

```
S = (A)+(B*2)+(C*4);

if (S==1)
  {
    digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,LOW);
        digitalWrite(MotorR,LOW);
        digitalWrite(MotorRB,LOW);
        delay (20);
  TurnRight60Degree ();
  }
else if (S==2)
 {
   digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,LOW);
        digitalWrite(MotorR,LOW);
        digitalWrite(MotorRB,LOW);
        delay (20);
  TurnLeft30Degree () ;
  }
else if (S==3)
 {
   digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,LOW);
        digitalWrite(MotorR,LOW);
        digitalWrite(MotorRB,LOW);
        delay (20);
  TurnLeft60Degree () ;
  }
else if (S==4)
 {
   digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,LOW);
        digitalWrite(MotorR,LOW);
        digitalWrite(MotorRB,LOW);
        delay (20);
  TurnRight30Degree ();
```

```
      }
    else if (S==5)
    {
      digitalWrite(MotorL,LOW);
           digitalWrite(MotorLB,LOW);
           digitalWrite(MotorR,LOW);
           digitalWrite(MotorRB,LOW);
           delay (20);
     TurnRight60Degree ();
      }
    else if (S==6)
    {

     MoveForward ();
      }
    else if (S==7)
    {
      digitalWrite(MotorL,LOW);
           digitalWrite(MotorLB,LOW);
           digitalWrite(MotorR,LOW);
           digitalWrite(MotorRB,LOW);
           delay (20);
     TurnRight60Degree ();
      }

  }
else //if path is clear
  {
    MoveForward();
  }

}
else
{
 digitalWrite(MotorL,LOW);
 digitalWrite(MotorLB,LOW);
 digitalWrite(MotorR,LOW);
 digitalWrite(MotorRB,LOW);
```

```
}
}
/************************ Sub Function **********************/
//control
void MoveForward (void)
   {
     D=0;
     digitalWrite(MotorL,HIGH);
     digitalWrite(MotorLB,LOW);
     digitalWrite(MotorR,HIGH);
     digitalWrite(MotorRB,LOW);
   }
   void MoveBackward (void)
   {
     D=1;
     digitalWrite(MotorL,LOW);
     digitalWrite(MotorLB,HIGH);
     digitalWrite(MotorR,LOW);
     digitalWrite(MotorRB,HIGH);
     delay (500);

   }
   void TurnLeft30Degree (void)
   {
     D=2;
     if (heading  < 30)
     {
       Angle = heading + 360;
       MoveAngle = Angle - 30;
     }
     else
     {
       Angle = heading;
       MoveAngle = Angle - 30;
     }

       if (heading  < 30)
       {
```

```
    while(Angle > MoveAngle)
    {
      heading = getHeading();
      if ((heading < 30) &&(heading >0))
        {
          Angle = heading + 360;
        }
      else
        {
          Angle = heading;
        }

      digitalWrite(MotorL,LOW);
      digitalWrite(MotorLB,HIGH);
      digitalWrite(MotorR,HIGH);
      digitalWrite(MotorRB,LOW);

    }
  }
  else
  {
    Angle = heading;
    while(Angle > MoveAngle)
    {
      heading = getHeading();
      Angle = heading;

      digitalWrite(MotorL,LOW);
      digitalWrite(MotorLB,HIGH);
      digitalWrite(MotorR,HIGH);
      digitalWrite(MotorRB,LOW);

    }
  }
}
```

```
void TurnLeft60Degree (void)
{
 D=2;
 if (heading  < 60)
 {
  Angle = heading + 360;
  MoveAngle = Angle - 60;
 }
 else
 {
  Angle = heading;
  MoveAngle = Angle - 60;
 }

 if (heading  < 60)
  {

    while(Angle > MoveAngle)
    {
     heading = getHeading();
     if ((heading < 60) &&(heading >0))
       {
         Angle = heading + 360;
       }
     else
       {
         Angle = heading;
       }

     digitalWrite(MotorL,LOW);
     digitalWrite(MotorLB,HIGH);
     digitalWrite(MotorR,HIGH);
     digitalWrite(MotorRB,LOW);

    }
 }
 else
 {
```

```
    Angle = heading;
    while(Angle > MoveAngle)
    {
      heading = getHeading();
      Angle = heading;

      digitalWrite(MotorL,LOW);
      digitalWrite(MotorLB,HIGH);
      digitalWrite(MotorR,HIGH);
      digitalWrite(MotorRB,LOW);

    }
  }
}

void TurnLeft90Degree (void)
  {
    D=2;
 if (heading  < 90)
 {
  Angle = heading + 360;
  MoveAngle = Angle - 90;
 }
 else
 {
  Angle = heading;
  MoveAngle = Angle - 90;
 }

 if (heading  < 90)
  {

    while(Angle > MoveAngle)
    {
      heading = getHeading();
      if ((heading < 90) &&(heading >0))
        {
          Angle = heading + 360;
```

```
        }
        else
        {
          Angle = heading;
        }

        digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,HIGH);
        digitalWrite(MotorR,HIGH);
        digitalWrite(MotorRB,LOW);
        }
    }
    else
    {
      Angle = heading;
      while(Angle > MoveAngle)
      {
        heading = getHeading();
        Angle = heading;

        digitalWrite(MotorL,LOW);
        digitalWrite(MotorLB,HIGH);
        digitalWrite(MotorR,HIGH);
        digitalWrite(MotorRB,LOW);

      }
    }
  }

//TurnLeft Algorithm
  void TurnRight90Degree (void)
  {
    D=2;
  if (heading  > 270)
  {
    Angle = heading - 360;
    MoveAngle = Angle + 90;
  }
```

```
else
{
  Angle = heading;
  MoveAngle = Angle + 90;
}

if (heading  > 270)
  {

    while(Angle < MoveAngle)
    {
      heading = getHeading();
      if ((heading > 270) &&(heading < 360))
        {
          Angle = heading - 360;
        }
      else
        {
          Angle = heading;
        }

        digitalWrite(MotorL,HIGH);
        digitalWrite(MotorLB,LOW);
        digitalWrite(MotorR,LOW);
        digitalWrite(MotorRB,HIGH);

    }
  }
  else
  {
    Angle = heading;
    while(Angle < MoveAngle)
    {
      heading = getHeading();
      Angle = heading;

      digitalWrite(MotorL,HIGH);
      digitalWrite(MotorLB,LOW);
```

```
      digitalWrite(MotorR,LOW);
      digitalWrite(MotorRB,HIGH);

    }
  }
}


void TurnRight30Degree (void)
{
 D=2;
if (heading  > 330)
{
  Angle = heading - 360;
  MoveAngle = Angle + 30;
}
else
{
  Angle = heading;
  MoveAngle = Angle + 30;
}

if (heading  > 330)
 {

   while(Angle < MoveAngle)
   {
     heading = getHeading();
     if ((heading > 330) &&(heading < 360))
       {
         Angle = heading - 360;
       }
     else
       {
         Angle = heading;
       }

     digitalWrite(MotorL,HIGH);
```

```
      digitalWrite(MotorLB,LOW);
      digitalWrite(MotorR,LOW);
      digitalWrite(MotorRB,HIGH);

    }
  }
  else
  {
    Angle = heading;
    while(Angle < MoveAngle)
    {
      heading = getHeading();
      Angle = heading;

      digitalWrite(MotorL,HIGH);
      digitalWrite(MotorLB,LOW);
      digitalWrite(MotorR,LOW);
      digitalWrite(MotorRB,HIGH);

    }
  }
}
void TurnRight60Degree (void)
{
  D=2;
  if (heading  > 300)
  {
    Angle = heading - 360;
    MoveAngle = Angle + 60;
  }
  else
  {
    Angle = heading;
    MoveAngle = Angle + 60;
  }
  if (heading  > 300)
  {
    while(Angle < MoveAngle)
```

```
        {
          heading = getHeading();
          if ((heading > 300) &&(heading < 360))
           {
             Angle = heading - 360;
           }
          else
           {
             Angle = heading;
           }

          digitalWrite(MotorL,HIGH);
          digitalWrite(MotorLB,LOW);
          digitalWrite(MotorR,LOW);
          digitalWrite(MotorRB,HIGH);

        }
      }
      else
      {
        Angle = heading;
        while(Angle < MoveAngle)
        {
          heading = getHeading();
          Angle = heading;

          digitalWrite(MotorL,HIGH);
          digitalWrite(MotorLB,LOW);
          digitalWrite(MotorR,LOW);
          digitalWrite(MotorRB,HIGH);

        }
      }
    }

//pathdistance
  void PulseL()
  {
```

```
  if(D==0)
  {
  pathdistanceL = pathdistanceL+0.0143;
 // Serial.print("pathdistanceL=");
  //Serial.println(pathdistanceL);
  }
  else if(D==1)
  {
  pathdistanceL = pathdistanceL-0.0143;
 // Serial.print("pathdistanceL=");
  //Serial.println(pathdistanceL);
  }
  else if(D==2)
  {
  pathdistanceL=0;
  //Serial.print("pathdistanceL=");
  //Serial.println(pathdistanceL);
  }
 }
void PulseR()
 {
  if(D==0)
  {
  pathdistanceR = pathdistanceR+0.0143;
  //Serial.print("pathdistanceR=");
  //Serial.println(pathdistanceR);
  }
  else if(D==1)
  {
  pathdistanceR = pathdistanceR-0.0143;
  //Serial.print("pathdistanceR=");
  //Serial.println(pathdistanceR);
  }
  else if(D==2)
  {
  pathdistanceR=0;
  //Serial.print("pathdistanceR=");
  //Serial.println(pathdistanceR);
```

```
    }
  }

//Compass
    void setupHMC5883L(){
 //Setup the HMC5883L, and check for errors
 int error;
 error = compass.SetScale(1.3); //Set the scale of the compass.
 if(error != 0) Serial.println(compass.GetErrorText(error)); //check if there is an error, and print if so
 error = compass.SetMeasurementMode(Measurement_Continuous); // Set the measurement mode to Continuous
 if(error != 0) Serial.println(compass.GetErrorText(error)); //check if there is an error, and print if so
}
float getHeading(){
 //Get the reading from the HMC5883L and calculate the heading
 MagnetometerScaled scaled = compass.ReadScaledAxis(); //scaled values from compass.
 float rawheading = atan2(scaled.YAxis, scaled.XAxis);

 // Correct for when signs are reversed.
 if(rawheading < 0) rawheading += 2*PI;
 if(rawheading > 2*PI) rawheading -= 2*PI;
 headingDegrees = rawheading * RAD_TO_DEG;
   if (headingDegrees >= 0 && headingDegrees < 35)
 {
   headingDegrees = map(headingDegrees,0,35,0,30);
 }
 else if (headingDegrees >= 35 && headingDegrees < 73)
 {
   headingDegrees =  map(headingDegrees,35,73,31,60);
 }
 else if (headingDegrees >= 73 && headingDegrees < 117)
 {
   headingDegrees =  map(headingDegrees,73,117,61,90);
 }
 else if (headingDegrees >= 117 && headingDegrees < 156)
```

```
{
  headingDegrees =  map(headingDegrees,117,156,91,120);
}
  else if (headingDegrees >= 156 && headingDegrees < 188)
{
  headingDegrees =  map(headingDegrees,156,188,121,150);
}
   else if (headingDegrees >= 188 && headingDegrees < 217)
{
  headingDegrees =  map(headingDegrees,188,217,151,180);
}
  else if (headingDegrees >= 217 && headingDegrees < 241)
{
  headingDegrees =  map(headingDegrees,217,241,181,210);
}
  else if (headingDegrees >= 241 && headingDegrees < 264)
{
  headingDegrees =  map(headingDegrees,241,264,211,240);
}
  else if (headingDegrees >= 264 && headingDegrees < 284)
{
  headingDegrees =  map(headingDegrees,264,284,241,270);
}
  else if (headingDegrees >= 284 && headingDegrees < 306)
{
  headingDegrees =  map(headingDegrees,284,306,271,300);
}
  else if (headingDegrees >= 306 && headingDegrees < 331)
{
  headingDegrees =  map(headingDegrees,306,331,301,330);
}
  else if (headingDegrees >= 331 && headingDegrees < 360)
{
  headingDegrees =  map(headingDegrees,331,360,331,360);
}
heading = headingDegrees;
return heading; //radians to degrees
}
```

ภาคผนวก ข
Source Code ที่ใช้ในการ Mapping

Map Exploration and Map Creation Program Source code in Visual Basic

SOFTWARE: VISUAL BASIC

```
Imports System.Math
Imports System.IO
Imports System.IO.Ports
Imports System.Threading

Public Class Form1

    ' Serial Setting
    Shared _continue As Boolean
    Shared _serialPort As SerialPort
    Dim value As String
    Dim output As String
    Dim i As Integer = 0
    Dim j As Integer = 0
    Dim k As Integer = 0
    Dim a As Integer = 0

    Dim x As Integer
    Dim Array(10000000) As String

    ' Center of picturebox.
    Dim StartPointX As Integer = 250
    Dim StartPointY As Integer = 200
    Dim CenterX As Integer = 250
    Dim CenterY As Integer = 200

    ' Sensor data.
    Dim Data As String
    Dim IR1 As Integer
    Dim IR2 As Integer
    Dim IR3 As Integer
    Dim Sonar1 As Integer
    Dim Sonar2 As Integer
    Dim Sonar3 As Integer
    Dim Sonar4 As Integer
    Dim Sensor2 As Integer
    Dim Sensor3 As Integer
    Dim Sensor4 As Integer
    Dim Sonar5 As Integer
    Dim Distance As Integer
    ' Angular data.
    Dim Compass1 As Integer

    Dim DIR1 As Double
    Dim DIR2 As Double
    Dim DIR3 As Double
    Dim DSonar1 As Double
```

```
    Dim DSonar2 As Double
    Dim DSonar3 As Double
    Dim DSonar4 As Double
    Dim DSensor2 As Double
    Dim DSensor3 As Double
    Dim DSensor4 As Double
    Dim DSonar5 As Double
    Dim DDistance As Double
    ' Angular data.
    Dim DCompass1 As Double
    Dim Compass2 As Integer = 0
    Dim MoveAngle As Integer

    ' Create start and sweep angles.
    Dim StartAngle1 As Integer = 170
    Dim StartAngle2 As Integer = 235
    Dim StartAngle3 As Integer = 260
    Dim StartAngle4 As Integer = 285
    Dim StartAngle5 As Integer = 350
    Dim SweepAngle As Integer = 20

    Dim bmp As New Drawing.Bitmap(500, 400)

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Communication Setting
        SerialPort1.Close()
        SerialPort1.PortName = "COM3"
        SerialPort1.BaudRate = 9600
        SerialPort1.DataBits = 8
        SerialPort1.Parity = IO.Ports.Parity.None
        SerialPort1.StopBits = IO.Ports.StopBits.One
        SerialPort1.Handshake = IO.Ports.Handshake.None
        SerialPort1.Encoding = System.Text.Encoding.Default 'very important!

        'Dim bmp As New Bitmap(PictureBox1.Width, PictureBox1.Height)
        'PictureBox1.Image = bmp
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        'Timer Setting
        Timer1.Interval = 50
        Timer1.Start()
        'Timer2.Interval = 20000
        'Timer2.Start()
        ' Center of picturebox.
        StartPointX = 250
        StartPointY = 200
        CenterX = 250
        CenterY = 200
        Distance = 0
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        SerialPort1.Open()
        a = 1
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        SerialPort1.Close()
        Timer1.Stop()
```

```
        a = 0
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        If k < 50 Then
            k += 1


            Dim i As Integer = 0
            Dim b As Integer = 0
            For i = 0 To 9
                If a = 1 Then
                    SerialPort1.Write(b)
                    Data = SerialPort1.ReadLine()
                    RawData.Text = Data
                    If b = 0 Then
                        Array(x) = Data
                        Infrared1.Text = Array(x)
                        b += 1
                    ElseIf b = 1 Then
                        Array(x) = Data
                        Infrared2.Text = Array(x)
                        b += 1
                    ElseIf b = 2 Then
                        Array(x) = Data
                        Infrared3.Text = Array(x)
                        b += 1
                    ElseIf b = 3 Then
                        Array(x) = Data
                        Ultrasonic1.Text = Array(x)
                        b += 1
                    ElseIf b = 4 Then
                        Array(x) = Data
                        Ultrasonic2.Text = Array(x)
                        b += 1
                    ElseIf b = 5 Then
                        Array(x) = Data
                        Ultrasonic3.Text = Array(x)
                        b += 1
                    ElseIf b = 6 Then
                        Array(x) = Data
                        Ultrasonic4.Text = Array(x)
                        b += 1
                    ElseIf b = 7 Then
                        Array(x) = Data
                        Ultrasonic5.Text = Array(x)
                        b += 1
                    ElseIf b = 8 Then
                        Array(x) = Data
                        TextBox6.Text = Array(x)
                        b += 1
                    ElseIf b = 9 Then
                        Array(x) = Data
                        TextBox7.Text = Array(x)
                        b = 0
                    End If
                    x += 1
                End If

            Next
```

```
Dim p As New System.Drawing.Pen(Color.Red, 2)
'Dim g As System.Drawing.Graphics

'Dim bmp As New Bitmap(PictureBox1.Width, PictureBox1.Height)
'bmp = New Bitmap(PictureBox1.Width, PictureBox1.Height)
PictureBox1.Image = bmp
Using g As Graphics = Graphics.FromImage(PictureBox1.Image)
    PictureBox1.Image = bmp

    'Get Data
    'IR1 = Infrared1.Text
    'IR2 = Infrared2.Text
    'IR3 = Infrared3.Text
    'Sonar1 = Ultrasonic1.Text
    'Sonar2 = Ultrasonic2.Text
    'Sonar3 = Ultrasonic3.Text
    'Sonar4 = Ultrasonic4.Text
    'Sonar5 = Ultrasonic5.Text
    'Distance = TextBox6.Text
    'Compass1 = TextBox7.Text

    'Get Data
    DIR1 = CDbl(Infrared1.Text)
    DIR2 = CDbl(Infrared2.Text)
    DIR3 = CDbl(Infrared3.Text)
    DSonar1 = CDbl(Ultrasonic1.Text)
    DSonar2 = CDbl(Ultrasonic2.Text)
    DSonar3 = CDbl(Ultrasonic3.Text)
    DSonar4 = CDbl(Ultrasonic4.Text)
    DSonar5 = CDbl(Ultrasonic5.Text)
    DDistance = CDbl(TextBox6.Text)
    DCompass1 = CDbl(TextBox7.Text)
    MoveAngle = Compass1 - Compass2

    'Get Data
    IR1 = CInt(DIR1)
    IR2 = CInt(DIR2)
    IR3 = CInt(DIR3)
    Sonar1 = CInt(DSonar1)
    Sonar2 = CInt(DSonar2)
    Sonar3 = CInt(DSonar3)
    Sonar4 = CInt(DSonar4)
    Sonar5 = CInt(DSonar5)
    Distance = CInt(DDistance)
    Compass1 = CInt(DCompass1)

    If Sonar2 > 30 Then
        Sonar2 = 30
    End If
    If Sonar3 > 30 Then
        Sonar3 = 30
    End If
    If Sonar4 > 30 Then
        Sonar4 = 30
    End If

    'Sonar or IR selecction
    'If Sonar2 >= IR1 Then
    'Sensor2 = Sonar2
    'Else : Sensor2 = IR1
    'End If
    'If Sonar3 >= IR2 Then
```

```
                'Sensor3 = Sonar3
                'Else : Sensor3 = IR2
                'End If
                'If Sonar4 >= IR3 Then
                'Sensor4 = Sonar4
                'Else : Sensor4 = IR3
                'End If

                'Compass2 = Compass1


                ' Create robot position.
                'PictureBox1.Refresh()

                If j > 0 Then
                    'g = PictureBox1.CreateGraphics
                    Dim width As Integer = 10
                    Dim height As Integer = 10
                    Dim OriginCenterX, OriginCenterY As Integer

                    'If MoveAngle <> 0 Then
                    'Distance = Distance * (-1)
                    'End If


                    If (((Compass1 >= 0) And (Compass1 < 90)) Or ((Compass1 <=
-300) And (Compass1 > -360))) Then
                        CenterX = CenterX + ((Math.Sin(MoveAngle * (PI / 180)))
* Distance)
                        CenterY = CenterY - ((Math.Cos(MoveAngle * (PI / 180)))
* Distance)
                    ElseIf (((Compass1 >= 90) And (Compass1 < 180)) Or
((Compass1 <= -180) And (Compass1 > -270))) Then
                        CenterX = CenterX + ((Math.Sin(MoveAngle * (PI / 180)))
* Distance)
                        CenterY = CenterY + ((Math.Cos(MoveAngle * (PI / 180)))
* Distance * (-1))
                    ElseIf (((Compass1 >= 180) And (Compass1 < 270)) Or
((Compass1 <= -90) And (Compass1 > -180))) Then
                        CenterX = CenterX - ((Math.Sin(MoveAngle * (PI / 180)))
* Distance * (-1))
                        CenterY = CenterY + ((Math.Cos(MoveAngle * (PI / 180)))
* Distance * (-1))
                    ElseIf (((Compass1 >= 270) And (Compass1 <= 360)) Or
((Compass1 <= 0) And (Compass1 > -90))) Then
                        CenterX = CenterX - ((Math.Sin(MoveAngle * (PI / 180)))
* Distance * (-1))
                        CenterY = CenterY - ((Math.Cos(MoveAngle * (PI / 180)))
* Distance)
                    End If

                    'CenterX = CenterX - ((Math.Sin(MoveAngle * (PI / 180))) *
Distance)
                    'CenterY = CenterY - ((Math.Cos(MoveAngle * (PI / 180))) *
Distance)

                    OriginCenterX = CenterX - (width / 2)
                    OriginCenterY = CenterY - (height / 2)
                    g.FillRectangle(Brushes.White, OriginCenterX,
OriginCenterY, width, height)
                    Dim blackPen As New Pen(Color.Black, 3)
```

```
                    ' Create solid brush.
                    Dim WriteBrush As New SolidBrush(Color.White)

                    ' Create location and size of ellipse.
                    Dim x1, x2, x3, x4, x5 As Integer
                    Dim y1, y2, y3, y4, y5 As Integer
                    Dim NewCenterXY1, NewCenterXY2, NewCenterXY3, NewCenterXY4,
NewCenterXY5 As Integer
                    NewCenterXY1 = Sonar1 / 2
                    NewCenterXY2 = Sonar2 / 2
                    NewCenterXY3 = Sonar3 / 2
                    NewCenterXY4 = Sonar4 / 2
                    NewCenterXY5 = Sonar5 / 2
                    x1 = CenterX - NewCenterXY1
                    x2 = CenterX - NewCenterXY2
                    x3 = CenterX - NewCenterXY3
                    x4 = CenterX - NewCenterXY4
                    x5 = CenterX - NewCenterXY5
                    y1 = CenterY - NewCenterXY1
                    y2 = CenterY - NewCenterXY2
                    y3 = CenterY - NewCenterXY3
                    y4 = CenterY - NewCenterXY4
                    y5 = CenterY - NewCenterXY5

                    'Area Graphics Plot

                    'g.FillPie(WriteBrush, x1, y1, Sonar1, Sonar1, StartAngle1
+ MoveAngle, SweepAngle)
                    g.FillPie(WriteBrush, x2, y2, Sonar2, Sonar2, StartAngle2 +
MoveAngle, SweepAngle)
                    g.FillPie(WriteBrush, x3, y3, Sonar3, Sonar3, StartAngle3 +
MoveAngle, SweepAngle)
                    g.FillPie(WriteBrush, x4, y4, Sonar4, Sonar4, StartAngle4 +
MoveAngle, SweepAngle)
                    'g.FillPie(WriteBrush, x5, y5, Sonar5, Sonar5, StartAngle5
+ MoveAngle, SweepAngle)
                End If
                j += 1
            End Using
        Else
            Timer1.Stop()
            MessageBox.Show("Time Out")
            k = 0
        End If
    End Sub

    Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles
PictureBox1.Click

    End Sub

    'Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles
Timer2.Tick
    'Dim c As Integer = 60
    'Timer1.Stop()
    'Timer2.Stop()
    'SerialPort1.Write(c)
    'MessageBox.Show("Time Out")
    'End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
```

```
        'Dim bmp As New Bitmap(PictureBox1.Width, PictureBox1.Height)
        PictureBox1.DrawToBitmap(bmp, New Rectangle(0, 0, PictureBox1.Width,
PictureBox1.Height))
        bmp.Save("C:\Test12.png", Imaging.ImageFormat.Png)
        'bmp = New Bitmap(PictureBox1.Width, PictureBox1.Height)
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
        Dim c As Integer = 10
        SerialPort1.Write(c)
    End Sub
End Class
```

ภาคผนวก ค

Source Code Image Processing ใน MATLAB

Image Processing

SOFTWARE: MATLAB

```
img1 = imread('C:\Test_3\Test10.png');
img2 = imread('C:\Test_3\Test11.png');
img3 = imread('C:\Test_3\Test12.png');
img5 = imcomplement(img1);
img6 = imcomplement(img2);
img7 = imcomplement(img3);
img8 = imadd(img6,img5);
img9 = imadd(img7,img8);
img4 = imcomplement(img9);
subplot(2,2,1); imshow(img1);
subplot(2,2,2); imshow(img2);
subplot(2,2,3); imshow(img3);
subplot(2,2,4); imshow(img4);
imwrite(img4,'C:\Test_1\CompleteMap1.png');
```

ภาคผนวก ง

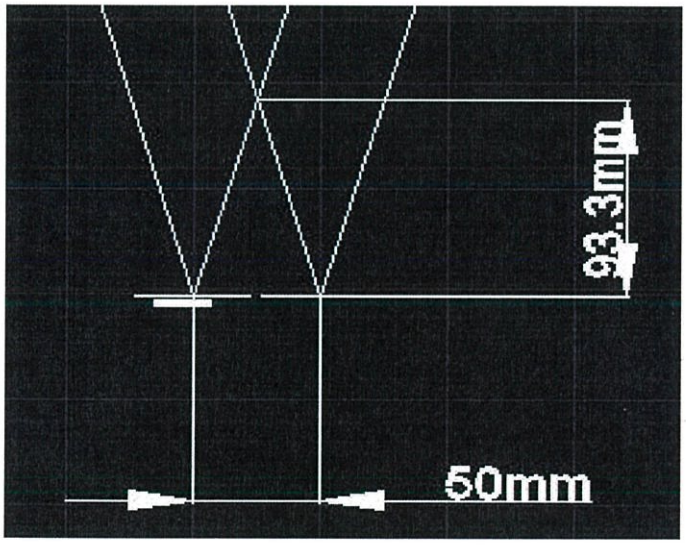
Sensing Profile

การทดลองหาผลกระทบจากการรบกวนของUltrasonic Sensor (เซนเซอร์ขนานกัน)

สมมติฐาน: ถ้าเซนเซอร์อัลตราโซนิกทั้งสองเริ่มมีย่านการวัดที่ทับซ้อนกัน ทำจะส่งผลให้ค่าที่ได้จากการวัดมีความผิดเพี้ยนไปจากเดิม

การทดลอง

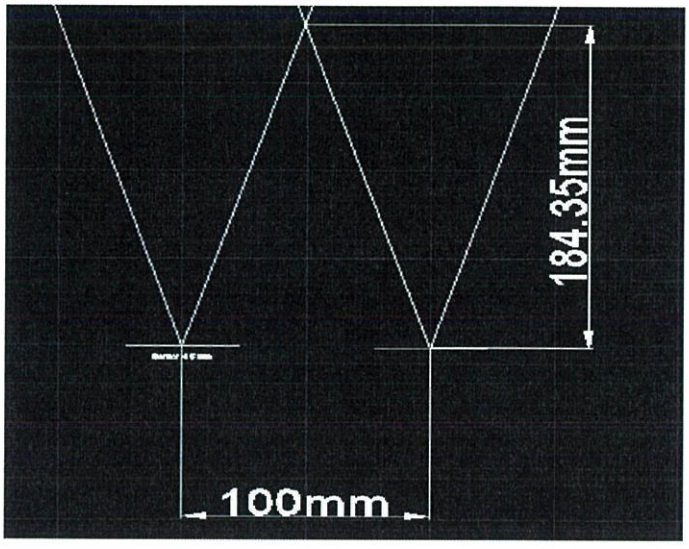
1. กำหนดให้เซนเซอร์วางห่างกัน 5 cm ทิศทางของการ Sensing เป็นแบบขนานกันไป ดังภาพ โดยเซนเซอร์ที่ใช้คือ HC-SR04 ซึ่งมีมุม operate อยู่ที่ประมาณ 30 องศา อ้างอิงจาก Datasheet จากการคำนวณ: จะเริ่มเกิดความผิดเพี้ยนจากการตรวจวัด ที่ระยะตั้งแต่ 9.33 เซนติเมตรเป็นต้นไป



ผลจากการวัด

	ค่าที่ได้		หลังการแก้ไขด้วยโปรแกรม ค่าจากการแก้ไขด้วยโปรแกรม	
ระยะอ้างอิง	Sensor1	Sensor2	Sensor1	Sensor2
3	3.21	3	3.16	3.44
4	4.3	4	4.3	4.4
6	6.07	4.52	6.53	6.89
7	7.46	4.05	7.51	7.3
8	8.69	3.28	8.54	7.87
10	10.6	2.82	10.27	10.48
12	12.51	2.04	12.39	12.29

2. กำหนดให้เซนเซอร์วางห่างกัน 10 cm ทิศทางของการ Sensing เป็นแบบขนานกันไป ดังภาพ โดย เซนเซอร์ที่ใช้คือ HC-SR04 ซึ่งมีมุม operate อยู่ที่ประมาณ 30 องศา อ้างอิงจาก Datasheet จากการคำนวณ: จะเริ่มเกิดความผิดเพี้ยนจากการตรวจวัด ที่ระยะตั้งแต่ 18.435 เซนติเมตรเป็นต้นไป



ผลจากการวัด			หลังการแก้ไขด้วยโปรแกรม		
ระยะอ้างอิง	Sensor1	Sensor2	ระยะอ้างอิง	Sensor1	Sensor2
27	26.89	27.03	27	26.24	26.25
37	36.63	36.12	37	36.48	36.81
42	42.04	41.31	42	41.98	41.55
44	43.44	43.21	44	43.44	44.12
45	44.88	30.24	45	45.02	45.36
46	45.48	33.18	46	45.92	45.88
48	46.77	32.8	48	48.32	47.89
53	52.51	36.77	53	52.16	52.11
63	63.06	47.8	63	63.14	62.22

ทดลองพบว่ามีค่าความผิดพลาดที่เกิดจากการวัดอย่างเห็นได้ชัดเจน ซึ่งเราสามารถสรุปได้ว่าเป็นผลมาจากการใช้เซนเซอร์ 2 ตัว ตรวจจับในบริเวณที่ใกล้เคียงกัน เพราะเมื่อเราทดลองตรวจจับด้วยเซนเซอร์ทีละตัว จะไม่พบค่าความผิดพลาดอย่างเช่นนี้เกิดขึ้นให้เห็น

ซึ่งค่าความผิดพลาดที่เกิดขึ้นนั้นมีค่าค่อนข้างมากและเห็นได้ชัดว่าเราไม่สามารถใช้ค่าที่ได้จากการวัดเช่นนี้มาใช้ในการตัดสินใจหรืออ้างอิงใดๆได้

แต่หลังจากการแก้ไขโปรแกรมแล้ว ทำให้ค่ากลับมาเชื่อถือได้

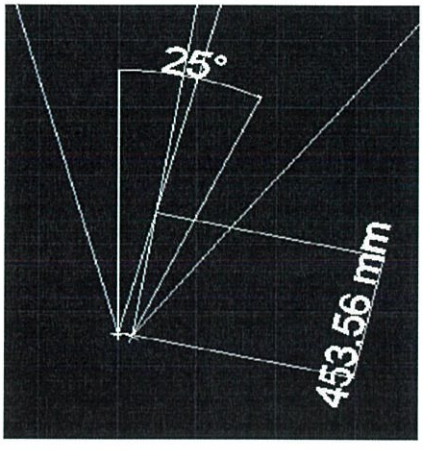
Ultrasonic interfering angle test

การทดลองหาผลกระทบจากการรบกวนของUltrasonic Sensor(เซนเซอร์ทำมุมขนาดต่างๆ)

สมมติฐาน : ถ้าเซนเซอร์อัลตราโซนิกทั้งสองเริ่มมีย่านการวัดที่ทับซ้อนกัน จะส่งผลให้ค่าที่ได้จากการวัดมีความผิดเพี้ยนไปจากเดิม

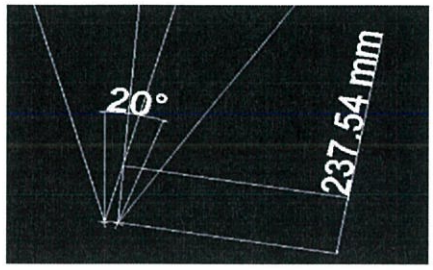
การทดลอง

1. กำหนดให้เซนเซอร์วางชิดกัน แต่ทำมุมต่อกัน 25 องศา ดังภาพ โดยเซนเซอร์ที่ใช้คือ HC-SR04 ซึ่งมีมุม operate อยู่ที่ประมาณ 30 องศาจากการคำนวณ จะเริ่มมีการรบกวนกันที่ระยะ 45.356 cm



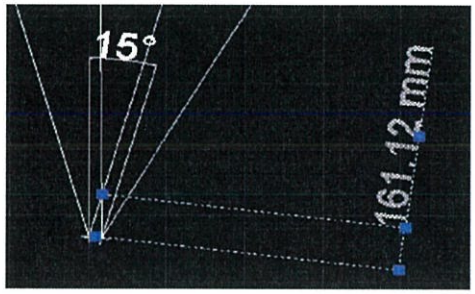
	เซนเซอร์ทำมุม 25องศา		ค่าเมื่อแก้ไขโปรแกรมแล้ว	
ระยะ	sensor1	sensor2	sensor1	sensor2
30	30.43	30.38	29.93	30.91
35	35.19	35.17		
40	40.29	40.91	39.83	40.74
45	45.25	45.24	0	
50	49.95	50.35	50.09	50.86
55	50.7	54.76		
60	51.53	60.86	59.52	60.14
65	51.05	64.3		
70	51.12	5.53/69.86	69.11	70.02
75	41.48/50.70	5.86/74.45		
80	49.05	4.73/78.49	79	79.67
85	84.83/60.34	84.66/5.09/15.81		
90	89.24	88.95	88.88	89.52
95	94.66	91.43		
100	99.04/84	99.11/91	98.32	99.16
105	104.71/	104.42/92.75		
110	109.14/56.33	109.46/89.28	108.38	109.21

2. กำหนดให้เซนเซอร์วางชิดกัน แต่ทำมุมต่อกัน 20 องศา ดังภาพ โดยเซนเซอร์ที่ใช้คือ HC-SR04 ซึ่งมีมุม operate อยู่ที่ประมาณ 30 องศาจากการคำนวณ จะเริ่มมีการรบกวนกันที่ระยะ 23.754 cm



	เซนเซอร์ทำมุม 20องศา		ค่าเมื่อแก้ไขโปรแกรมแล้ว	
ระยะ	sensor1	sensor2	sensor1	sensor2
30	30.03	30.69	30.27	31.01
35	34.97	35.07		
40	39.67	40.43	39.3	40.7
45	44.47	45.53		
50	49.73	49.66	49	50.69
55	54.97	55.02		
60	59.52	60.24	59.95	60.55
65	64.55	65.03		
70	69.24	69.78	69	71.01
75	74.05	74.24		
80	79.09	79.6	78.33	79.42
85	84.42	83.75		
90	89.17	88.42	88.47	89
95	94.11	92.21		
100	99.11	98.06	97.71	99.71
105	104.69	103.77/90.79		
110	109.21	89.71	108.08	109.3
115	112.7	111.77/94.46		
120	117.15	52.18/116.53	117.66	118.68
125	123.49	121.58/92.74		
130	127.25	126.89/91.84	128.11	129.61
135	132.63	131.12/121		
140	137.17	135.42/90.91	138.02	138.73
145	142.35/132.32	141.34/124.97		
150	147.20/132.15	147.12/6.67/93/122/23.42	148.42	148.53

3. กำหนดให้เซนเซอร์วางชิดกัน แต่ทำมุมต่อกัน 15 องศา ดังภาพ โดยเซนเซอร์ที่ใช้คือ HC-SR04 ซึ่งมีมุม operate อยู่ที่ประมาณ 30 องศาจากการคำนวณ จะเริ่มมีการรบกวนกันที่ระยะ 16.112 cm



	เซนเซอร์ทำมุม 15 องศา		ค่าเมื่อแก้ไขโปรแกรมแล้ว	
ระยะ	sensor1	sensor2	sensor1	sensor2
30	28.99	29.29	29.76	30.09
35	33.64	33.78		
40	38.32	39.5	39.67	40.07
45	43.28	43.78		
50	48.38	49.93	49.45	50.86
55	53.38	54.47		
60	58.28	59.04	59.9	60.43
65	63.13	63.06/48.59		
70	68.49	52.56/68.75	69.04	70.19
75	72.9	58.26/60.16/72.85		
80	78.82	77.56	78.32	79.11
85	82.16/61.54	82.54		
90	88.68	88.47	88.85	89.11
95	94.16	93.32		
100	98.68	98.43	98.63	99.66
105	104.21	103.79/95.61		
110	109.11	109.81/97.76	108.85	108.59
115	114.04	112.75/93.37		
120	118.95	114.43/95.19	118.69	119.67
125	123.83	122.49/92.67/88.34		
130	128.37	127.51/89.06	128.52	128.69
135	133.37	131.53/115.09/117.94		
140	138.21	137.75/122.22/95.18	139.64	138.75
145	144.55	142.20/122.78/92.24		
150	147.65	120.26/149	149.21	150.12

จากการทดลองทำให้เราทราบว่า มุมต่างๆที่เซนเซอร์กระทำต่อกันนั้น มีผลต่อค่าความผิดพลาดที่เกิดขึ้นอันเนื่องมาจากการรบกวนกันของเซนเซอร์อย่างเห็นได้ชัด แม้ส่วนใหญ่ระยะการวัดแรกที่เริ่มมีการผิดเพี้ยน จะไม่เป็นจริงตามการคำนวณ แต่มันก็ยังเกิดขึ้นที่ระยะทางหลังจากนั้น ซึ่งจะทำให้ผลของการวัดไม่สามารถนำไปใช้ในการอ้างอิง หรือตัดสินใจใดๆได้

แต่หลังจากได้ทำการแก้ตัวโปรแกรมแล้วก็พบว่า สามารถแก้ปัญหาการรบกวนที่เกิดขึ้นได้เป็นอย่างดี ทำให้ค่าที่ได้จากการวัดสามารถนำไปใช้อ้างอิงและตัดสินใจได้

Ultrasonic 2 objects interfering test

การทดลองหาผลกระทบจากการรบกวนระหว่าง Ultrasonic Sensor เมื่อมีวัตถุสองชิ้นในระยะการวัดที่แตกต่าง

ทดลองด้วยการกำหนดให้เซนเซอร์ทำมุมกัน 15 องศา จากนั้น ใช้วัตถุชิ้นแรกวางไว้ที่ระยะที่คงที่ค่าหนึ่งของเซนเซอร์ตัวที่ 1 โดยข้อกำหนดคือ วัตถุแต่ละชิ้นจะต้องถูกวางในแนวตั้งฉากกับแนวการวัดของเซนเซอร์แต่ละตัว หมายความว่า เซนเซอร์ตัวที่1 จะ Sensing วัตถุชิ้นที่1 ในแนวตั้งฉากกับวัตถุนั้น เช่นเดียวกันกับเซนเซอร์ตัวที่2และวัตถุชิ้นที่2

*** อีกเงื่อนไขที่สำคัญคือ ห้ามให้วัตถุชิ้นที่1 ถูกตรวจจับด้วยเซนเซอร์ตัวที่2 และในขณะเดียวกัน วัตถุชิ้นที่2 ห้ามถูกตรวจจับด้วยเซนเซอร์ตัวที่ 1 เช่นเดียวกันก็จะไม่มีความหมายใดๆ ไม่ว่าจะอยู่ในระยะใดก็ตาม มันจะไม่ถูกตรวจจับด้วยเซนเซอร์ตัวที่มันทำมุมตั้งฉากอยู่ เพราะเซนเซอร์ตัวนั้นไปตรวจจับวัตถุชิ้นแรกไปแล้ว นี่จึงเป็นเหตุผลที่ต้องตั้งข้อกำหนดเช่นนี้***

การทดลอง

1. กำหนดให้วัตถุชิ้นที่ 1 อยู่ที่ระยะ 30 cm

เซนเซอร์ทำมุมกัน 15 องศา			
ระยะวัตถุ1	sensor1	ระยะวัตถุ2	sensor2
30	30.6	30	31.01
30	30.6	40	40.8
30	30.6	50	50.7
30	30.6	60	60.6
30	30.6	70	69.6
30	30.6	80	79.8
30	30.6	90	89.4
30	30.6	120	119.3
30	30.6	150	148.8
30	30.6	180	179/150/114/123
30	30.6	210	211/5.96/---

กำหนดให้วัตถุชิ้นที่อยู่กับที่ อยู่ที่ระยะ 40 cm

เซนเซอร์ทำมุมกัน 15 องศา			
ระยะวัตถุ1	sensor1	ระยะวัตถุ2	sensor2
40	40.8	40	40.55
40	40.8	60	60.1
40	40.8	90	89.52
40	40.8	120	119.2
40	40.8	150	149.9
40	40.8	180	179.3
40	40.8	210	210/179/122/159
40	40.8	240	178/241/5.98/180

2. กำหนดให้วัตถุชิ้นที่อยู่กับที่ อยู่ที่ระยะ 50 cm

เซนเซอร์ทำมุมกัน 25 องศา			
ระยะวัตถุ1	sensor1	ระยะวัตถุ2	sensor2
50	50.8	50	50.3
50	50.8	60	60.1
50	50.8	90	90.3
50	50.8	120	119.1
50	50.8	150	149.5
50	50.8	180	180.2/256.3/5.98
50	50.8	210	209.3/257
50	50.8	240	240/5.98

หลังจากการทดลองทำให้เราทราบว่า ระยะของวัตถุ 2 ชนิดที่แตกต่างกันไม่ส่งผลกตารางบันทึกผลในบริเวณที่ค่ามีความผิดพลาดไปนั้นเนื่องมาจากความผิดพลาดของตัวเซนเซอร์เอง ซึ่งพวกเราได้ทำการทดลองแยกอีกครั้งเพื่อทำให้เกิดความมั่นใจแล้ว ได้ข้อสรุปว่าที่ระยะประมาณ 150 cm ขึ้นไป ตัวเซนเซอร์จะเริ่ม ระทบใดๆต่อการตรวจจับ ดังนั้นจึงไม่ได้มีการทดลองที่มุมอื่นๆอีก แต่ที่เราเห็นจาก มีค่าความผิดพลาดจ่ากการตรวจจับ คาดว่าน่าจะมาจากคุณภาพของเซนเซอร์ (แม้ว่า Datasheet จะระบุว่ามันสามารถตรวจจับได้มากที่สุดถึง 4 เมตรก็ตาม แต่จากการทดลองทำให้เราเห็นได้ว่า ระยะที่เชื่อมั่นได้อยู่ที่ประมานไม่เกิน 150cm เท่านั้น)

Offset Test

ULTRASONIC SENSOR NO.1

ULTRASONIC SENSOR NO.1

วัดที่ระยะ 10 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms

วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	9.73	0.27	2.7
ครั้งที่ 2	9.69	0.31	3.1
ครั้งที่ 3	9.64	0.36	3.6
ครั้งที่ 4	9.5	0.5	5
ครั้งที่ 5	9.62	0.38	3.8
ครั้งที่ 6	9.57	0.43	4.3
ครั้งที่ 7	9.45	0.55	5.5
ครั้งที่ 8	9.88	0.12	1.2
ครั้งที่ 9	9.73	0.27	2.7
ครั้งที่ 10	9.57	0.43	4.3
ค่าเฉลี่ย	9.638	0.362	3.62

วัดที่ระยะ 15cm วัตถุตั้งฉากกับ sensor , scan time = 500ms

วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	14.48	0.52	3.466667
ครั้งที่ 2	14.33	0.67	4.466667
ครั้งที่ 3	14.4	0.6	4
ครั้งที่ 4	14.45	0.55	3.666667
ครั้งที่ 5	14.48	0.52	3.466667
ครั้งที่ 6	14.48	0.52	3.466667
ครั้งที่ 7	14.28	0.72	4.8
ครั้งที่ 8	14.45	0.55	3.666667
ครั้งที่ 9	14.4	0.6	4
ครั้งที่ 10	14.38	0.62	4.133333
ค่าเฉลี่ย	14.413	0.587	3.913333

วัดที่ระยะ 30cm วัตถุตั้งฉากกับ sensor , scan time = 500ms			
วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	28.75	1.25	4.166667
ครั้งที่ 2	28.53	1.47	4.9
ครั้งที่ 3	28.33	1.67	5.566667
ครั้งที่ 4	28.9	1.1	3.666667
ครั้งที่ 5	28.38	1.62	5.4
ครั้งที่ 6	28.99	1.01	3.366667
ครั้งที่ 7	28.83	1.17	3.9
ครั้งที่ 8	28.38	1.62	5.4
ครั้งที่ 9	28.47	1.53	5.1
ครั้งที่ 10	28.75	1.25	4.166667
ค่าเฉลี่ย	28.631	1.369	4.563333

ULTRASONIC SENSOR NO.2

ULTRASONIC SENSOR NO.2			
วัดที่ระยะ 10 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms			
วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	10.14	0.14	1.4
ครั้งที่ 2	10.14	0.14	1.4
ครั้งที่ 3	10.34	0.34	3.4
ครั้งที่ 4	10.13	0.13	1.3
ครั้งที่ 5	10.38	0.38	3.8
ครั้งที่ 6	10.45	0.45	4.5
ครั้งที่ 7	10.31	0.31	3.1
ครั้งที่ 8	10.11	0.11	1.1
ครั้งที่ 9	10.34	0.34	3.4
ครั้งที่ 10	10.13	0.13	1.3
ค่าเฉลี่ย	10.247	0.247	2.47

วัดที่ระยะ 15 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms			
วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	14.71	0.29	1.933333
ครั้งที่ 2	14.7	0.3	2
ครั้งที่ 3	14.76	0.24	1.6
ครั้งที่ 4	14.76	0.24	1.6
ครั้งที่ 5	15.22	0.22	1.466667
ครั้งที่ 6	14.74	0.26	1.733333
ครั้งที่ 7	15.17	0.17	1.133333
ครั้งที่ 8	15.14	0.14	0.933333
ครั้งที่ 9	14.76	0.24	1.6
ครั้งที่ 10	15.17	0.17	1.133333
ค่าเฉลี่ย	14.913	0.227	1.513333

วัดที่ระยะ 30 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms			
วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	28.69	1.31	4.366667
ครั้งที่ 2	29.18	0.82	2.733333
ครั้งที่ 3	29.26	0.74	2.466667
ครั้งที่ 4	29.14	0.86	2.866667
ครั้งที่ 5	29.18	0.82	2.733333
ครั้งที่ 6	28.75	1.25	4.166667
ครั้งที่ 7	28.78	1.22	4.066667
ครั้งที่ 8	29.21	0.79	2.633333
ครั้งที่ 9	29.3	0.7	2.333333
ครั้งที่ 10	29.21	0.79	2.633333
ค่าเฉลี่ย	29.07	0.93	3.1

ULTRASONIC SENSOR NO.3

ULTRASONIC SENSOR NO.3

วัดที่ระยะ 10 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms

วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	10.26	0.26	2.6
ครั้งที่ 2	10.24	0.24	2.4
ครั้งที่ 3	10.36	0.36	3.6
ครั้งที่ 4	9.95	0.05	0.5
ครั้งที่ 5	10.24	0.24	2.4
ครั้งที่ 6	9.95	0.05	0.5
ครั้งที่ 7	9.95	0.05	0.5
ครั้งที่ 8	10.36	0.36	3.6
ครั้งที่ 9	10.24	0.24	2.4
ครั้งที่ 10	9.81	0.19	1.9
ค่าเฉลี่ย	10.136	0.204	2.04

วัดที่ระยะ 15 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms

วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	15.14	0.14	0.933333
ครั้งที่ 2	15.14	0.14	0.933333
ครั้งที่ 3	14.69	0.31	2.066667
ครั้งที่ 4	15.29	0.29	1.933333
ครั้งที่ 5	15.21	0.21	1.4
ครั้งที่ 6	14.68	0.32	2.133333
ครั้งที่ 7	15.24	0.24	1.6
ครั้งที่ 8	15.25	0.25	1.666667
ครั้งที่ 9	15.31	0.31	2.066667
ครั้งที่ 10	15.33	0.33	2.2
ค่าเฉลี่ย	15.128	0.254	1.693333

วัดที่ระยะ 30 cm วัตถุตั้งฉากกับ sensor , scan time = 500ms			
วัดครั้งที่	ค่าที่วัดได้	Error	%Error
ครั้งที่ 1	29.4	0.6	2
ครั้งที่ 2	29.28	0.72	2.4
ครั้งที่ 3	29.28	0.72	2.4
ครั้งที่ 4	29.26	0.74	2.466667
ครั้งที่ 5	29.26	0.74	2.466667
ครั้งที่ 6	29.66	0.34	1.133333
ครั้งที่ 7	29.24	0.76	2.533333
ครั้งที่ 8	29.28	0.72	2.4
ครั้งที่ 9	29.26	0.74	2.466667
ครั้งที่ 10	29.36	0.64	2.133333
ค่าเฉลี่ย	29.328	0.672	2.24

MATERIAL SHAPE TEST

ULTRASONIC SENSOR NO.3

วัดที่ระยะ 30 cm ด้วยวัสดุพลาสติกรูปทรงโค้งนูน

วัดครั้งที่	ค่าที่วัดได้		
1	30.38		
2	29.93		
3	30.05		
4	30.07		
5	29.93		
6	29.97		
7	29.93		
8	30.05		
9	29.93		
10	30.36	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	30.06	0.06	0.2

วัดที่ระยะ 15 cm ด้วยวัสดุพลาสติกรูปทรงโค้งนูน

วัดครั้งที่	ค่าที่วัดได้		
1	15.29		
2	15.41		
3	14.86		
4	15.29		
5	15.29		
6	14.86		
7	15.29		
8	15.27		
9	15.41		
10	15.22	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	15.219	0.219	1.46

วัดที่ระยะ 30 cm ด้วยวัสดุพลาสติกรูปทรงโค้งเว้า

วัดครั้งที่	ค่าที่วัดได้		
1	29.86		
2	29.9		
3	29.47		
4	29.47		
5	29.47		
6	29.91		
7	29.47		
8	29.86		
9	29.48		
10	29.93	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	29.682	0.318	1.06

วัดที่ระยะ 15 cm ด้วยวัสดุพลาสติกรูปทรงโค้งเว้า

วัดครั้งที่	ค่าที่วัดได้		
1	15.19		
2	15.21		
3	14.78		
4	14.81		
5	14.78		
6	14.81		
7	14.81		
8	15.19		
9	14.81		
10	15.19	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	14.958	0.042	0.28

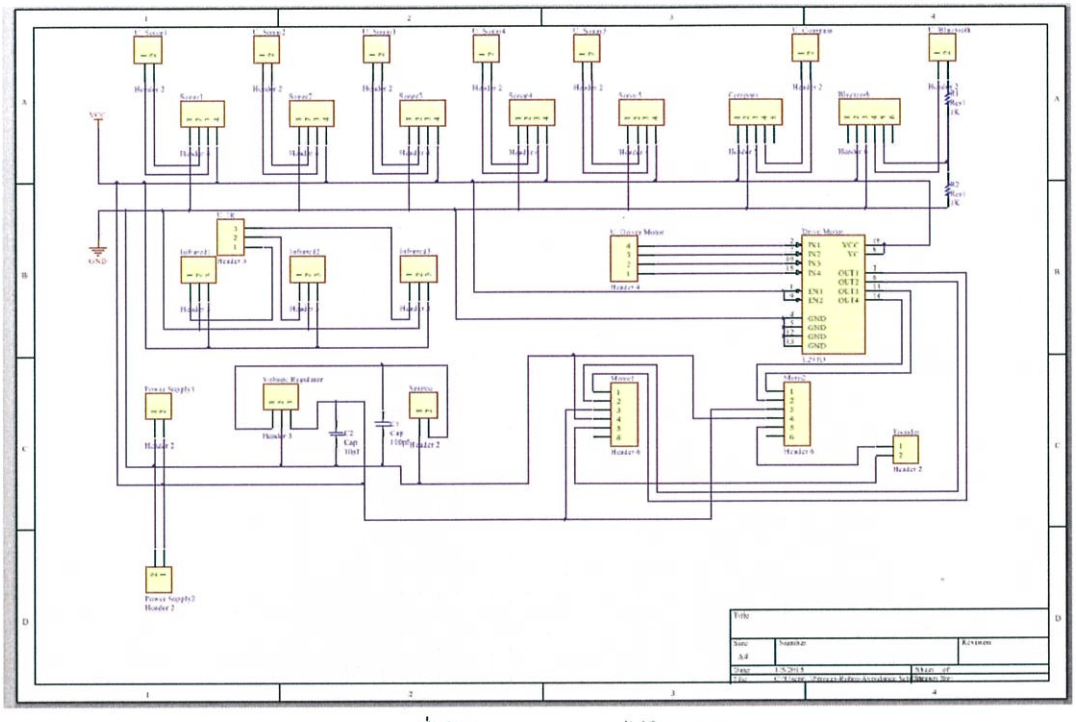
วัดที่ระยะ 30 cm ด้วยวัสดุพลาสติกทรงสามเหลี่ยมมุมฉาก (วัดด้านใน)

วัดครั้งที่	ค่าที่วัดได้		
1	30.86		
2	31.36		
3	30.86		
4	30.84		
5	31.27		
6	31.86		
7	31.72		
8	30.13		
9	30.86		
10	30.98	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	31.074	1.074	3.58

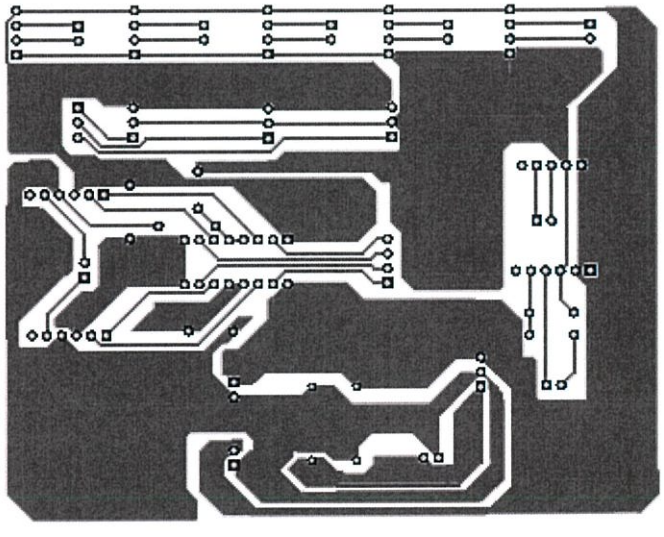
วัดที่ระยะ 15 cm ด้วยวัสดุพลาสติกทรงสามเหลี่ยมมุมฉาก (วัดด้านใน)

วัดครั้งที่	ค่าที่วัดได้		
1	15.14		
2	15.55		
3	15.65		
4	15.14		
5	15.22		
6	15.57		
7	15.55		
8	15.22		
9	15.57		
10	15.64	error	%error
ค่าเฉลี่ยทั้ง 10 ครั้ง >>>	15.425	0.425	2.833333

ภาคผนวก จ
วงจรลายปริ้น



ภาพวงจรที่ได้ทำการออกแบบไว้ใน Altium



ภาพวงจรลายปริ้นที่ได้ทำการเขียนจาก Altium