



โครงการพัฒนาระบบสร้างแผนที่โดยใช้อัลตราโซนิกเซ็นเซอร์

**DEVELOPMENT OF MAP BUILDING SYSTEM
USING ULTRASONIC SENSOR**

นายติณณ์ พรมโชติ รหัส 53363430

นายภาณุพงศ์ กิตติสุภลักษณ์ รหัส 53363775

นายภัทรยุทธ ปริมาณนท์ รหัส 53364017

ใบเสร็จรับเงิน
เลขที่ 28 พง 57
16551361
พ.ร.
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ๓๑๓๕๙

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ ระบบสร้างแผนที่โดยใช้อัลตราโซนิกเซ็นเซอร์

ผู้ดำเนินโครงการ นายติณณ์ พรมโชติ รหัส 53363430

นายภาณุพงศ์ กิตติสุภลักษณ์ รหัส 53363775

นายภัทรยุทธ ปรมานนท์ รหัส 53364017

ที่ปรึกษาโครงการ ดร. พันธ์ นัถฤทธิ์

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2556

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ

(ดร. พันธ์ นัถฤทธิ์)

.....กรรมการ

(อาจารย์รัฐภูมิ วรรณสาสน์)

.....กรรมการ

(อาจารย์เศรษฐา ตั้งคำวานิช)

ชื่อหัวข้อโครงการ

ผู้ดำเนินโครงการ นายดิณณ์ พรมโชติ รหัส 53363430
นายภาณุพงศ์ กิตติศุภลักษณ์ รหัส 53363775
นายภัทรยุทธ ปริมาณนท์ รหัส 53364017

ที่ปรึกษาโครงการ คร.พนัส นัถฤทธิ

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2556

บทคัดย่อ

โครงการนี้เป็นการพัฒนาระบบสร้างแผนที่ โดยอาศัยข้อมูลของระยะทางและทิศทางจากเซ็นเซอร์ที่ติดตั้งอยู่กับหุ่นยนต์เคลื่อนที่ขณะทำการสำรวจพื้นที่ โดยระบบนี้สามารถแสดงข้อมูลระยะทางจากตัวหุ่นยนต์เคลื่อนที่จนถึงพื้นผิวของวัตถุ โดยสามารถระบุระยะทางได้ตั้งแต่ 1 เซนติเมตร จนถึง 4 เมตร และแสดงค่าองศาของมุมเอียงได้ตั้งแต่ 0 องศา ถึง 359.9 องศา โดยอ้างอิงจากแกนแม่เหล็กโลก และข้อมูลที่ได้จากระบบนี้จะนำไปสร้างแผนที่และแสดงผลออกมาทางหน้าจอคอมพิวเตอร์เพื่อความสะดวกและปลอดภัยในการสำรวจพื้นที่ในบริเวณที่มีความเสี่ยงต่อมนุษย์ โดยหุ่นยนต์ที่ใช้ในการเคลื่อนที่ใช้ระบบคลื่นวิทยุในการควบคุมและมีการสื่อสารกับระบบเมื่อมีการเปลี่ยนแปลงตำแหน่งเพื่อให้เกิดความเสถียรภาพในการสร้างแผนที่มากยิ่งขึ้น

ผลลัพธ์ที่ได้จากโครงการนี้คือระบบที่สามารถสร้างแผนที่ในรูปแบบของข้อมูลระยะทางและทิศทางโดยแสดงผลเป็นภาพออกทางหน้าจอแสดงผลบนคอมพิวเตอร์ส่วนบุคคล

Project Title

Name Mr. Tin Promchot ID. 53363430
Mr. Panupong Kittisupaluk ID. 53363775
Mr. Bhattarayudh Poramatanonta ID. 53364017

Project advisor Dr. Panus Nattharith

Major Computer Engineering

Department Electrical and Computer Engineering

Academic 2013

Abstract

This project is to build a map using information systems by distance and direction from sensors mounted on mobile robot while exploring the area. The system can display the distance from the robot to the surface of the object. This can identify the distance from 1 cm to 4 meters, and shows the tilt angle from 0 degrees to 359.9 degrees, based on the magnetic core. The data obtained from this system will be used to build the map and display on developed Graphic User Interface (GUI) running on a personal computer (PC).

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นเรื่องเกี่ยวกับการออกแบบระบบการสร้างแผนที่ ซึ่งจะสำเร็จไม่ได้
เลยถ้าไม่ได้รับความช่วยเหลือจากบุคคลต่อไปนี้

ขอขอบพระคุณบิดา มารดา ผู้ที่เลี้ยงดูอบรมบ่มนิสัยของคณะผู้ดำเนินโครงการมาเป็นอย่างดี
ถ้าไม่มี 2 ท่านนี้แล้ว ทางคณะผู้ดำเนินโครงการคงไม่มีวันนี้ ซึ่งเป็นพระคุณที่หาใครเปรียบไม่ได้

ขอขอบพระคุณ ดร.พนัส นัถฤทธิ เป็นอย่างสูงที่กรุณาเป็นอาจารย์ที่ปรึกษาให้แก่โครงการนี้
อาจารย์ได้เสียสละเวลาในการให้คำปรึกษาพร้อมทั้งให้แนวคิดในการทำงาน รวมทั้งแนะนำและจัดท
างประมาณเพื่อจัดซื้ออุปกรณ์ทางฮาร์ดแวร์ต่างๆ ที่จำเป็นสำหรับการสร้างระบบขึ้นมา

ขอขอบพระคุณคณาจารย์ทุกท่าน ที่ได้มอบความรู้ต่างๆ ให้แก่คณะผู้ดำเนินโครงการ ซึ่งเป็น
ส่วนสำคัญช่วยให้ผู้ดำเนินโครงการอย่างมากในการทำโครงการนี้

ขอขอบคุณมหาวิทยาลัยนเรศวรที่ให้งบประมาณและสถานที่แก่คณะผู้ดำเนินโครงการในการ
ดำเนินโครงการในครั้งนี้

ขอขอบคุณทุกๆ ท่านที่ไม่ได้เอ่ยชื่อนามมา ณ ที่นี้ ที่มีส่วนช่วยเหลือในการดำเนินโครงการนี้ให้
สำเร็จลุล่วงไปด้วยดี

คณะผู้ดำเนินโครงการ

ดิฉัน พรมโชติ

ภาณุพงศ์ กิตติสุภลักษณ์

ภัทรยุทธ ปรมานนท์

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของ โครงการงาน.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบข่ายของ โครงการงาน.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ตารางแสดงกิจกรรมการดำเนินงาน.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	4
1.7 งบประมาณ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 สถาปัตยกรรมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ (Mobile Robot Architecture).....	5
2.2 ระบบ Localization.....	9
2.3 ระบบ MapBuilding.....	11
2.4 ระบบ SLAM (Simultaneous Localization And Map building).....	11
2.5 ศรี โคนมิติ.....	11
2.6 ความหมายและความสำคัญของ GUI.....	14
2.7 บอร์ด ไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2).....	15
2.8 โมดูลตรวจจับและวัดระยะทางด้วยอัลตราโซนิกSRF05.....	18
2.9 โมดูลเข็มทิศและวัดความเอียงแบบมีการชดเชยค่า CMPS10.....	25

สารบัญ (ต่อ)

เรื่อง	หน้า
2.10 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3.....	29
2.11 วงจรควบคุมมอเตอร์ Motor Drive Shield (L293D).....	31
2.12 วงจรรับคลื่นวิทยุบังคับ.....	33
2.13 ภาษา C#.....	34
2.14 AForge.NET.....	35
2.15 บทสรุป.....	36
บทที่ 3 วิธีการดำเนินการ.....	37
3.1 ภาพรวมของระบบ (System Overview).....	37
3.2 การเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมการเคลื่อนที่.....	38
3.3 การเขียนโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์เพื่อใช้ในการอ่านค่าจากเซ็นเซอร์ที่ใช้สำหรับวัดระยะทางและทิศทาง.....	44
3.4 การเขียนโปรแกรมสร้างหน้าจอ GUI สำหรับแสดงผล.....	49
3.5 บทสรุป.....	62
บทที่ 4 การทดลองและผลการทดลอง.....	63
4.1 ระบบที่พัฒนาขึ้น.....	63
4.2 ผลการทดลองการคำนวณหาระยะทางจากค่าที่ได้รับจากเซ็นเซอร์.....	65
4.3 ผลการทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตราโซนิกเซ็นเซอร์ SRF05.....	68
4.4 ผลการทดลองการวัดขนาดลำแสง (Beam width) ของอัลตราโซนิกเซ็นเซอร์ SRF05.....	71
4.5 ผลการทดลองการวัดมุมเอียงของเซ็นเซอร์ CMPS10.....	73
4.6 ผลการทดลองจากการส่งสัญญาณระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino กับ PIC.....	78
4.7 ผลการทดลองการทำงานร่วมกันของอัลตราโซนิกเซ็นเซอร์ SRF05 กับ โมดูลเข็มทิศดิจิทัล CMPS10.....	81
4.8 บทสรุป.....	89

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 สรุปผลการทดลอง.....	90
5.1 สรุปผลการทดลอง.....	90
5.2 วิเคราะห์ผลการทดลอง.....	91
5.3 การพัฒนาโครงการต่อในอนาคต.....	91
5.4 ข้อเสนอแนะ.....	92
เอกสารอ้างอิง.....	93
ภาคผนวก.....	100
ก. ขั้นตอนการติดตั้งโปรแกรม Microsoft VisualC# Studio 2010 Express.....	100
ข. ขั้นตอนการติดตั้งโปรแกรม MicroC PRO for PIC.....	104
ค. คู่มือการใช้งานโปรแกรมสำหรับวาดแผนที่.....	106
ประวัติคณะผู้ดำเนินโครงการ.....	111

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน.....	3
ตารางที่ 2.1 แสดงการคาดเดา present local และ future local.....	10
ตารางที่ 3.1 รายละเอียดของรีจิสเตอร์ TICON.....	45
ตารางที่ 4.1 แสดงผลการทดลองระยะทางจริงเปรียบเทียบกับระยะทางที่วัดได้จากอัลตราโซนิก เซ็นเซอร์ SRF05.....	66
ตารางที่ 4.2 แสดงผลการทดลองการคำนวณหาระยะทางจากอัลตราโซนิกเซ็นเซอร์ SRF05.....	68
ตารางที่ 4.3 แสดงผลการทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตราโซนิก เซ็นเซอร์RF05.....	69
ตารางที่ 4.4 ผลการทดลองการวัดขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05.....	71
ตารางที่ 4.5 แสดงผลการทดลอง ค่าเฉลี่ย และค่าความคลาดเคลื่อนของผลการทดลองของการวัดมุม เอียงในของทิศทาง.....	74

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 หุ่นยนต์ที่ใช้ในโรงงานอุตสาหกรรมยานยนต์ Honda [4].....	5
รูปที่ 2.2 Shakey หุ่นยนต์เคลื่อนที่ตัวแรกด้วยเซ็นเซอร์.....	6
รูปที่ 2.3 Planner-based system [5].....	7
รูปที่ 2.4 Reactive system.....	8
รูปที่ 2.5 Hybrid system.....	9
รูปที่ 2.6 ตัวอย่างการหาตำแหน่ง.....	10
รูปที่ 2.7 ฟังก์ชันตรีโกณมิติ.....	12
รูปที่ 2.8 ค่าตรีโกณมิติ.....	13
รูปที่ 2.9 มุมก้มและมุมเงย.....	14
รูปที่ 2.10 บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2).....	15
รูปที่ 2.11 สถาปัตยกรรมภายในของ PIC 16F877.....	17
รูปที่ 2.12 ตำแหน่งขาสัญญาณต่างของ PIC 16F877.....	17
รูปที่ 2.13 โมดูลตรวจจับและวิเคราะห์ทางควมัลตราโซนิก SRF05.....	18
รูปที่ 2.14 การส่งสัญญาณของอัลตราโซนิกเซ็นเซอร์.....	19
รูปที่ 2.15 ขาสัญญาณของ SRF05.....	20
รูปที่ 2.16 ไคอะแกรมเวลาแสดงสัญญาณที่ส่งไปยัง SRF05 และสัญญาณที่ตอบรับกลับมาจาก SRF05 ในกรณีติดต่อแบบ 2 สัญญาณ.....	21
รูปที่ 2.17 ไคอะแกรมเวลาแสดงสัญญาณที่ส่งไปยัง SRF05 และสัญญาณที่ตอบรับกลับมาจาก SRF05 ในกรณีติดต่อแบบ 1 สัญญาณ (ขา MODE ต้องต่อลงกราวด์).....	21
รูปที่ 2.18 การทำงานของอัลตราโซนิกเซ็นเซอร์.....	22
รูปที่ 2.19 การเกิด Critical Angle.....	23
รูปที่ 2.20 Crosstalk.....	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 2.21 โมดูล CMPS10 [14].....	25
รูปที่ 2.22 แสดงการต่อขาการใช้งาน CMPS10 ในโหมด I2C.....	26
รูปที่ 2.23 แสดงการต่อขาการใช้งาน CMPS10 ในโหมดบัสสื่อสารอนุกรม.....	26
รูปที่ 2.24 แสดงการต่อขาการใช้งาน CMPS10 ในโหมดบัสสื่อสารอนุกรม PWM.....	27
รูปที่ 2.25 รูปแบบการเขียน/อ่านข้อมูลแบบ I2C Bus [15].....	27
รูปที่ 2.26 I2C Bus (Control Byte) [15].....	28
รูปที่ 2.27 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3.....	29
รูปที่ 2.28 ตำแหน่งขาสัญญาณต่างของ ATmega328.....	30
รูปที่ 2.29 ส่วนประกอบบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3.....	30
รูปที่ 2.30 รูปวงจรควบคุมมอเตอร์ Motor Drive Shield (L293D)	32
รูปที่ 2.31 ผังการทำงานของเครื่องส่ง.....	33
รูปที่ 2.32 วงจรภาครีบริดิวซ์.....	34
รูปที่ 3.1 ภาพรวมของระบบ (System Overview)	37
รูปที่ 3.2 หุ่นยนต์เคลื่อนที่.....	39
รูปที่ 3.3 วงจรหุ่นยนต์.....	40
รูปที่ 3.4 ตำแหน่งการต่อมอเตอร์ที่ใช้ใช้งาน Motor Drive Shield (L293d)	41
รูปที่ 3.5 แสดงการทำงานของโปรแกรมควบคุมมอเตอร์.....	42
รูปที่ 3.6 แสดงรายละเอียดของรีจิสเตอร์ TICON.....	44
รูปที่ 3.7 โปรโตคอลสำหรับส่งข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ไปยังตัวโปรแกรมสร้างแผนที่	50
รูปที่ 3.8 แสดงการวาดรูปภาพบน Picture Box และแนวแกน X , Y.....	51
รูปที่ 3.9 แสดง Flowchart การทำงานของโปรแกรมหน้าจอ GUI สำหรับแสดงผล.....	55

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.10 ปุ่มที่ใช้ในการควบคุม.....	58
รูปที่ 4.1 ภาพรวมของระบบ.....	63
รูปที่ 4.2 หน้าจอ GUI ของระบบ.....	64
รูปที่ 4.3 สถานที่ทำการทดลอง.....	65
รูปที่ 4.4 การติดตั้งเซ็นเซอร์กับตัวหุ่นยนต์เคลื่อนที่.....	66
รูปที่ 4.5 แสดงความสัมพันธ์ระหว่างระยะทางจริงกับค่าที่นำมาใช้คำนวณ.....	67
รูปที่ 4.6 การทดลองหา Critical Angle.....	68
รูปที่ 4.7 ปัญหา Critical Angle.....	70
รูปที่ 4.8 ลักษณะลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05.....	72
รูปที่ 4.9 ผลการทดลองการหาขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05.....	73
รูปที่ 4.10 วิธีการทดลองวัดมุมเอียงของทิศทาง.....	74
รูปที่ 4.11 การกระจายข้อมูลของค่าความคลาดเคลื่อนจาก โมดูลเข็มทิศดิจิทัลCMPS10.....	76
รูปที่ 4.12 แสดงความสัมพันธ์ระหว่างค่าของค่าความคลาดเคลื่อนกับค่ามุมเอียงของทิศทาง.....	77
รูปที่ 4.13 ระบบที่พัฒนาขึ้นเพื่อทำการทดสอบการสร้างแผนที่.....	78
รูปที่ 4.14 ผลทดสอบการเดินหน้าและถอยหลัง.....	79
รูปที่ 4.15 สถานะของ Compass เมื่อมีการเดินหน้าหรือถอยหลัง.....	79
รูปที่ 4.16 ผลการทดสอบการเลี้ยวขวา.....	80
รูปที่ 4.17 สถานะของ Compass เมื่อเริ่มมีการเลี้ยว.....	80
รูปที่ 4.18 อุปกรณ์ที่ใช้ในการทดลอง.....	81
รูปที่ 4.19 การทดลองรูปสามเหลี่ยม.....	82
รูปที่ 4.20 ผลการจำลองรูปสามเหลี่ยม.....	82
รูปที่ 4.21 การทดลองรูปสี่เหลี่ยมขนาด 1 x 1 แผ่น.....	83

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.22 ผลการจำลองรูปสี่เหลี่ยมขนาด 1 x 1 แผ่น.....	83
รูปที่ 4.23 การทดลองรูปสี่เหลี่ยมขนาด 2 x 2 แผ่น.....	84
รูปที่ 4.24 ผลการจำลองรูปสี่เหลี่ยมขนาด 2 x 2 แผ่น.....	84
รูปที่ 4.25 การทดลองรูปสี่เหลี่ยมขนาด 3 x 3 แผ่น.....	85
รูปที่ 4.26 ผลการจำลองรูปสี่เหลี่ยมขนาด 3 x 3 แผ่น.....	85
รูปที่ 4.27 การทดลองรูปแปดเหลี่ยม.....	86
รูปที่ 4.28 ผลการจำลองรูปแปดเหลี่ยม.....	86
รูปที่ 4.29 การทดลองรูปสิบเอ็ดเหลี่ยม.....	87
รูปที่ 4.30 ผลการจำลองรูปสิบเอ็ดเหลี่ยม.....	87
รูปที่ 4.31 การทดลองรูปตัวแอล.....	88
รูปที่ 4.32 ผลการจำลองรูปตัวแอล.....	88

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

จากหลักฐานทางประวัติศาสตร์ที่เก่าแก่ที่สุดเท่าที่จะหาได้ในปัจจุบัน ระบุว่าศาสตร์ของการสำรวจนั้นได้เริ่มขึ้นในประเทศอียิปต์ [1] โดยมีสาเหตุมาจากการแบ่งที่ดินของประเทศออกเป็นผืนๆ เพื่อประโยชน์ในการเก็บภาษี แต่บางส่วนของที่ดินที่อยู่ตามฝั่งแม่น้ำในลุ่มน้ำกักตุนน้ำหายไปทุกปีจึงได้มีการมอบหมายหน้าที่ในการสำรวจซ่อมแซมหลักเขตที่ดินให้แก่พนักงานสำรวจในสมัยนั้น เรียกว่า “ผู้ขึงเชือก” (rope stretcher) โดยการใช้เชือกซึ่งทำเป็นปมๆเป็นเครื่องมือในการวัดที่ดิน ต่อมาชาวกรีกได้พัฒนาวิชาเรขาคณิต (geometry) ขึ้นมาเพื่อใช้คำนวณต่างๆ เพื่อการเขียนแบบแปลน แผนที่ ซึ่งยุคของการสำรวจที่แท้จริงนั้นเริ่มขึ้นในคริสต์ศตวรรษ ที่ 15 - 17 [2] ในช่วงนั้นชาวยุโรปเริ่มออกเดินทางสำรวจทางทะเลในโลกว้างออกไป โดยมีจุดประสงค์เพื่อหาผู้ค้าและแสวงหาดินแดนโดยใช้เข็มทิศในการกำหนดและระบุเส้นทางเพื่อสร้างแผนที่ ปัจจุบันการสำรวจได้รับการพัฒนาก้าวหน้าไปมาก โดยเฉพาะเครื่องมือที่ใช้ในการสำรวจ ได้ถูกพัฒนาให้มีประสิทธิภาพสูงขึ้น ปลอดภัยมากขึ้น และลดความเสี่ยงในการสำรวจ จนกระทั่งสามารถสร้างหุ่นยนต์เข้าไปสำรวจแทนมนุษย์ในสภาพแวดล้อมที่เข้าถึงยากได้

โครงการพัฒนาระบบสร้างแผนที่โดยใช้อัลตราโซนิกเซ็นเซอร์ (Development of Map Building System using Ultrasonic Sensor) นี้เป็นโครงการที่มีวัตถุประสงค์เพื่อพัฒนาต้นแบบของระบบที่ใช้สำหรับการสำรวจลักษณะทางกายภาพของสิ่งแวดล้อม โดยข้อมูลที่วัดได้จากเซ็นเซอร์จะถูกส่งไปประมวลผลที่เครื่องคอมพิวเตอร์ เพื่อแสดงผลออกมาทางหน้าจอ GUI (Graphic User Interface) ที่ได้ทำการพัฒนาขึ้น

โดยในการสร้างแผนที่นี้ จะใช้ข้อมูลจากเซ็นเซอร์ 2 ชนิด คือข้อมูลระยะทางจากอัลตราโซนิกเซ็นเซอร์ และข้อมูลทิศทางจากเข็มทิศดิจิทัล

การใช้เซ็นเซอร์สำหรับวัดระยะทางที่นิยมในปัจจุบันนั้น มีอยู่ 3 ชนิด คือ อินฟราเรดเซ็นเซอร์ [3] อัลตราโซนิกเซ็นเซอร์ [4] และเลเซอร์เซ็นเซอร์ [5] ซึ่งมีความสามารถในการทำงานแตกต่างกัน โดยอินฟราเรดเซ็นเซอร์นั้นมีประสิทธิภาพการทำงานค่อนข้างต่ำ มีความผิดพลาดเกิดขึ้นบ่อย ส่วนเลเซอร์เซ็นเซอร์นั้นมีประสิทธิภาพในการทำงานที่สูง และเกิดปัญหาน้อย แต่ตามมาคือราคาที่สูงมาก ทางคณะผู้ดำเนินโครงการจึงเลือกใช้อัลตราโซนิกเซ็นเซอร์ซึ่งคุณภาพการทำงานอยู่ในเกณฑ์ที่ดีในการดำเนินโครงการนี้

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาอัลตราโซนิกเซ็นเซอร์ที่ใช้สำหรับวัดระยะทางและโมดูลเข็มทิศดิจิทัลที่ใช้สำหรับตรวจสอบทิศทาง

1.2.2 เพื่อศึกษาการเขียนโปรแกรมไมโครคอนโทรลเลอร์สำหรับการประมวลผลค่าที่ได้รับมาจากอัลตราโซนิกเซ็นเซอร์และ โมดูลเข็มทิศดิจิทัลโดยโปรแกรมจะทำการแปลงสัญญาณที่ได้เป็นระยะทางและทิศทางตามลำดับ แล้วส่งข้อมูลที่ไปยังเครื่องคอมพิวเตอร์

1.2.3 เพื่อศึกษาการเขียนโปรแกรมการแสดงผลบนหน้าจอบนคอมพิวเตอร์สำหรับสร้างแผนที่

1.2.4 เพื่อศึกษาการควบคุมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่

1.3 ขอบข่ายของโครงการ

1.3.1 พัฒนาระบบที่ใช้สำหรับสร้างแผนที่ 2 มิติสำหรับแสดงผลบนคอมพิวเตอร์ โดยแสดงรายละเอียดของระยะทางและมุมที่ได้รับจากเซ็นเซอร์

1.3.2 ระบบที่พัฒนาขึ้นสามารถใช้งานร่วมกับหุ่นยนต์เคลื่อนที่ได้

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาการเขียนหน้าตาแสดงผลการสร้างแผนที่บนคอมพิวเตอร์

1.4.2 เขียนโปรแกรมแสดงผลการสร้างแผนที่พร้อมทดสอบ

1.4.3 ศึกษาวิธีการสร้างหุ่นยนต์เคลื่อนที่พร้อมทั้งอุปกรณ์ควบคุมระยะไกล

1.4.4 ศึกษาการทำงานของเซ็นเซอร์ทั้ง 2 ชนิด

1.4.5 ศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์

- 1.4.6 เขียน โปรแกรมควบคุมการทำงานของเซ็นเซอร์พร้อมทดสอบการทำงาน
- 1.4.7 สร้างหุ่นยนต์เคลื่อนที่พร้อมทั้งทดสอบการทำงานกับอุปกรณ์ควบคุมระยะไกล
- 1.4.8 ทดสอบการทำงานร่วมกันของไมโครคอนโทรลเลอร์และหน้าตาแสดงผล
- 1.4.9 สรุปผลการทดลองและจัดทำรายงาน

1.5 ตารางแสดงกิจกรรมการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กำหนดการดำเนินงาน	2556							
	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาการเขียนหน้าตาแสดงผลการสร้างแผนที่บนคอมพิวเตอร์								
2. เขียนโปรแกรมแสดงผลการสร้างแผนที่พร้อมทดสอบ								
3. ศึกษาวิธีการสร้างหุ่นยนต์เคลื่อนที่พร้อมทั้งอุปกรณ์ควบคุมระยะไกล								
4. ศึกษาการทำงานของเซ็นเซอร์ทั้ง 2 ชนิด								
5. ศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์								
6.เขียนโปรแกรมควบคุมกาทำงานของเซ็นเซอร์พร้อมทดสอบการทำงาน								
7. สร้างหุ่นยนต์เคลื่อนที่พร้อมทั้งทดสอบการทำงานกับอุปกรณ์ควบคุมระยะไกล								

กำหนดการดำเนินงาน	2556							
	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
8. ทดสอบการทำงานร่วมกันของไมโครคอนโทรลเลอร์และหน้าตาแสดงผล								
9.สรุปผลการทดลองและจัดทำรายงาน								

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้ระบบที่สามารถสร้างแผนที่ และแสดงผลบนหน้าจอกอมพิวเตอร์ได้

1.7 งบประมาณ

1.7.1 CP-PIC V3/877 (ICD2)	1,180	บาท
1.7.2 CMPS10 - Tilt Compensated Compass Module	1,835	บาท
1.7.3 SRF05 Ultrasonic sensor	3,000	บาท
1.7.3 Arduino Uno R3	500	บาท
1.7.4 motor drive shield (I293d)	500	บาท
1.7.5 Arduino Car	600	บาท
1.7.6 USB Serial Converter	500	บาท
1.7.7 วงจรวิทยุบังคับ	300	บาท
1.7.8 แบตเตอรี่	180	บาท
1.7.9 สายไฟ	120	บาท
1.7.10 ค่าเอกสารและจัดทำรูปเล่ม	200	บาท
รวมเป็นเงิน	8,915	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

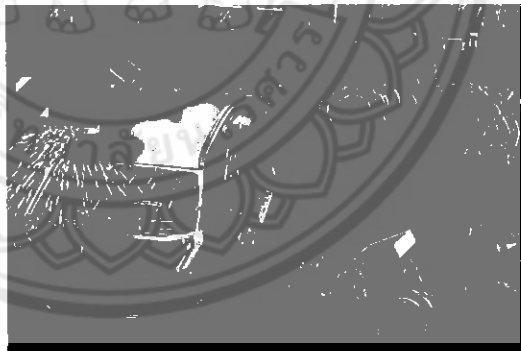
บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงสถาปัตยกรรมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ตรี โจนมิติเบื้องต้น การคำนวณหาระยะทางโดยใช้ตรี โจนมิติ ความสำคัญของหน้าจอแสดงผล (GUI) อุปกรณ์ทางฮาร์ดแวร์ที่นำมาใช้ในการสร้างระบบ และสุดท้ายจะกล่าวถึงซอฟต์แวร์ที่ใช้ในการพัฒนาระบบและไลบรารีที่เกี่ยวข้อง

2.1 สถาปัตยกรรมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ (Mobile Robot Architecture)

หุ่นยนต์ [6] หรือ Robot คือเครื่องจักรกลชนิดหนึ่ง มีลักษณะ โครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่าง ๆ ตามการควบคุมของมนุษย์ โดยการควบคุมและสั่งงานระหว่างหุ่นยนต์และมนุษย์ สามารถทำได้โดยทางกึ่งอัตโนมัติและอัตโนมัติ โดยทั่วไปหุ่นยนต์ถูกสร้างขึ้นเพื่อสำหรับงานที่มีความยากลำบากเช่น งานสำรวจในพื้นที่บริเวณแคบหรืองานสำรวจดวงจันทร์ดาวเคราะห์ที่ไม่มีสิ่งมีชีวิต ปัจจุบันเทคโนโลยีของหุ่นยนต์เจริญก้าวหน้าอย่างรวดเร็ว หุ่นยนต์ถูกแบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งาน คือ



รูปที่ 2.1 หุ่นยนต์ที่ใช้ในโรงงานอุตสาหกรรมยานยนต์ Honda [7]

1. หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) เป็นหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง มีลักษณะเป็นแขนกล สามารถขยับและเคลื่อนไหวได้เฉพาะแต่ละข้อต่อ ภายในตัวเองเท่านั้น มักนำไปใช้ในโรงงานอุตสาหกรรม เช่น โรงงานประกอบรถยนต์ ดังแสดงในรูปที่ 2.1



รูปที่ 2.2 Shakey หุ่นยนต์เคลื่อนที่ตัวแรกด้วยเซ็นเซอร์

2. หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) หุ่นยนต์ประเภทนี้จะแตกต่างจากหุ่นยนต์ที่ติดตั้งอยู่กับที่ เพราะสามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง โดยการใช้ล้อหรือการใช้ขา ซึ่งหุ่นยนต์ประเภทนี้ปัจจุบันยังเป็นงานวิจัยที่ทำการศึกษายู่ภายในห้องทดลอง เพื่อพัฒนาออกมาใช้งานในรูปแบบต่าง ๆ เช่น หุ่นยนต์สำรวจดาวอังคารขององค์การนาซ่า Shakey หุ่นยนต์เคลื่อนที่ตัวแรกด้วยเซ็นเซอร์ ดังแสดงในรูปที่ 2.2

ตามที่ได้กล่าวไว้แล้วว่าหุ่นยนต์เคลื่อนที่คือหุ่นยนต์ที่สามารถเคลื่อนที่ได้เอง โดยไม่ต้องอาศัยการควบคุมของมนุษย์ซึ่งการที่จะทำให้หุ่นยนต์มีความสามารถได้ต้องอาศัยส่วนสำคัญหลายส่วนซึ่งแบ่งเป็นระดับการประมวลผลได้ดังนี้

- Global Control มีหน้าที่จัดการรายละเอียดและขั้นตอนของพฤติกรรมต่างๆ ซึ่งจะเป็นตัวกำหนดความซับซ้อนของหุ่นยนต์
- Global Planning เป็นการวางแผนหรือการตัดสินใจเพื่อให้ไปถึงเป้าหมายที่ต้องการ
- Navigation เป็นการวางแผนหรือการตัดสินใจในการที่จะเคลื่อนที่
- Real-World Model เป็นการแทนสิ่งต่างๆที่อยู่ในโลกแห่งความเป็นจริงด้วยภาษาที่เป็นสัญลักษณ์ ซึ่งข้อมูลเหล่านี้จะรับมาจาก sensor และ action ที่หุ่นยนต์กระทำ

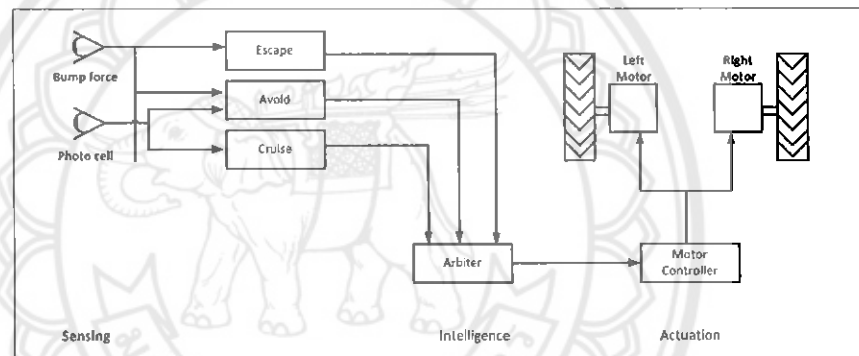
- Sensor Interpretation เป็นการติดต่อกับ sensor เพื่อวัดค่าต่างๆของสิ่งแวดล้อมแล้วทำการปรับแต่งให้เหมาะสมกับการประมวลผล

- Robot Control เป็นการควบคุมการทำงานทางกายภาพของหุ่นยนต์

โดยในแต่ละระดับจะประกอบด้วยหลายๆ โมดูลซึ่งจะมีความยืดหยุ่นตามการประยุกต์ใช้งานของหุ่นยนต์นั้นๆ

Robot control architectures หรือสถาปัตยกรรมควบคุมหุ่นยนต์จะมีอยู่ 3รูปแบบ คือ Deliberative (Planner-based), Reactive และ Hybrid

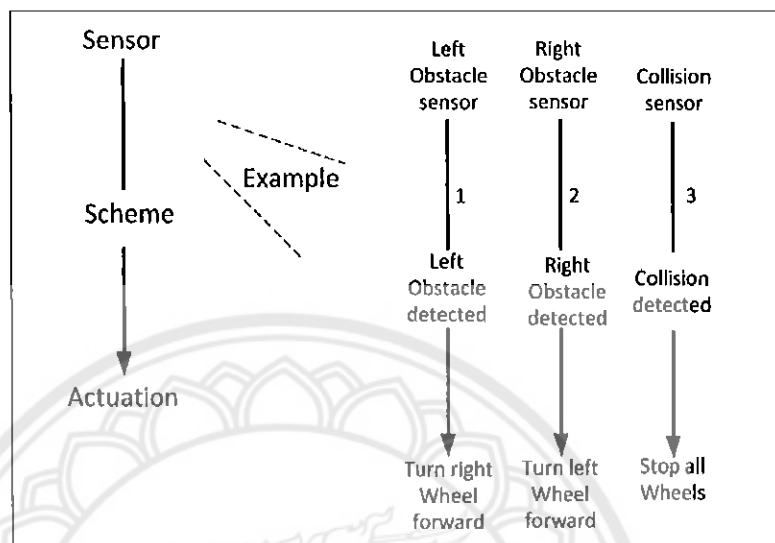
2.1.1 สถาปัตยกรรมการเคลื่อนที่แบบ Deliberative (Deliberative Architecture)



รูปที่ 2.3 Planner-based system [8]

จากรูปที่ 2.3 เป็นภาพของระบบพื้นฐานของการวางแผนซึ่งมีหลักการทำงานดังนี้ ระบบจะประกอบด้วยสามส่วน คือ ส่วนรับรู้ (Sensing) ส่วนความคิด (Intelligence) และส่วนทำงาน (Actuation) โดยจะมีการส่งค่าสภาพแวดล้อมมาจากเซ็นเซอร์ จากนั้นจะไปสู่ขั้นตอนของการคิดหรือ (Intelligence) เพื่อให้หุ่นยนต์คิดว่าต่อไปควรจะทำอย่างไรเมื่อมีการรับรู้มาจากภาคเซ็นเซอร์ และเมื่อหุ่นยนต์ได้มีการคิดแล้วก็จะไปสู่ขั้นตอนของการทำงานซึ่งเป็นขั้นตอนสุดท้าย และจะมีโมดูลที่รับข้อมูลจากเซ็นเซอร์ทั้งหมดสามตัวซึ่งในแต่ละตัวจะทำงานเป็นอิสระต่อกัน แต่ในบางครั้งโมดูลอาจจะต้องการสั่งงานพร้อมๆกันทำให้เกิดความขัดแย้งในการของใช้ทรัพยากร ดังนั้นจึงต้องมีโมดูลพิเศษขึ้นมาคือ โมดูล Arbiter เพื่อทำหน้าที่คอยจัดการหรือจัดเรียงระดับความสำคัญของโมดูลต่างๆ โมดูลใดมีความสำคัญกว่าก็จะถูกนำไปใช้งานก่อนหรือทำงานก่อน

2.1.2 สถาปัตยกรรมการเคลื่อนที่แบบ Reactive (Reactive Architecture)

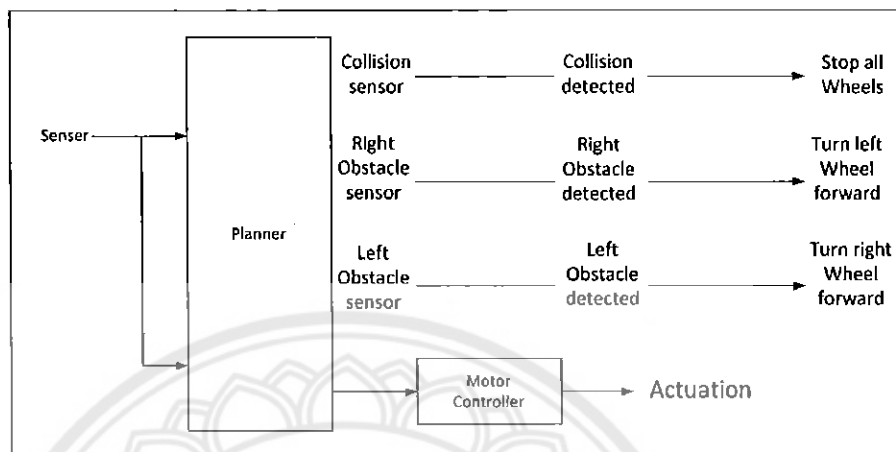


รูปที่ 2.4 Reactive system

จากรูปที่ 2.4 เป็นระบบตอบโต้หรือปฏิกิริยาตอบสนอง จะมีส่วนประกอบ 3 ส่วน คือส่วนรับรู้ (Sensor) ส่วนแบบแผน (Scheme) และส่วนทำงาน (Actuation) เมื่อมีการรับรู้จากส่วนรับรู้แล้ว ระบบจะแสดงผลลัพธ์ในทันทีที่มีหลักการทำงานดังนี้

1. เมื่อหุ่นยนต์มีการรับรู้ว่ามีสิ่งกีดขวางอยู่ทางซ้ายก็จะทำการหลบหลีกสิ่งกีดขวางแล้วทำการเลี้ยวขวา
2. เมื่อหุ่นยนต์มีการรับรู้ว่ามีสิ่งกีดขวางอยู่ทางขวาก็จะทำการหลบหลีกสิ่งกีดขวางแล้วทำการเลี้ยวซ้าย
3. เมื่อมีการรับรู้ว่ามีกรชนก็จะทำการตรวจจับการชนแล้วทำการหยุด

2.1.3 สถาปัตยกรรมการเคลื่อนที่แบบ Hybrid (Hybrid Architecture)

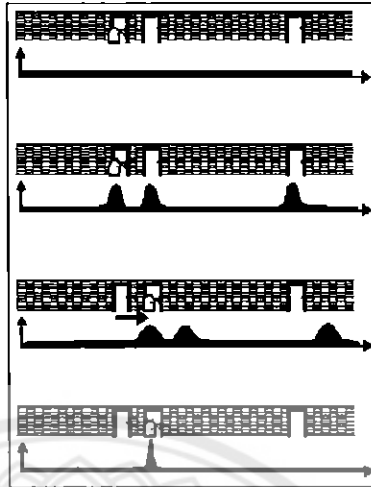


รูปที่ 2.5 Hybrid system

จากรูปที่ 2.5 เป็นระบบแบบผสมที่ได้มีการนำเอา Deliberative หรืออีกชื่อ Planner system เข้ามาช่วยในการทำงาน ระบบผสม (Hybrid) จะแบ่งออกเป็นสองส่วนคือเป็นระดับบนและระดับล่าง โดยระดับบน (higher level) จะเป็นในส่วนของ Planner ในส่วนนี้จะช่วยในการตัดสินใจ และระดับล่าง (lower level) จะเป็นในส่วนของปฏิกิริยาตอบสนอง reactive

2.2 ระบบ Localization

หลักการของ Markov Localization [9] หรือ Particle Filter Localization คือการหาค่าความน่าจะเป็นของตำแหน่งที่ในแผนที่ของสภาพแวดล้อมหนึ่งๆ จากการวัดค่าจากสภาพแวดล้อมจริง ซึ่งในการวัดค่าครั้งแรกผลที่ได้ อาจหมายถึงตำแหน่งที่ใดก็ได้ในแผนที่ หมายความว่าค่าความน่าจะเป็นมีการกระจายเท่ากันหมดในแผนที่ และเมื่อหุ่นยนต์ มีการเคลื่อนที่ไปจากเดิมจะทำให้มีการสะสมการเปลี่ยนแปลงของค่าที่วัดได้มากขึ้นเรื่อยๆ ส่งผลให้พื้นที่ต่างๆ ที่มีค่าความน่าจะเป็นของตำแหน่งที่ค่อยๆ ลดลงจนกระทั่งเหลือเพียงจุดที่มีความน่าจะเป็นมากที่สุดเพียงจุดเดียวที่เป็นตำแหน่งของตำแหน่งนั้นๆ



รูปที่ 2.6 ตัวอย่างการหาดำแหน่ง

จากรูปที่ 2.6 เมื่อมีการเคลื่อนที่ของหุ่นยนต์ เซ็นเซอร์จะวัดค่าสภาพแวดล้อมทำให้รู้ว่าหุ่นยนต์อยู่บริเวณประตู ความน่าจะเป็นของตำแหน่งของหุ่นยนต์จึงเพิ่มขึ้นที่บริเวณประตูทั้งหมด หลังจากนั้นจะมีการคาดเดาตำแหน่งจากรางที่

ตารางที่ 2.1 แสดงการคาดเดา present local และ future local

Presentlocal	Future local
ประตูที่1	ประตู
ประตูที่2	กำแพง
ประตูที่3	กำแพง

แต่เมื่อหุ่นยนต์เคลื่อนที่ไป เซ็นเซอร์จะวัดค่าสภาพแวดล้อมอีกครั้ง ทำให้รู้ว่าหุ่นยนต์อยู่บริเวณประตู ทำให้รู้ตำแหน่งจริงของหุ่นยนต์

2.3 ระบบ MapBuilding

การสร้างแผนที่ คือการที่หุ่นยนต์ต้องการปฏิบัติการกิจในสถานที่ที่ไม่มีแผนที่ในการอ้างอิง ตำแหน่งของหุ่นยนต์ หุ่นยนต์จะกำหนดตำแหน่งพิกัดของตัวหุ่นยนต์โดยอ้างอิงจากวัตถุหรือตำแหน่ง พิกัดที่ใกล้เคียงเช่นมุมห้อง ประตู หรือจุดหมายปลายทาง ซึ่งสถานที่หรือวัตถุเหล่านี้ อาจจะมีการ กำหนดพิกัดล่วงหน้า เพื่อใช้เป็นจุดเริ่มต้นให้การรับข้อมูลจากเซ็นเซอร์ [10] ซึ่งจากตัวอย่างในหัวข้อที่ 2.2 ระบบ Localization ได้มีการใช้ประตูเป็นจุดอ้างอิง เนื่องจากระบบไม่รู้ว่าตำแหน่งประตูอยู่ใน ตำแหน่งใด เมื่อทำการอ้างอิงจากพิกัดแล้วจะทำการรับรับข้อมูลจากเซ็นเซอร์ซึ่งอาจจะเป็นระยะทาง ของตัวเซ็นเซอร์กับวัตถุ แล้วนำมาใช้สร้างพื้นที่ว่างที่สามารถเคลื่อนที่ได้ของแผนที่แต่ละจะเป็นพิกัดใน ระบบอ้างอิงของพิกัดที่ตัวรถใช้อ้างอิง

2.4 ระบบ SLAM (Simultaneous Localization And Map building)

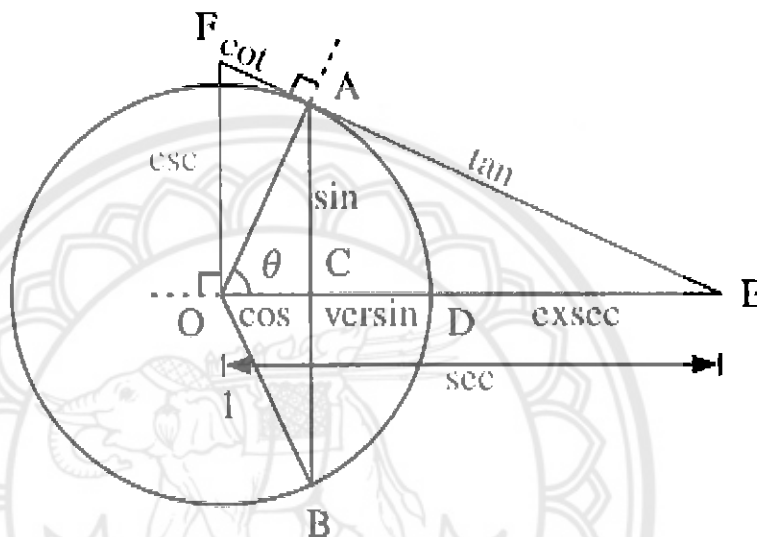
วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) [11] เป็นกระบวนการที่หุ่นยนต์จะสร้างแผนที่ของสภาพแวดล้อมในขณะที่กำลังเคลื่อนที่ และระบุ ตำแหน่งของตัวเองในเวลาพร้อมๆกัน โดยที่หุ่นยนต์นั้น ไม่มีข้อมูลของสิ่งแวดล้อมมาก่อน ซึ่ง SLAM นั้นมีความสำคัญเป็นอย่างมากสำหรับหุ่นยนต์ที่ต้องการการโต้ตอบแบบทันที

การระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) นั้น มีความยุ่งยากมากกว่าการระบุตำแหน่ง หรือการสร้างแผนที่เพียงอย่างเดียว ทั้งนี้เนื่องจากว่าในการระบุตำแหน่งนั้น จำเป็นต้องมีแผนที่ในการ อ้างอิง เพื่ออธิบายการวางตัวของหุ่นยนต์ ส่วนในการสร้างแผนที่นั้นก็จำเป็นต้องอาศัยตำแหน่งของ อุปกรณ์วัดค่า (ซึ่งมักจะติดตั้งบนตัวหุ่นยนต์) เพื่อใช้ในการจัดเรียง โครงสร้างของแผนที่อย่างถูกต้องจะ เห็นได้ว่าในปัญหา SLAM นั้นมีงานสองงานผูกพันกันอยู่ ถ้าหากงานใดมีความผิดพลาด ก็ย่อมส่งผล กระทบถึงอีกงานหนึ่งด้วย ด้วยเหตุนี้จึงทำให้ SLAM นั้น ไม่มีความทนทานต่อความผิดพลาด

2.5 ทริโกณมิติ

ตรีโกณมิติ (Trigonometry) เรียกอีกอย่างหนึ่งว่า Circular Function [12] คือ ฟังก์ชันของมุม ซึ่งมีความสำคัญในการศึกษารูปสามเหลี่ยมและปรากฏการณ์ในลักษณะเป็นคาบ ฟังก์ชันอาจนิยามด้วย อัตราส่วนของด้าน 2 ด้านของรูปสามเหลี่ยมมุมฉาก หรืออัตราส่วนของพิกัดของจุดบนวงกลมหนึ่ง หน่วยหรือนิยามในรูปทั่วไปเช่นอนุกรมอนันต์ [13] หรือสมการเชิงอนุพันธ์ [14] รูปสามเหลี่ยมที่

นำมาใช้จะอยู่ในระนาบแบบยูคลิด [15] ดังนั้นผลรวมของมุมทุกมุมจึงเท่ากับ 180° เสมอ ในปัจจุบัน มีฟังก์ชันตรีโกณมิติอยู่ 6 ฟังก์ชันที่นิยมใช้กันคือ sine cosine tangent cotangent secant และ cosecant ดังแสดงในรูปที่ 2.7



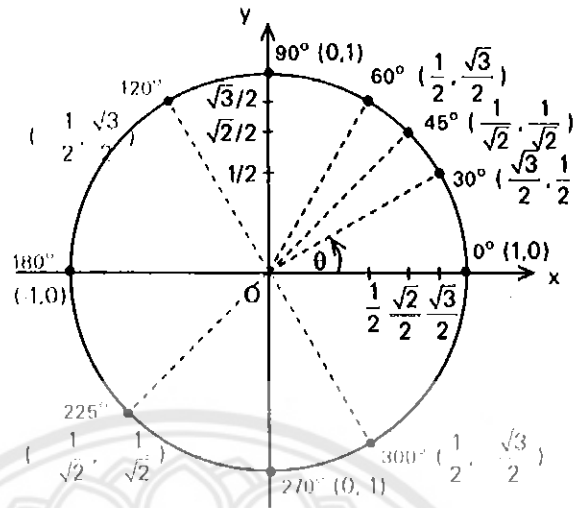
รูปที่ 2.7 ฟังก์ชันตรีโกณมิติ

การกำหนดค่าของฟังก์ชันตรีโกณมิตินั้น สามารถทำได้โดยการใช้วงกลมรัศมี 1 หน่วย มีจุดศูนย์กลางอยู่ที่จุดกำเนิดและเราจะเรียกวงกลมดังกล่าวว่า วงกลมหนึ่งหน่วย (The unit circle) เมื่อเรากำหนดจำนวนจริง θ (ทีตา) จาก $(1,0)$ วักระยะไปตามส่วนโค้งของวงกลม โดยมีข้อตกลงดังนี้

- ถ้า $\theta > 0$ จะวัดส่วนโค้งจากจุด $(1,0)$ ไปในทิศทางทวนเข็มนาฬิกา
- ถ้า $\theta < 0$ จะวัดส่วนโค้งจากจุด $(1,0)$ ไปในทิศทางตามเข็มนาฬิกา
- ถ้า $\theta = 0$ จุดปลายส่วนโค้งคือจุด $(1,0)$

จะได้ว่า เมื่อกำหนดจำนวนจริง θ สามารถหาจุด (x,y) ซึ่งเป็นจุดปลายส่วนโค้งได้เพียงจุดเดียวเท่านั้น

ถ้า $|\theta| > 2\pi$ แสดงว่า วัดส่วนโค้งเกิน 1 รอบ เพราะเส้นรอบวงของวงกลมยาว 2π หน่วย



รูปที่ 2.8 ค่าตรีโกณมิติ

จากรูปที่ 2.8 เมื่อ (x,y) เป็นจุดปลายส่วน โค้งของวงกลม

$$y = \sin \theta$$

$$x = \cos \theta$$

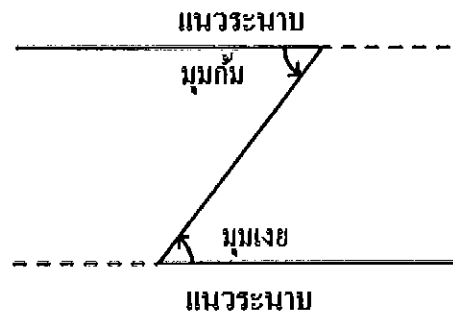
ฟังก์ชันไซน์และฟังก์ชันโคไซน์นั้น เป็นจำนวนจริง ตั้งแต่ -1 ถึง 1 นั่นคือ เรนจ์ของฟังก์ชันไซน์และโคไซน์ คือ เซตของจำนวนจริง ตั้งแต่ -1 ถึง 1 และ โดเมนของฟังก์ชันทั้งสองคือเซตของจำนวนจริง

2.5.1 การคำนวณหาระยะทางโดยใช้ตรีโกณมิติ

ในการวัดระยะทางและความสูงของสิ่งใดก็ตาม บางครั้งจะใช้เครื่องมือสำหรับวัดมาใช้ในการวัดโดยตรงไม่ได้ เช่น การวัดสถานที่สองแห่งที่มีสิ่งกีดขวางกั้นตรงกลาง หรือการวัดความสูงของภูเขา เป็นต้น เราสามารถนำความรู้ในเรื่องฟังก์ชันตรีโกณมิติ มาประยุกต์ใช้ในการคำนวณได้อันได้แก่

- มุมก้ม (Angle of Depression) คือ มุมที่วัดลงไปจากแนวราบ (ระดับสายตา)
- มุมเงย (Angle of Elevation) คือ มุมที่วัดขึ้นจากแนวราบ ดังแสดงในภาพ 2.9

รวมถึงการใช้กฎของไซน์และโคไซน์มาช่วยในการคำนวณ



รูปที่ 2.9 มุมก้มและมุมเงย

2.6 ความหมายและความสำคัญของ GUI

GUI ย่อมาจาก Graphical User Interface [16] คือการติดต่อกับผู้ใช้โดยใช้ภาพสัญลักษณ์เป็นการออกแบบส่วนของโปรแกรมคอมพิวเตอร์ให้มีการโต้ตอบกับผู้ใช้ โดยการใช้รูปภาพ และสัญลักษณ์อื่นๆ เพื่อแทนลักษณะต่างๆ ของโปรแกรมแทนที่ผู้ใช้จะพิมพ์คำสั่งต่างๆ ในการทำงานช่วยให้ผู้ใช้สามารถทำงานได้ง่าย และรวดเร็วขึ้น ไม่จำเป็นต้องจดจำคำสั่งต่างๆ ของโปรแกรมมากนักถือเป็นวิธีการที่ให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ให้ติดต่อกับระบบโดยผ่านทางภาพ เช่น ใช้เมาส์กดเลือก Icon แทนการพิมพ์คำสั่งตั้งแต่ก่อน โดยเฉพาะในบางโปรแกรมที่มีคำสั่งมากๆ เช่น โปรแกรม Autocad [17] ที่ใช้ในการวาดแบบ ซึ่งจะมี คำสั่งต่างๆ ที่ใช้ในการสร้างรูปมากมาย ผู้ใช้สามารถใช้เมาส์ เลือกคำสั่งที่ต้องการจะวาดจากสัญลักษณ์รูปภาพ ที่ปรากฏในโปรแกรมและใช้งานได้ โดยไม่ต้องพิมพ์คำสั่งต่างๆ ทางแป้นพิมพ์ช่วยให้เกิดความรวดเร็วในการทำงานและไม่ต้องเสียเวลาในการเรียนรู้และจดจำคำสั่งที่ต้องการมากนัก เพียงดูจากสัญลักษณ์รูปภาพ ที่ปรากฏในโปรแกรมก็สามารถใช้งานได้ทันที

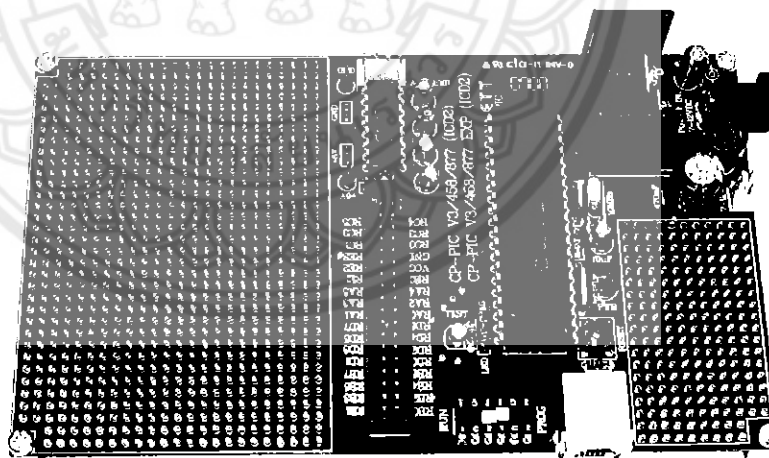
GUI ถูกพัฒนาขึ้นโดยนักวิจัยจากสถาบันวิจัยสแตนฟอร์ดนำโดย คัก เอนเกลบาร์ต (Doug Engelbart) [18] โดยการใช้งานร่วมกับไฮเปอร์ลิงก์และเมาส์ ซึ่งภายหลังได้นำมาวิจัยต่อที่ศูนย์วิจัยซีรีออกซ์พาร์ค (Xerox PARC) โดยใช้งานระบบกราฟิกแทนที่ระบบตัวอักษร โดยบางคนจะเรียกระบบนี้ว่า PARC User Interface หรือ PUI [19] ปลายคริสต์ศักราชที่ 1970 แอปเปิลคอมพิวเตอร์ [20] ได้นำมาใช้กับเครื่องแมคอินทอชซึ่งภายหลังทางไมโครซอฟต์ [21] ได้นำมาใช้กับระบบปฏิบัติการวินโดวส์ ซึ่งปัจจุบันนี้ระบบ GUI เป็นที่นิยมอย่างมากโดยสามารถเห็นได้ในทุกระบบปฏิบัติการไม่ว่าจะเป็นระบบปฏิบัติการวินโดวส์ แมคอินทอช หรือลินุกซ์ [22] ก็ตาม

ระบบ GUI สามารถนำเอามาประยุกต์ใช้กับระบบการสร้างแผนที่ในการแสดงข้อมูลระยะทางและทิศทางได้แบบ Real Time ซึ่งสามารถทำให้ผู้ควบคุมหุ่นยนต์สามารถรู้ได้ถึงจุดที่กำลังสำรวจ ส่งผลให้ผู้ควบคุมหุ่นสามารถสร้างแผนที่ออกมาได้อย่างครบถ้วน

2.7 บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2)

CP-PIC V3.0 (ICD2) [23] เป็นบอร์ดไมโครคอนโทรลเลอร์ซึ่งออกแบบวงจรเฉพาะส่วนพื้นฐานที่จำเป็นเช่นแหล่งจ่ายไฟวงจรรีเซตวงจรกำเนิดความถี่สัญญาณนาฬิกาพอร์ตสำหรับ Download โปรแกรมและวงจรสื่อสารอนุกรมดังแสดงในรูปที่ 2.10 ส่วนวงจร I/O ภายนอกนั้นจะไม่ได้จัดเตรียมไว้ให้ด้วยแต่จะทำการต่อสัญญาณ I/O ต่างๆจาก CPU มาไว้ยังขั้วต่อ Connector สำหรับให้ผู้ใช้นำไปเชื่อมต่อกับอุปกรณ์ I/O ภายนอกได้โดยง่ายและยังมีพื้นที่เอนกประสงค์สำหรับให้ผู้ใช้ออกแบบวงจร I/O และต่อวงจร I/O เพิ่มเติมได้เองเหมาะสำหรับผู้ใช้ที่ต้องการนำบอร์ดไปใช้พัฒนางานต้นแบบโดยการสร้าง I/O ต่างๆขึ้นมาใช้งานเองซึ่งในบอร์ดจะประกอบด้วย

- RS – 232 1 แชนแนล
- ETT CON 34PIN (ET BUS I/O 34PIN)
- 5 Volt Regulator On Board



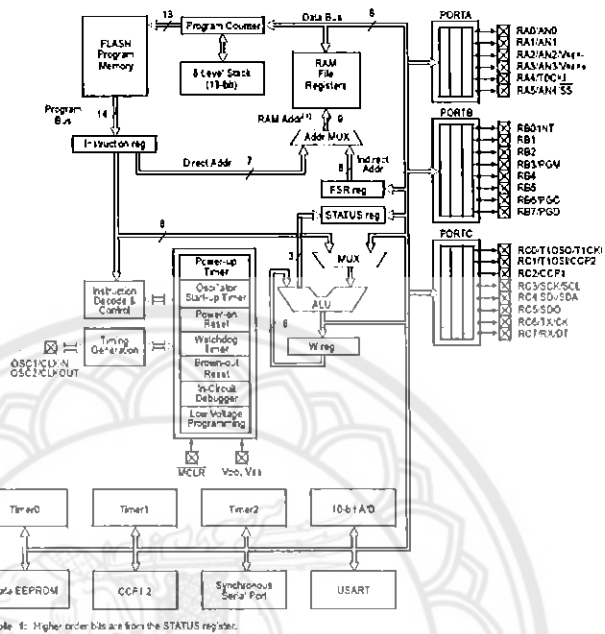
รูปที่ 2.10 บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2)

2.7.1 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ของ PIC 16F877

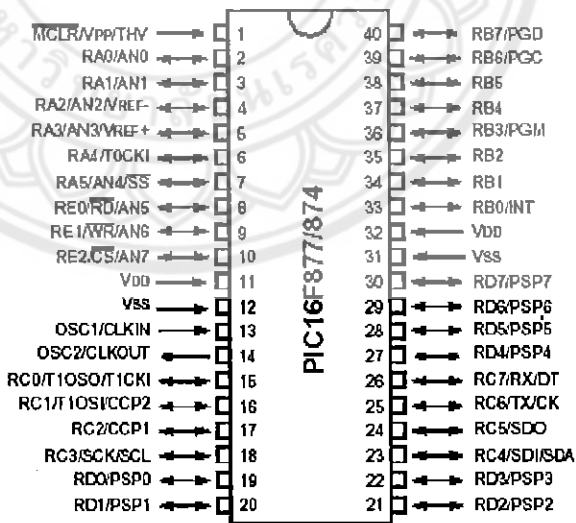
สถาปัตยกรรมของ PIC 16F877 เป็น CPU ของบริษัท Microchip Technology โดย IC เบอร์นี้เป็นเบอร์ที่มีความสามารถทำงานได้ในระดับปานกลาง และประกอบไปด้วยฟังก์ชันต่างๆดังนี้

- มีคำสั่งให้ใช้งาน 35 คำสั่ง
- คำสั่งต่างๆใช้เวลาทำงาน 1 ถึง 2 Cycle
- ทำงานได้สูงสุดที่ 20 MHz
- ทำงานแบบ Pipe-line
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8K
- มี RAM ขนาด 368 ไบต์
- มี EEPROM ขนาด 256 ไบต์
- ตอบสนองกับ Interrupt ได้ทั้งหมด 14 แหล่ง
- มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
- มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up timer
- Watchdog timer
- มีระบบ Code Protection
- มีโหมดประหยัดพลังงาน(Sleep Mode)
- สัญญาณนาฬิกาามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
- สามารถโปรแกรมด้วยไฟ +5V ได้
- ใช้การโปรแกรมแบบ In-Circuit Serial Programming
- ทำงานที่ไฟเลี้ยง 2.0 V ถึง 5.0 V
- Current Sink และ Current Source อยู่ที่ 25mA
- มี Timer/Counter 3 ตัว Timer0 –Timer2
- มีโมดูล Capture/Compare/PWM อีก 2 ชุด
- มี A-TO-D Converter แบบ 10 บิต จำนวน 8 ช่องนำเข้า
- มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232
- มีระบบตรวจระดับไปเลี้ยง (Brown-out reset)
- มี I/O พอร์ตทั้งหมด 5 พอร์ต

ทั้งนี้ยังแสดงส่วนประกอบภายในของ PIC 16F877 และตำแหน่งขาสัญญาณดังรูปที่ 2.11 และ 2.12 ตาลำดับ



รูปที่ 2.11 สถาปัตยกรรมภายในของ PIC 16F877

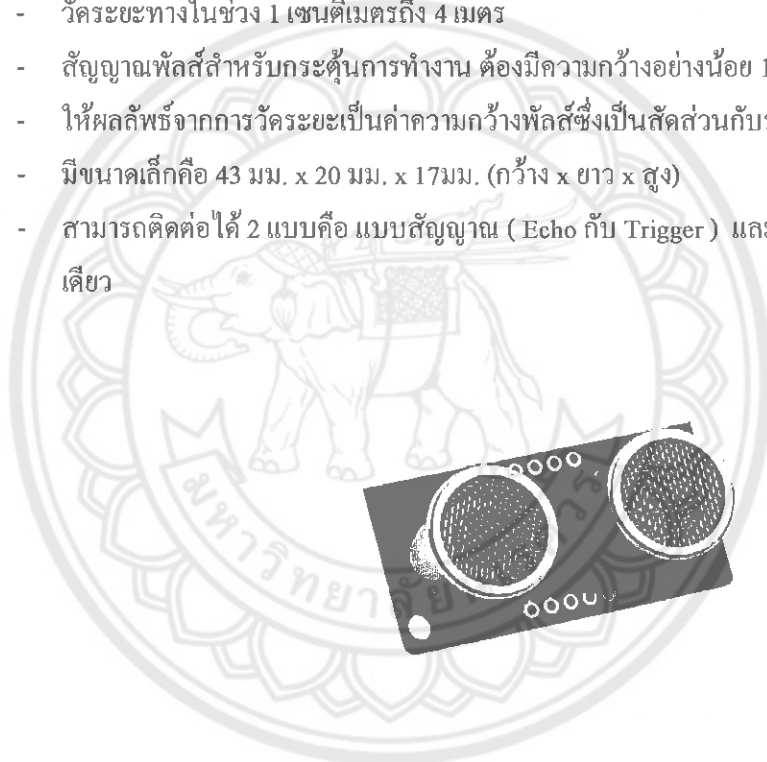


รูปที่ 2.12 ตำแหน่งขาสัญญาณต่างของ PIC 16F877

2.8 โมดูลตรวจจับและวัดระยะทางด้วยอัลตราโซนิกSRF05

SRF05 เป็นแผงวงจรตรวจจับและวัดระยะทางด้วยคลื่นอัลตราโซนิก (ดังแสดงในรูปที่ 2.13) ที่มีความเที่ยงตรงสูง โดยสามารถวัดระยะได้ตั้งแต่ 1 เซนติเมตรไปจนถึง 4 เมตร [24] SRF05 ถูกออกแบบมาให้ใช้งานกับไมโครคอนโทรลเลอร์อย่างง่ายโดยเชื่อมต่อเพียง 1 หรือ 2 ขา ขึ้นอยู่กับรูปแบบการทำงานทางฮาร์ดแวร์ เหมาะอย่างยิ่งกับการประยุกต์ใช้งานด้านหุ่นยนต์โดยมีคุณสมบัติดังนี้

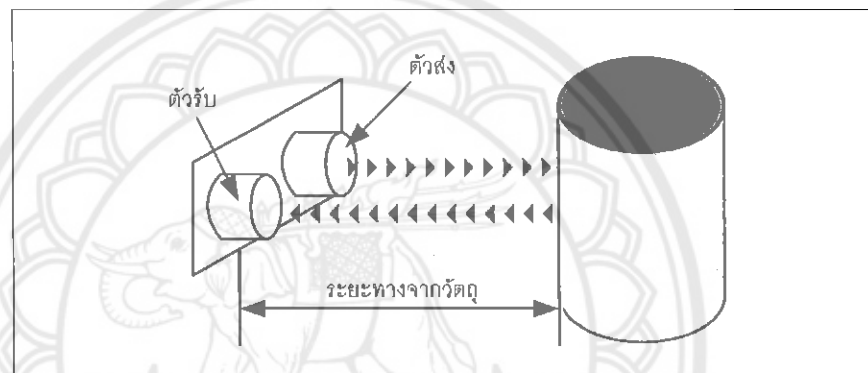
- ใช้ไฟเลี้ยง +5V ต้องการกระแสไฟฟ้า 30 mA
- ใช้ตัวรับและส่งคลื่นอัลตราโซนิก ใช้ความถี่ 40 kHz ในการทำงาน
- วัดระยะทางในช่วง 1 เซนติเมตรถึง 4 เมตร
- สัญญาณพัลส์สำหรับกระตุ้นการทำงาน ต้องมีความกว้างอย่างน้อย 10 ไมโครวินาที
- ให้ผลลัพธ์จากการวัดระยะเป็นค่าความกว้างพัลส์ซึ่งเป็นสัดส่วนกับระยะทางที่วัดได้
- มีขนาดเล็กคือ 43 มม. x 20 มม. x 17 มม. (กว้าง x ยาว x สูง)
- สามารถติดต่อได้ 2 แบบคือ แบบสัญญาณ (Echo กับ Trigger) และแบบอนุกรมสัญญาณเส้นเดียว



รูปที่ 2.13 โมดูลตรวจจับและวัดระยะทางด้วยอัลตราโซนิก SRF05

2.8.1 หลักการทำงาน

SRF05 จะทำการส่งสัญญาณคลื่นอัลตราโซนิกออกไป แล้ววัดระยะเวลาที่มีสัญญาณสะท้อนตอบกลับมา เอาค่าพุดที่ได้จาก SRF05 จะอยู่ในรูปของความกว้างพัลส์ซึ่งสัมพันธ์กับระยะทางของวัตถุที่ตรวจจับได้ ความถี่สัญญาณอัลตราโซนิกของ SRF05 คือ 40 kHz ถูกส่งออกไปในอากาศด้วยความเร็ว 1.125 ฟุตต่อมิลลิวินาที (ประมาณ 346 เมตรต่อวินาที) ดังนั้นเมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น และเวลาที่รับเสียงสะท้อนกลับมา จึงสามารถคำนวณหาระยะทางได้ ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 การส่งสัญญาณของอัลตราโซนิกเซ็นเซอร์

2.8.2 จุดต่อใช้งานของ SRF05

อัลตราโซนิกเซ็นเซอร์ SRF05 มีจุดต่อสำหรับใช้งานอยู่ทั้งหมด 5 จุด ดังแสดงในรูป 2.15 โดยมีรายละเอียดดังนี้

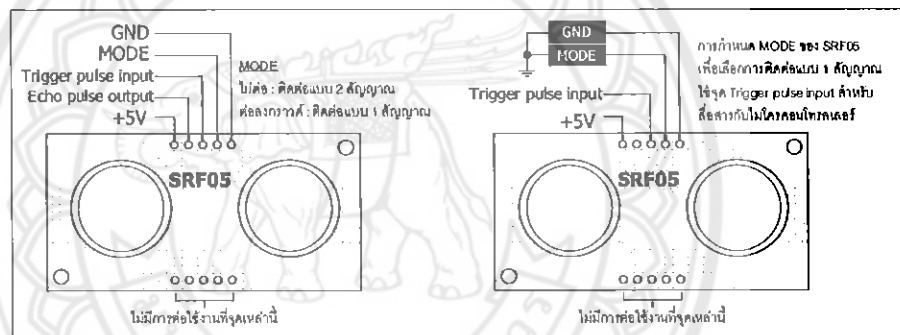
1. ขาไฟเลี้ยง (+5) สำหรับต่อไฟเลี้ยงแรงดัน +5 V
2. ขา Echo Pulse Output (ECHO) เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจาก SRF05 ซึ่งการใช้งานจะนำขานี้ไปต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์ที่ส่งออกมาเพื่อแปลความหมายออกมาเป็นระยะทางอีกครั้งหนึ่ง
3. ขา Trigger Pulse Input (TRIGGER) เป็นขาอินพุตรับสัญญาณพัลส์ที่มีความกว้างอย่างน้อย 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตราโซนิกความถี่ 40 kHz ออกสู่อากาศจากตัวส่ง ดังนั้น

เมื่อคลื่นความถี่ดังกล่าวนี้เคลื่อนที่ไปกระทบสิ่งกีดขวางที่อยู่เบื้องหน้าก็จะเกิดการสะท้อนกลับเข้ามายังตัวรับ และถูกแปลงออกมาเป็นความกว้างของสัญญาณพัลส์ที่จะส่งออกไปทางขา Echo Pulse Input นอกจากนี้ในโหมด 1 สัญญาณ จะใช้จุดนี้เป็นจุดสื่อสารข้อมูลอนุกรมเพื่อรับส่งค่า

4. ขา MODE สำหรับเลือกรูปแบบการติดต่อกับ SRF05

- ปลั๊กลอยไว้ (NC) : เลือกให้ติดต่อบน 2 สัญญาณ ผ่านจุดต่อ ECHO และ TRIGGER
- ต่อดงกราวด์ : เลือกให้ติดต่อบน 1 สัญญาณ ผ่านจุดต่อ TRIGGER

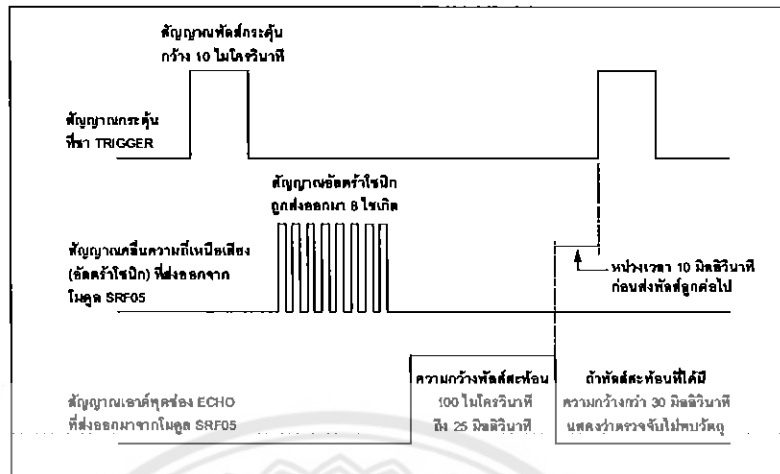
5. ขา GND สำหรับต่อดงกราวด์



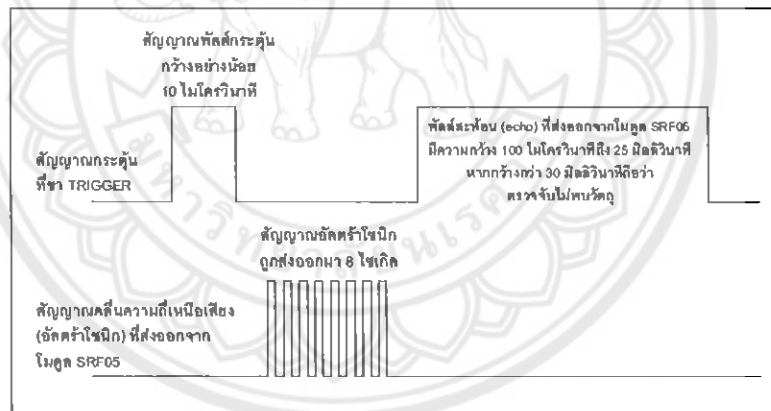
รูปที่ 2.15 ขาสัญญาณของ SRF05

สำหรับการทำงานอัลตราโซนิกเซ็นเซอร์ SRF05 นั้นสามารถทำงานได้ 2 แบบคือ

1. การติดต่อบน 2 สัญญาณ (Echo กับ Trigger) จะเริ่มทำงานก็ต่อเมื่อ มีการส่งสัญญาณพัลส์ขนาด 10 ไมโครวินาทีเข้ามาผ่านทางขา Trigger เพื่อเป็นการกระตุ้นให้อัลตราโซนิกเซ็นเซอร์ SRF 05 เริ่มทำงาน ต่อจากนั้นเซ็นเซอร์ก็จะทำการปล่อยสัญญาณอัลตราโซนิกผ่านออกมาเพื่อวัดระยะจากเซ็นเซอร์ถึงพื้นผิวของวัตถุและรับสัญญาณสะท้อนกลับเข้ามา และเซ็นเซอร์จะส่งผลลัพธ์ออกมาผ่านทางขา Echo ดังแสดงในรูปที่ 2.16
2. การติดต่อบนสัญญาณอนุกรมเส้นเดียว จะมีการทำงานโดยใช้ขา Trigger เพียงขาเดียว โดยมีลักษณะการทำงานคล้ายกัน แต่จะรับ-ส่งสัญญาณกันเพียงช่องทางเดียว ดังแสดงในรูปที่ 2.17



รูปที่ 2.16 ไคอะแกรมเวลาแสดงสัญญาณที่ส่งไปยัง SRF05 และสัญญาณที่ตอบรับกลับมาจาก SRF05 ในกรณีติดต่อแบบ 2 สัญญาณ



รูปที่ 2.17 ไคอะแกรมเวลาแสดงสัญญาณที่ส่งไปยัง SRF05 และสัญญาณที่ตอบรับกลับมาจาก SRF05 ในกรณีติดต่อแบบ 1 สัญญาณ (ขา MODE ต้องต่อลงกราวด์)

2.8.3 Criticle angle

เมื่อเสียงเดินทางจากตัวกลางที่มีครรชนีหักเหมากไปยังตัวกลางที่มีครรชนีหักเหน้อยรังสีหักเหจะเบนออกจากเส้นแนวฉากกรณีที่มุมหักเห เป็น 90 องศา รังสีหักเหจะขนานกับผิวรอยต่อเรียกมุมตกกระทบนี้ว่ามุมวิกฤติ(Critical Angle) [25] ซึ่งปัญหาดังกล่าวจะเกิดขึ้นกับการใช้งานอัลตราโซนิกเซ็นเซอร์เมื่อคลื่นอัลตราโซนิกเกิดการสะท้อนกับวัตถุรอบข้าง และย้อนกลับมายังตัวรับสัญญาณ ส่งผลให้ระยะทางที่ได้รับจากการคำนวณคลาดเคลื่อนไป เมื่อเทียบกับระยะทางจริง



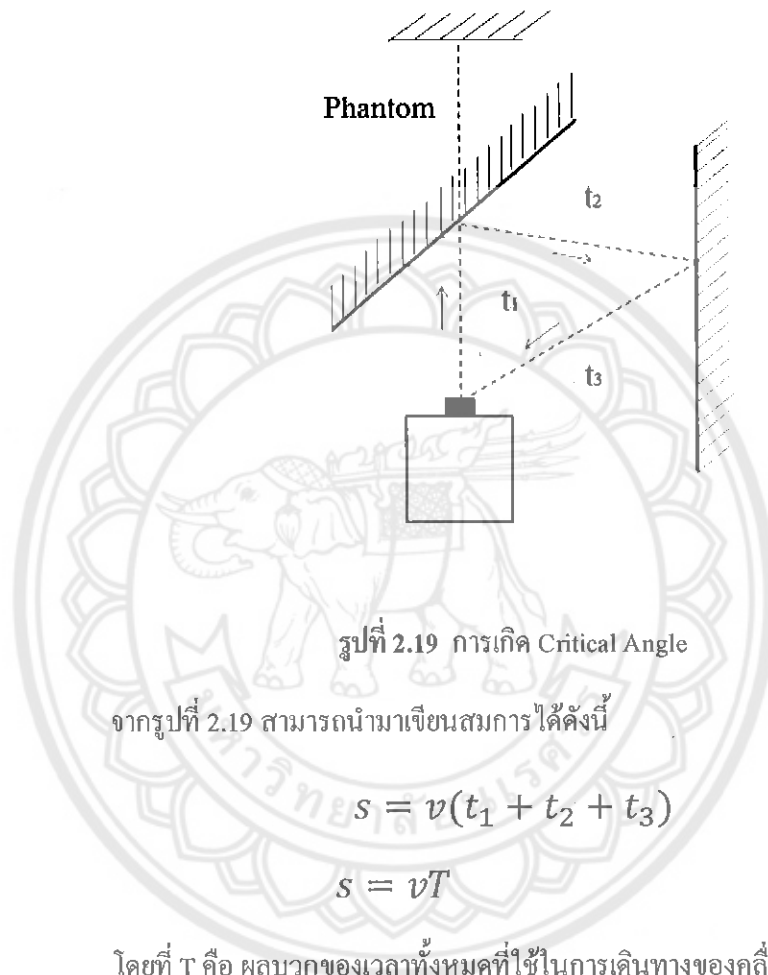
จากรูป 2.18 แสดงให้เห็นถึงการทำงานในสภาวะปกติของอัลตราโซนิกเซ็นเซอร์ โดยเริ่มจากการปล่อยคลื่นอัลตราโซนิกออกไปกระทบกับพื้นผิวของวัสดุ แล้วสะท้อนกลับมายังตัวรับสัญญาณ โดยแสดงสมการการหาระยะทางได้ดังนี้

$$s = vt$$

โดยเมื่อคำนวณผลคูณระหว่างความเร็วของคลื่นกับเวลา ก็จะทราบระยะทางระหว่างอัลตราโซนิกเซ็นเซอร์กับวัตถุ

ส่วนกรณีที่อัลตราโซนิกเซ็นเซอร์เกิดปัญหา Critical Angle นั้น เนื่องมาจากการที่คลื่นอัลตราโซนิกถูกปล่อยออกจากตัวเซ็นเซอร์ไปกระทบกับพื้นผิวของวัสดุ แล้วเกิดการสะท้อนไปยังทิศทางอื่น ก่อนไปกระทบกับพื้นผิวของวัสดุรอบข้างแล้วจึงสะท้อนกลับมายังตัวรับสัญญาณ ทำให้

เวลาที่นำมาใช้ในการคำนวณมากขึ้น ส่งผลให้ระยะทางระหว่างอัลตราโซนิกเซ็นเซอร์กับวัตถุนั้นมากขึ้นไปด้วย ดังแสดงในรูป 2.19



โดยที่ T คือ ผลบวกของเวลาทั้งหมดที่ใช้ในการเดินทางของคลื่น ทำให้ระยะทางที่ได้นั้นมีความมากเกินความจริง

2.8.4 Crosstalk

การแทรกสัญญาณข้ามหรือสัญญาณแทรกข้าม (Crosstalk) หมายถึงสัญญาณไฟฟ้าข้ามไปรบกวนกัน เป็นปรากฏการณ์ที่เกิดขึ้นได้ในการส่งสัญญาณในขณะที่สัญญาณเดินทางไปในสายทองแดงด้วยความเร็วสูง ปรากฏการณ์นี้มักพบได้เมื่อสื่อที่ใช้ในการส่งสัญญาณเป็นสายทองแดงตั้งแต่สองคู่ขึ้นไป เช่น โทศัพท์ซึ่งมีสายทองแดงมัดรวมกันเป็นคู่ นอกจากนี้ปรากฏการณ์นี้ยังสามารถเกิดจากสนามแม่เหล็กภายในสายทองแดงเองขณะกำลังส่งสัญญาณทำให้เกิดสัญญาณเสียงข้ามไปยังคู่สนทนาคู่อื่นได้แม้ว่าจะมีการหุ้มแผ่นโลหะไว้เพื่อป้องกันสัญญาณรบกวนแล้วก็ตาม

Crosstalk สามารถเกิดขึ้นกับอัลตราโซนิกเซ็นเซอร์ได้ ในกรณีที่มีการติดตั้งอัลตราโซนิกเซ็นเซอร์หลายตัวไว้บนอุปกรณ์เดียวกัน เมื่อสั่งให้อัลตราโซนิกเริ่มทำงานก็จะเริ่มทำการส่งคลื่นอัลตราโซนิกออกไป แต่คลื่นนั้นกลับสะท้อนกับพื้นผิวของวัสดุ แล้วย้อนกลับไปเข้าตัวรับสัญญาณของอัลตราโซนิกอีกตัว ทำให้ผลลัพธ์ที่ได้ออกมาผิดพลาดไปจากความเป็นจริง ดังแสดงตัวอย่างการเกิดปัญหา Crosstalk ดังรูปที่ 2.20



รูปที่ 2.20 Crosstalk

2.8.5 การประยุกต์ใช้งานอัลตราโซนิกเซ็นเซอร์SRF05

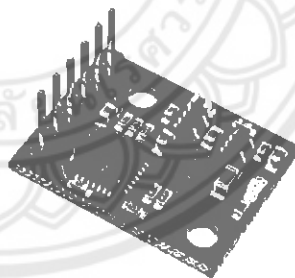
เนื่องจากการทำงานของอัลตราโซนิกเซ็นเซอร์ SRF05 สามารถทำงานได้ 2 รูปแบบ ในที่นี้ทางคณะผู้ดำเนินโครงการจึงขออธิบายในส่วนที่ได้ใช้งานจริง นั่นคือการติดต่อแบบ 1 สัญญาณ คือ เมื่อมีการส่งสัญญาณพัลส์ TRIGGER ขนาด 10 ไมโครวินาที ก็จะมีการส่งคลื่นอัลตราโซนิกจำนวน 8

ไซเคิล และจะได้รับพัลส์สะท้อนกลับมาจากวัตถุขนาด 100 ไมโครวินาทีถึง 25 มิลลิวินาที ซึ่งถ้าหากได้รับค่าเกิน 30 มิลลิวินาทีจะถือว่าตรวจสอบไม่พบวัตถุ และจะได้ค่าออกมาขนาด 16 บิต โดยแบ่งเป็นไบต์สูงและไบต์ต่ำตามลำดับ โดยนำค่าที่ได้มาแปลงเป็นเลขฐานสิบ แล้วหารด้วยค่าตามช่วงที่เหมาะสมจะได้ค่าเป็นหน่วยเซนติเมตรออกมา

2.9 โมดูลเข็มทิศและวัดความเอียงแบบมีการชดเชยค่า CMPS10

โมดูล CMPS10 [26] คือ โมดูลเข็มทิศดิจิทัลและ โมดูลวัดความเอียงในตัวเดียวกัน(ดังแสดงในรูปที่ 2.21) ภายในมีเซ็นเซอร์วัดสนามแม่เหล็ก 3 แกนและเซ็นเซอร์วัดความเร่งอีก 3 แกน ใช้ไมโครคอนโทรลเลอร์ 16 บิต (PIC24F) ที่มีประสิทธิภาพออกแบบมาเพื่อลดความผิดพลาดที่เกิดขึ้นจากการเอียงของแผ่นวงจรพิมพ์ด้วย CMPS10 จะให้เอาต์พุตออกมาหลายรูปแบบดังนี้

- ค่าทิศทางแบบหยาบ 0-255 สำหรับทิศทางรอบตัว
- ค่าทิศทางแบบละเอียด 0-3599 ใช้แทนทิศรอบตัว 0 ถึง 359.9 องศา
- ค่าดิบของเซ็นเซอร์วัดค่าสนามแม่เหล็กทั้ง 3 แกน 16 บิต
- ค่าดิบของเซ็นเซอร์วัดค่าความเอียงทั้ง 3 แกน 16 บิต
- ค่า มุมยก (pitch) มีค่าระหว่าง -85 ถึง 85 องศา เทียบกับแนวระนาบ
- ค่ามุมเอียง (roll) มีค่าระหว่าง -85 ถึง 85 องศา เทียบกับแนวระนาบ

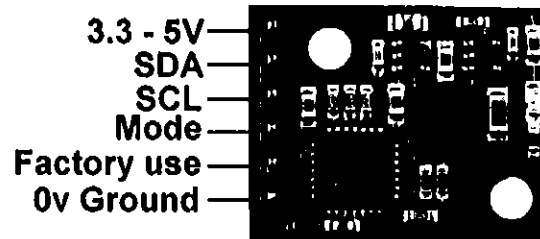


รูปที่ 2.21 โมดูล CMPS10 [14]

2.9.1 รูปแบบการใช้งานโมดูล CMPS10

CMPS10 ใช้ไฟเลี้ยง 3.3-5V กินกระแส 25mA สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้ 3 รูปแบบคือ I²C บัส, บัสสื่อสารอนุกรม, และ PWM โดยแต่ละโหมดการเชื่อมต่อจะมีรูปแบบดังนี้

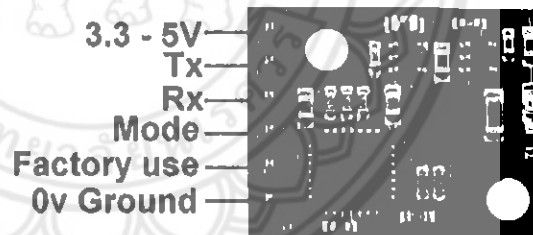
1. I²C [27] บัสมีรูปแบบขาใช้งานดังรูปที่ 2.22 ในโหมดนี้ขา Mode ต้องปล่อยลอยไว้ ขาใช้งานหลัก ๆ มี 4 ขา



รูปที่ 2.22 แสดงการต่อขาการใช้งาน CMPS10 ในโหมด I²C

- +3.3-5V ขาไฟเลี้ยง
- SDA ขารับส่งข้อมูล (serial data)
- SCL ขาสัญญาณนาฬิกา (serial clock)
- GND ขากราวด์

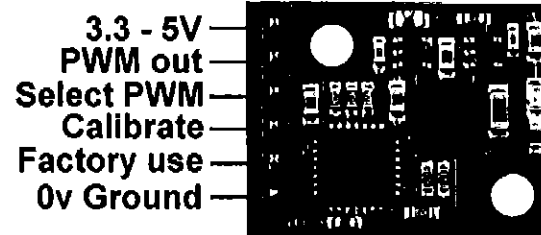
2. สื่อสารอนุกรม [28] มีขาต่อใช้งานดังรูปที่ 2.23 โดยขา Mode จะต้องต่อลงกราวด์ มีขาใช้งาน 4 ขาเช่นกันดังนี้



รูปที่ 2.23 แสดงการต่อขาการใช้งาน CMPS10 ในโหมดบัสสื่อสารอนุกรม

- +3.3-5V ขาไฟเลี้ยง
- Tx ขาส่งข้อมูล
- Rx ขารับข้อมูล
- GND ขากราวด์

3. การอ่านค่าโดยตรวจสอบค่าความกว้างพัลส์ใช้ขา Select PWM [29] ต่อลงกราวด์ ขาใช้งานมีเพียง 3 ขา ดังแสดงในรูปที่ 2.24



รูปที่ 2.24 แสดงการต่อขาการใช้งาน CMPS10 ในโหมดบัสสื่อสารอนุกรม PWM

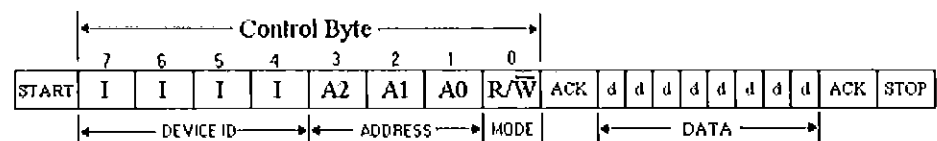
- +3.3-5V ขาไฟเลี้ยง
- PWM out ขารับค่าความกว้างพัลส์
- GND ขากราวด์

2.9.2 การประยุกต์ใช้โมดูล CMPS10

เนื่องจากการทำงานของโมดูลเข็มทิศและวัดความเอียงแบบมีการชดเชยค่า CMPS10 สามารถทำงานได้ 3 รูปแบบ ในที่นี้ทางคณะผู้ดำเนินโครงการจึงขออธิบายในส่วนที่ได้ใช้งานจริง นั่นคือการใช้งานในโหมด I²C

2.9.2.1 I²C

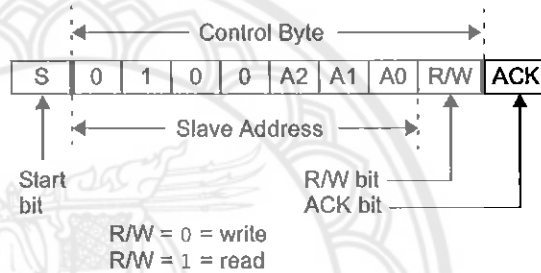
I²C ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆ ว่า I²C Bus เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) [30] เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) [31] กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors [32] โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) สามารถเชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น โดยมีรูปแบบการสื่อสารกับอุปกรณ์ดังนี้



รูปที่ 2.25 รูปแบบการเขียน/อ่านข้อมูลแบบ I²C Bus [15]

จากรูปที่ 2.25 การรับส่งข้อมูลแบบ I²C จะเริ่มต้นการส่งข้อมูลด้วยการ

- ส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส
- ตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัสประจำตัวอุปกรณ์ Device ID, Device Address, และ Mode ในการเขียนหรืออ่านข้อมูล
- เมื่ออุปกรณ์ รับทราบว่า MCU ต้องการ จะติดต่อกับบัสก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ส่งมามีความถูกต้อง
- และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่าสิ้นสุดการใช้บัส



รูปที่ 2.26 I²C Bus (Control Byte) [15]

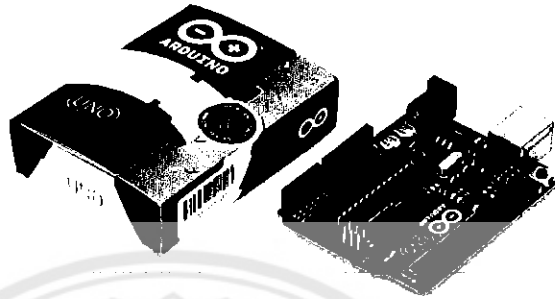
จากรูปที่ 2.26 แสดงรหัสควบคุมของ I²C Bus ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิต 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่าน

- รหัสประจำตัวของอุปกรณ์ ประกอบด้วยรหัสประจำตัวจากผู้ผลิต Product ID 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้สามารถกำหนดเองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุตัวอุปกรณ์ ที่ต่ออยู่บนบัส จะมีค่าซ้ำกันไม่ได้
- บิตควบคุมการเขียนอ่าน (Mode) บิต 0 เมื่อ MCU ต้องการเขียนข้อมูลไปยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการอ่านข้อมูลจากอุปกรณ์ก็กำหนดให้บิตนี้เป็น 1

2.9.3 การอ่านค่าทิศทางจากเอาต์พุตของสัญญาณพัลส์

การอ่านค่าทิศทางจากเอาต์พุตของสัญญาณพัลส์นั้น เป็นการนำค่าความกว้างของสัญญาณพัลส์ที่ได้มาระบุตำแหน่งองศาจาก 0 – 3599 เมื่อทำการหารด้วยสิบ จึงได้ผลลัพธ์ออกมาเป็น 0 – 359.9 องศา

2.10 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3



ARDUINO UNO R3

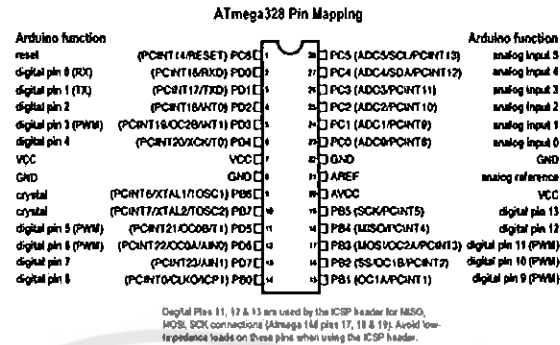
รูปที่ 2.27 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

จากรูปที่ 2.27 คือ Arduino Uno R3 [33] เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ ATmega328 เป็น MCUหลัก ซึ่งจะมีขา ดิจิตอล 14 ขา I/O สามารถทำเป็น PWM ได้ถึง 6 ขา และมีขา Analog อินพุตได้อีก 6 ขาซึ่งหลักการของ Arduino Board นี้ทำมาเพื่อความสะดวกง่ายในการเชื่อมต่อเข้ากับคอมพิวเตอร์ สามารถต่อ USB เข้ากับช่องคอมพิวเตอร์เหมาะสำหรับผู้ที่กำลังเริ่มต้นศึกษาไมโครคอนโทรลเลอร์

2.10.1 คุณสมบัติของบอร์ด ARDUINO UNO R3 โดยทั่วไป

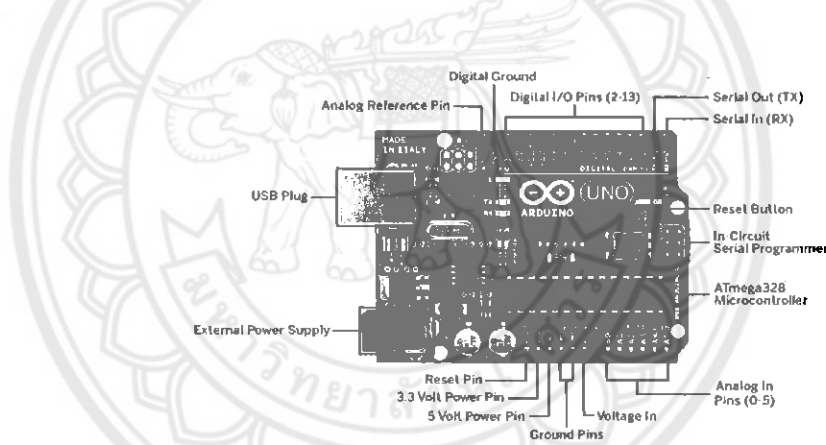
- ใช้ CPU Atmega[34]ตระกูล mx8 (168หรือ 328), ทำงานที่ความถี่16.00 เมกกะเฮิร์ต
- บนบอร์ดมี LED 4ดวง ซึ่งจะใช้แสดง power
- มีขาแอนาล็อก6 ขา
- มีขาดิจิทัล 14 ขา
- ขาดิจิทัลที่13ใช้สำหรับทดสอบการทำงานของบอร์ด โดยการต่อ LED จะแสดงสถานะ High เมื่อมีการใช้งานของบอร์ด
- บนบอร์ดมี LED 2 ดวงสำหรับแสดงการทำงานของขา TX และ RX
- ขาบนบอร์ดเป็นแบบมาตรฐานตามบอร์ดของ ATmega328 ดังแสดงในรูปที่2.28
- 6-pin standard ICSP header
- Auto-reset capability

Pin Diagram



รูปที่ 2.28 ตำแหน่งขาสัญญาณต่างของ ATmega328

2.10.2 หน้าที่การใช้งานพอร์ต I/O ของ ARDUINO UNO R3



รูปที่ 2.29 ส่วนประกอบบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

จากรูปที่ 2.29 เป็นการแสดงส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยมีขาสำหรับการใช้งานในรูปแบบ I/O อยู่ 3 ส่วนคือ ขาซีเรียลขาดิจิทัลและขาแอนาล็อก อธิบาย ดังนี้

- ขา 0 เป็นขาสัญญาณที่ทำหน้าที่ในการรับข้อมูล (Rx) ในโหมดการสื่อสารอนุกรม
- ขา 1 เป็นขาสัญญาณที่ทำหน้าที่ในการส่งข้อมูล (Tx) ในโหมดการสื่อสารอนุกรม

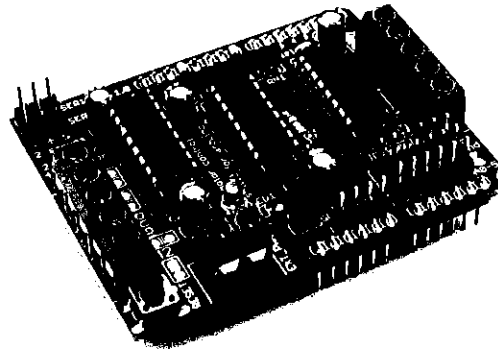
- ขา 2 - 13 เป็นขาสัญญาณที่ทำหน้าที่เป็น Digital I/O ปกติและมีบางขาที่สามารถทำงานในรูปแบบของ PWM ได้ คือขา 3 5 6 9 10 และ 11
- ขา A0-A5 เป็นขาสัญญาณที่ทำหน้าที่เป็น Analog I/O ปกติ

2.10.3 การประยุกต์การใช้งานบอร์ด Arduino Uno R3

บอร์ด Arduino Uno R3 ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ ซึ่งทำงานแยกออกจากบอร์ด PIC ที่ทำหน้าที่ควบคุมเซ็นเซอร์ เนื่องจากเมื่อเริ่มทำการศึกษาค้นคว้าพัฒนาได้เลือกใช้ Bluetooth ในการควบคุมจึงทำให้ระบบจะมีการใช้งานในส่วนของพอร์ตอนุกรมในการสื่อสารถึง 2 ส่วนคือ RS232 และ Bluetooth เพื่อลดการสื่อสารของพอร์ตอนุกรมจึงเปลี่ยนมาใช้วงจรรับคลื่นวิทยุบังคับ เพื่อลดการสื่อสารระหว่างคอมพิวเตอร์และระบบ

2.11 วงจรควบคุมมอเตอร์ Motor Drive Shield (L293D)

การควบคุมมอเตอร์กระแสตรงนั้น จะทำได้โดยการปรับแรงดันที่จ่ายให้มอเตอร์ ส่วนการควบคุมแรงบิด ทำโดยการควบคุมกระแสที่ผ่านขดลวดอาร์เมเจอร์ และที่จ่ายให้กับขดลวดสเตเตอร์ในกรณีสเตเตอร์แบบใช้ขดลวดพัน สำหรับการควบคุมการหมุน หรือการสลับทิศทางการหมุนนั้น ในกรณีมอเตอร์ไฟตรง สามารถทำได้โดยการสลับขั้วแหล่งจ่ายไฟที่จ่ายให้แก่มอเตอร์ สำหรับกรณีของมอเตอร์ไฟฟ้ากระแสตรงขนาดเล็กแบบใช้แม่เหล็กถาวรเป็นสเตเตอร์ จะไม่สามารถควบคุมอะไรได้มากนัก การควบคุมความเร็วของมอเตอร์จะถูกจำกัดด้วยขนาดแรงดัน โดยสามารถปรับได้ไม่เกินค่าแรงดันสูงสุดที่จ่ายให้มอเตอร์ ส่วนการควบคุมแรงบิด อาจทำได้โดยใช้ตัวต้านทานปรับค่าแบบขดลวด (Wire Wound Resistor [35]) แต่ก็จะมีกำลังไฟฟ้าสูญเสียที่ตัวต้านทาน ฉะนั้นในปัจจุบันจึงมักนิยมใช้การควบคุมด้วยวงจรพัลส์วิธมอดูเลเตอร์ (Pulse Width Modulator) ซึ่งจะใช้วิธีจ่ายไฟให้แก่มอเตอร์เป็นช่วง ๆ โดยการควบคุมแรงดัน คือการปรับช่วงกว้างของพัลส์ที่จ่ายให้มันเอง วิธีนี้จะทำให้ลดกำลังสูญเสียได้มาก สำหรับการกลับทางหมุนของมอเตอร์ อาจใช้วิธีสลับขั้วด้วยมือ หรือใช้วงจรรีเลย์หรืออิเล็กทรอนิกส์เข้าไปควบคุม



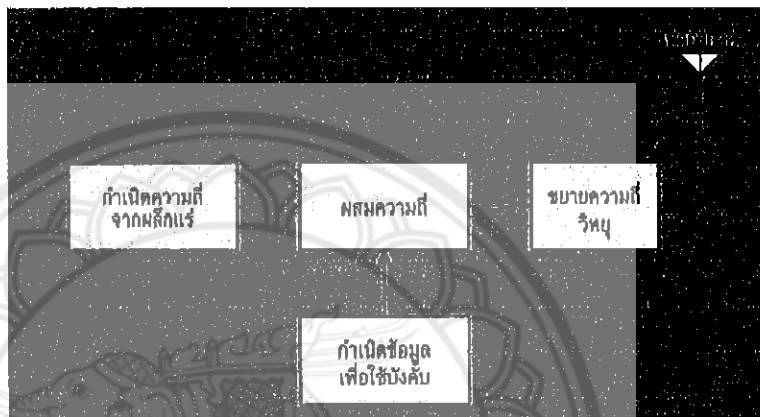
รูปที่ 2.30 รูปวงจรถามุมมอเตอร์ Motor Drive Shield (L293D)

จากรูป 2.30คือ Motor Drive Shield (L293D) [36] ทำหน้าที่ควบคุมแรงดันที่ส่งให้มอเตอร์โดยควบคุมจากบอร์ดมีคุณสมบัติดังนี้

- ใช้ไฟเลี้ยง 5V จากไมโครคอนโทรลเลอร์
- ใช้ IC L293D [37] ในการควบคุมมอเตอร์
- ขับมอเตอร์ด้วยแรงดัน 6.5-12v (VIN Power Supply), 4.5-25V (External Power Source)
- ควบคุมได้ 2 ทิศทาง
- จ่ายกระแสสำหรับควบคุม 36mA
- จ่ายกระแสให้มอเตอร์ได้มากที่สุด 0.6A
- ควบคุมมอเตอร์ได้สูงสุด 4 ตัว
- ควบคุมการทำงาน โดยการใช้ libraryAFMotor.h
- สัญญาณควบคุม
 - High level: $2.3V \leq V_{in} \leq 5V$
 - Low level: $-0.3V \leq V_{in} \leq 1.5V$

2.12 วงจรรับคลื่นวิทยุบังคับ

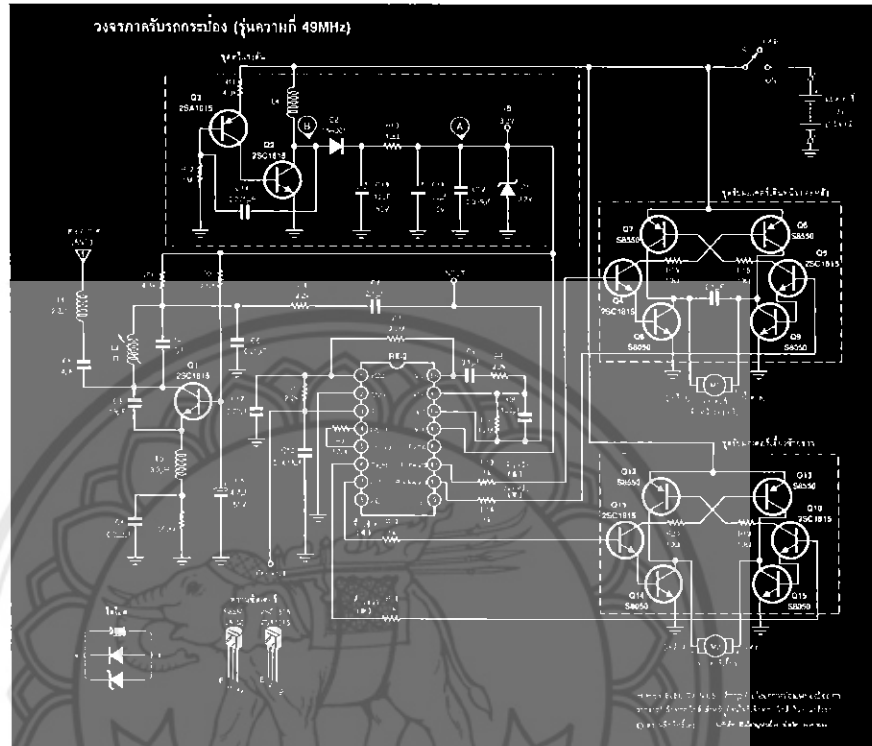
คลื่นวิทยุ (Radio Frequency) [38] เป็นคลื่นแม่เหล็กไฟฟ้าความถี่สูง ซึ่งมีคุณสมบัติกระจายไปได้เป็นระยะทางไกล เครื่องส่งวิทยุจะทำหน้าที่สร้างคลื่นแม่เหล็กไฟฟ้าความถี่สูงหรือคลื่นวิทยุ (RF) ผสมกับคลื่นเสียง (Audio Frequency -AF) แล้วส่งกระจายออกไปดังในรูปที่ 2.31 เครื่องรับวิทยุ จะทำหน้าที่รับคลื่นวิทยุและแยกคลื่นเสียงออกจากคลื่นวิทยุให้รับฟังเป็นเสียงปกติได้



รูปที่ 2.31 ฟังก์ชันการทำงานของเครื่องส่ง

ย่านความถี่ที่ใช้กันมากในการบังคับด้วยวิทยุก็คือย่าน 27 MHz นอกจากนี้ยังมีย่าน 40 MHz, 53 MHz, 72 MHz และ 75 MHz กำลังส่งออกอากาศอยู่ในระหว่าง 300 มิลลิวัตต์ ถึง 1 วัตต์ ซึ่งได้ระยะในการบังคับมากเพียงพอต่อการใช้งาน ส่วนระบบผสมคลื่นที่ใช้กันในวิทยุบังคับมีทั้งระบบ AM และระบบ FM ที่จะกล่าวถึงในที่นี้เป็นระบบ AM ซึ่งมีข้อดีคือทำการผสมคลื่น (Modulation) ได้ง่าย เพียงแต่การปิด-เปิดให้เป็นพัลส์ของคลื่นพาห์ออกอากาศไปเป็นห่วง ๆ การดีเทคเตอร์ในภาครับก็ไม่ยุ่งยาก

2.12.1 การประยุกต์ใช้วงจรรับคลื่นวิทยุบังคับ



รูปที่ 2.32 วงจรภาครับวิทยุบังคับ

ในการใช้งานนั้น ภาครับของวงจร ใช้ในการจับคลื่นวิทยุจากรีโมทเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยทำการนำสัญญาณเอาต์พุตจาก IC RX-2 ที่ขา 6 1 10 และ 11 ดังรูปที่ 2.32 ต่อเข้ากับบอร์ด Arduino Uno R3 เพื่อรับสัญญาณควบคุม

2.13 ภาษา C#

C# [39] เป็นภาษา object-oriented programming [40] จาก Microsoft ที่มีจุดมุ่งหมายในการรวมความสามารถคำนวณของ C++ [41] ด้วยการโปรแกรมง่ายกว่าของ Visual Basic [42] โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java [43]

C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคืออำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บและทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ใน

ขนาดกระทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) [44] และ Simple Object Access Protocol (SOAP) [45] ซึ่งยอมให้เข้าถึงอ็อบเจกต์ของโปรแกรมหรือเมธอดโดยปราศจากความจำเป็นต้องให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอนเนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ C# ได้รับการคาดหวังว่าทำให้เร็วขึ้นและต้นทุนต่ำในทำผลิตภัณฑ์และบริการเข้าสู่ตลาด

2.14 AForge.NET

AForge.net เป็นเฟรมเวิร์คที่ถูกพัฒนาสำหรับภาษา C# สำหรับใช้ในการวิจัยและพัฒนาในด้านคอมพิวเตอร์วิทัศน์และระบบปัญญาประดิษฐ์ (Artificial Intelligence) [46] เช่นการประมวลผลภาพ (Image Processing) [47] โครงข่ายประสาท (Neural Network) [48] การประยุกต์ใช้ในพันธุกรรมคอมพิวเตอร์ (Genetic Algorithms) [49] การเรียนรู้ของเครื่อง (Machine Learning) [48] และวิทยาการหุ่นยนต์ (Robotics) เป็นต้นตัวอย่างของชุดคำสั่งไลบรารีของ AForge.NET ประกอบไปด้วย

- AForge.Imaging เป็นชุดคำสั่งที่ใหญ่ที่สุดในขอบเขตงานประกอบด้วยชุดคำสั่งการประมวลผลภาพต่างๆซึ่งช่วยในการปรับแต่งภาพและการประมวลผลภาพต่างๆ
- AForge.Vision เป็นชุดคำสั่งที่ประกอบด้วยเครื่องมือตรวจจับการเคลื่อนไหวหรือท่าทางต่างๆ และชุดคำสั่งการประมวลผลการเคลื่อนไหว
- AForge.Video เป็นชุดคำสั่งที่ประกอบด้วยคลาสต่างๆที่ใช้ในการเข้าถึงข้อมูลวิดีโอ
- AForge.Robotics เป็นชุดคำสั่งที่ใช้ควบคุมอุปกรณ์เกี่ยวกับหุ่นยนต์
- AForge.Neuro เป็นชุดคำสั่งที่ประกอบด้วยวิธีการทางโครงสร้างเครือข่ายประสาทเทียมและขั้นตอนวิธีการเรียนรู้ต่างๆ
- AForge.Genetic เป็นชุดคำสั่งที่ประกอบด้วยคลาสที่ใช้แก้ปัญหงานต่างๆจาก Genetic Algorithms, Genetic Programming และ Gene Expression Programming
- AForge.MachineLearning เป็นชุดคำสั่งที่ประกอบด้วยคลาสที่เกี่ยวกับการเรียนรู้ของเครื่อง

โดยในโครงการนี้มีการนำฟังก์ชัน AForge.Video มาใช้ในการติดตั้ง webcam เพื่อเปรียบเทียบสถานที่ทำการทดลองจริงกับผลลัพธ์ของระบบ

2.15 บทสรุป

ในบทที่ 2 นี้ กล่าวถึงหลักการและทฤษฎีทั้งหมดที่เกี่ยวข้องสำหรับการสร้างระบบ เริ่มต้นตั้งแต่สถาปัตยกรรมการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ระบบ Localizationระบบ MapBuildingระบบ SLAMตรีโกณมิติ ความหมายและความสำคัญของ GUIบอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2) โมดูลตรวจจับและวัดระยะทางด้วยอัลตราโซนิกSRF05 โมดูลเข็มทิศและวัดความเอียงแบบมีการชดเชยค่า CMPS10บอร์ดไมโครคอนโทรลเลอร์ARDUINO UNO R3 วงจรควบคุมมอเตอร์ Motor Drive Shield (L298P) และวงจรรับคลื่นวิทยุบังคับ และสุดท้ายกล่าวถึง C# ซึ่งเป็น Software ที่ใช้ในการพัฒนาระบบสร้างแผนที่ รวมทั้งไลบรารี AForge.NET



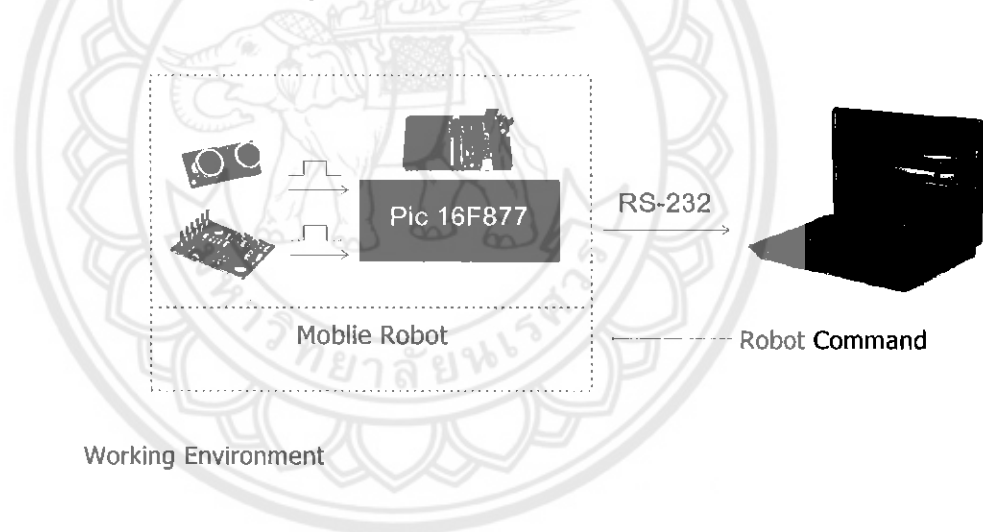
บทที่ 3

วิธีการดำเนินการ

ในบทนี้คณะผู้ดำเนินโครงการจะกล่าวถึงรายละเอียดการดำเนินโครงการ โดยแบ่งเป็น 3 ส่วน ดังนี้

1. ภาพรวมของระบบ จะกล่าวถึงภาพรวมของระบบในแต่ละส่วนว่าสัมพันธ์กันอย่างไร และมีระบบการทำงานอย่างไร
2. การเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์เพื่อใช้ในการอ่านเซ็นเซอร์ที่ใช้สำหรับวัดระยะทางและทิศทาง
3. การเขียนโปรแกรมสร้างหน้าจอ GUI สำหรับแสดงผล

3.1 ภาพรวมของระบบ (System Overview)



รูปที่ 3.1 ภาพรวมของระบบ (System Overview)

จากรูปที่ 3.1 เป็นการแสดงภาพรวมของระบบได้มีการใช้อัลตราโซนิกเซ็นเซอร์ SRF05 เป็นตัววัดระยะทางจากตำแหน่งของอัลตราโซนิกเซ็นเซอร์กับวัตถุ ส่วนทิศทางนั้นจะได้จาก โมดูลเข็มทิศดิจิทัลCMPS10 โดยมีความคุมการทำงานด้วย CP-PIC V3.0 (ICD2) ซึ่งเป็นไมโครคอนโทรลเลอร์ทำหน้าที่ในการประมวลผลของสัญญาณที่ได้มาจากเซ็นเซอร์ทั้ง 2 ตัว จะได้ผลลัพธ์ออกมาเป็นระยะทาง

และทิศทางตามลำดับ แล้วส่งผลลัพธ์ที่ได้ขึ้นไปวาดแผนที่และแสดงผลออกทางหน้าจอคอมพิวเตอร์ ในส่วนของ Mobile Robot นั้นให้ในการเปลี่ยนตำแหน่งพิกัดและทิศทางของอัลตราโซนิกเซ็นเซอร์ และโมดูลเซ็นเซอร์ดิจิทัล CMPS10 ซึ่งมีหลักการทำงานในการเปลี่ยนพิกัดตำแหน่งด้วยการใช้มอเตอร์ในการเคลื่อนที่ โดยมีมนุษย์เป็นผู้ควบคุม ภายใต้วงแวดล้อมที่มีการทำงานตลอดเวลา

3.2 การเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมการเคลื่อนที่

3.2.1 การสร้างหุ่นยนต์เคลื่อนที่

การเคลื่อนที่ของหุ่นยนต์นั้นส่วนใหญ่เป็นการเคลื่อนที่ในแนวราบ อุปกรณ์ที่ใช้ในการเคลื่อนที่ที่ดีที่สุดก็จะเป็น ล้อ, ราง ที่นำมาใช้กับระบบการเคลื่อนที่ของหุ่น มันสามารถทำงานในภูมิประเทศที่เป็นสภาพทุรกันดาร มีการเคลื่อนที่ได้อย่างต่อเนื่องยิ่งถ้ามีหลายล้อความมีเสถียรภาพในการเคลื่อนที่ก็มีมาก

3.2.1.1 อุปกรณ์

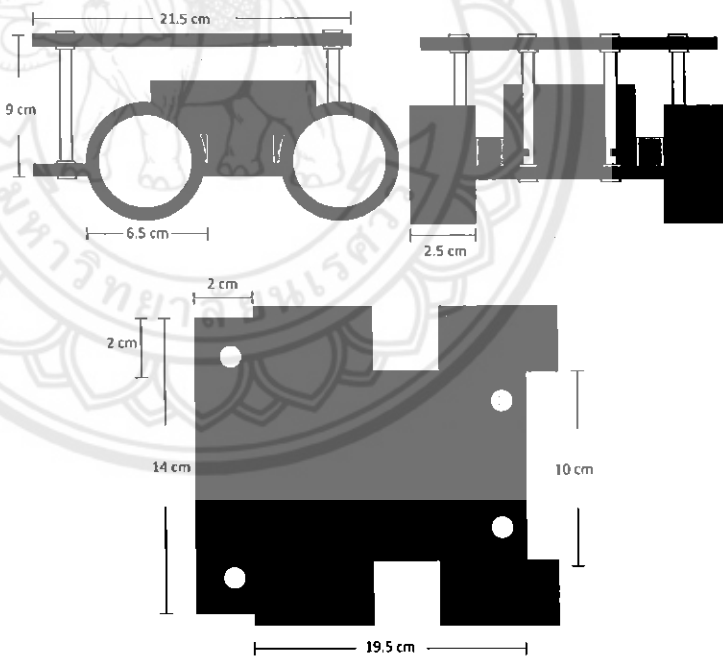
1. Dc motor	4ตัว
2. แผ่นอะคริลิก ขนาด 10cm x 5cm	2แผ่น
3. วงจรควบคุมมอเตอร์ Motor Drive Shield (L293d)	1 วงจร
4. ล้อ	4 ล้อ
5. น็อตตัวผู้ เบอร์ M4 (4 มม. x 100 มม.)	4ตัว
6. น็อตตัวเมีย เบอร์ M4 (4 มม.)	12ตัว
7. วงจรส่งสัญญาณ RX	1 วงจร
8. วงจร volt regulator 5V	1 วงจร
9. แบตเตอรี่ ขนาด 12v	1 ก้อน
10. บอร์ด Arduino Uno R3	1 วงจร

3.2.1.2 โครงสร้างของหุ่นยนต์

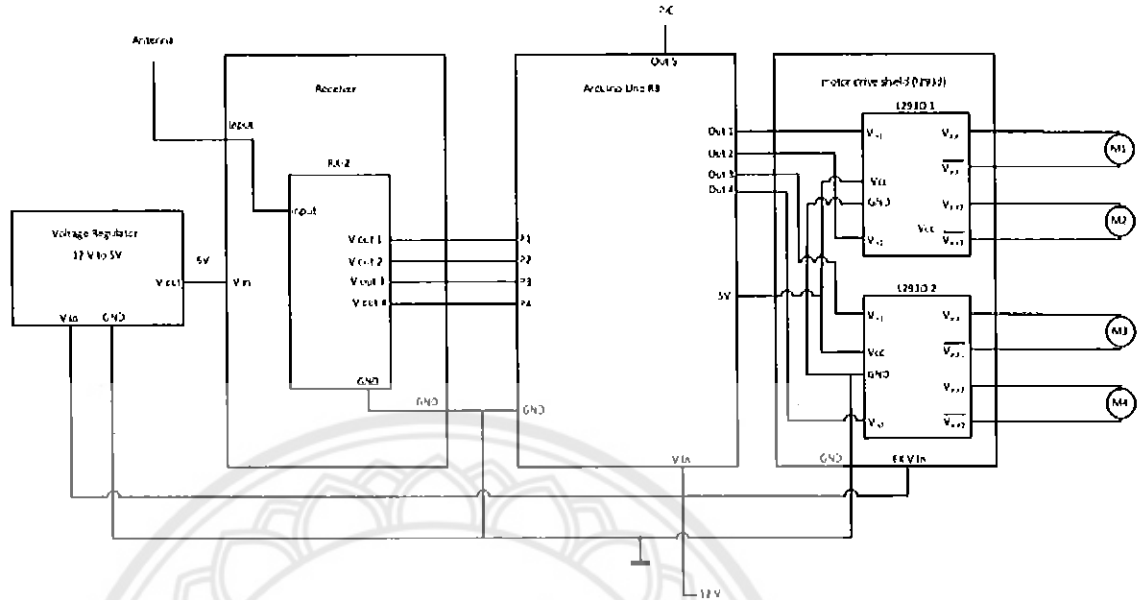
ในการประกอบหุ่นยนต์นั้น หุ่นยนต์ที่ทำการสร้างขึ้นเพื่อใช้ในการเคลื่อนที่ สิ่งสำคัญคือส่วนที่ทำให้หุ่นยนต์มีการเปลี่ยนตำแหน่ง โดยโครงสร้างพื้นฐานของหุ่นยนต์แบ่งออกเป็น 5 ส่วนดังนี้

1. เครื่องส่งควบคุมด้วยมือมีหน้าที่ส่งคลื่นวิทยุไปที่เครื่องรับ
2. เครื่องรับ ประกอบด้วยเสาอากาศ และแผงวงจรไฟฟ้า เพื่อคอยรับสัญญาณส่งจากเครื่องส่ง และกระตุ้นให้มอเตอร์ภายในเครื่องเล่นทำงาน
3. บอร์ด ไมโครคอนโทรลเลอร์ทำหน้าที่รับคำสั่งจากเครื่องรับ เพื่อควบคุมการเปลี่ยนแปลงของมอเตอร์
4. มอเตอร์ทำหน้าที่หมุนล้อและเลี้ยวสำหรับหุ่นยนต์
5. แหล่งจ่ายพลังงานคือแบตเตอรี่ หรือถ่านไฟฉาย

โดยทำการสร้างหุ่นยนต์ตามสัดส่วนดังรูปที่ 3.2



รูปที่ 3.2 หุ่นยนต์เคลื่อนที่



รูปที่ 3.3 วงจรหุ่นยนต์

จากรูปที่ 3.3 เป็นการต่อวงจรที่ใช้ควบคุมหุ่นยนต์จะแบบออกเป็น 4 ส่วนด้วยกัน 1.ตัวรับสัญญาณ 2.บอร์ด Arduino 3.วงจรควบคุมมอเตอร์ และ 4.แหล่งจ่ายไฟ

1. ตัวรับสัญญาณ เนื่องจากการใช้วงจรถ่ายของรถบังคับวิทยุมาใช้งาน โดยทำการต่อเอาต์พุตจาก IC Rx-2 ที่ทำหน้าที่ Decoder สัญญาณที่รับเข้ามา กับบอร์ด Arduino

2. บอร์ด Arduino นั้นทำหน้าที่ควบคุมการทำงานซึ่งมี 2 ส่วน

1. ส่งสัญญาณให้แก่บอร์ด PIC

2. ควบคุมการทำงานของวงจรควบคุมมอเตอร์

จึงมีการต่อเอาต์พุตไปยังบอร์ด PIC ส่วนวงจรถควบคุมนั้นเนื่องจากเป็น Shield จึงต่อโดยการสวมลงบนบอร์ด Arduino ได้

3. วงจรควบคุมมอเตอร์นั้น เนื่องจากใช้ควบคุมมอเตอร์จึงทำมอเตอร์มาต่อทั้ง 4 ตัวดังจากรูปที่

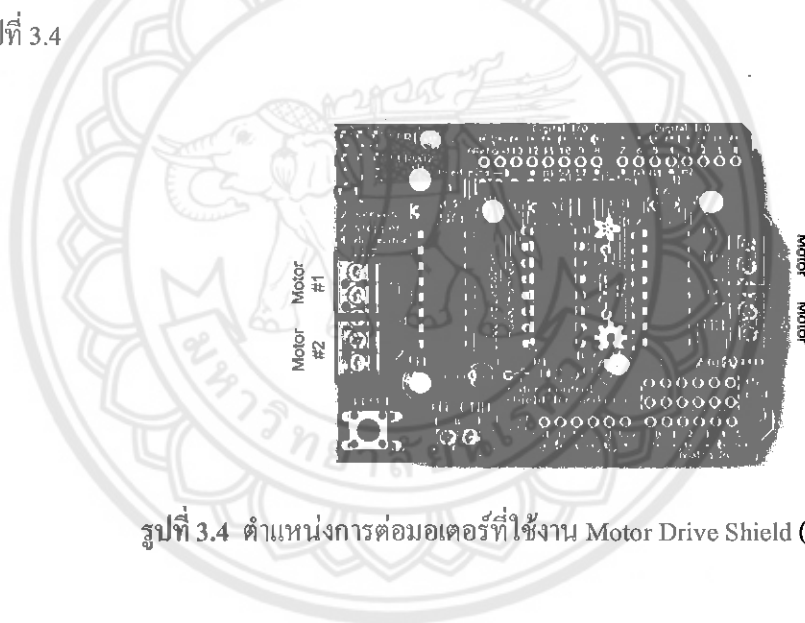
4. แหล่งจ่ายไฟนั้นใช้แบตเตอรี่ ขนาด12V ต่อเข้ากับ แหล่งจ่ายไฟภายนอกของวงจรควบคุมมอเตอร์ เป็นแหล่ง Power ของบอร์ด Arduinoและวงจรแปลงความต่างศักย์ที่แปลงให้กับตัวรับสัญญาณ

3.2.2 การใช้วงจรควบคุมมอเตอร์ Motor Drive Shield (L293d)

เนื่องจากวงจรควบคุมมอเตอร์มีหลักการทำงาน โดยใช้สัญญาณจากบอร์ด Arduino สามารถควบคุมการทำงานด้วยการใช้ LibraryAFMotor.h มีหลักการใช้งานแบ่งได้ 2 ส่วน คือ

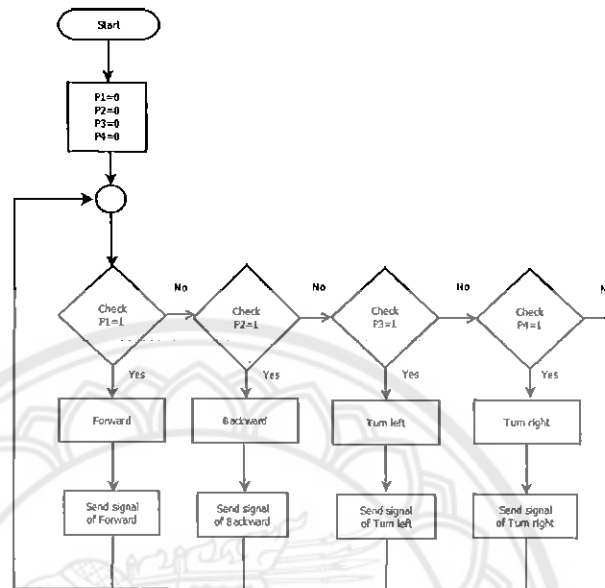
- 1.ควบคุมความเร็ว สามารถทำได้โดยการใช้คำสั่ง “setSpeed”
- 2.ควบคุมทิศทางของมอเตอร์ สามารถทำได้โดยการใช้คำสั่ง “RUN”

ในการต่อเชื่อมกับมอเตอร์ Motor Drive Shield (L293d) สามารถต่อกับมอเตอร์ได้สูงสุด4ตัว ดังรูปที่ 3.4



รูปที่ 3.4 ตำแหน่งการต่อมอเตอร์ที่ใช้งาน Motor Drive Shield (L293d)

3.2.3 แสดง Pseudo code ในการใช้วงจรควบคุมมอเตอร์



รูปที่ 3.5 แสดงการทำงานของโปรแกรมควบคุมมอเตอร์

จากรูปที่ 3.5 เป็นการแสดงลักษณะการทำงานของโปรแกรมในการควบคุมมอเตอร์ เมื่อเริ่มต้นจะมีการรอสัญญาณจากวงจรภาครับ โดยเมื่อมีสัญญาณเข้ามาจะทำการตรวจสอบสัญญาณของ P1 หากมีสัญญาณ P1 เข้ามาจะมีการส่งสัญญาณให้รถเดินหน้า แล้วกลับไปยังค่าเริ่มต้นในการรอสัญญาณใหม่เข้า แต่หาไม่มีสัญญาณเข้ามา โปรแกรมจะทำการรอสัญญาณต่อไป

ตัวอย่าง Pseudo code

```

01: void loop() {
02: if (P1 == HIGH) { //ตรวจสอบสัญญาณ P1
03: Move_Forward(); //เมื่อมีสัญญาณ P1 เข้ามา จะทำการเรียกฟังก์ชัน
Move_Forward() เพื่อสั่งให้รถเดินหน้า
04: }
05: else if (P2 == HIGH) { //ตรวจสอบสัญญาณ P2
06: Move_Backward(); //เมื่อมีสัญญาณ P2 เข้ามา จะทำการเรียกฟังก์ชัน
Move_Backward() เพื่อสั่งให้รถถอยหลัง
07: }
  
```

```

08: else if (P3 == HIGH) { //ตรวจสอบสัญญาณ P3
09: Move_Left(); //เมื่อมีสัญญาณP3เข้ามา จะทำการเรียกฟังก์ชัน
Move_Left() เพื่อสั่งให้รถเลี้ยวซ้าย
10: }
11: else if (P4 ==HIGH) { //ตรวจสอบสัญญาณ P4
12:Move_Right(); //เมื่อมีสัญญาณP4เข้ามา จะทำการเรียกฟังก์ชัน
Move_Right() เพื่อสั่งให้รถเลี้ยวขวา
13: }

```

ในฟังก์ชัน Move ทั้งหมดในข้างต้นจะเป็นการสั่งให้มอเตอร์ทำงานทิศทางที่แตกต่างกัน
เช่น

Move_Forward() จะเป็นการสั่งให้มอเตอร์หมุน ไปทิศข้างหน้าทั้ง 4 ตัว
Move_Backward()จะเป็นการสั่งให้มอเตอร์หมุน ไปทิศข้างหลังทั้ง 4 ตัว
Move_Left() จะเป็นการสั่งให้มอเตอร์ที่อยู่ด้านซ้ายทั้ง 2 ตัวหมุนไปทิศข้างหลัง ส่วน
ด้านขวาหมุนไปข้างหน้าทั้ง 2 ตัว
Move_Right() จะเป็นการสั่งให้มอเตอร์ที่อยู่ด้านขวาทั้ง 2 ตัวหมุนไปทิศข้างหลัง
ส่วนด้านซ้ายหมุนไปข้างหน้าทั้ง 2 ตัว

3.2.4 แสดง Pseudo code ในการส่งสัญญาณให้บอร์ด PICจากบอร์ด Arduino

ในส่วนนี้เป็นส่วนที่ใช้ในการส่งสัญญาณระหว่างบอร์ดเพื่อให้มีการรับรู้สถานะของแต่ละบอร์ดมีการเปลี่ยนแปลงอย่างไรบ้าง โดยเมื่อมีการสั่งให้เดินหน้าหรือถอยหลัง จะมีการส่งสัญญาณ LOW หากเป็นการเลี้ยวจะเป็นการส่งสัญญาณ HIGH

```

01: check(){
02: if((turn left)||( turn right)){ //ตรวจสอบการเลี้ยวของหุ่นยนต์
03: digitalWrite(pic, HIGH); //กำหนดสัญญาณเพื่อสั่งให้บอร์ดPIC เป็น HIGH
04: }
05: else{
06: digitalWrite(pic, LOW);} //หากไม่มีการเลี้ยวให้ส่งสัญญาณ LOW
07: }

```

3.3 การเขียนโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์เพื่อใช้ในการอ่านค่าจากเซ็นเซอร์ที่ใช้สำหรับวัดระยะทางและทิศทาง

3.3.1 การใช้โมดูลTimer ของ PIC 16F877 ในการจับเวลา

หลักการในการประมวลผลสัญญาณจากเซ็นเซอร์ SRF05 นั้นมีหลักการคือต้องใช้ Timer เป็นตัวจับเวลาหาคาบเวลาของสัญญาณพัลส์ที่มี logic เป็น “1” ผู้ดำเนินโครงการกล่าวถึงวิธีการใช้ Timer และการเอาค่าที่ได้จาก Timer มาแทนค่าลงในสมการเพื่อคำนวณค่าระยะทางของเซ็นเซอร์

Timer เป็นวงจรที่มีอยู่ในชิพ PIC จะทำการนับสัญญาณพัลส์ ซึ่งเป็นสัญญาณที่ได้จากสัญญาณนาฬิกาจากเซ็นเซอร์ SRF05 ที่ป้อนให้กับ PIC โดยในที่นี่จะเลือกใช้ Timer1 เป็นตัววัดคาบซึ่ง Timer1 เป็น Timer ขนาด 2 byte ประกอบด้วยรีจิสเตอร์ TMR1H ขนาด 1 byte และ TMR1L ขนาด 1 byte ซึ่งรูปที่ 3.6 แสดงถึงรายละเอียดของรีจิสเตอร์ TICON [51] และตารางที่ 3.1 แสดงรายละเอียดของรีจิสเตอร์ TICON

000 00	000 00	000 00	000 00	000 00	000 00	000 00	000 00
T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR1ON
0	0	0	0	0	0	0	0

รูปที่ 3.6 แสดงรายละเอียดของรีจิสเตอร์ TICON

ตารางที่ 3.1 รายละเอียดของรีจิสเตอร์ TICON

บิต	ชื่อ	คำอธิบาย
7	T1GINV	ให้ค่าเป็น '0'
6	TMRIGE	ให้ค่าเป็น '0'
5	TICKPS1	ใช้เลือกอัตราส่วนในการหารความถี่ของ Prescaler
4	TICKPS0	ใช้เลือกอัตราส่วนในการหารความถี่ของ Prescaler
3	TIOSCEN	ใช้เปิด/ปิดการทำงานของออสซิลเลเตอร์ของ Timer1 <ul style="list-style-type: none"> ● 1 = ปิดการใช้งานออสซิลเลเตอร์ของ Timer1 ● 0 = เปิดการใช้งานออสซิลเลเตอร์ของ Timer1
บิต	ชื่อ	คำอธิบาย
2	T1SYNC	ใช้เปิด/ปิดการ Synchronize ของสัญญาณพัลส์อินพุตจากภายนอก <ul style="list-style-type: none"> ● 1 = ปิดการ Synchronize ● 0 = เปิดการ Synchronize
3	TMR1CS	ใช้เลือกแหล่งจ่ายสัญญาณพัลส์ให้กับ Timer 1 <ul style="list-style-type: none"> ● 1 = ใช้เลือกแหล่งจ่ายสัญญาณพัลส์อินพุตของขาขึ้นจากภายนอกผ่านขา TICKI/TIOSO ● 0 = ใช้เลือกแหล่งจ่ายสัญญาณพัลส์อินพุตจากคริสตอล
0	TMR1ON	ใช้เปิด/ปิดการทำงานของ Timer 1 <ul style="list-style-type: none"> ● 1 = เปิดการทำงานของ Timer 1 ให้ Timer 1 เริ่มนับ ● 0 = ปิดการทำงานของ Timer 1 ให้ Timer 1 หยุดนับ

3.3.2 หลักการประมวลผลสัญญาณที่ได้จากอัลตราโซนิกเซ็นเซอร์SRF05

จากการทำงานของอัลตราโซนิกเซ็นเซอร์ คือ จะทำการส่งสัญญาณอัลตราโซนิกออกไป แล้ววัดระยะเวลาที่มีสัญญาณสะท้อนตอบกลับมา เอาต์พุตที่ได้จาก SRF05 จะอยู่ในรูปของความกว้างพัลส์ซึ่งมีความสัมพันธ์กับระยะทางของวัตถุที่ตรวจจับได้ ความถี่สัญญาณอัลตราโซนิกของ SRF05 คือ 40 kHz ถูกส่งออกไปในอากาศด้วยความเร็ว 1.125 ฟุตต่อมิลลิวินาที (ประมาณ 346 เมตรต่อวินาที) ดังนั้นเมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น เวลาเริ่มต้นส่งคลื่น และเวลาที่รับเสียงสะท้อนกลับมา จึงสามารถคำนวณหาระยะทางได้ดังนี้

$$S = Vt$$

โดยที่

S	คือระยะทางที่ต้องการทราบค่า (เมตร)
V	คือ ความเร็วในการเคลื่อนที่ (เมตรต่อวินาที)
t	คือ ระยะเวลาจากการสะท้อนของเสียงหารสอง (วินาที)

3.3.3 หลักการประมวลผลสัญญาณที่ได้จากโมดูลเข็มทิศCMPS10

จากเลขที่ได้กล่าวไว้ในหัวข้อ 2.9.3 นั้น การอ่านค่าทิศทางจากเอาต์พุตของสัญญาณพัลส์ นั้น เป็นการนำค่าความกว้างของสัญญาณพัลส์ที่ได้มาระบุตำแหน่งองศาจาก 0 – 3599 เมื่อทำการหารด้วยสิบ จึงได้ผลลัพธ์ออกมาเป็น 0 – 359.9 องศา

3.3.4 Pseudo code ของการควบคุมไมโครคอนโทรลเลอร์

ในส่วนของหัวข้อนี้จะขอแยกออกเป็น 3 ส่วนคือ

3.3.4.1 Pseudo code ในการอ่านค่าสัญญาณพัลส์ของ logic = I จากอัลตราโซนิกเซ็นเซอร์ SRF05

3.3.4.2 Pseudo code ในการอ่านค่า 0-3599 จาก โมดูลเข็มทิศดิจิทัลCMPS10

3.3.4.3 Pseudo code ในการรับค่าจากบอร์ด Arduinoและการส่งค่าไปยังโปรแกรมสร้างแผนที่

3.3.4.1 Pseudo code ในการอ่านค่าสัญญาณพัลส์ของ logic = 1 จากอัลตราโซนิก

เซ็นเซอร์SRF05

ในการทำงานของ Code ส่วนนี้จะแบ่งเป็น 2 ฟังก์ชัน ได้แก่ TRIGGER(), ECHO() ตามลำดับ โดยในฟังก์ชันแรกจะเริ่มต้นด้วยการกำหนดให้พอร์ต B ของบอร์ด PIC 16F877 เป็นเอาต์พุต และส่งสัญญาณพัลส์ขนาด 10 ไมโครวินาที ไปยังอัลตราโซนิกเซ็นเซอร์ เพื่อร้องขอระยะทาง(กระตุ้นการทำงาน) และขั้นต่อไป ฟังก์ชัน ECHO() ก็จะเริ่มทำงานโดยเริ่มกำหนดให้พอร์ต B เปลี่ยนหน้าที่เป็นอินพุต เพื่อรับสัญญาณพัลส์ที่ได้จากตัวเซ็นเซอร์ โดยจะได้สัญญาณพัลส์ขนาด 2 ไบต์

แสดงPseudo code ในการส่งค่า Trigger ให้กับอัลตราโซนิกเซ็นเซอร์SRF05

01 : TRIGGER()
 02 : กำหนดให้ Timer 1 มีค่าเท่ากับ 0
 03 : กำหนดให้พอร์ต B บิต 0 เป็นเอาต์พุต
 04 : เปิดการส่งสัญญาณไปยังเซ็นเซอร์ SRF05
 05 : หน่วงเวลา 10 ไมโครวินาที
 06 : ปิดการส่งสัญญาณไปยังเซ็นเซอร์ SRF05

แสดงPseudo code ในการรับค่า ECHO จากอัลตราโซนิกเซ็นเซอร์SRF05

01 : ECHO()
 02 : กำหนดให้พอร์ต B บิต 0 เป็นอินพุต
 03 : วนลูปค่าพอร์ต B บิต 0 เท่ากับ 0 รอจนกว่าจะมีค่าเป็น 1
 04 : สั่งให้ Timer 1 เริ่มนับ
 05 : วนลูปค่าพอร์ต B บิต 0 เท่ากับ 1 รอจนกว่าจะมีค่าเป็น 0
 06 : สั่งให้ Timer 1 หยุดนับ
 07 : กำหนดให้ ULH เก็บค่าใน Timer 1 ไบต์ High
 08 : กำหนดให้ ULL เก็บค่าใน Timer 1 ไบต์ Low

3.3.4.2 Pseudo code ในการอ่านค่า 0-3599 จากโมดูล CMPS10

ในส่วนของการอ่านค่าจากโมดูล CMPS10 นั้นจะใช้หลักการอ้างอิงแอดเดรสของตัวโมดูล CMPS10 ในขั้นแรกนั้นจะต้องทำการกำหนดพอร์ต SCL (serial clock), SDA (serial data) ก่อน โดยในที่นี้จะกำหนดให้เป็นพอร์ต RC3 และ RC4 ของบอร์ด PIC 16F877 ตามลำดับ

- 01 : กำหนดให้มีการใช้ SCL ในโหมด I²C
- 02 : กำหนดให้มีการใช้ SDA ในโหมด I²C
- 03 : กำหนดให้พอร์ต C บิต 3 เป็นขาสัญญาณ SCL
- 04 : กำหนดให้พอร์ต C บิต 4 เป็นขาสัญญาณ SDA
- 05 : PC_CMPS10()
- 06 : เปิดการใช้งานโหมด I²C
- 07 : ส่งค่าแอดเดรสของโมดูล CMPS10
- 08 : ส่งค่ารีจิสเตอร์ 2 ไปยังโมดูล CMPS10
- 09 : เริ่มทำงาน
- 10 : อ่านค่าจากโมดูล CMPS10
- 11 : กำหนดให้ CMPH เก็บค่าในไบต์ High
- 12 : กำหนดให้ CMPL เก็บค่าในไบต์ Low
- 13 : หยุดทำงาน

3.3.4.3 Pseudo code ในการรับค่าจากบอร์ด Arduino Uno R3 และการส่งค่าไปยังตัว

โปรแกรมสร้างแผนที่

ในการทำงานทั้ง 2 ส่วนนี้จะอยู่ในฟังก์ชันเดียวกัน โดยจะมีขั้นตอนของการส่งค่าของเซ็นเซอร์ SRF05 และโมดูล CMP10 ไปยังโปรแกรมสร้างแผนที่ และการรับค่าจากบอร์ด Arduino Uno R3 โดยจะกำหนดให้พอร์ต B บิต 1 ของบอร์ด PIC 16F877 รับสัญญาณจากบอร์ด Arduino Uno R3 เมื่อหุ่นยนต์เคลื่อนที่หมุนจะมีสัญญาณเข้ามา บอร์ด PIC 16F77 ก็จะทำการส่งค่า 1 ไปยังตัวโปรแกรมสร้างแผนที่ แต่ถ้าไม่มีการส่งสัญญาณเข้ามาก็จะทำการค่า 0 ไปยังโปรแกรมสร้างแผนที่ซึ่งจะส่งเป็นไบต์สุดท้าย

- 01 : SEND()
 02 : ส่งตัวอักษร CMPH
 03 : ส่งค่าที่อยู่ในตัวแปร CMPH
 04 : ส่งตัวอักษร CMPL
 05 : ส่งค่าที่อยู่ในตัวแปร CMPL
 06 : ส่งตัวอักษร ULH
 07 : ส่งค่าที่อยู่ในตัวแปร ULH
 08 : ส่งตัวอักษร ULL
 09 : ส่งค่าที่อยู่ในตัวแปร ULL
 10 : ถ้าพอร์ต B บิต 1 มีค่าเท่ากับ 1
 11 : ส่งค่า 1
 13: ถ้าพอร์ต B บิต 1 มีค่าไม่เท่ากับ 1
 14: ส่งค่า 0
 15: ขึ้นบรรทัดใหม่เพื่อเริ่มส่งข้อมูลชุดต่อไป

3.4 การเขียนโปรแกรมสร้างหน้าจอ GUI สำหรับแสดงผล

3.4.1 โพรโตคอลในการรับส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์กับตัว

โปรแกรมสร้างแผนที่

จากหัวข้อที่ 3.3.4 ได้นำเสนอในเรื่องการส่งค่ามายัง โปรแกรมสร้างแผนที่ในหัวข้อนี้จะเป็นการแสดงโปรโตคอลในการรับส่งค่าระหว่างบอร์ดไมโครคอนโทรลเลอร์กับตัวโปรแกรมดังนี้

'CMPH'	Angle_Z_H	'CMPL'	Angle_Z_L	'ULH'	Distance_H	'ULL'	Distance_L	Ctrl
(4 byte)	(6 byte)	(4 byte)	(6 byte)	(4 byte)	(6 byte)	(4 byte)	(6 byte)	(1 byte)

สำหรับรายละเอียดของโปรโตคอลเป็นดังนี้

- 'CMPH' มีขนาด 4 ไบต์ ใช้แสดงถึงค่าที่ได้ต่อจากข้อความนี้คือ ค่าที่วัดได้จาก CMPS10 ในไบต์ High
- Angle_Z_H มีขนาด 6 ไบต์ คือค่าที่วัดได้จาก CMPS10 ในไบต์ High

- 'CMPL' มีขนาด 4 ไบต์ ใช้แสดงถึงค่าที่ได้ต่อจากข้อความนี้คือ ค่าที่วัดได้จาก CMPS10 ในไบต์ Low
- Angle_Z_L มีขนาด 6 ไบต์ คือค่าที่วัดได้จาก CMPS10 ในไบต์ Low
- 'ULH' มีขนาด 4 ไบต์ ใช้แสดงถึงค่าที่ได้ต่อจากข้อความนี้คือ ค่าที่วัดได้จาก SRF05 ในไบต์ High
- Distance_H มีขนาด 6 ไบต์ คือค่าที่วัดได้จาก SRF05 ในไบต์ High
- 'ULL' มีขนาด 4 ไบต์ ใช้แสดงถึงค่าที่ได้ต่อจากข้อความนี้คือ ค่าที่วัดได้จาก SRF05 ในไบต์ Low
- Distance_L มีขนาด 6 ไบต์ คือค่าที่วัดได้จาก SRF05 ในไบต์ Low
- Ctrl มีขนาด 1 ไบต์ คือค่าสำหรับควบคุมการเปิดการทำงานของ SRF05 และ CMPS10

การรับข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์มายังโปรแกรมนั้นจำเป็นต้องกำหนด Baud Rate [52] ให้เท่ากันซึ่งคณะผู้ดำเนินโครงการได้เลือก Baud Rate เท่ากับ 9600 เพื่อใช้ในการส่งข้อมูลมายังตัวโปรแกรม ส่วนในการนำข้อมูลมาประมวลผลนั้นทางคณะผู้ดำเนินโครงการจะอธิบายในหัวข้อที่ 3.4.3.3

ตัวอย่างการส่งข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ไปยังตัวโปรแกรม ดังแสดงในรูปที่ 3.7

```
CMPL000013CMPL000091ULH000025ULL0002430
CMPL000013CMPL000091ULH000025ULL0002430
CMPL000013CMPL000091ULH000025ULL0002430
CMPL000013CMPL000091ULH000025ULL0002431
CMPL000013CMPL000091ULH000025ULL0002431
CMPL000013CMPL000091ULH000025ULL0002471
CMPL000013CMPL000091ULH000025ULL0002431
```

รูปที่ 3.7 โปรโตคอลสำหรับส่งข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ไปยังตัวโปรแกรมสร้างแผนที่
หมายเหตุ สาเหตุที่ส่งเป็น byte เนื่องจากส่งข้อมูลเป็นชุดของ String เพื่อความสะดวกในการโปรแกรม

3.4.2 การวาดรูปภาพแสดงผล

การวาดรูปภาพนั้นทางคณะผู้ดำเนินโครงการได้เลือกให้วาดรูปภาพบน Picture Box ขนาด 600 x 600 เพื่อตอบสนองการทำงานในส่วนของฮาร์ดแวร์และซอฟต์แวร์ ซึ่งจะทำให้

ประสิทธิภาพในการวาดรูปภาพรวดเร็วยิ่งขึ้น โดยจะประกอบด้วยฟังก์ชันที่ใช้ในการวาดรูปภาพอยู่ 6 ฟังก์ชัน ดังนี้

3.4.2.1 ฟังก์ชันวาดรูปภาพหุ่นยนต์เคลื่อนที่

3.4.2.2 ฟังก์ชันวาดรูปภาพเซ็นเซอร์

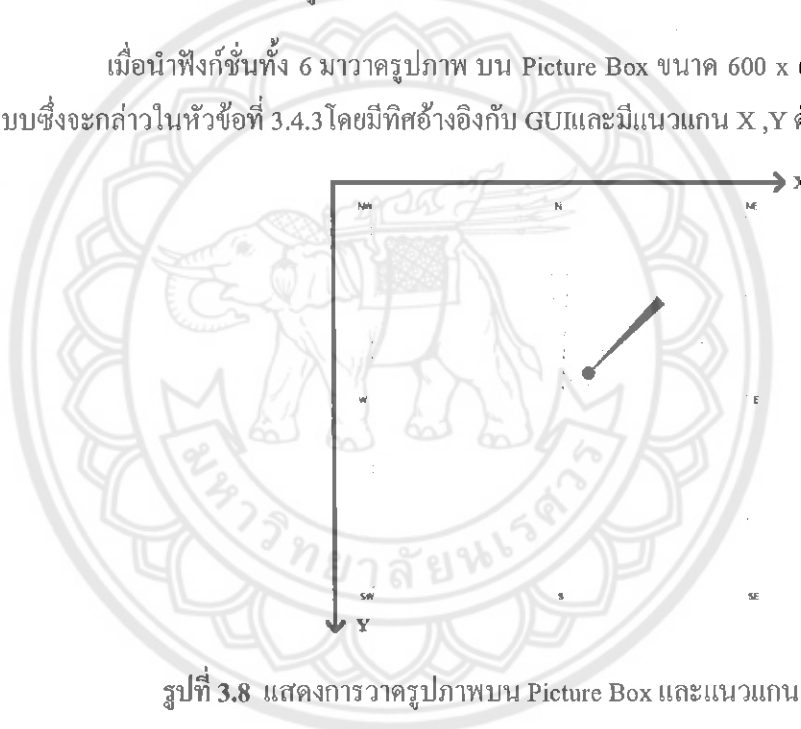
3.4.2.3 ฟังก์ชันวาดรูปภาพลำแสงของเซ็นเซอร์

3.4.2.4 ฟังก์ชันวาดรูปภาพแผนที่ของหุ่นยนต์เคลื่อนที่

3.4.2.5 ฟังก์ชันวาดรูปภาพแผนที่ของเซ็นเซอร์

3.4.2.6 ฟังก์ชันวาดรูปภาพแผนที่ของลำแสงเซ็นเซอร์

เมื่อนำฟังก์ชันทั้ง 6 มาวาดรูปภาพ บน Picture Box ขนาด 600 x 600 ผ่านอัลกอริทึมของระบบซึ่งจะกล่าวในหัวข้อที่ 3.4.3 โดยมีทิศอ้างอิงกับ GUI และมีแนวแกน X, Y ดังรูปที่ 3.8



รูปที่ 3.8 แสดงการวาดรูปภาพบน Picture Box และแนวแกน X, Y

3.4.2.1 ฟังก์ชันวาดรูปภาพหุ่นยนต์เคลื่อนที่

ในการวาดรูปภาพหุ่นยนต์เคลื่อนที่นั้นจะใช้ฟังก์ชัน Car เพื่อใช้ในการวาดรูปภาพหุ่นยนต์เคลื่อนที่ โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่ 2 ค่าคือค่าพิกัดในแนวแกน X และ Y เมื่อรับค่ามาแล้ว จึงทำการวาดรูปภาพหุ่นยนต์เคลื่อนที่ซึ่งจะแทนด้วยวงกลม ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชัน Car รับค่า 2 ค่า
 - ค่าที่ 1 พิกัดของหุ่นยนต์เคลื่อนที่ในแนวแกน X
 - ค่าที่ 2 พิกัดของหุ่นยนต์เคลื่อนที่ในแนวแกน Y
- วาดวงกลมสีเขียวรัศมี 10 โดยมีตำแหน่งวงกลมอยู่ที่พิกัด X และ Y

3.4.2.2 ฟังก์ชันวาดรูปภาพเซ็นเซอร์

ในการวาดรูปภาพเซ็นเซอร์นั้นจะใช้ฟังก์ชัน Sensor เพื่อใช้ในการวาดรูปภาพเซ็นเซอร์ โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่ 4 ค่าคือค่าพิกัดเริ่มต้นของเซ็นเซอร์ในแนวแกน X และ Y และพิกัดสิ้นสุดของเซ็นเซอร์ในแนวแกน X และ Y เมื่อรับค่ามาแล้วจึงทำการวาดรูปภาพเซ็นเซอร์ซึ่งจะแทนด้วยเส้นตรง ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชัน Sensor รับค่า 4 ค่า
 - ค่าที่ 1 พิกัดเริ่มต้นของเซ็นเซอร์ในแนวแกน X
 - ค่าที่ 2 พิกัดเริ่มต้นของเซ็นเซอร์ในแนวแกน Y
 - ค่าที่ 3 พิกัดสิ้นสุดของเซ็นเซอร์ในแนวแกน X
 - ค่าที่ 4 พิกัดสิ้นสุดของเซ็นเซอร์ในแนวแกน Y
- วาดเส้นตรงสีแดง โดยมีพิกัดเริ่มต้นและพิกัดสิ้นสุดในแนวแกน X และ Y

3.4.2.3 ฟังก์ชันวาดลำแสงของเซ็นเซอร์

ในการวาดรูปภาพลำแสงของเซ็นเซอร์นั้นจะใช้ฟังก์ชัน Sensor_Beam เพื่อใช้ในการวาดรูปภาพลำแสงของเซ็นเซอร์ โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่ 4 ค่าคือ ArrayList [53] พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน X และ Y และ ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน X และ Y เมื่อรับค่ามาแล้วจึงทำการวาดรูปภาพลำแสงของเซ็นเซอร์ซึ่งจะแทนด้วยเส้นตรงหลายเส้น ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชัน Sensor_Beam รับค่า 4 ค่า
 - ค่าที่ 1 ArrayList พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน X
 - ค่าที่ 2 ArrayList พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน Y
 - ค่าที่ 3 ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน X

ค่าที่ 4 ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน Y

- วนลูป

วาดเส้นตรงสีแดง โดยมีพิกัดเริ่มต้นและพิกัดสิ้นสุดในแนวแกน X และ Y ของArrayListแต่ละรอบการทำงาน

3.4.2.4 ฟังก์ชันวาดแผนที่ของหุ่นยนต์เคลื่อนที่

ในการวาดแผนที่ของหุ่นยนต์เคลื่อนที่นั้นจะใช้ฟังก์ชัน CreateMap_Car เพื่อใช้ในการวาดแผนที่ของหุ่นยนต์เคลื่อนที่ โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่ 2 ค่าคือArrayListพิกัดในแนวแกน X และ Y เมื่อรับค่ามาแล้วจึงทำการวาดแผนที่ของหุ่นยนต์เคลื่อนที่ซึ่งจะแทนด้วยวงกลมหลายวง ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชันCreateMap_Car รับค่า 2 ค่า

ค่าที่1 ArrayListพิกัดของหุ่นยนต์เคลื่อนที่ในแนวแกน X

ค่าที่ 2 ArrayListพิกัดของหุ่นยนต์เคลื่อนที่ในแนวแกน Y

- วนลูป

วาดวงกลมสีขาวยาวรัศมี 10 โดยมีพิกัดในแนวแกน X และ Y ของArrayListแต่ละรอบการทำงาน

3.4.2.5 ฟังก์ชันวาดแผนที่ของเซ็นเซอร์

ในการวาดแผนที่ของเซ็นเซอร์นั้นจะใช้ฟังก์ชัน CreateMap_Sensor เพื่อใช้ในการวาดแผนที่ของเซ็นเซอร์ โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่3 ค่าคือArrayListพิกัดเริ่มต้นและสิ้นสุดของเซ็นเซอร์ในแนวแกน Xและ ArrayListพิกัดเริ่มต้นและสิ้นสุดของเซ็นเซอร์ในแนวแกน Y และ Index หรือรอบการทำงานของโปรแกรม เมื่อรับค่ามาแล้วจึงทำการวาดแผนที่ของเซ็นเซอร์ซึ่งจะแทนด้วยเส้นตรงหลายเส้น ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชันCreateMap_Sensor รับค่า 3 ค่า

ค่าที่1 ArrayListพิกัดเริ่มต้นและสิ้นสุดของเซ็นเซอร์ในแนวแกน X

ค่าที่ 2 ArrayListพิกัดเริ่มต้นและสิ้นสุดของเซ็นเซอร์ในแนวแกน Y

ค่าที่ 3Index หรือรอบการทำงานของ โปรแกรม

- วนลูป

วาดเส้นตรงสีขาว โดยมีพิกัดเริ่มต้นและพิกัดสิ้นสุดในแนวแกน X และ Y ของ ArrayList แต่ละรอบการทำงาน

3.4.2.6 ฟังก์ชันวาดแผนที่ของลำแสงเซ็นเซอร์

ในการวาดแผนที่ของลำแสงเซ็นเซอร์นั้นจะใช้ฟังก์ชัน CreateMap_Beam เพื่อใช้ในการวาดแผนที่ของลำแสงเซ็นเซอร์โดยฟังก์ชันนี้จะรับค่าหลักๆอยู่ 5 ค่าคือ ArrayList พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน X, Y และ ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน X, Y และ Index หรือรอบการทำงานของโปรแกรม เมื่อรับค่ามาแล้วจึงทำการวาดแผนที่ของลำแสงเซ็นเซอร์ซึ่งจะแทนด้วยเส้นตรงหลายเส้น ทางคณะผู้ดำเนินโครงการได้อธิบายเป็น Pseudo code ดังนี้

- ฟังก์ชัน CreateMap_Beam รับค่า 5 ค่า
 - ค่าที่ 1 ArrayList พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน X
 - ค่าที่ 2 ArrayList พิกัดเริ่มต้นของลำแสงเซ็นเซอร์ในแนวแกน Y
 - ค่าที่ 3 ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน X
 - ค่าที่ 4 ArrayList พิกัดสิ้นสุดของลำแสงเซ็นเซอร์ในแนวแกน Y
 - ค่าที่ 5 Index หรือรอบการทำงานของโปรแกรม
- วนรูป
 - วาดเส้นตรงสีขาว โดยมีพิกัดเริ่มต้นและพิกัดสิ้นสุดในแนวแกน X และ Y ของ ArrayList แต่ละรอบการทำงาน

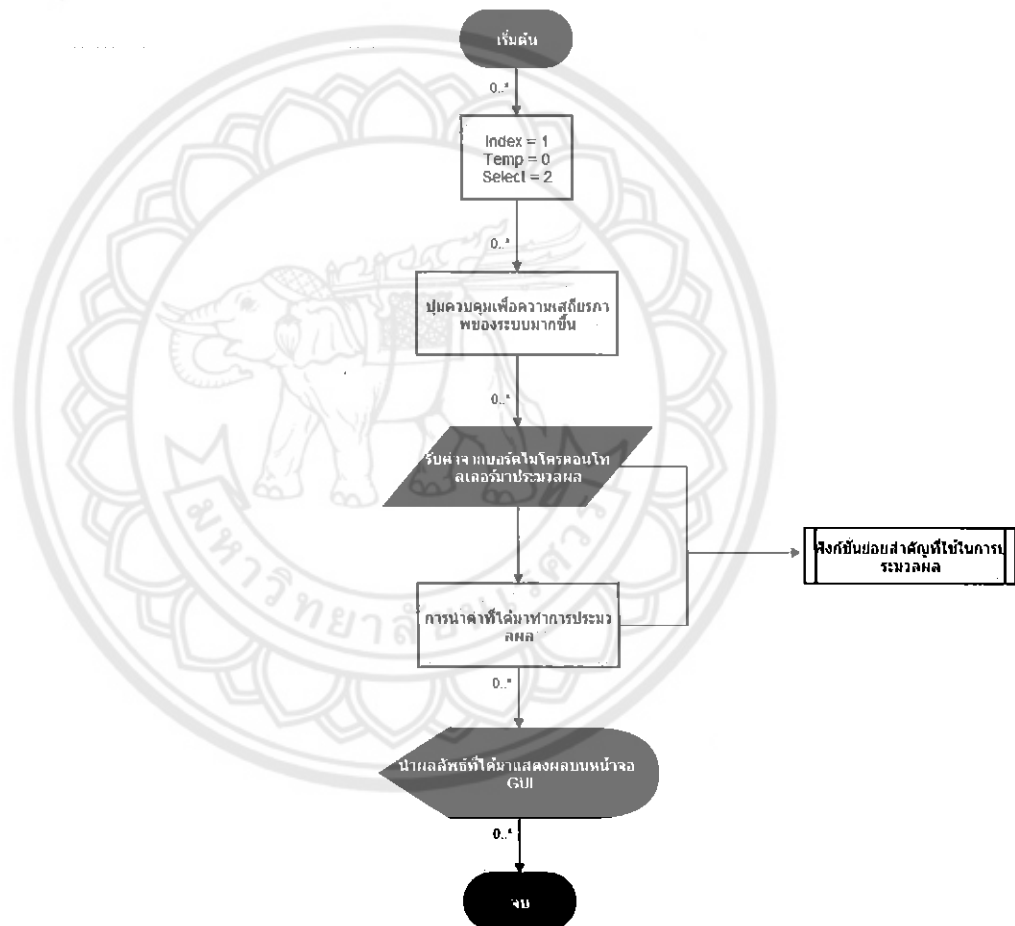
3.4.3 Pseudo code ของการเขียนโปรแกรมหน้าจอ GUI สำหรับแสดงผล

การทำงานในส่วนของโปรแกรมหน้าจอ GUI สำหรับแสดงผลนั้นจำเป็นจะต้องมีการกำหนดตัวแปรสำคัญในการทำงานของโปรแกรมซึ่งมีอยู่ 3 ตัวแปรสำคัญดังนี้

- Index รอบการทำงานของโปรแกรมซึ่งจะถูกเพิ่มค่าทีละ 1 ในแต่ละรอบการทำงานโดยจะกำหนดให้ค่าเริ่มต้นของตัวแปร Index เท่ากับ 1
- Select เป็นตัวควบคุมอัลกอริทึมในการนำค่ามาประมวลผล ซึ่งตัวแปร Select จะมีค่าตั้งแต่ 0 ถึง 2 โดยจะกำหนดให้ค่าเริ่มต้นของตัวแปร Select เท่ากับ 2
- Temp เป็นตัวควบคุมของโปรแกรมในส่วนการรับค่าจากบอร์ดไมโครคอนโทรลเลอร์มาประมวลผล ซึ่งตัวแปร Temp จะมีค่าอยู่ 2 ค่าคือ 0 และ -1 โดยจะกำหนดให้ค่าเริ่มต้นของตัวแปร Temp เท่ากับ 0

เมื่อมีการกำหนดตัวแปรแล้วในหัวข้อนี้จะกล่าวแยกออกเป็น 5 ส่วนคือ

- 3.4.3.1 Pseudo code สำหรับฟังก์ชันย่อยสำคัญที่ใช้ในการประมวลผล
- 3.4.3.2 Pseudo code สำหรับบำรุงความคุมเพื่อความปลอดภัยของระบบ
- 3.4.3.3 Pseudo code สำหรับการรับค่าจากบอร์ดไมโครคอนโทรลเลอร์มาประมวลผล
- 3.4.3.4 Pseudo code สำหรับการนำค่าที่ได้มาทำการประมวลผล
- 3.4.3.5 Pseudo code สำหรับนำผลลัพธ์ที่ได้จากข้อที่ 3.4.3.4 มาแสดงผลบนหน้าจอ GUI โดยการโปรแกรมทั้ง 5 ส่วนนี้จะมีความสัมพันธ์เกี่ยวเนื่องกันดัง Flowchart [54] รูปที่ 3.9



รูปที่ 3.9 แสดง Flowchart การทำงานของ โปรแกรมหน้าจอ GUI สำหรับแสดงผล

3.4.3.1 Pseudo code สำหรับฟังก์ชันย่อยสำคัญที่ใช้ในการประมวลผล

ในระบบจำเป็นต้องมีฟังก์ชันย่อยเพื่อความสะดวกในการเขียนโปรแกรมซึ่งมีถึง 10 ฟังก์ชัน ในที่นี้คณะผู้ดำเนินโครงการจะขอยกเฉพาะฟังก์ชันย่อยสำคัญ 3 ฟังก์ชัน มากล่าวดังนี้

3.4.3.1.1 Pseudo code ฟังก์ชันคำนวณหาระยะทางจากอัลตราโซนิกเซ็นเซอร์

การหาระยะทางจากอัลตราโซนิกเซ็นเซอร์เมื่ออัลตราโซนิกเริ่มทำงานและเมื่อได้ค่า 2 ไบต์มาแล้วจึงทำการแปลงค่าโดยการกำหนดตัวหารที่เหมาะสมในแต่ละช่วงเพื่อให้ได้ค่า 0 ถึง 400 ที่นำมาใช้ประมวลผลกับโปรแกรมสร้างแผนที่ในหน่วยเซนติเมตร ทางคณะผู้ดำเนินโครงการจึงได้สร้างฟังก์ชัน Cal_length เพื่อหาระยะทางจากอัลตราโซนิกเซ็นเซอร์อธิบายเป็น Pseudo code ได้ดังนี้

- ฟังก์ชัน Cal_lengthรับค่า 1 ค่าผ่านตัวแปรtwo_bytes
กำหนดให้ตัวแปรtwo_bytesเท่ากับค่า 2 ไบต์ที่ได้รับจากอัลตราโซนิกเซ็นเซอร์
- ถ้าค่าของtwo_bytesอยู่ในช่วง 0 ถึง 6935
 $two_bytes = two_bytes / 154$
- ถ้าค่าของtwo_bytesอยู่ในช่วง 6935 ถึง 13506
 $two_bytes = two_bytes / 150$
- ถ้าค่าของ two_bytes อยู่ในช่วง 13506 ถึง 41730
 $two_bytes = two_bytes / 149$
- ถ้าค่าของ two_bytes อยู่ในช่วง 41730 ถึง 53272
 $two_bytes = two_bytes / 148$
- ถ้าค่าของ two_bytes อยู่ในช่วง 53272 ถึง 58808
 $two_bytes = two_bytes / 147$
- ถ้าไม่ใช่ทุกกรณี
 $two_bytes = -1$
- Return two_bytes

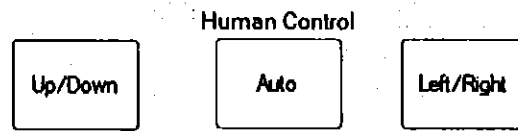
3.4.3.1.2 ฟังก์ชันคำนวณหาพิกัดในแนวแกน X และ Y

การหาพิกัดในแนวแกน X และ Y นั้นเป็นส่วนที่สำคัญมากกับโปรแกรมซึ่งทั้ง 2 ฟังก์ชันนี้สามารถนำมาคำนวณหาพิกัดได้หลากหลายเช่น พิกัดของเซ็นเซอร์ พิกัดของลำแสงเซ็นเซอร์ พิกัดของหุ่นยนต์เคลื่อนที่ และพิกัดที่ใช้ในการวาดแผนที่ในแบบต่างๆ ซึ่งพิกัดทั้งหมดผ่านขั้นตอนการประมวลผลทั้ง 2 ฟังก์ชันนี้ทั้งสิ้น โดยจะใช้หลักการตรีโกณมิติที่ได้กล่าวไว้ในหัวข้อที่ 2.5 ซึ่งทั้ง 2 ฟังก์ชันนี้จะรับค่าอยู่ 3 ค่า คือ ค่าพิกัดเริ่มต้น และค่าระยะทาง 0-400 จากอัลตราโซนิกเซ็นเซอร์และทิศทาง 0-359.9 จากโมดูล CMPS10 มาประมวลผล ทางคณะผู้ดำเนินโครงการจึงได้สร้างฟังก์ชัน Cal_x เพื่อคำนวณหาพิกัดในแนวแกน X และฟังก์ชัน Cal_y เพื่อคำนวณหาพิกัดในแนวแกน Y อธิบายเป็น Pseudo code ได้ดังนี้

- ฟังก์ชัน Cal_x รับค่า 3 ค่า
 - ค่าที่ 1 พิกัดในแนวแกน X ให้เท่ากับตัวแปร x
 - ค่าที่ 2 ระยะทางจากอัลตราโซนิกเซ็นเซอร์ให้เท่ากับตัวแปร l
 - ค่าที่ 3 ทิศทางจากโมดูล CMPS10 ให้เท่ากับตัวแปร ag
- คำนวณหาพิกัด $x = x + (l * \text{Cosine}(ag))$
- Return x
- ฟังก์ชัน Cal_y รับค่า 3 ค่า
 - ค่าที่ 1 พิกัดในแนวแกน Y ให้เท่ากับตัวแปร y
 - ค่าที่ 2 ระยะทางจากอัลตราโซนิกเซ็นเซอร์ให้เท่ากับตัวแปร l
 - ค่าที่ 3 ทิศทางจากโมดูล CMPS10 ให้เท่ากับตัวแปร ag
- คำนวณหาพิกัด $y = y + (l * \text{Sine}(ag))$
- Return y

3.4.3.2 Pseudo code สำหรับปุ่มควบคุมเพื่อความสะดวกของระบบ

ปุ่มที่จะใช้ในการควบคุมนั้นจะมีอยู่ 3 ปุ่มคือ Auto, Up/Down และ Left/Right ดังรูปที่ 3.10 ซึ่งแต่ละปุ่มจะมีหน้าที่แตกต่างกันไป ในที่นี้จะขออธิบายปุ่ม Auto, Up/Down และ Left/Right ตามลำดับ



รูปที่ 3.10 ปุ่มที่ใช้ในการควบคุม

3.4.3.2.1 Pseudo code การทำงานของปุ่ม Auto

การทำงานของปุ่ม Auto จำเป็นจะต้องทำงานควบคู่กับการทำงานของบอร์ด Arduino Uno R3 และบอร์ด PIC 16F877 เนื่องจากเมื่อมีการส่งสัญญาณบังคับให้หุ่นยนต์เคลื่อนที่หมุน บอร์ด Arduino Uno R3 จะส่งสัญญาณมายังบอร์ด PIC 16F877 และบอร์ด PIC 16F877 จะส่งสัญญาณมายัง Computer โดย ปุ่ม Auto จะกำหนดค่า Temp เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และกำหนดค่า Select เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และ 3.4.3.4 ทางคณะผู้ดำเนินโครงการอธิบายเป็น Pseudo code ดังนี้

- เมื่อมีการคลิกปุ่ม Auto
 - กำหนดให้ค่า Temp เท่ากับ 0
 - กำหนดให้ค่า Select เท่ากับ 2

3.4.3.2.2 Pseudo code การทำงานของปุ่ม Up/Down

การทำงานของปุ่ม Up/Down นั้นไม่จำเป็นจะต้องมีการส่งสัญญาณจากบอร์ด Arduino Uno R3 มายังบอร์ด PIC 16F877 แต่จำเป็นจะต้องบังคับให้หุ่นยนต์เคลื่อนที่เดินหน้า ถอยหลังหรืออยู่กับที่เท่านั้น โดยปุ่ม Up/Down จะกำหนดค่า Temp เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และกำหนดค่า Select เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และ 3.4.3.4 ทางคณะผู้ดำเนินโครงการอธิบายเป็น Pseudo code ดังนี้

- เมื่อมีการคลิกปุ่ม Up/Down
 - กำหนดให้ค่า Temp เท่ากับ -1
 - กำหนดให้ค่า Select เท่ากับ 0

3.4.3.2.3 Pseudo code การทำงานของปุ่ม Up/Down

การทำงานของปุ่ม Left/Right นั้นไม่จำเป็นจะต้องมีการส่งสัญญาณจากบอร์ด Arduino Uno R3 มายังบอร์ด PIC 16F877 แต่จำเป็นจะต้องบังคับให้หุ่นยนต์เคลื่อนที่โดยการหมุนเท่านั้น โดยปุ่ม Left/Right จะกำหนดค่า Temp เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และกำหนดค่า Select เพื่อนำไปควบคุมการประมวลผลในหัวข้อที่ 3.4.3.3 และ 3.4.3.4 ทางคณะผู้ดำเนินโครงการอธิบายเป็น Pseudo code ดังนี้

- เมื่อมีการคลิกปุ่ม Left/Right
- กำหนดให้ค่า Temp เท่ากับ -1
- กำหนดให้ค่า Select เท่ากับ 1

3.4.3.3 Pseudo code สำหรับการรับค่าจากบอร์ดไมโครคอนโทรลเลอร์มาประมวลผล

จากที่เคยกล่าวในหัวข้อที่ 3.4.1 เรื่องโปรโตคอลที่ใช้ในการรับส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์กับตัวโปรแกรมในหัวข้อนี้จะอธิบายถึงการนำโปรโตคอลมาใช้ในการประมวล การเช็คค่าโปรโตคอลส่งมาถูกต้องหรือไม่ รวมไปถึงจนถึงตัวแปร Temp และ Select ว่ามีการควบคุมอย่างไรบ้าง การนำค่า 2 ไบต์จากโมดูล CMPS10 มาคำนวณหาค่า 0-359.9 และการนำฟังก์ชันคำนวณหาระยะทางจากอัลตราโซนิกเซ็นเซอร์มาคำนวณหาค่า 0-400 ทางคณะผู้ดำเนินโครงการจึงได้อธิบายเป็น Pseudo code ดังนี้

- ให้ receive คือชุดของข้อมูลที่อ่านได้จาก Serial Port
- ถ้าข้อมูลมาครบ
- ถ้าข้อมูลข้อมูล 4 ไบต์แรกเท่ากับ "CMPH"
- กำหนดให้ค่า in_cmph คือไบต์ที่ 4 ถึงไบต์ที่ 10
- กำหนดให้ค่า in_cmpl คือไบต์ที่ 14 ถึงไบต์ที่ 20
- กำหนดให้ค่า in_ulh คือไบต์ที่ 23 ถึงไบต์ที่ 29
- กำหนดให้ค่า in_ull คือไบต์ที่ 32 ถึงไบต์ที่ 38
- ถ้าค่า Temp ไม่เท่ากับ -1

กำหนดให้ค่า Select คือไบต์ที่ 38 ถึงไบต์ที่ 39

- คำนวณหาค่าจากโมดูล CMPS10

$$\text{in_cmph} = (256 * \text{in_cmph} + \text{in_cmpl}) / 10$$

- คำนวณหาค่าจากอัลตราโซนิกเซ็นเซอร์

$$\text{in_ulh} = \text{Cal_length}(256 * \text{in_ulh} + \text{in_ull})$$

- ถ้าค่า in_ulh อยู่ในช่วง 0 ถึง 400

- ถ้าค่า Select เท่ากับ 0

ให้ค่า length เท่ากับ in_ulh

- ถ้าค่า Select ไม่เท่ากับ 0

ให้ค่า angle เท่ากับค่า in_cmph และ

ค่า length เท่ากับค่า in_ulh

3.4.3.4 Pseudo code สำหรับนำค่าที่ได้มาทำการประมวลผล

ในโปรแกรมการประมวลผลนั้นจะเป็นการนำค่าระยะทาง 0-400 จากอัลตราโซนิกเซ็นเซอร์และค่าทิศทาง 0-359.9 จากโมดูล CMPS10 มาประมวลผลโดยจะมีค่าของ Index เป็นตัวกำหนดการทำงานของโปรแกรมซึ่งการทำงานในส่วนนี้จะเป็นการคำนวณหาพิคคของเซ็นเซอร์ พิคคของลำแสงเซ็นเซอร์ พิคคของหุ่นยนต์เคลื่อนที่และพิคคของแผนที่ต่างๆ โดยตัวแปร Select ก็จะถูกนำมาใช้ควบคุมการทำงานในส่วนนี้เหมือนกัน ทางคณะผู้ดำเนินโครงการอธิบายเป็น Pseudo code ดังนี้

Process()

- ถ้า Index เท่ากับ 1

กำหนดให้หุ่นยนต์เคลื่อนที่อยู่กับที่

ag[Index] เก็บค่าของของโมดูล CMPS10

li[Index] เก็บค่าของอัลตราโซนิกเซ็นเซอร์

- ถ้า Index ไม่เท่ากับ 1

ag[Index] เก็บค่าของของโมดูล CMPS10

li[Index] เก็บค่าของอัลตราโซนิกเซ็นเซอร์

คำนวณหาทิศทางหรือองศาที่เปลี่ยนแปลงจาก Index ก่อนหน้านี้

- ถ้า Select เท่ากับ 1 แสดงว่าหุ่นยนต์เคลื่อนที่มีการหมุน
กำหนดให้หุ่นยนต์เคลื่อนที่อยู่กับที่
- ถ้า Select ไม่เท่ากับ 1 แสดงว่าหุ่นยนต์เคลื่อนที่มีการเดินหน้า ถอยหลัง หรืออยู่กับที่
คำนวณหาพิกัดใหม่ของหุ่นยนต์เคลื่อนที่
- คำนวณหาพิกัดของเซ็นเซอร์ ลำแสงของเซ็นเซอร์

3.4.3.5 Pseudo code สำหรับนำผลลัพธ์ที่ได้จากข้อที่ 3.4.3.4 มาแสดงผลบนหน้าจอ GUI

การวาดภาพแสดงผลนั้นคณะผู้ดำเนินโครงการได้เลือกให้การวาดอยู่ในฟังก์ชัน `pictureBox2_Paint` โดยจะนำฟังก์ชันในหัวข้อที่ 3.4.3.4 และฟังก์ชันทั้งหมดที่ได้กล่าวในหัวข้อที่ 3.4.2 มาใช้ ทางคณะผู้ดำเนินโครงการอธิบายเป็น Pseudo code ดังนี้

`pictureBox2_Paint(object sender, PaintEventArgs e)`

- ถ้า Index เท่ากับ 1
ทำงานฟังก์ชัน `Process`
- ถ้า Index ไม่เท่ากับ 1
ทำงานฟังก์ชัน `Process`
ทำการวาดแผนที่ของเซ็นเซอร์โดยใช้ฟังก์ชัน `CreateMap_Sensor`
ทำการวาดแผนที่ของลำแสงเซ็นเซอร์โดยใช้ฟังก์ชัน `CreateMap_Beam`
ทำการวาดแผนที่ของหุ่นยนต์เคลื่อนที่โดยใช้ฟังก์ชัน `CreateMap_Car`
- ทำการวาดรูปภาพของเซ็นเซอร์โดยใช้ฟังก์ชัน `Sensor`
- ทำการวาดรูปภาพลำแสงของเซ็นเซอร์โดยใช้ฟังก์ชัน `Sensor_Beam`
- ทำการวาดรูปภาพหุ่นยนต์เคลื่อนที่โดยใช้ฟังก์ชัน `Car`
- เพิ่มค่า Index ที่ละ 1

3.5 บทสรุป

ในบทที่ 3 นี้กล่าวถึงการดำเนินงานทั้งหมด ซึ่งอันดับแรกจะกล่าวถึงภาพรวมของระบบ (System Overview) ว่าระบบนั้นมีองค์ประกอบอะไรและมีหลักการทำงานอย่างไร จากนั้นจะกล่าวถึงการเขียนโปรแกรมลงบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เพื่อใช้ในการควบคุมการเคลื่อนที่ การสร้างตัวหุ่นยนต์เคลื่อนที่ที่ใช้ในการเคลื่อนที่การใช้วงจรควบคุมมอเตอร์ และยังกล่าวถึงการเขียนโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์ PIC16F877 เพื่อใช้ในการอ่านค่าจากเซ็นเซอร์ที่ใช้สำหรับวัดระยะทางและทิศทาง ว่ามีวิธีการอย่างไร เริ่มจากการใช้โมดูล Timer1 ในการจับเวลา หลักการประมวลผลสัญญาณที่ได้จากอัลตราโซนิกเซ็นเซอร์ SRF05 และโมดูล CMPS10 พร้อมทั้งกล่าวถึงโปรโตคอลในการรับส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์กับตัวโปรแกรมและ Pseudo Code ที่สำคัญของขั้นตอนต่างๆ ดังที่ได้กล่าวมา รวมทั้ง การเขียนโปรแกรมสร้างหน้าจอ GUI สำหรับแสดงผล พร้อมทั้ง Pseudo Code สำคัญในขั้นตอนต่างๆของการสร้างแผนที่จำลองขึ้นมา



บทที่ 4

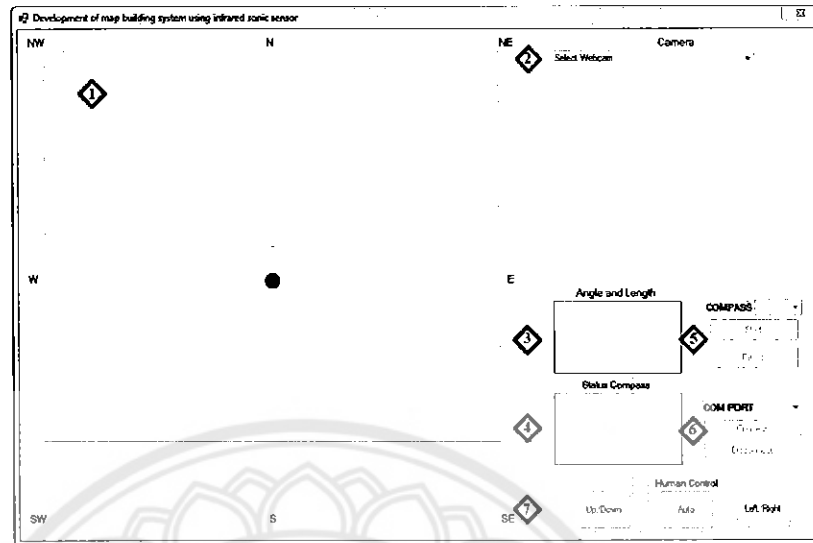
การทดลองและผลการทดลอง

4.1 ระบบที่พัฒนาขึ้น



รูปที่ 4.1 ภาพรวมของระบบ

จากรูปที่ 4.1 แสดงภาพของระบบที่พัฒนาขึ้น คือระบบสร้างแผนที่ ซึ่งจะทำหน้าที่วัดระยะทางระหว่างหุ่นยนต์เคลื่อนที่ถึงสิ่งกีดขวางรวมทั้งทิศทางและทำการประมวลผลจากนั้นจะส่งค่าที่วัดได้ไปยังเครื่องคอมพิวเตอร์ (Laptop) ผ่าน Serial Port เพื่อไปสร้างเป็นแผนที่จำลองขึ้นมาบนหน้าจอ GUI โดยจะแสดงการจำลองทั้งทิศทางและขนาดของพื้นที่ที่หุ่นยนต์ได้เคลื่อนที่ไป



รูปที่ 4.2 หน้าจอ GUI ของระบบ

หน้าจอ GUI ของระบบที่พัฒนาขึ้นจะประกอบไปด้วย 4 องค์ประกอบ ดังแสดงในรูปที่ 4.2 ดังนี้

- หมายเลข 1. คือส่วนที่ใช้สำหรับการแสดงผล ซึ่งคือภาพจำลองของระบบ โดยที่จะแสดงการสร้างแผนที่จากข้อมูลที่วัดได้จากเซ็นเซอร์ทั้ง 2 ตัว (SRF05, CMPS10) ทั้งทิศทางและระยะทาง
- หมายเลข 2. Camera คือส่วนที่ใช้แสดงผลภาพจากกล้องที่ใช้ต่อกับคอมพิวเตอร์ เพื่อเปรียบเทียบกับแผนที่ที่จริงกับแผนที่ที่ GUI สร้างขึ้น
- หมายเลข 3. Angle and Length คือส่วนที่จะแสดงระยะทางและทิศทางที่วัดได้ มีหน่วยเป็นเซนติเมตร และองศาตามลำดับ
- หมายเลข 4. Status Compass คือส่วนแสดงสถานะของ CMPS10 ว่าพร้อมทำงานหรือไม่
- หมายเลข 5. Compass คือส่วนที่ใช้เลือกทิศทางแรกที่ต้องการให้ระบบเริ่มวาดแผนที่
- หมายเลข 6. COM PORT คือส่วนที่ใช้สำหรับเลือกพอร์ตที่ต่อกับระบบ
- หมายเลข 7. Human Control คือส่วนที่ใช้บังคับการทำงานของ SRF05 และ CMPS10 เพื่อควบคุมเสถียรภาพของระบบ โดยแยกเป็น 3 ส่วนดังนี้
- Up/Down คือ การใช้งานเฉพาะค่าที่ได้รับจาก SRF05 เพื่อลดปัญหาของการแกว่งของค่าที่ได้รับจาก CMPS10 ซึ่งการทำงานในโหมดนี้จะใช้ระหว่างการเดินทางหรือถอยหลังของรถเท่านั้น

- Auto คือ โหมดการทำงานร่วมกับหุ่นยนต์เคลื่อนที่ โดยจะเปิดการใช้งานค่าที่ได้รับจากเซ็นเซอร์ทั้ง 2 ตัว (SRF05 และ CPMS10) และทำงานตามอัลกอริทึมของโปรแกรม
- Left/Right คือ เปิดการใช้งานค่าที่ได้รับจากเซ็นเซอร์ทั้ง 2 ตัว (SRF05 และ CPMS10) ซึ่งการทำงานในโหมดนี้จะใช้ระหว่างการเปลี่ยนทิศทางของหุ่นยนต์เคลื่อนที่เท่านั้น

4.2 ผลการทดลองการคำนวณหาระยะทางจากค่าที่ได้รับจากเซ็นเซอร์

4.2.1 วิธีการทดลอง

สำหรับผลการทดลองทุกการทดลอง ทางคณะผู้ดำเนินโครงการได้ทำการทดสอบในอาคาร ซึ่งเป็นห้องโถง มีขนาด 8 x 12 เมตร ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 สถานที่ทำการทดลอง

การทดลองนั้นจะใช้อุปกรณ์วัดระยะทางเข้ามาช่วยในการเปรียบเทียบระยะทางตั้งแต่ 45 เซนติเมตร จนถึง 4 เมตร กับค่าที่ได้จากตัวอัลตราโซนิกเซ็นเซอร์ SRF05

สำหรับการเปรียบเทียบนั้น จะนำค่าที่ได้จากการวัดจริงไปเปรียบเทียบกับค่าที่ได้รับจากเซ็นเซอร์ โดยแสดงรายละเอียดของวิธีการคำนวณ เพื่อให้ได้มาซึ่งค่าที่มีความใกล้เคียงกับค่าความจริงให้มากที่สุด โดยการติดตั้งตัวเซ็นเซอร์กับหุ่นยนต์เคลื่อนที่นั้น จะต้องให้ตัวเซ็นเซอร์ทำมุมเงยขึ้นจากแนวระนาบเป็นมุม 30 องศา [24] ซึ่งจะทำให้ได้ค่าระยะทางจากหุ่นยนต์เคลื่อนที่ถึงวัตถุที่ถูกต้อง ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 การติดตั้งเซ็นเซอร์กับตัวหุ่นยนต์เคลื่อนที่

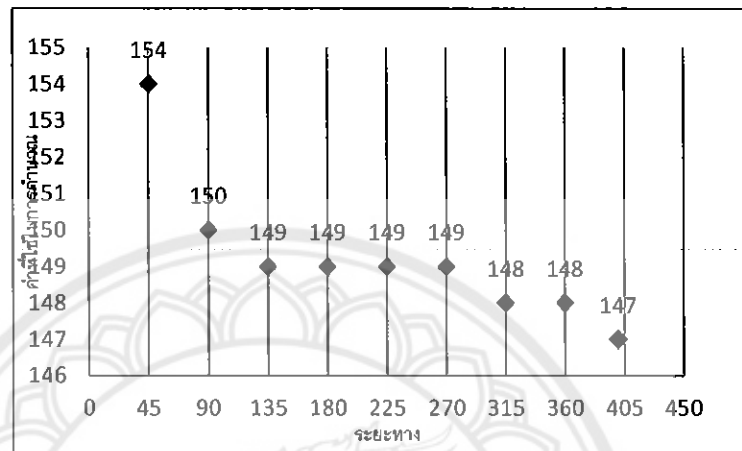
4.2.2 ผลการทดลอง

จากการทดลองพบว่า เมื่อทำการเปรียบเทียบระยะทางจริงกับค่าที่ได้รับจากอัลตราโซนิกเซ็นเซอร์ ทำให้สามารถหาค่าปรับเทียบออกมาได้ ซึ่งทางคณะผู้ดำเนินโครงการได้แบ่งค่าปรับเทียบนั้นออกเป็นช่วงๆ เพื่อเพิ่มความแม่นยำในการทำงานของอัลตราโซนิกเซ็นเซอร์ยิ่งขึ้น

ตารางที่ 4.1 แสดงผลการทดลองระยะทางจริงเปรียบเทียบกับระยะทางที่วัดได้จากอัลตราโซนิกเซ็นเซอร์ SRF05

ระยะจริง (เซนติเมตร)	ไบต์สูง	ไบต์ต่ำ	(ไบต์สูง * 256) + ไบต์ต่ำ	ค่าไบต์/ระยะจริง	ค่าคงที่ในการปรับเทียบ
45	27	23	6935	154.111	154
90	52	194	13506	150.067	150
135	78	154	20122	149.052	149
180	104	204	26828	149.044	149
225	130	254	33534	149.040	149
270	162	258	41730	149.036	149
315	182	22	46614	147.981	148
360	208	24	53272	147.978	148
400	229	184	58808	147.020	147

จากตารางที่ 4.1 แสดงให้เห็นถึงการเปรียบเทียบข้อมูลที่ได้รับจากอัลตราโซนิคเซ็นเซอร์ SRF05 กับระยะทางจริง ซึ่งจะทำให้ได้ค่าคงที่ในการเปรียบเทียบ ส่งผลให้ระยะทางวัดที่ได้มีความแม่นยำมากขึ้น



รูปที่ 4.5 แสดงความสัมพันธ์ระหว่างระยะทางจริงกับค่าที่นำมาใช้คำนวณ

จากรูปที่ 4.5 แสดงกราฟความสัมพันธ์ระหว่างระยะทางจริงกับค่าที่นำมาใช้คำนวณ ซึ่งพบว่าค่าคงที่ที่ใช้ในการเปรียบเทียบนั้นแบ่งออกเป็นช่วง โดยในระยะ 0 - 45 เซนติเมตร จะใช้ค่า 154 ในการเปรียบเทียบ ระยะ 46 - 90 เซนติเมตร จะใช้ค่า 150 ในการเปรียบเทียบ ระยะ 91 - 270 เซนติเมตร จะใช้ค่า 149 ในการเปรียบเทียบ ระยะ 271 - 360 เซนติเมตร จะใช้ค่า 148 ในการเปรียบเทียบ และระยะ 361 - 400 เซนติเมตร จะใช้ค่า 147 ในการเปรียบเทียบ

4.2.3 วิเคราะห์ผลการทดลองการคำนวณหาระยะทางจากอัลตราโซนิคเซ็นเซอร์

SRF05

จากตารางที่ 4.1 และรูปที่ 4.3 จะพบว่าค่าคงที่ที่ต้องนำมาใช้ในการคำนวณนั้นไม่คงที่ โดยต้องแบ่งระยะออกเป็นช่วงดังนี้

ตารางที่ 4.2 แสดงผลการทดลองการคำนวณหาระยะทางจากอัลตราโซนิกเซ็นเซอร์ SRF05

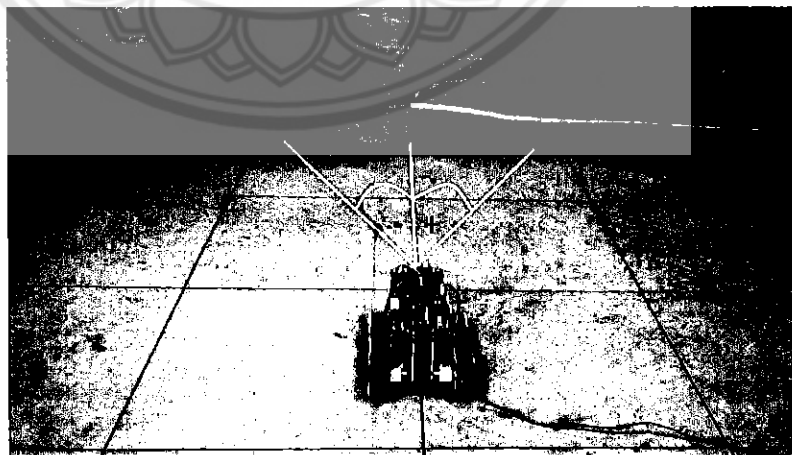
ระยะทาง (เซนติเมตร)	จะได้ค่าที่นำมาเปรียบเทียบในแต่ละช่วงดังนี้	
	ค่า 2 ไบต์	ค่าที่นำมาหาร
0 - 45	0 - 6935	154
46 - 90	6936 - 13506	150
91 - 270	13507 - 41730	149
271 - 360	41731 - 53272	148
361 - 400	53273 - 58808	147

จากผลลัพธ์ที่ได้ เมื่อทำการเปรียบเทียบแล้วจะได้ค่าคงที่ในการเปรียบเทียบหลายค่า โดยแบ่งออกเป็นช่วง ดังตารางที่ 4.2 เมื่อเปรียบเทียบกับข้อมูลในคู่มือการใช้งาน ซึ่งใช้ค่า 148 ในการเปรียบเทียบเพียงค่าเดียว ทำให้เกิดความคลาดเคลื่อนของระยะทางที่วัดได้มากขึ้น

4.3 ผลการทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตราโซนิกเซ็นเซอร์ SRF05

4.3.1 วิธีการทดลอง

การทดลองนี้จะทำการเปรียบเทียบระยะทางที่ได้จากเซ็นเซอร์กับระยะจริงในช่วงที่ค่ามีการเปลี่ยนแปลงครั้งละมากๆ หรือเรียกอีกอย่างว่าเกิดปัญหา Critical Angle นั้นเอง



รูปที่ 4.6 การทดลองหา Critical Angle

สำหรับการทดลองนั้น จะทำการหันหน้าหุ่นยนต์เคลื่อนที่เข้าหาผนัง โดยเว้นระยะห่างจากตัวหุ่นยนต์ถึงผนังไม่เป็นระยะทาง 100 เซนติเมตร และทำการหมุนรถไปทางขวาทีละ 5 องศา จนกว่าการทำงานของเซ็นเซอร์จะเกิดปัญหา Critical Angle ขึ้น แล้วทำการเปลี่ยนหมุนไปทางซ้ายทีละ 5 องศาเช่นเดียวกัน จนการทำงานของเซ็นเซอร์เกิดปัญหา Critical Angle ขึ้น ดังแสดงในรูปที่ 4.6

4.3.2 ผลการทดลอง

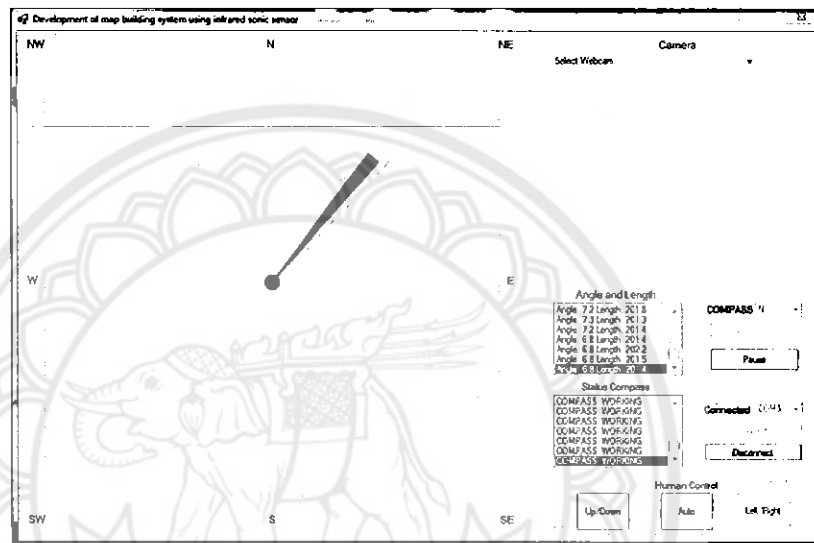
จากการทดลองพบว่า ผลลัพธ์ที่ได้จากการทดลองทั้ง 5 ครั้ง มีแนวโน้มไปทางเดียวกัน คือ เมื่อทำการวัดมุมเอียงระหว่าง -35 องศา ถึง + 35 องศา ระยะทางที่วัดได้จากอัลตราโซนิกเซ็นเซอร์ยังผิดพลาดจากระยะทางจริงไม่มากนัก แต่เมื่ออัลตราโซนิกเซ็นเซอร์ทำมุมเอียงกับผนังตั้งแต่ ± 40 องศาขึ้นไป ระยะทางที่อัลตราโซนิกเซ็นเซอร์วัดได้ เกินกว่าระยะทางจริงมาก หรือเกิดปัญหา Critical Angle นั้นเองดังนั้นในหัวข้อนี้จะขอแสดงผลทางทดลองจากการทดสอบครั้งล่าสุด ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 แสดงผลการทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตราโซนิกเซ็นเซอร์ SRF05

มุมเอียง (องศา)	ระยะทางจากการคำนวณ (เซนติเมตร)	ระยะทางจากเซ็นเซอร์ (เซนติเมตร)
40	130.7	201.4
35	122.2	108.1
30	115.6	107.1
25	110.3	105.4
20	106.4	104.9
15	103.6	104.7
10	101.6	104
5	100.4	103.9
0	100.7	101.8
-5	100.4	103.8
-10	101.5	103.9
-15	104.3	104.4
-20	106.6	104.6
-25	110.7	106.1
-30	115.7	106.5

-35	122.2	107.5
-40	130.9	229.6

จากตารางที่ 4.3 แสดงถึงการเปรียบเทียบระหว่างระยะทางที่อัลตราโซนิกเซ็นเซอร์วัดได้กับระยะทางที่วัดได้จริง ซึ่งทำให้ได้ข้อสรุปว่า มุมที่มีขนาดโตกว่า ± 40 องศาจะทำให้เกิดปัญหา Critical Angle ขึ้น



รูปที่ 4.7 ปัญหา Critical Angle

หมายเหตุ: ทำการทดลองนี้กับผนังไม้

จากรูปที่ 4.7 แสดงรูปของ Critical Angle ที่เกิดขึ้น ส่งผลแผนที่ที่ได้จากการจำลองของระบบมีความคลาดเคลื่อน เมื่อเทียบกับสถานที่ทำการทดลองจริง

4.3.3 วิเคราะห์ผลการทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตรา

โซนิคเซ็นเซอร์ SRF05

จากตารางที่ 4.3 จะพบว่าในเบื้องต้น ค่าที่ได้รับจากอัลตราโซนิคเซ็นเซอร์นั้นมีความใกล้เคียงกับระยะทางจริงระหว่างค่ามุม -35 องศา ถึงมุม +35 องศา โดยถ้าอัลตราโซนิคเซ็นเซอร์ทำมุมกับวัตถุในมุมที่มีขนาดโคกว่า ± 40 องศา จะทำให้เกิดปัญหา Critical Angle ดังรูปที่ 4.7 ทำให้แผนที่ที่ได้จากการจำลองขอรระบบเกิดความผิดพลาดจากความเป็นจริงไปมาก

4.4 ผลการทดลองการวัดขนาดลำแสง (Beam width) ของอัลตราโซนิคเซ็นเซอร์ SRF05

จุดประสงค์ของการทดลองนี้คือ เพื่อวัดขนาดของลำแสงอัลตราโซนิคเซ็นเซอร์ เพื่อนำไปเพิ่มประสิทธิภาพของระบบ โดยนำขนาดของลำแสงอัลตราโซนิคที่ได้จากการทดลอง ไปปรับเทียบกับขนาดของลำแสงที่ใช้ในการวาด ส่งผลให้ระบบสามารถจำลองแผนที่ได้รวดเร็วและชัดเจนยิ่งขึ้น

4.4.1 วิธีการทดลอง

การทดลองนี้จะเป็นการวัดหาขนาดความกว้างลำแสงของอัลตราโซนิคเซ็นเซอร์ที่ปล่อยออกไปขณะทำงาน โดยเริ่มต้นด้วยการวัดระยะในแนวตั้งฉากกับลำแสงของอัลตราโซนิคเซ็นเซอร์แล้วเริ่มขยับวัตถุที่ใช้ทำการทดลองออกไปด้านขวาเรื่อยๆ จนพบจุดที่อัลตราโซนิคเซ็นเซอร์ไม่สามารถตรวจจับวัตถุได้ หลังจากนั้นเปลี่ยนไปทำการทดลองเช่นเดียวกันนี้ในด้านซ้ายของอัลตราโซนิคเซ็นเซอร์

4.4.2 ผลการทดลอง

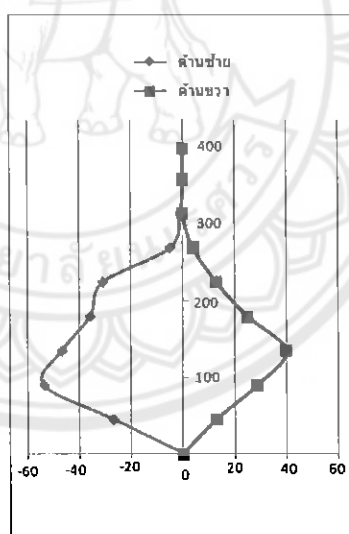
จากการทดลองพบว่า ในระยะ 100 เซนติเมตรเมื่อวัดจากหุ่นยนต์เคลื่อนที่ถึงวัตถุ อัลตราโซนิคจะมีขนาดของลำแสงกว้างถึง 30 องศาในด้านซ้าย และ 15 องศา ในด้านขวา และหลังจากนั้นขนาดของลำแสงก็จะลดลงมา เมื่อระยะทางมากกว่า 300 เซนติเมตรขึ้นไป ขนาดของลำแสงของอัลตราโซนิคนั้นเริ่มจะกลายเป็นเส้นตรง

ตารางที่ 4.4 ผลการทดลองการวัดขนาดลำแสงของอัลตราโซนิคเซ็นเซอร์ SRF05

ระยะทาง (เซนติเมตร)	ระยะทางด้านซ้าย (เซนติเมตร)	ระยะทางด้านขวา (เซนติเมตร)
0	0	0
45	-27	13
90	-54	29

135	-47	40
180	-36	25
225	-31	13
270	-5	4
315	-1	0
360	0	0
400	0	0

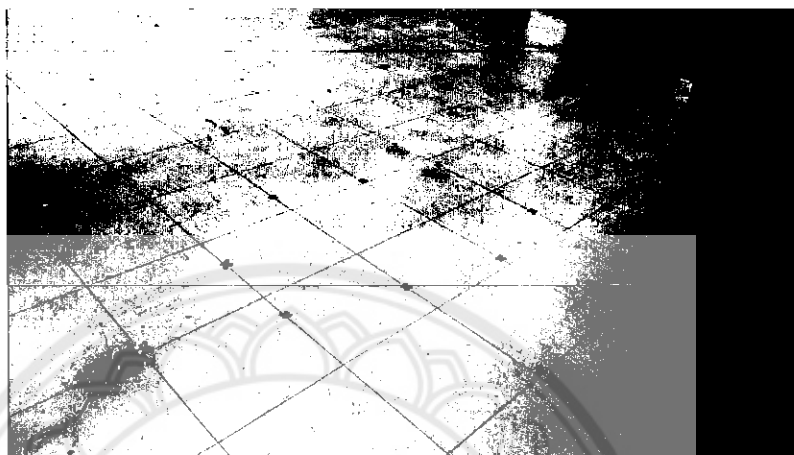
จากตารางที่ 4.4 พบว่าในระยะ 45 เซนติเมตร ขนาดของลำแสงจากอัลตราโซนิกเซ็นเซอร์นั้นจะกระจายตัวออกไปทางด้านซ้ายถึง 27 เซนติเมตร หรือทำมุม 30 องศา และในระยะเดียวกันขนาดลำแสงจากอัลตราโซนิกเซ็นเซอร์นั้นจะกระจายตัวออกไปทางด้านขวาถึง 13 เซนติเมตรหรือทำมุม 15 องศา โดยขนาดของมุมทั้งสองด้านจะเป็นเช่นนี้ไปจนถึงระยะทาง 100 เซนติเมตร และเมื่อระยะทางมากกว่า 100 เซนติเมตรขึ้นไป ค่าขนาดของลำแสงจากอัลตราโซนิกก็จะเริ่มลดลง



รูปที่ 4.8 ลักษณะลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05

จากรูปที่ 4.8 แสดงเส้นกราฟจากการทดลองหาขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์ ซึ่งสามารถอธิบายได้ว่า ในระยะแรกนั้น ขนาดของลำแสงจากอัลตราโซนิกเซ็นเซอร์จะมีความกว้าง

ค่อนข้างมาก แต่เมื่อระยะไกลออกไป ขนาดของลำแสงจากอัลตราโซนิกเซ็นเซอร์ ก็จะเริ่มลดลงจนแทบจะเป็นเส้นตรง



รูปที่ 4.9 ผลการทดลองการหาขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05

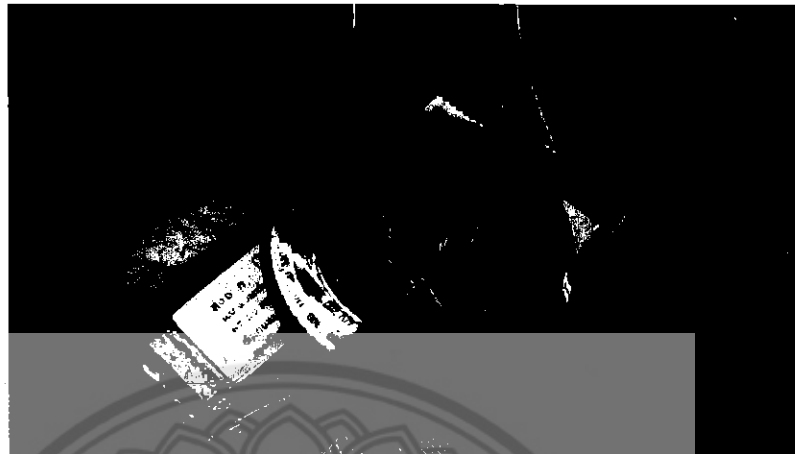
จากภาพที่ 4.9 แสดงถึงผลการทดลองจากสถานที่ทำการทดลองจริง ซึ่งขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์จากการทดลอง เมื่อเปรียบเทียบกับขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์จากคู่มือพบที่มีความใกล้เคียงกัน แต่ขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์จากคู่มือ จะมีความสมคูลมากกว่านี้เนื่องจากการทำการทดลองในคู่มือนั้น เป็นการทำการทดลองในสถานที่ที่มีสภาพแวดล้อมที่ปราศจากสิ่งรบกวนจากภายนอก หรือกล่าวคือ เป็นการทดลองในอุดมคติ ทำให้ได้ผลการทดลองที่สมบูรณ์มาก

4.5 ผลการทดลองการวัดมุมเอียงของเซ็นเซอร์ CMPS10

4.5.1 วิธีการทดลอง

การทดลองนี้จะใช้อุปกรณ์ในการวัดเป็นฐานวงกลมที่มีสเกลสำหรับการวัดตั้งแต่ 0 – 359 องศา โดยใช้สำหรับวัดมุมของทิศทาง เพื่อเปรียบเทียบกับค่าที่ระบบวัดได้ ดังแสดงในรูปที่ 4.10

สำหรับการวัดค่ามุมเอียงนั้น จะเริ่มวัดตั้งแต่ 0 องศาโดยเทียบกับทิศเหนือ และปรับเพิ่มค่าทีละ 10 องศา จนถึงสุดการวัดที่ 350 องศา โดยแต่ละครั้งที่วัดนั้น จะทำการวัดซ้ำทั้งหมด 10 ค่า



รูปที่ 4.10 วิธีการทดลองวัดมุมเอียงของทิศทาง

4.5.2 ผลการทดลอง

จากการทดลองแสดงให้เห็นถึงศักยภาพการทำงานของโมดูลเข็มทิศดิจิทัลว่ามีประสิทธิภาพการทำงานสูง มีความผิดพลาดน้อยมาก โดยหลังจากทำการทดลองมีการบันทึกค่าและคำนวณค่าเฉลี่ยของความผิดพลาดออกมา ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 แสดงผลการทดลอง ค่าเฉลี่ย และค่าความคลาดเคลื่อนของการทดลองของ การวัดมุมเอียงในของทิศทาง

มุมเอียง (องศา)	ค่ามุมที่วัดได้ (องศา)										ค่าเฉลี่ย (องศา)	ค่าความคลาด เคลื่อน (องศา)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10		
0	0.1	0.1	0.3	0.1	0.2	0.2	0.1	0.0	0.1	0.1	0.13	0.13
10	10.2	10.2	10.1	10.2	10.2	10.1	10.0	10.1	10.1	10.0	10.12	0.12
20	20.2	20.1	20.0	20.1	20.2	20.1	20.1	20.0	20.0	20.1	20.09	0.09
30	30.0	30.1	29.9	29.9	29.8	29.8	30.0	30.0	30.1	30.1	29.97	0.03
40	40.0	39.9	40.0	40.1	40.1	40.1	40.0	40.0	40.1	40.1	40.04	0.04
50	50.0	50.1	50.0	50.0	50.1	50.0	50.0	50.0	50.0	50.1	50.03	0.03
60	60.2	60.1	60.2	60.3	60.3	60.2	60.1	60.2	60.1	60.1	60.18	0.18
70	69.8	70.0	69.9	70.0	69.9	69.9	69.9	69.9	69.8	69.8	69.89	0.11
80	79.9	80.0	80.2	80.0	79.6	79.8	79.8	79.7	79.8	79.9	79.87	0.13

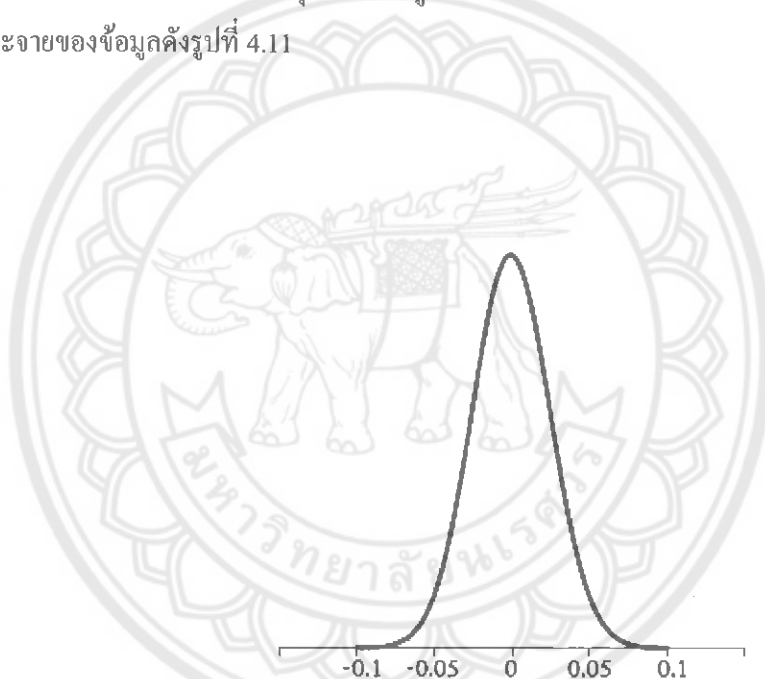
90	90.2	89.9	90.1	90.2	90.1	90.1	90.0	89.9	89.9	89.8	90.02	0.02
มุมเอียง (องศา)	ค่ามุมที่วัดได้ (องศา)										ค่าเฉลี่ย (องศา)	ค่าความคลาด เฉลี่ย (องศา)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10		
100	99.8	100.0	100.0	100.0	100.0	99.9	100.1	100.2	100.0	100.0	100.00	0.00
110	109.9	110.1	110.2	110.0	109.9	109.9	109.8	110.0	110.0	110.0	109.98	0.02
120	119.9	119.9	120.0	120.0	120.1	120.0	119.9	120.0	120.0	120.0	119.98	0.02
130	130.2	130.2	130.2	130.2	130.2	130.2	130.0	130.0	130.2	130.2	130.16	0.16
140	140.0	139.9	139.9	139.9	140.0	140.0	140.1	140.2	140.1	140.0	140.01	0.01
150	149.9	149.8	149.8	149.9	150.0	149.9	149.9	150.1	149.9	149.9	149.91	0.09
160	160.0	160.0	160.2	160.1	160.2	160.1	160.1	160.0	160.0	160.0	160.07	0.07
170	170.1	170.1	170.1	170.0	170.0	170.0	170.1	170.2	170.2	170.0	170.08	0.08
180	180.2	180.3	180.2	180.2	180.3	180.2	180.2	180.2	180.0	180.1	180.19	0.19
190	190.1	190.0	190.0	189.9	190.0	189.9	189.8	189.8	189.8	189.8	189.91	0.09
200	200.2	200.3	200.2	200.1	200.2	200.1	200.0	200.2	200.3	200.2	200.18	0.18
210	209.9	210.0	210.1	210.0	210.1	210.1	210.1	210.2	210.3	210.2	210.10	0.10
220	220.0	220.2	220.1	220.0	220.1	220.2	220.0	220.0	220.1	220.2	220.09	0.09
230	229.9	229.9	229.8	229.8	229.8	229.8	229.8	229.9	229.9	229.9	229.85	0.15
240	240.1	239.9	239.9	240.0	240.0	240.0	240.1	240.0	240.1	240.0	240.01	0.01
250	249.9	249.8	249.9	249.9	250.0	249.9	250.0	249.9	249.9	249.9	249.91	0.09
260	260.0	260.0	260.2	260.1	260.2	259.9	260.1	260.0	260.0	260.0	260.05	0.05
270	270.1	270.1	270.1	270.1	270.0	270.0	270.2	270.2	270.2	270.0	270.10	0.10
280	280.2	280.3	280.2	280.2	280.1	280.2	280.2	280.3	280.0	280.1	280.18	0.18
290	290.1	290.0	290.1	289.9	290.0	289.9	289.8	289.8	289.8	289.8	289.92	0.08
300	300.2	300.3	300.2	300.1	300.0	300.1	300.0	300.2	300.4	300.3	300.18	0.18
310	309.9	310.0	310.1	310.0	309.9	310.1	310.1	310.2	310.3	310.2	310.08	0.08
320	320.0	320.2	319.9	320.0	320.1	320.2	320.1	320.0	320.1	320.2	320.08	0.08
330	329.9	329.9	329.8	329.8	329.8	329.6	329.8	330.0	330.0	329.9	329.85	0.15
340	340.1	339.9	339.9	340.0	340.0	340.0	340.1	340.0	340.1	340.0	340.01	0.01
350	349.9	349.8	350.0	349.9	350.0	349.9	350.0	349.9	349.9	349.9	349.92	0.08
ค่าความคลาดเฉลี่ย												0.09

จากตารางที่ 4.5 เมื่อนำค่าความคลาดเคลื่อนเฉลี่ยของแต่ละมุมมาคำนวณหาค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ได้จากสูตร

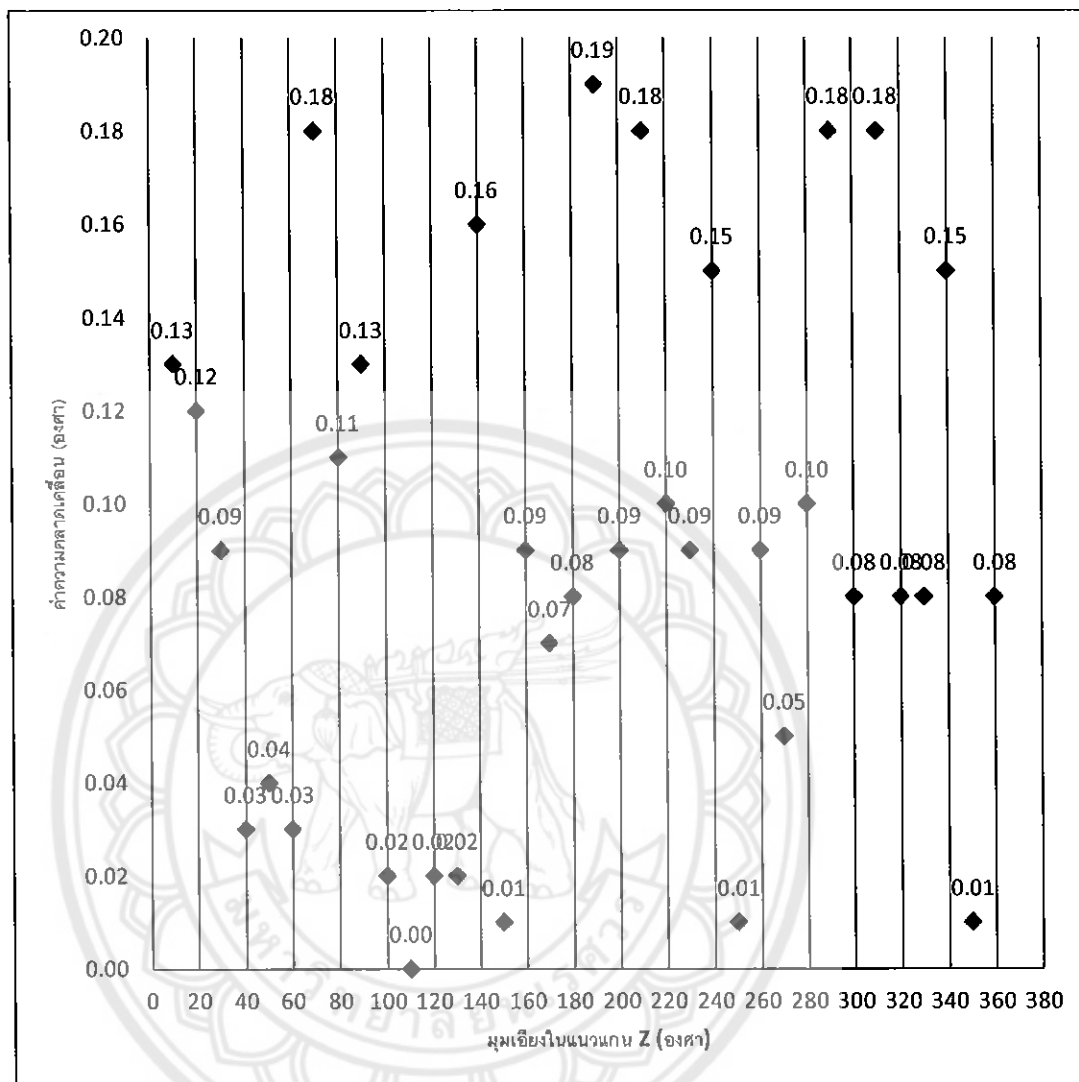
$$\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n-1}}$$

$$\sigma = 0.056918$$

เมื่อพิจารณาค่าเบี่ยงเบนมาตรฐานที่ได้เทียบกับค่าเฉลี่ย พบว่ามีการกระจายตัวของค่าความคลาดเคลื่อนน้อย จึงสามารถสรุปได้ว่าโมดูลเข็มทิศดิจิทัลมีประสิทธิภาพที่ดี โดยสามารถแสดงการกระจายของข้อมูลดังรูปที่ 4.11



รูปที่ 4.11 การกระจายข้อมูลของค่าความคลาดเคลื่อนจาก โมดูลเข็มทิศดิจิทัล CMPS10



รูปที่ 4.12 แสดงความสัมพันธ์ระหว่างค่าของค่าความคลาดเคลื่อนกับค่ามุมเอียงของทิศทาง

จากรูปที่ 4.12 แสดงกราฟความสัมพันธ์ระหว่างค่าของค่าความคลาดเคลื่อนกับค่ามุมเอียงของทิศทาง โดยจะพบว่า ในแต่ละช่วงของการวัด ค่าเฉลี่ยในแต่ละครั้งจะมีค่าไม่เกิน 0.2 องศา

4.5.3 วิเคราะห์ผลการทดลองการวัดมุมเอียงโดยใช้เซ็นเซอร์ CMPS10

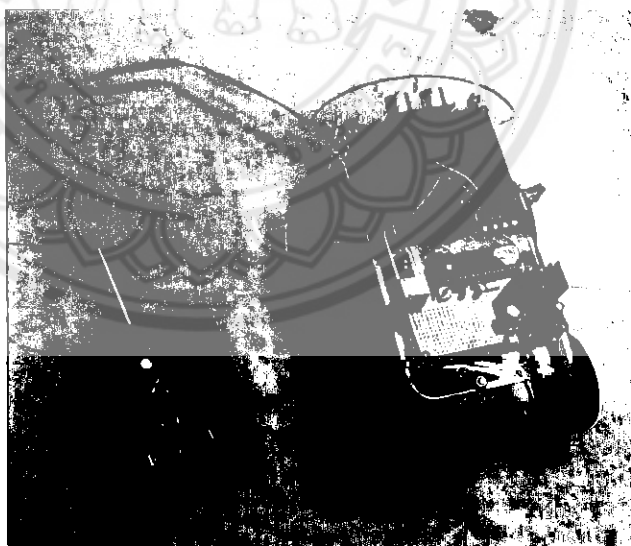
จากตารางที่ 4.5 รูปที่ 4.11 และ รูปที่ 4.12 นั้น จะพบว่าค่าที่ได้จาก CMPS10 นั้นมีเสถียรภาพค่อนข้างสูง กล่าวคือได้ค่าที่มีความคลาดเคลื่อนต่ำ และมีการกระจายตัวของข้อมูลน้อย แม้จะทำการวัดซ้ำหลายรอบแล้วก็ตาม

4.6 ผลการทดลองจากการส่งสัญญาณระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino กับ PIC

4.6.1 วิธีการทดลอง

การทดลองนี้จะใช้รีโมทควบคุมหุ่นยนต์ให้เคลื่อน(ดังแสดงในรูปที่ 4.13) ไปใน 4 ทิศทาง นั่นคือ เดินหน้า ถอยหลัง เลี้ยวซ้ายและเลี้ยวขวา เพื่อสังเกตการเปลี่ยนแปลงของค่าสัญญาณที่บอร์ด Arduino ส่งให้ บอร์ด PIC

สำหรับการเปลี่ยนแปลงที่เกิดขึ้น สามารถดูได้จากการใช้งานในส่วน of CMPS10 ในโปรแกรม เพราะเมื่อมีการส่งสัญญาณจากบอร์ด Arduino Uno R3 ส่งให้ บอร์ด PIC จะเป็นการเลือกการใช้งาน CMPS10 ในแต่ละช่วงของการเคลื่อนที่



รูปที่ 4.13 ระบบที่พัฒนาขึ้นเพื่อทำการทดสอบการสร้างแผนที่

4.6.2 ผลการทดลอง

เมื่อทำการเดินหน้าหรือถอยหลังดังในรูปที่ 4.14 จะมีการส่งสัญญาณจากบอร์ด Arduino ไปยัง PIC เพื่อให้ระบบที่พัฒนาขึ้นไม่นำข้อมูลของ CMPS10 มาใช้เพื่อเพิ่มความแม่นยำของข้อมูล เนื่องจากการเปลี่ยนแปลงเมื่อมีการเคลื่อนไหวในทิศทางตรงไม่จำเป็นต้องใช้งานข้อมูลดังกล่าว จะเห็นได้จากการแสดงการทำงานของ Status Compass เป็น null ในรูปที่ 4.15



รูปที่ 4.15 สถานะของ Compass เมื่อมีการเดินหน้าหรือถอยหลัง

อย่างไรก็ตามเมื่อเริ่มทำการเลี้ยวหุ่นยนต์คั้งในรูปที่ 4.16 จะมีส่งสัญญาณจากบอร์ด Arduino ไปยัง PIC เพื่อให้ให้นำข้อมูลที่รับเข้ามาใช้งานในส่วนของ CMPS10 เนื่องจากมีการเปลี่ยนแปลงของทิศทาง จะเห็นได้จากการแสดงการทำงานของ Status Compass เป็น Working ในรูปที่ 4.17



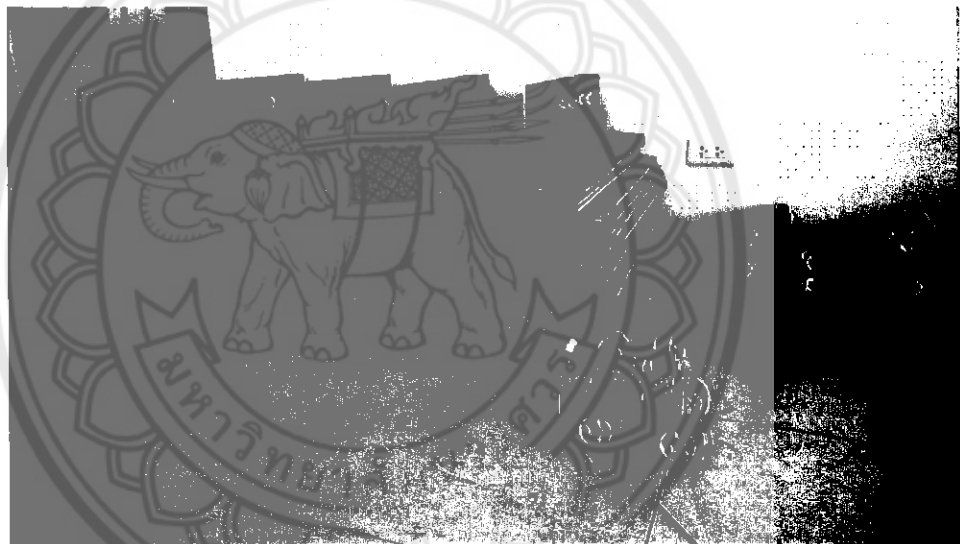
เนื่องจากต้องการเพิ่มความแม่นยำและต้องการสื่อสารตลอดเวลาจึงออกแบบให้บอร์ดทั้ง 2 รู้สถานะการเปลี่ยนแปลงของกันและกัน

4.7 ผลการทดลองการทำงานร่วมกันของอัลตราโซนิคเซ็นเซอร์ SRF05 กับโมดูลเข็มทิศ ดิจิตอลCMPS10

4.7.1 วิธีทำการทดลอง

การทดลองนี้จะเป็นการทดสอบประสิทธิภาพการทำงานของระบบ โดยการกำหนดลักษณะของสภาพแวดล้อมที่ต้องการเข้าไปสำรวจและสั่งให้ระบบจำลองแผนที่นั้นออกมา

สำหรับการทดลองนี้ จะจำลองแผนที่ที่จะทำการทดสอบโดยใช้แผ่นไม้อัดตั้งวางเป็นรูปทรงต่างๆ โดยเพิ่มระดับความท้าทายขึ้นเรื่อยๆ โดยในโครงการนี้จะทำการทดสอบทั้งหมด 5 รูปแบบ ได้แก่ รูปสามเหลี่ยม รูปสี่เหลี่ยม รูปแปดเหลี่ยม รูปสิบเอ็ดเหลี่ยม และรูปตัวแอล

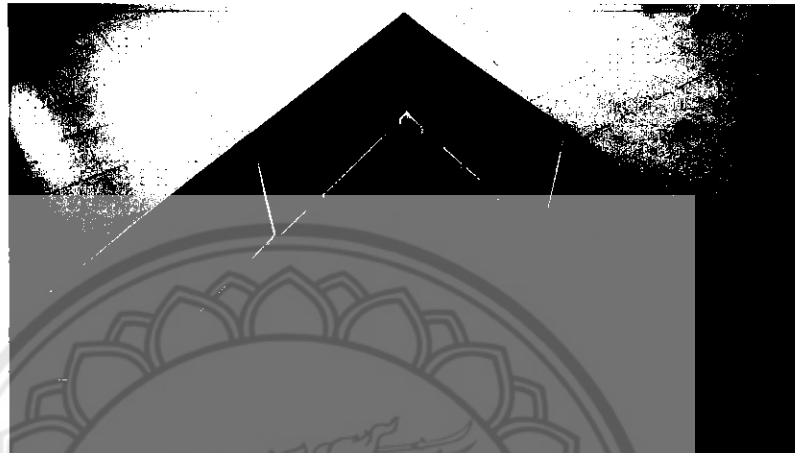


รูปที่ 4.18 อุปกรณ์ที่ใช้ในการทดลอง

จากรูปที่ 4.18 แสดงอุปกรณ์ที่ใช้ในการทดลอง ซึ่งเป็นไม้อัดขนาด 60 x 100 เซนติเมตร และฐานรองสำหรับตั้งไม้อัดพร้อมนี่อดสำหรับยึด

4.7.2 ผลการทดลอง

4.7.2.1 ผลการทดลองรูปสามเหลี่ยม

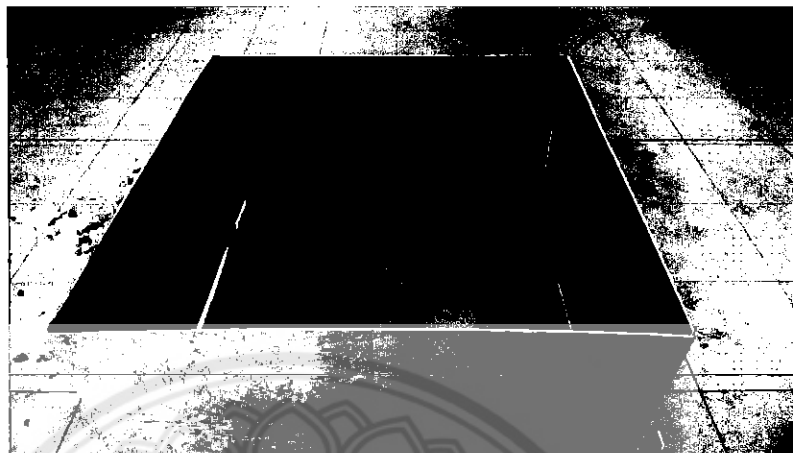


รูปที่ 4.19 การทดลองรูปสามเหลี่ยม

รูปที่ 4.20 ผลการจำลองรูปสามเหลี่ยม

จากรูปที่ 4.19 และ รูปที่ 4.20 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผนทีรูปสามเหลี่ยม โดยผลลัพธ์ที่ได้จะเกิดปัญหา Critical Angle ขึ้นบริเวณมุมทั้งสามของสามเหลี่ยม โดยมีสาเหตุมาจากมุมทุกมุมของรูปสามเหลี่ยมนั้นเป็นมุมแหลม ซึ่งทำให้เกิดการสะท้อนกันของคลื่นอัลตราโซนิก ส่งผลให้ระยะเวลาในการคำนวณหาระยะทางของอัลตราโซนิกเซ็นเซอร์นั้นเพิ่มขึ้น เป็นผลให้ได้ระยะทางที่มากเกินไปกว่าระยะทางจริง

4.7.2.2 ผลการทดลองรูปสี่เหลี่ยม

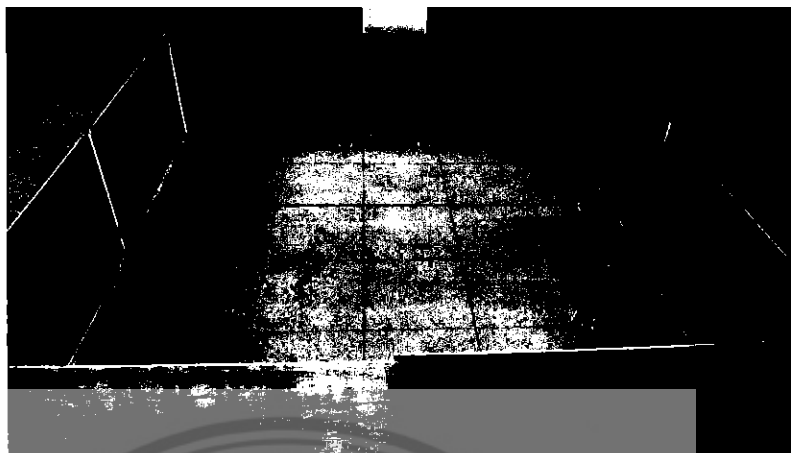


รูปที่ 4.21 การทดลองรูปสี่เหลี่ยมขนาด 1 x 1 แผ่น

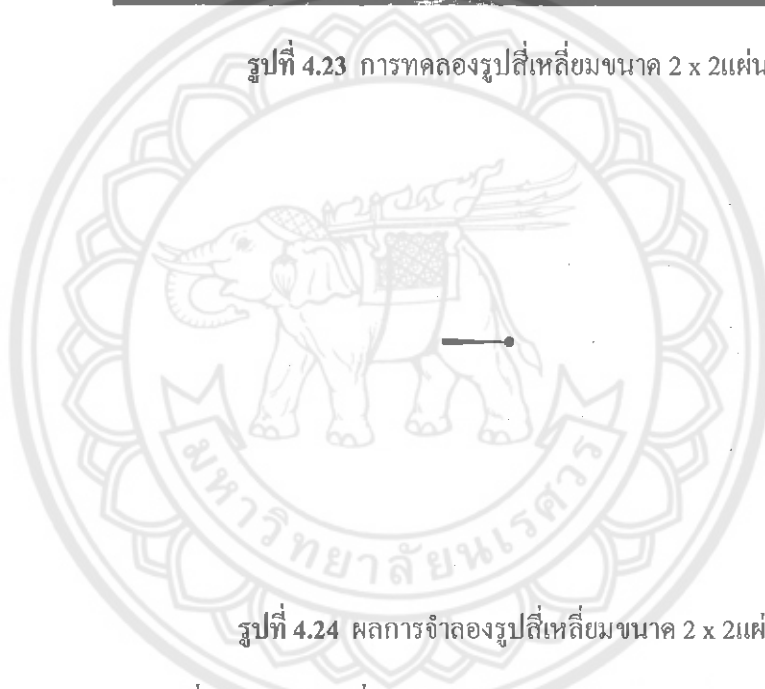


รูปที่ 4.22 ผลการจำลองรูปสี่เหลี่ยมขนาด 1 x 1 แผ่น

จากรูปที่ 4.21 และ รูปที่ 4.22 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผ่นที่รูปสี่เหลี่ยมขนาด 1 x 1 แผ่น พบว่าผลลัพธ์ที่ได้จากระบบนั้นอยู่ในเกณฑ์ดี กล่าวคือสามารถบอกรายละเอียดของแผ่นที่ได้อย่างครบถ้วน แต่ยังมีปัญหา Critical Angle ในบริเวณมุมทุกมุมของรูปสี่เหลี่ยมเพียงเล็กน้อย

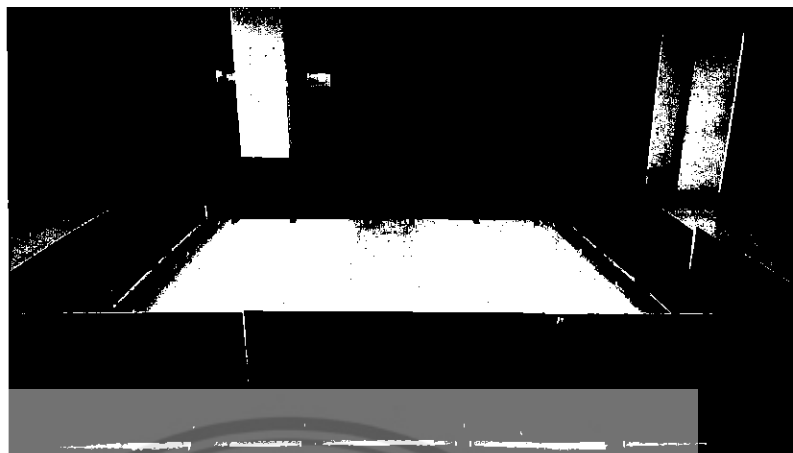


รูปที่ 4.23 การทดลองรูปสี่เหลี่ยมขนาด 2 x 2 แผ่น

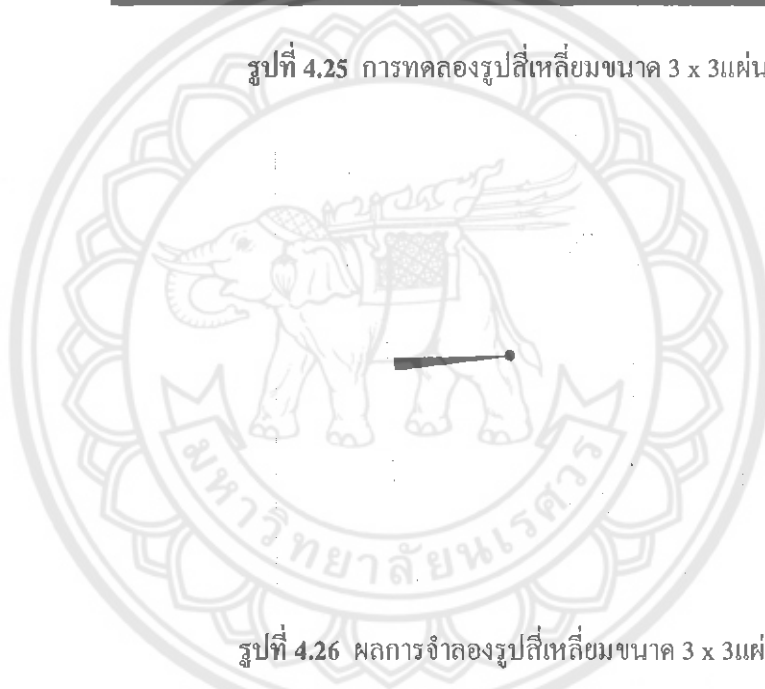


รูปที่ 4.24 ผลการจำลองรูปสี่เหลี่ยมขนาด 2 x 2 แผ่น

จากรูปที่ 4.23 และ รูปที่ 4.24 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผ่นที่รูปสี่เหลี่ยมขนาด 2 x 2 แผ่น พบว่าผลลัพธ์ที่ได้จากระบบนั้นอยู่ในเกณฑ์ดี และยังสามารถบอกรายละเอียดของแผ่นที่ได้อย่างครบถ้วน แต่พบปัญหา Critical Angle มากกว่าผลลัพธ์ของการจำลองแผ่นที่รูปสี่เหลี่ยมขนาด 1 x 1 แผ่น



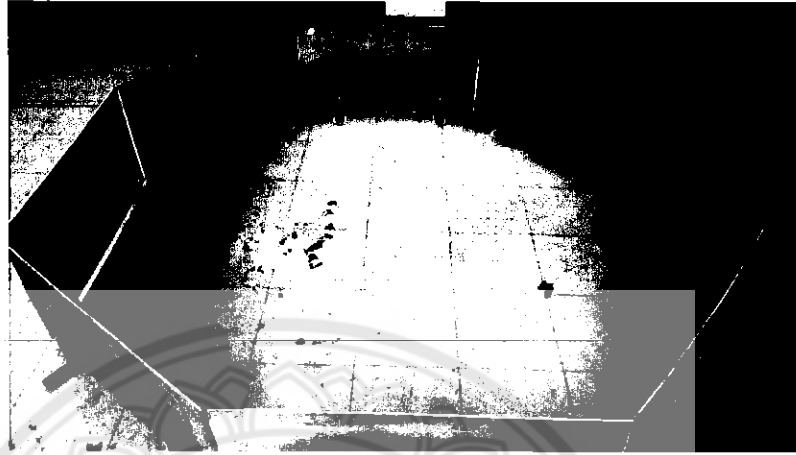
รูปที่ 4.25 การทดลองรูปสี่เหลี่ยมขนาด 3 x 3 แผ่น



รูปที่ 4.26 ผลการจำลองรูปสี่เหลี่ยมขนาด 3 x 3 แผ่น

จากรูปที่ 4.25 และ รูปที่ 4.26 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผนที่รูปสี่เหลี่ยมขนาด 3 x 3 แผ่นเมื่อพิจารณาผลลัพธ์ที่ได้ทั้งหมดจากการทดสอบรูปสี่เหลี่ยมขนาดต่างๆ สามารถสรุปได้ว่า ถ้าระยะทางจากตัวหุ่นยนต์เคลื่อนที่จนถึงพื้นผิวของวัตถุนั้นมีค่ามาก จะส่งผลต่อการเกิดปัญหา Critical Angle มากยิ่งขึ้นด้วย

4.7.2.3 ผลการทดลองรูปแปดเหลี่ยม



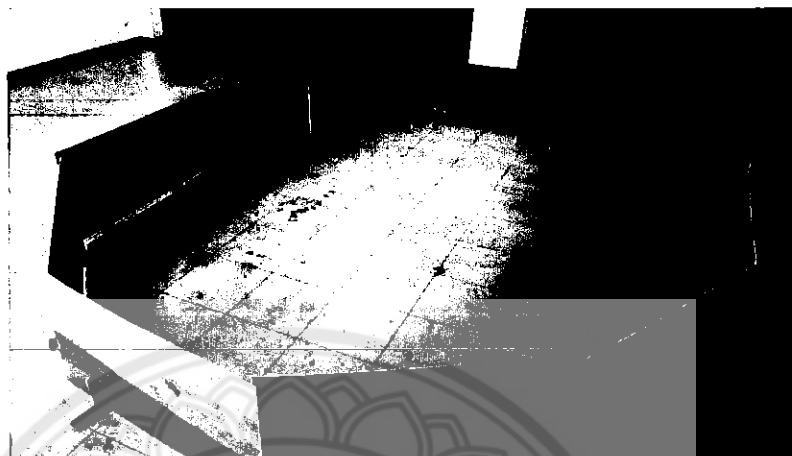
รูปที่ 4.27 การทดลองรูปแปดเหลี่ยม



รูปที่ 4.28 ผลการจำลองรูปแปดเหลี่ยม

จากรูปที่ 4.27 และ รูปที่ 4.28 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผนที่รูปแปดเหลี่ยม พบว่าผลลัพธ์ที่ได้สามารถบอกรายละเอียดของแผนที่ได้อย่างครบถ้วน และไม่เกิดปัญหา Critical Angle ขึ้นเลย เนื่องจากในแต่ละมุมของรูปแปดเหลี่ยมนั้นเป็นมุมป้าน จึงทำให้ไม่เกิดการสะท้อนระหว่างกันในแต่ละมุม

4.7.2.4 ผลการทดลองรูปสิบเอ็ดเหลี่ยม



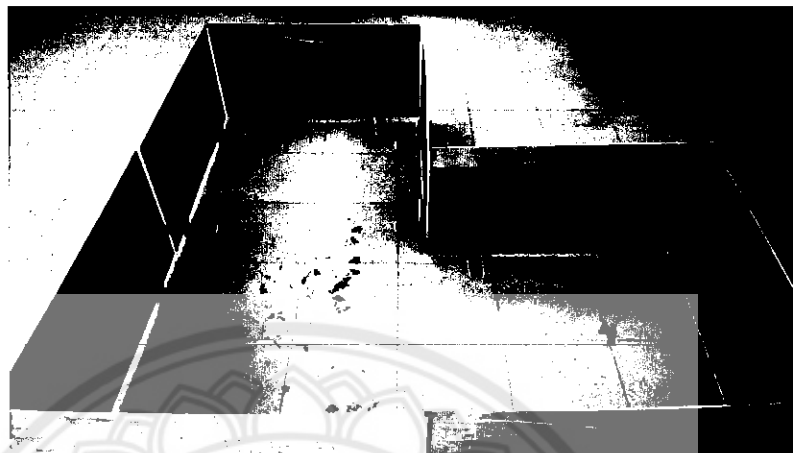
รูปที่ 4.29 การทดลองรูปสิบเอ็ดเหลี่ยม



รูปที่ 4.30 ผลการจำลองรูปสิบเอ็ดเหลี่ยม

จากรูปที่ 4.29 และ รูปที่ 4.30 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผนที่รูปสิบเอ็ดเหลี่ยม พบว่าผลลัพธ์ที่นั้นอยู่ในเกณฑ์พอใช้ กล่าวคือสามารถบอกรายละเอียดของแผนที่ที่จะลองได้บางส่วน แต่ไม่สามารถแยกมุมทั้งสิบเอ็ดมุมออกมาได้อย่างชัดเจน และไม่เกิดปัญหา Critical Angle เนื่องจากในแต่ละมุมของรูปสิบเอ็ดเหลี่ยมนั้นเป็นมุมป้าน จึงทำให้ไม่เกิดการสะท้อนระหว่างกันในแต่ละมุม

4.7.2.5 ผลการทดลองรูปตัวแอล



รูปที่ 4.31 การทดลองรูปตัวแอล



รูปที่ 4.32 ผลการจำลองรูปตัวแอล

จากรูปที่ 4.31 และ รูปที่ 4.32 แสดงการเปรียบเทียบระหว่างสถานที่การทดลองจริงกับการทำงานของระบบในการจำลองแผ่นที่รูปตัวแอล พบว่าผลลัพธ์จากการทดลองนี้จะเกิดปัญหา Critical Angle อย่างเห็นได้ชัด เนื่องจากในแต่ละมุมของสถานที่ทำการทดลองนั้นเป็นมุมฉาก ซึ่งทำให้เกิดการสะท้อนของคลื่นอัลตราโซนิกในแต่ละมุมได้

4.7.3 วิเคราะห์ผลการทดลองการทำงานร่วมกันของอัลตราโซนิกเซ็นเซอร์ SRF05 กับโมดูลเข็มทิศดิจิทัลCMPS10

เมื่อพิจารณาจากผลการทดลองการทำงานร่วมกันของอัลตราโซนิกเซ็นเซอร์กับ โมดูลเข็มทิศดิจิทัล ทั้ง 5 ผลการทดลองแล้วสามารถสรุปได้ เมื่ออัลตราโซนิกเซ็นเซอร์ทำงานในบริเวณที่เป็นมุมที่มีขนาด 0 องศา ถึง 90 องศา หรือในมุมแหลมจนถึงมุมฉาก จะทำให้เกิดปัญหา Critical Angle ขึ้น ส่วนถ้ามุมมีขนาดมากกว่า 90 องศาขึ้นไป หรือเป็นมุมป้าน จะไม่เกิดปัญหา Critical Angle

4.8 บทสรุป

ในบทนี้กล่าวถึงวิธีการทดลองและผลการทดลองของเซ็นเซอร์ทั้ง 2 ตัว คืออัลตราโซนิกเซ็นเซอร์ SRF05 และ โมดูลเข็มทิศดิจิทัลCMPS10 โดยทำการทดลองตามลำดับดังนี้ การทดลองการคำนวณหาระยะทางจากค่าที่ได้รับเซ็นเซอร์ ซึ่งจะพบว่า ระยะการทำงานของเซ็นเซอร์นั้นทำงานได้ตั้งแต่ 1 เซนติเมตร จนถึง 4 เมตร และยังพบว่าค่าคงตัวในการคำนวณแต่ละช่วงของระยะทางนั้นแตกต่างกันออกไป ต่อด้วย การทดลองการหามุมที่ทำให้เกิดปัญหา Critical Angle จากอัลตราโซนิกเซ็นเซอร์ SRF05 ซึ่งจะพบว่าขนาดของมุมที่เริ่มทำให้เกิดปัญหา Critical Angle นั้นคือมุมที่มีขนาดโดกว่า ± 40 องศา อีกทั้งยังทำการทดลองการวัดขนาดลำแสงของอัลตราโซนิกเซ็นเซอร์ SRF05 โดยได้ข้อสรุปเป็นรูปลักษณะของลำแสงนั้นเป็นทรงกรวยปลายแหลม และสุดท้ายได้ทำการทดลองการวัดมุมเอียงของเซ็นเซอร์ CMPS10 พบว่าการทำงานของ โมดูลนี้มีความเสถียรค่อนข้างมาก ในส่วนของหุ่นยนต์เคลื่อนที่ได้ทำการทดลองจากการส่งสัญญาณระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino กับ PIC พบว่ามีประสิทธิภาพการทำงาน ได้ดี และเมื่อนำทุกส่วนมากประกอบเข้าด้วยกัน เพื่อทำการทดลองการทำงานร่วมกันของอัลตราโซนิกเซ็นเซอร์ SRF05 กับ โมดูลเข็มทิศดิจิทัลCMPS10 ได้ผลลัพธ์ที่น่าพอใจ โดยแผนที่จำลองที่ระบบได้สร้างออกมานั้นยังมีปัญหาในความคิดพลาดเนื่องจากผลของ Critical Angle ทำให้แผนที่จำลองนั้นยังไม่ชัดเจนมากนัก

บทที่ 5

สรุปผลการทดลอง

โครงการพัฒนาระบบสร้างแผนที่โดยใช้อัลตราโซนิกเซ็นเซอร์ (DEVELOPMENT OF MAP BUILDING SYSTEM USING ULTRASONIC SENSOR) มีวัตถุประสงค์เพื่อพัฒนาต้นแบบสำหรับระบบสร้างแผนที่ในรูปแบบของข้อมูลระยะทางและทิศทางในขณะที่ทำการสำรวจอยู่ โดยข้อมูลที่วัดได้จะถูกส่งไปยังเครื่องคอมพิวเตอร์ เพื่อแสดงผลออกทางหน้าจอ GUI (Graphic User Interface) ที่ได้พัฒนาขึ้น

คณะผู้ดำเนินโครงการได้ออกแบบโดยใช้โปรแกรม Visual studio 2010 ในการพัฒนาโปรแกรม โดยเลือกใช้ภาษา C# เนื่องจากเป็นภาษาที่ง่ายต่อการศึกษาและมีประสิทธิภาพ

5.1 สรุปผลการทดลอง

จากผลการทดลองในบทที่ 4 นั้น สามารถแยกสรุปได้เป็น 2 ส่วน ดังนี้

1. อัลตราโซนิกเซ็นเซอร์ SRF05 มีความแม่นยำในการวัดสูง และค่าที่วัดได้นั้นมีเสถียรภาพสูง กล่าวคือมีความผิดพลาดของระยะทางที่คำนวณได้น้อยมาก อย่างไรก็ตามผลการทดลองแสดงให้เห็นถึงความผิดพลาดของระยะ ซึ่งเกิดมาจาก Critical Angle ส่งผลให้เกิดความคลาดเคลื่อนของระยะทางที่เซ็นเซอร์วัดได้ เมื่อเทียบกับระยะทางจริง
2. โมดูล CMPS10 มีความแม่นยำในการวัดสูง กล่าวคือมีการวัดค่าออกมาได้อย่างแม่นยำและตอบสนองต่อการเคลื่อนที่อย่างรวดเร็ว เนื่องจากตัวโมดูลดังกล่าวมีระบบชดเชยมุมที่เสียไป ทำให้ค่าที่วัดได้มีเสถียรภาพสูง อย่างไรก็ตามยังมีการแกว่งของสัญญาณในบางช่วง เนื่องจากโมดูลเข็มทิศดิจิทัล CMPS10 นั้นเป็นโมดูลที่ทำงานโดยการตรวจจับสนามแม่เหล็กโลก จึงอาจถูกคลื่นสัญญาณรบกวนจากสื่อการของอุปกรณ์หรือการทำงานของมอเตอร์ที่ใช้ขับเคลื่อนหุ่นยนต์ได้

5.2 วิจารณ์ผลการทดลอง

1. ความถูกต้องที่ได้จากอัลตราโซนิกเซ็นเซอร์SRF05 นั้นมีเสถียรภาพสูง แต่เนื่องจากการทำงานนั้นเป็นการส่งคลื่นอัลตราโซนิกออกไปกระทบกับวัตถุแล้วรอรับสัญญาณกลับเข้ามา จึงอาจทำให้เกิดปัญหา Critical Angle ขึ้น เมื่อมุมระหว่างอัลตราโซนิกกับวัตถุที่วัดนั้นมีขนาดโตกว่า ± 40 องศาเป็นต้นไป ซึ่งค่าที่วัดได้จะมากกว่าระยะทางจริงมาก จึงทำให้ระบบสร้างแผนที่ออกมาไม่เป็นไปตามสถานที่ที่ใช้ทำการทดสอบจริง
2. ความผิดพลาดของค่าที่ได้รับจากโมดูลเข็มทิศดิจิทัลCMPS10 นั้นเนื่องมาจาก โมดูลดังกล่าวนี้ทำงานโดยตรวจจับนามแม่เหล็กโลก แล้วจึงรับค่าสัญญาณจากตัวตรวจจับมาประมวลผลเป็นข้อมูลดิจิทัลสำหรับการแสดงผลทิศทาง ซึ่งสนามแม่เหล็กโลกอาจถูกรบกวนด้วยการสื่อสารกันของอุปกรณ์หรือมอเตอร์ที่ใช้ควบคุมการเคลื่อนที่ของหุ่นยนต์ อีกทั้งทางคณะผู้ดำเนินโครงการยังได้ทำการทดสอบภายในอาคาร ซึ่งอาจทำให้โมดูลดังกล่าวตรวจจับสนามแม่เหล็กโลกได้ไม่ดีเหมือนการทดสอบในพื้นที่โล่ง ทำให้ผลลัพธ์ที่ได้เกิดความคลาดเคลื่อน

5.3 การพัฒนาโครงการต่อในอนาคต

1. ค่าที่ได้จากอัลตราโซนิกเซ็นเซอร์ SRF05 ที่ใช้สำหรับวัดระยะทางนั้น ถึงแม้จะมีเสถียรภาพสูง แต่อาจเกิดปัญหา Critical Angle ตามมาเมื่อใช้งานในพื้นที่ที่มีลักษณะเฉพาะ ซึ่งสามารถแก้ไขได้โดยการเขียนซอฟต์แวร์ชดเชยค่าดังกล่าวซึ่งปัญหานี้สามารถเกิดกับเซ็นเซอร์ตัวอื่นๆ ได้เหมือนกันขึ้นอยู่กับลักษณะพื้นที่ที่ใช้งาน
2. แก้ปัญหา Critical Angle หรือมุมวิกฤต ซึ่งปัญหานี้มีกระบวนการทางคณิตศาสตร์ โดยนำทฤษฎีต่างๆ มาใช้แก้ไขปัญหา Critical Angle

5.4 ข้อเสนอแนะ

การพัฒนาโครงการต่อไปในอนาคตนั้นควรมีความรู้พื้นฐานในเรื่องต่อไปนี้

1. ศึกษาการแก้ปัญหา Critical Angle ทำการทดลองกับมุมที่เป็นปัญหา Critical Angle ของแผนที่ โดยทำการหมุนหุ่นยนต์เคลื่อนที่และอ่านค่าระยะทางและทิศทางที่ได้จากเซ็นเซอร์ในแต่ละองศา เพื่อให้รู้ถึงการเปลี่ยนแปลงของทิศทางและระยะทางก่อนที่จะเกิด Critical Angle และหลังจากเกิด Critical Angle นำผลการทดลองมา plot กราฟ
2. เขียนซอฟต์แวร์เพื่อการแก้ปัญหา Critical Angle เมื่อเจอกับมุมลักษณะต่างๆที่เกิดปัญหา Critical Angle แล้วชดเชยค่าดังกล่าวแทนระยะทางที่ได้จากอัลตราโซนิกเซ็นเซอร์
3. แก้ปัญหา Critical Angle โดยวิธีการ update ระยะทางมุมที่เกิดปัญหา Critical Angle เปรียบเทียบกับระยะทางในกรณีที่ว่าหุ่นยนต์เคลื่อนที่ทำมุมปกติกับวัตถุ เพื่อให้ได้ระยะทางที่แท้จริง

เอกสารอ้างอิง

- [1] B.T.C. **SURVEYING GROUP.** ประวัติของการสำรวจ. สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก <http://www.btcsurvey.com/proline/proline.php?lang=thai>
- [2] Wikipedia. **ยุคแห่งการสำรวจ.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก <http://th.wikipedia.org/w/index.php?title=%E0%B8%A2%E0%B8%B8%E0%B8%84%E0%B9%81%E0%B8%AB%E0%B9%88%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%AA%E0%B8%B3%E0%B8%A3%E0%B8%A7%E0%B8%88&oldid=4940254>
- [3] Wikipedia. **Proximity sensor.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก <http://th.wikipedia.org/w/index.php?title=%E0%B8%A2%E0%B8%B8%E0%B8%84%E0%B9%81%E0%B8%AB%E0%B9%88%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%AA%E0%B8%B3%E0%B8%A3%E0%B8%A7%E0%B8%88&oldid=4940254>
- [4] Wikipedia. **Ultrasonic sensor** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก http://en.wikipedia.org/wiki/Ultrasonic_sensor
- [5] Wikipedia. **Laser rangefinder.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก http://en.wikipedia.org/wiki/Laser_rangefinder

- [6] Wikipedia. **หุ่นยนต์.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก
<http://th.wikipedia.org/w/index.php?title=%E0%B8%AB%E0%B8%B8%E0%B9%88%E0%B8%99%E0%B8%A2%E0%B8%99%E0%B8%95%E0%B9%8C&oldid=5202354>

เอกสารอ้างอิง (ต่อ)

- [7] Jimme. **โรงงาน Honda.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก
<http://topicstock.pantip.com/ratchada/topicstock/2006/10/V4765922/V4765922.html>
- [8] Gordon Cheng. **Vision-based Mobile Robot Navigation.** สืบค้นเมื่อ
 6 พฤษภาคม 2556, จาก
http://infoeng.rsise.anu.edu.au/files/gordon_cheng_thesis.pdf
- [9] Dieter Fox. **Wolfram Bugard, Sebastain Thrun. Markov Localization for mobile robot in Dynamic Enviroments.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก
http://robotics.caltech.edu/~jerma/research_papers/MarkovLocalization.pdf
- [10] พันัส นัถฤทธิ. **Introduction to AutomousMoblle Robot.** สืบค้นเมื่อ
 6 พฤษภาคม 2556, จาก
http://web2.eng.nu.ac.th/nuej/file/journal/NUEJ_Vol6_2_2011_paper05.pdf
- [11] ชุทธนา สุทธสุภา. **Large Scale SLAM using Wide-Angle Camera.** สืบค้นเมื่อ
 6 พฤษภาคม 2556, จาก
<http://isl2.cp.eng.chula.ac.th/research-file/aun/doctor/aun-proposal.pdf>
- [12] Wikipedia. **Trigonometric functions.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก
http://en.wikipedia.org/wiki/Trigonometric_functions

- [13] Wikipedia. **อนุกรม.** สืบค้นเมื่อ 6 พฤษภาคม 2556, จาก
<http://th.wikipedia.org/wiki/%E0%B8%AD%E0%B8%99%E0%B8%B8%E0%B8%81%E0%B8%A3%E0%B8%A1>

เอกสารอ้างอิง (ต่อ)

- [14] Wikipedia. **สมการเชิงอนุพันธ์.** สืบค้นเมื่อ 8 พฤษภาคม 2556, จาก
<http://th.wikipedia.org/wiki/%E0%B8%AA%E0%B8%A1%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%8A%E0%B8%B4%E0%B8%87%E0%B8%AD%E0%B8%99%E0%B8%B8%E0%B8%9E%E0%B8%B1%E0%B8%99%E0%B8%98%E0%B9%8C>
- [15] Wikipedia. **รูปสามเหลี่ยม.** สืบค้นเมื่อ 8 พฤษภาคม 2556, จาก
<http://th.wikipedia.org/wiki/%E0%B8%A3%E0%B8%B9%E0%B8%9B%E0%B8%AA%E0%B8%B2%E0%B8%A1%E0%B9%80%E0%B8%AB%E0%B8%A5%E0%B8%B5%E0%B9%88%E0%B8%A2%E0%B8%A1>
- [16] Wikipedia. **Graphical user interface.** สืบค้นเมื่อ 2 มิถุนายน 2556, จาก
http://en.wikipedia.org/wiki/Graphical_user_interface
- [17] สภาวิศวกร. **AutoCAD.** สืบค้นเมื่อ 15 มิถุนายน 2556, จาก
http://www.coc.or.th/e_engineers/knc_detail.php?id=17
- [18] Wikipedia. **DougEngelbart.** สืบค้นเมื่อ 17 มิถุนายน 2556, จาก
http://en.wikipedia.org/wiki/Doug_Engelbart
- [19] Wikipedia. **PARC(company).** สืบค้นเมื่อ 19 มิถุนายน 2556, จาก
http://en.wikipedia.org/wiki/PARC_User_Interface#The_GUI
- [20] Wikipedia. **Apple Inc.** สืบค้นเมื่อ 7 กรกฎาคม 2556, จาก

- http://en.wikipedia.org/wiki/Apple_computer
- [21] Wikipedia. **Microsoft.** สืบค้นเมื่อ 20 กรกฎาคม 2556, จาก
<http://en.wikipedia.org/wiki/Microsoft>

เอกสารอ้างอิง (ต่อ)

- [22] Wikipedia. **Linux.** สืบค้นเมื่อ 29 กรกฎาคม 2556, จาก
<http://en.wikipedia.org/wiki/Linux>
- [23] ETTeam. **CP-PICV3/877 (ICD2).** สืบค้นเมื่อ 4 สิงหาคม 2556, จาก
<http://www.etteam.com/product/pic/cp-pic-v3-877-icd2.html>
- [24] Innovative Experiment. **SRF05 .** สืบค้นเมื่อ 19 สิงหาคม 2556, จาก
<http://www.es.co.th/schemetic/pdf/NX-SFR05.pdf>
- [25] Wikipedia. **การสะท้อนกลับทั้งหมด.** สืบค้นเมื่อ 19 สิงหาคม 2556, จาก
<http://th.wikipedia.org/w/index.php?title=%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%AA%E0%B8%B0%E0%B8%97%E0%B9%89%E0%B8%AD%E0%B8%99%E0%B8%81%E0%B8%A5%E0%B8%B1%E0%B8%9A%E0%B8%97%E0%B8%B1%E0%B9%89%E0%B8%87%E0%B8%AB%E0%B8%A1%E0%B8%94&oldid=4926831>
- [26] Wikipedia. **Snell's law.** สืบค้นเมื่อ 19 สิงหาคม 2556, จาก
http://en.wikipedia.org/w/index.php?title=Snell%27s_law&oldid=584870840
- [27] INEX. **CMPS10 .** สืบค้นเมื่อ 8 กันยายน 2556, จาก
<http://doc.inex.co.th/cmeps10-tilt-compensated-compass-module>
- [28] Wikipedia. **I²C.** สืบค้นเมื่อ 10 กันยายน 2556, จาก
<http://en.wikipedia.org/wiki/I%C2%B2C>
- [29] Wikipedia. **Serial communication.** สืบค้นเมื่อ 23 กันยายน 2556, จาก

http://en.wikipedia.org/wiki/Serial_communication

- [30] Wikipedia. **Pulse-width modulation.** สืบค้นเมื่อ 23 กันยายน 2556, จาก
http://en.wikipedia.org/wiki/Pulse-width_modulation

เอกสารอ้างอิง (ต่อ)

- [31] Wikipedia. **Synchronous.** สืบค้นเมื่อ 23 กันยายน 2556, จาก
<http://en.wikipedia.org/wiki/Synchronous>
- [32] Wikipedia. **Microcontroller.** สืบค้นเมื่อ 30 กันยายน 2556, จาก
<http://en.wikipedia.org/wiki/Microcontroller>
- [33] Wikipedia. **NXP Semiconductors.** สืบค้นเมื่อ 9 ตุลาคม 2556, จาก
http://en.wikipedia.org/wiki/Philips_Semiconductors
- [34] Arduino. **Arduino Uno.** สืบค้นเมื่อ 9 ตุลาคม 2556, จาก
<http://arduino.cc/en/Main/ArduinoBoardUno>
- [35] Arduino. **Pin diagram.** สืบค้นเมื่อ 9 ตุลาคม 2556, จาก
<http://arduino.cc/en/Hacking/PinMapping168>
- [36] Resistorguide. **Wire Wound Resistor.** สืบค้นเมื่อ 21 ตุลาคม 2556, จาก
<http://www.resistorguide.com/wirewound-resistor>
- [37] Arduino. **Adafruit Motor Shield.** สืบค้นเมื่อ 21 ตุลาคม 2556, จาก
<http://playground.arduino.cc/Main/AdafruitMotorShield>
- [38] Sparkfun. **L298P.** สืบค้นเมื่อ 1 พฤศจิกายน 2556, จาก
https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
- [39] นกแสบ. **วิทยุบังคับ.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://www.vcharkarn.com/vcafe/186885>

- [40] Wikipedia. **C Sharp (programming language).** สืบค้นเมื่อ
5 พฤศจิกายน 2556, จาก
[http://en.wikipedia.org/w/index.php?title=C_Sharp_\(programming_language\)&oldid=586609967](http://en.wikipedia.org/w/index.php?title=C_Sharp_(programming_language)&oldid=586609967)

เอกสารอ้างอิง (ต่อ)

- [41] Wikipedia. **การเขียนโปรแกรมเชิงวัตถุ.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://th.wikipedia.org/w/index.php?title=%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%80%E0%B8%82%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B9%82%E0%B8%9B%E0%B8%A3%E0%B9%81%E0%B8%81%E0%B8%A3%E0%B8%A1%E0%B9%80%E0%B8%8A%E0%B8%B4%E0%B8%87%E0%B8%A7%E0%B8%B1%E0%B8%95%E0%B8%96%E0%B8%B8&oldid=5203531>
- [42] Wikipedia. **C.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/w/index.php?title=C%2B%2B&oldid=582491572>
- [43] Wikipedia. **Visual Basic.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Visual_Basic&oldid=586537696
- [44] Wikipedia. **Java.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/w/index.php?title=Java&oldid=585389335>
- [45] Wikipedia. **XML.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/w/index.php?title=XML&oldid=586460807>
- [46] Wikipedia. **SOAP.** สืบค้นเมื่อ 5 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/w/index.php?title=SOAP&oldid=586382124>

- [47] Wikipedia. **Artificial intelligence.** สืบค้นเมื่อ 15 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Artificial_intelligence&oldid=586638636
- [48] Wikipedia. **Image processing.** สืบค้นเมื่อ 15 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Image_processing&oldid=585134403

เอกสารอ้างอิง (ต่อ)

- [49] Wikipedia. **Artificial neural network.** สืบค้นเมื่อ 15 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=586638699
- [50] Wikipedia. **Genetic algorithm.** สืบค้นเมื่อ 15 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Genetic_algorithm&oldid=585005163
- [51] Wikipedia. **Machine learning.** สืบค้นเมื่อ 15 พฤศจิกายน 2556, จาก
http://en.wikipedia.org/w/index.php?title=Machine_learning&oldid=586518404
- [52] Wikipedia. **TICON.** สืบค้นเมื่อ 22 พฤศจิกายน 2556, จาก
<http://www.mikroe.com/chapters/view/5/>
- [53] Wikipedia. **Baud Rate.** สืบค้นเมื่อ 22 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/wiki/Baud>
- [54] Dotnetperls. **ArrayList.** สืบค้นเมื่อ 25 พฤศจิกายน 2556, จาก
<http://www.dotnetperls.com/arraylist>
- [55] Wikipedia. **Flowchart.** สืบค้นเมื่อ 26 พฤศจิกายน 2556, จาก
<http://en.wikipedia.org/wiki/Flowchart>

ภาคผนวก

ก. ขั้นตอนการติดตั้งโปรแกรม Microsoft VisualC# Studio 2010 Express

1. เข้าไปที่เว็บไซต์ <http://www.visualstudio.com/en-us/downloads#d-2010-express>
คลิก Install now เพื่อทำการดาวน์โหลด



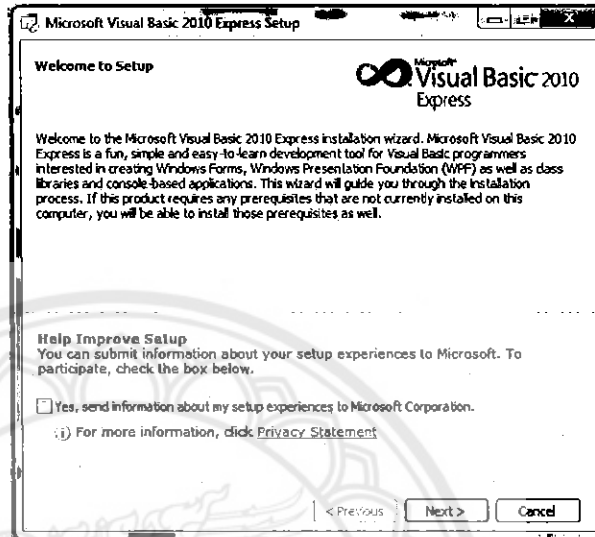
รูปที่ ก.1 ดาวน์โหลด Microsoft VisualC# Studio 2010 Express

2. หลังการดาวน์โหลดสำเร็จ จะได้ไฟล์ vcs_web.exe ให้ทำการดับเบิลคลิกเพื่อที่จะเริ่มทำการดาวน์โหลดและติดตั้งโปรแกรม



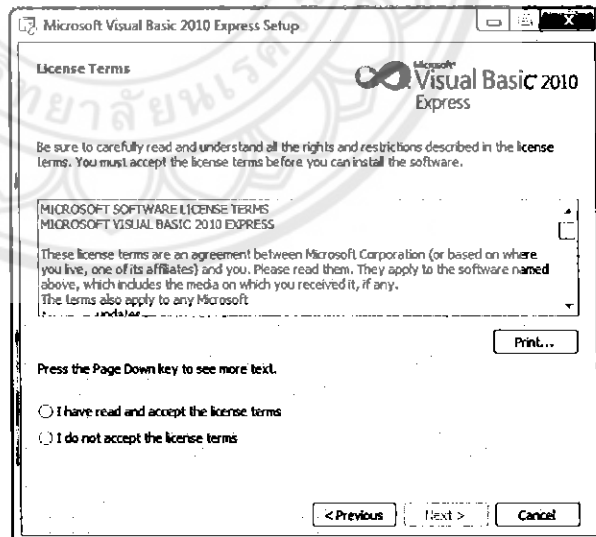
รูปที่ ก.2 VS2010Express1.iso

3. หลังจากนั้นจะมีหน้าต่างเริ่มการติดตั้งแสดงขึ้นมา กดปุ่ม Next



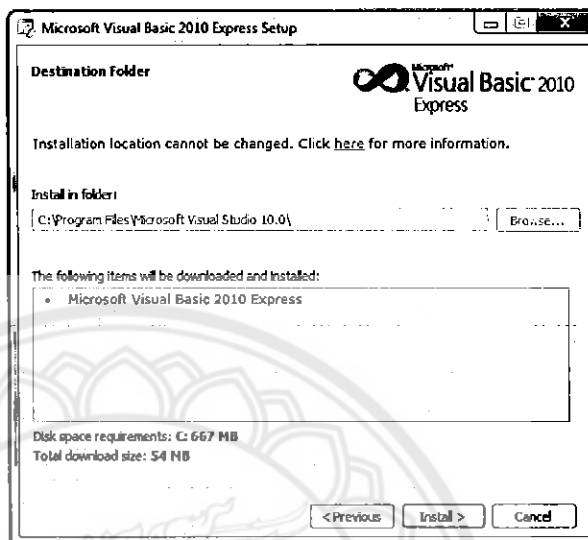
รูปที่ ก.3 หน้าต่างแสดงการเริ่มติดตั้ง โปรแกรม

4. ทำการอ่านเงื่อนไขในการติดตั้ง กดยอมรับแล้วกดปุ่ม Next



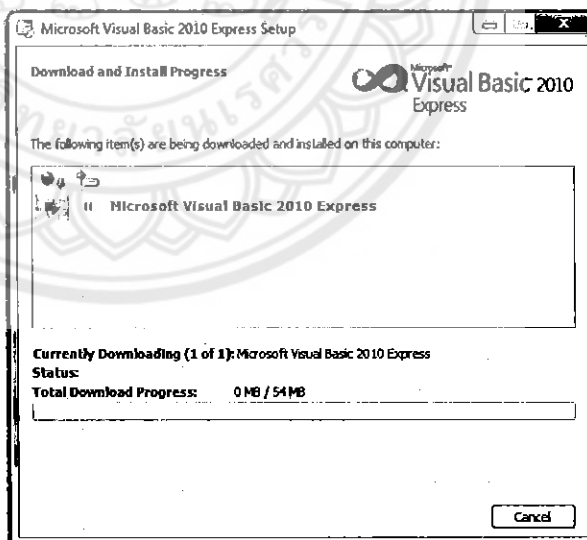
รูปที่ ก.4 หน้าต่างแสดงเงื่อนไขในการติดตั้งโปรแกรม

5. ทำการเลือก Directory ในการติดตั้งโปรแกรมแล้วกลุ่ม Install



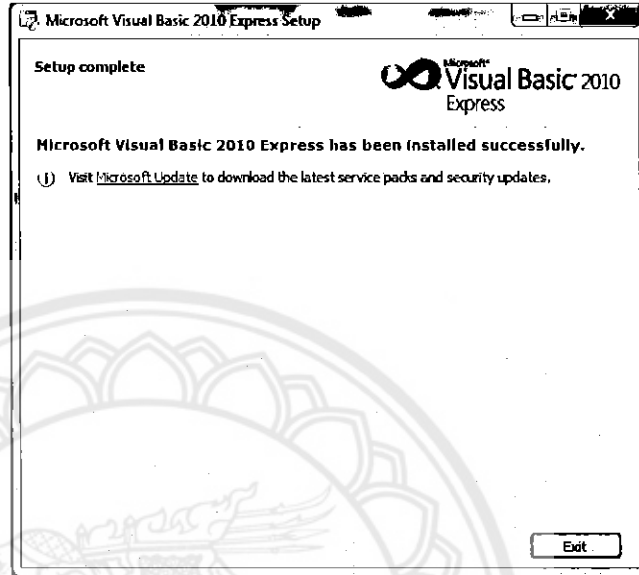
รูปที่ ก.5 เลือก Directory ในการติดตั้ง โปรแกรม

6. คลิก Install ใช้เนื้อที่ 3.5G



รูปที่ ก.6 เลือก โปรแกรมที่ต้องการติดตั้ง

7. โปรแกรมจะทำการติดตั้งจนเสร็จสมบูรณ์ ดังรูปที่ ก.7



รูปที่ ก.7 สิ้นสุดการติดตั้งโปรแกรม

ข. ขั้นตอนการติดตั้งโปรแกรม MicroC PRO for PIC

1. การติดตั้งโปรแกรม MicroC PRO for PIC สามารถดาวน์โหลดโปรแกรมได้ที่ <http://www.mikroe.com/>
2. ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมาดังรูป

Name

(๖) mikroC_PRO_PIC_2011_Build.5.01

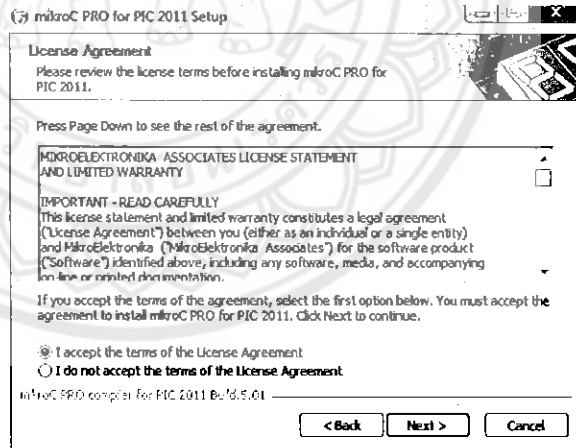
รูปที่ ข.1 การติดตั้ง MicroC PRO for PIC

3. หลังจากนั้นจะมีหน้าต่างเริ่มการติดตั้งแสดงขึ้นมา กดปุ่ม Next



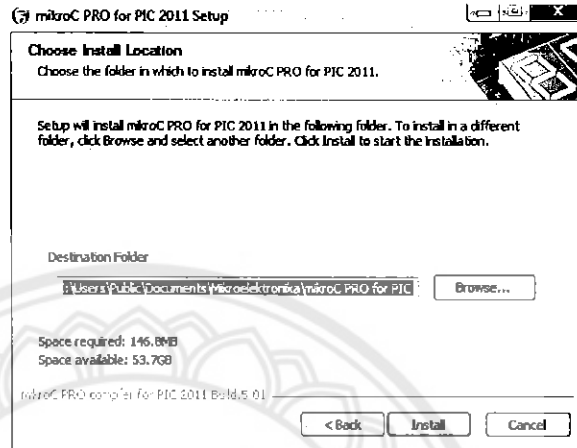
รูปที่ ข.2 หน้าต่างแสดงการเริ่มติดตั้ง โปรแกรม

4. อ่านเงื่อนไขในการติดตั้ง พร้อมทั้งกดยอมรับเงื่อนไขและกดปุ่ม Next



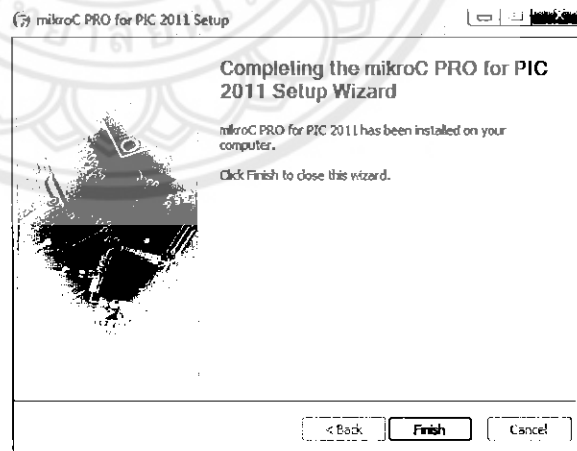
รูปที่ ข.3 หน้าต่างแสดงเงื่อนไขในการติดตั้งโปรแกรม

5. เมื่อกดปุ่ม Next มาเรื่อยๆ จนถึงหน้าต่างสำหรับเลือก Directory ให้ทำการเลือก Directory ในการติดตั้งโปรแกรมแล้วกดปุ่ม Install



รูปที่ ข.4 หน้าต่างเลือก Directory ที่ต้องการสำหรับติดตั้งโปรแกรม

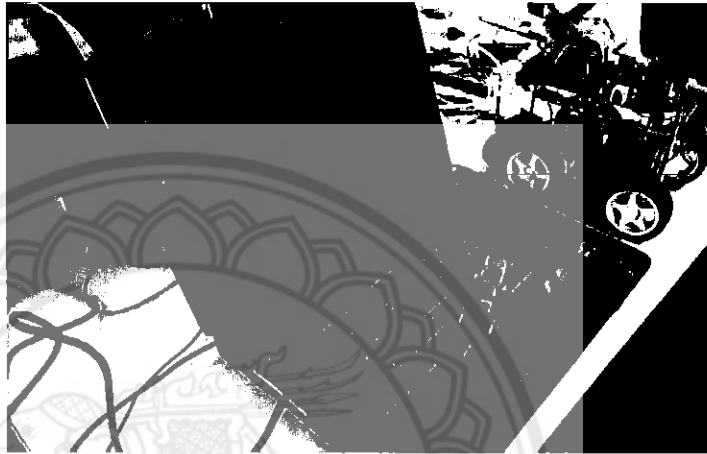
6. โปรแกรมจะทำการติดตั้งจนเสร็จสมบูรณ์ เมื่อ โปรแกรมติดตั้งสำเร็จจะได้หน้าต่าง ดังรูปที่ ข.5



รูปที่ ข.5 หน้าต่างแสดงการติดตั้งโปรแกรมเสร็จสมบูรณ์

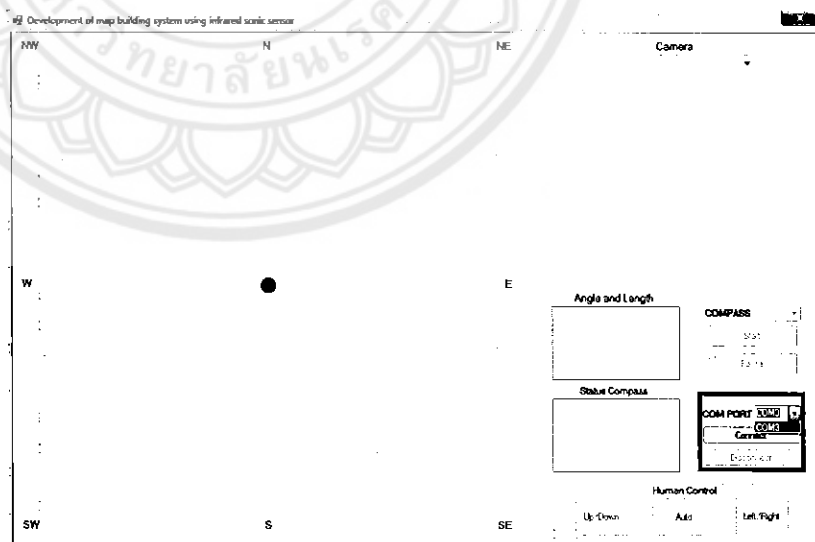
ค. คู่มือการใช้งานโปรแกรมสำหรับวาดแผนที่

1. เสียบ PORT RS232 เข้ากับบอร์ด PIC V3.0 16F877 และเสียบ PORT USB เข้ากับเครื่องคอมพิวเตอร์ดังรูป ค.1



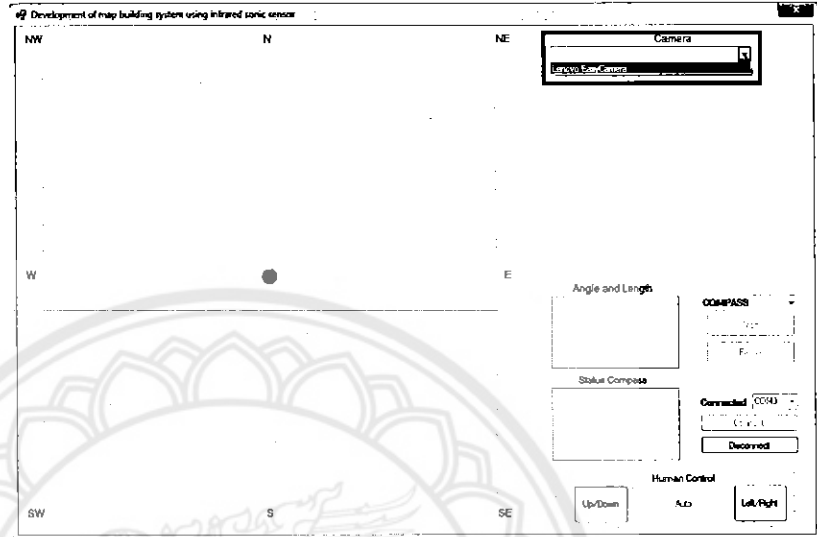
รูปที่ ค.1 การต่อ COMPORT

2. เปิดโปรแกรมขึ้นมา คลิกเลือก COMPORT ที่ได้ต่อเข้ากับคอมพิวเตอร์ แล้วกด Connect เพื่อให้สามารถติดต่อกับบอร์ดไมโครคอนโทรลเลอร์ได้



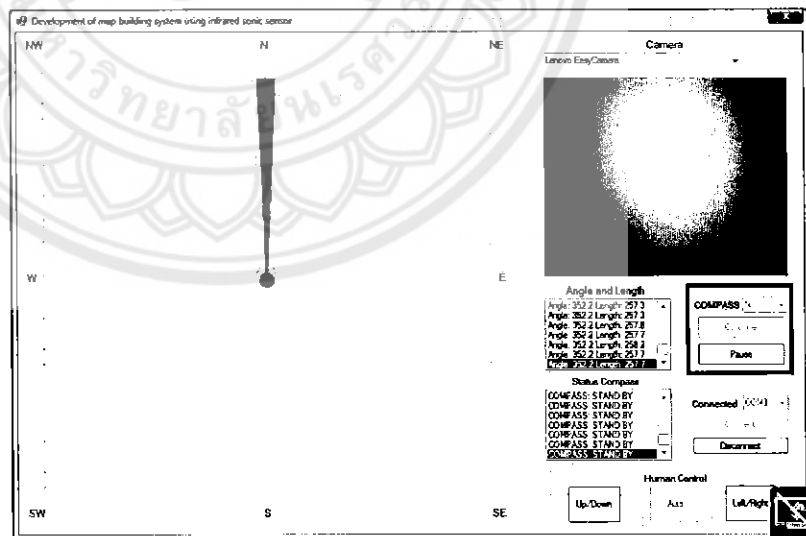
รูปที่ ค.2 วิธีการเลือก COMPORT

3. หากมีการติดตั้งกล้องเพื่อใช้เปรียบเทียบแผนที่จริงกับแผนที่ที่ระบบจำลองขึ้นมา ให้ทำการเลือกกล้องที่ Select Webcam



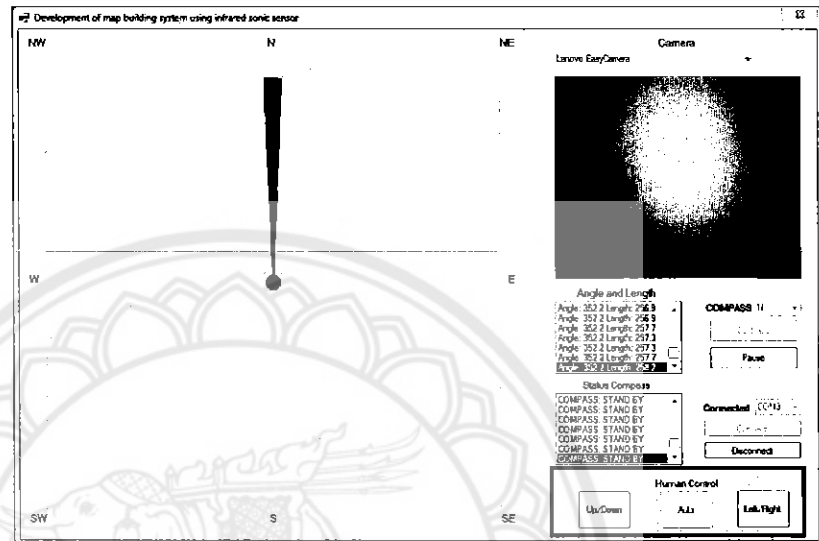
รูปที่ ค.3 วิธีการเลือก Webcam

4. ทำการเลือกทิศที่ต้องการสร้างแผนที่ แล้วกดปุ่ม Start บอร์ด ไมโครคอนโทรลเลอร์จะเริ่มทำการส่งข้อมูลทั้งระยะทางและทิศทางเข้ามายังโปรแกรม



รูปที่ ค.4 เริ่มต้นการทำงานของระบบ

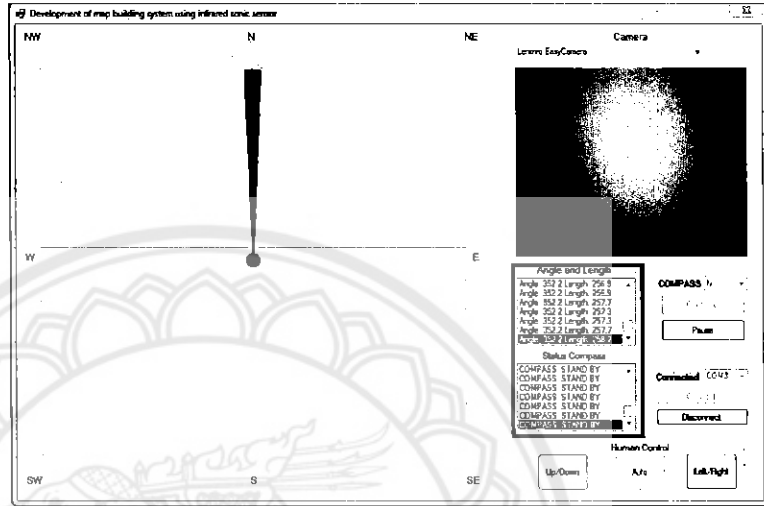
5. ทำการเลือกโหมดในแถบ Human Control เพื่อให้การทำงานของโปรแกรมมีเสถียรภาพมากขึ้น โดยแบ่งได้ดังนี้



รูปที่ ค.5 การใช้งานโหมดต่างๆ

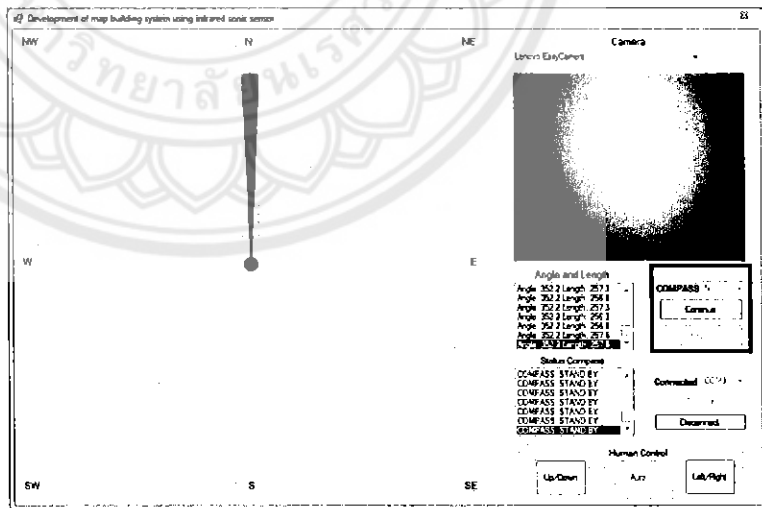
- 5.1 Up/Down การทำงานของโหมดนี้จะมีการทำงานเฉพาะกรณีหุ่นยนต์เดินหน้าหรือถอยหลังเท่านั้น
- 5.2 Auto การทำงานของโหมดนี้จะเปิดให้การควบคุมหุ่นยนต์ได้อย่างอิสระ
- 5.3 Left/Right การทำงานของโหมดนี้จะมีการทำงานเฉพาะหุ่นยนต์หมุนอยู่กับที่เท่านั้น

- 6. ผู้ใช้สามารถตรวจสอบข้อมูลที่นำมาใช้ในการจำลองแผนที่ได้ โดยดูจากกล่องข้อความ Angle and Length และตรวจสอบสถานการณ์ทำงานของ โมดูลเข็มทิศดิจิทัล ได้จากกล่องข้อความ Status Compass โดยสามารถย้อนกลับไปดูได้ถึงค่าแรกที่รับเข้ามา



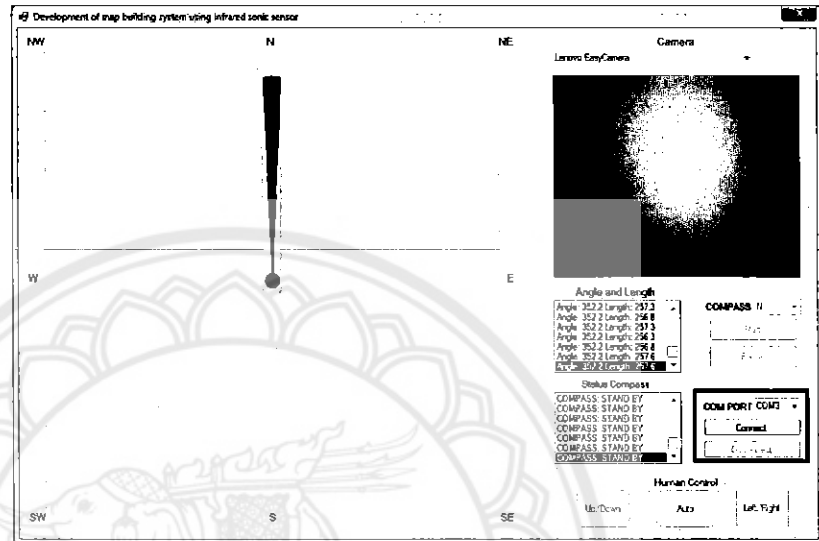
รูปที่ ค.6 ตรวจสอบข้อมูล

- 7. สามารถกดปุ่ม Pause เพื่อหยุดการทำงานและสามารถกลับมาทำงานต่อได้ โดยกดปุ่ม Continue



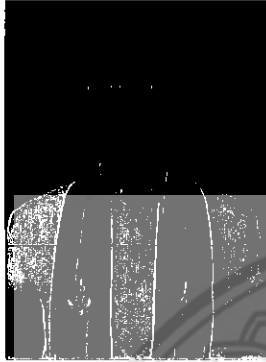
รูปที่ ค.7 หยุดการทำงานชั่วคราว

8. เมื่อต้องการจบการทำงานของระบบ สามารถยกเลิกการเชื่อมต่อได้โดยการกดปุ่ม Disconnect ก็จะทำการยกเลิกการเชื่อมต่อกันระหว่าง โปรแกรมกับบอร์ด ไมโครคอนโทรลเลอร์



รูปที่ 8 จบการทำงาน

ประวัติคณะผู้ดำเนินโครงการ



ชื่อ นายตองกบณ์ พรมโชติ
ภูมิลำเนา บ้านเลขที่ 340/2 หมู่ 10 ตำบลบึงพระ อำเภอเมือง
จังหวัดพิษณุโลก 65000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email: atonkub@gmail.com



ชื่อ นายปานพงษ์ กิตติสุภลักษณ์
ภูมิลำเนา บ้านเลขที่ 22/12 ซอย 1 อำเภอตะพานหิน
จังหวัดพิจิตร 66110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนโพธิ์สารศึกษานครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email: panupongkitti@gmail.com

ประวัติคณะผู้ดำเนินโครงการ (ต่อ)



ชื่อ นายภัทรบุษ ประมานนท์
ภูมิลำเนา 281/25 หมู่ 5 ตำบลหัวรอ อำเภอเมือง
จังหวัดพิษณุโลก 65000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email: benphoenix_x@hotmail.com