



การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
(The Control of an Omnidirectional Wheel Robot)

นายอุดมศักดิ์	สมภักดี
นายณัฐชัย	ชุ่มประดิษฐ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ ภาควิชาครุศาสตร์เครื่องกล
คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2556

การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
(The Control of an Omnidirectional Wheel Robot)

นายอุดมศักดิ์	สมภักดี
นายณัฐชัย	ชุ่มประดิษฐ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ ภาควิชาครุศาสตร์เครื่องกล
คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2556



ใบรับรองปริญญาานิพนธ์
ภาควิชาครุศาสตร์เครื่องกล คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
(The Control of an Omnidirectional Wheel Mobile Robot)
โดย นายอุดมศักดิ์ สมภักดี
นายณัฐชัย ชุ่มประดิษฐ์

โครงการนี้ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

.....หัวหน้าภาควิชาครุศาสตร์เครื่องกล
(ผศ. ดร.บัณฑิต สุขสวัสดิ์)

คณะกรรมการสอบปริญญาานิพนธ์

.....อาจารย์ที่ปรึกษา
(ผศ. ดร. อนันต์ สืบสำราญ)

.....ประธานกรรมการกรรมการ
(ผศ. ดร.จิระศักดิ์ วิตตะ) (อาจารย์ ดร.อภัยวงศ์ จันทร์ช่างพูด)

.....กรรมการกรรมการ
(ผศ. ดร.พรจิต ประทุมสุวรรณ) (อาจารย์ ลันติ หุตะมาน)

ชื่อปริญญาบัตร	: การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
ชื่อ	: นายอุดมศักดิ์ สมภักดี นายณัฐชัย ชุ่มประดิษฐ์
อาจารย์ที่ปรึกษาปริญญาบัตร	: ผศ.ดร.อนันต์ สืบสำราญ
สาขาวิชา	: วิศวกรรมแมคคาทรอนิกส์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา	: 2556

บทคัดย่อ

ปริญญาบัตรนี้นำเสนอ การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางพัฒนา ซึ่งประกอบไปด้วย 3 ส่วนคือ 1. การออกแบบหุ่นยนต์ 2. การพัฒนาโปรแกรม C# 3. พัฒนาโปรแกรม Android เพื่อควบคุมหุ่นยนต์ โดยที่การทำงานจะแบ่งออกเป็น 2 ระบบคือ การเคลื่อนที่แบบบุคคลเป็นผู้บังคับและแบบอัตโนมัติ ในการเคลื่อนที่แบบบุคคลเป็นผู้บังคับจะเป็นการเคลื่อนที่แบบอิสระตามที่ผู้บังคับต้องการที่จะให้ไป และการควบคุมแบบอัตโนมัติจะเป็นการควบคุมโดยสั่งเป็นแบบกำหนดรัศมีกับมุม โดยที่ระบบอัตโนมัติจะมีระบบหลักสิ่งกีดขวาง ซึ่งจะใช้โปรแกรม C# และโปรแกรม Android เป็นตัวแสดงผลแล้วจึงส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ โดยผ่านระบบไร้สาย หลังจากนั้น ไมโครคอนโทรลเลอร์ จะทำการควบคุมการทำงานของหุ่นยนต์โดยจะมีสัญญาณป้อนกลับทั้งหมด 2 ส่วนด้วยกัน คือเอ็นโค้ดเดอร์ (Encoder) ใช้ควบคุมการเคลื่อนที่กับระยะทาง และอัลตราโซนิก (Ultrasonic) จะใช้เป็นส่วนที่หลักสิ่งกีดขวาง โดยที่ระบบจะควบคุมแบบพีซี และ ทำการทดสอบการเคลื่อนที่ ในพื้นที่ที่มีแรงเสียดทานต่ำ กับพื้นที่ที่มีแรงเสียดทานสูงจากการทดลอง

ผลการทดลองที่ได้จากการทดสอบ พบว่า สามารถควบคุมให้หุ่นยนต์เคลื่อนที่หลายทิศทางนั้นสามารถเคลื่อนที่ไปยังตำแหน่งที่กำหนด และสามารถหลบสิ่งกีดขวางได้โดยมีความผิดพลาดจากระยะที่ต้องการไม่เกิน 10%

Project Report Title : The Control of an Omnidirectional Wheel Robot
Name : Mr. Udomsak Sompakdee
: Mr. Nutthachai Chumpradith
Project Report Advisor : Asst. Prof. Dr. Anan Suebsomran
Department of : Teacher Training in Mechanical Engineering
Academic Year : 2014

Abstract

This project is the control of an omnidirectional wheel robot which consist of 3 element: Robot, C#, and Android development with 2 operating mode: manual mode and autonomous mode. In normal mode, user will be able to control position of a robot to their wish. While autonomous mode, user will need to enter both distance and angle to command a robot. After this, robot will start to move according to user input while trying to avoid any obstacle that block its path. User can use C# or Android Program as display and communicator which will send data to Arduino microcontroller via wireless LAN. When data at microcontroller is retrieved, it will control the transmission of a robot and send back 2 feedback signal: encoder signal – used in distance control and ultrasonic signal – used in obstacle avoidance. This robot used fuzzy control system to control overall direction of a robot and tested on area with high friction.

The result of this project is, Omnidirectional wheel robot is be able to move to user-specific position and avoid obstacle along the way with 10% error.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลงได้โดยได้รับความช่วยเหลืออย่างดีจาก ผศ.ดร. อนันต์ สิบสำราญ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ให้คำแนะนำและข้อคิดต่าง ๆ มาโดยตลอด นอกจากนั้นแล้ว ยังได้รับความช่วยเหลือและกำลังใจจาก อาจารย์และเจ้าหน้าที่ ประจำภาควิชาครุศาสตร์ศรีสะเกษทุกท่าน ทำให้การทำปริญญานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี ทางผู้จัดทำจึงขอขอบคุณทุกท่านเป็นอย่างสูง

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ที่ได้สนับสนุนทางการเงินและคอยให้กำลังใจมาโดยตลอด จนเป็นแรงบันดาลใจให้ตั้งใจทำโครงงานนี้ ผู้จัดทำโครงงานจึงขอขอบพระคุณทุกๆ ท่านที่ให้ความอนุเคราะห์ มาไว้ ณ โอกาสนี้

นาย อุดมศักดิ์ สมภักดี

นาย ญัฐชัย ชุ่มประดิษฐ์

ผู้จัดทำโครงงาน

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	3
1.3 ขอบเขตของโครงการ	3
1.4 นิยามศัพท์เฉพาะ	4
1.5 ประโยชน์ที่ได้รับจากโครงการ	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	7
2.1 การเคลื่อนที่ของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel	7
2.2 แนวทางการควบคุมเคลื่อนตัวของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel	9
2.3 เอ็นโคเดอร์ (Encoder)	9
2.4 ระบบอัลตราโซนิก	11
2.5 โปรแกรม Microsoft Visual Studio	13
2.6 ส่วนประกอบของโปรแกรม Microsoft Visual Studio	14
2.7 โปรแกรม Android Studio	23
2.8 การหาความเร็วมอเตอร์	24
2.9 ระบบเครือข่ายไร้สาย (Wireless LAN Technology)	24
2.10 รูปแบบการเชื่อมต่อของระบบเครือข่ายไร้สาย	25
2.11 ฟัซซีลอจิก (Fuzzy Logic)	28
2.12 งานวิจัยที่เกี่ยวข้อง	38
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน	39
3.1 ศึกษาและรวบรวมทฤษฎีที่เกี่ยวข้องกับปริญญานิพนธ์	43
3.2 การออกแบบทางกลของหุ่นยนต์	43
3.3 วงจรของหุ่นยนต์	55
3.4 ออกแบบโปรแกรมควบคุมด้วยมือ (Manual)	62
3.5 ออกแบบโปรแกรมควบคุมแบบอัตโนมัติ (Automatic)	63

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน	81
4.1 หุ่นยนต์ที่ประกอบสมบูรณ์	81
4.2 หน้าต่างโปรแกรมของการควบคุมหุ่นยนต์ล้อยหลายทิศทาง	82
4.2 ผลการทดสอบการควบคุมหุ่นยนต์โดยไม่มีสิ่งกีดขวาง	85
4.3 ผลการทดสอบการควบคุมหุ่นยนต์โดยมีสิ่งกีดขวาง	93
บทที่ 5 สรุปผลการทดลอง	95
5.1 สรุปผลของโครงงาน	95
5.2 ปัญหาและแนวทางแก้ไข	95
5.3 ข้อเสนอแนะ	96
บรรณานุกรม	97
ภาคผนวก ก	99
ภาคผนวก ข	107
ภาคผนวก ค	173
ภาคผนวก ง	189
ประวัติผู้จัดทำ	205

สารบัญตาราง

ตารางที่	หน้า
3-1 ระยะเวลาการดำเนินโครงการ	41
3-2 ตัวประกอบความล้า (Fatigue Factor)	46
3-3 ขนาดกำหนดของเพลตามมาตรฐาน ISO/R 775-1969	47
3-4 อายุงานลูกกลิ้งแบร์ริง	48
3-5 ค่าตัวประกอบแรงในแนวรัศมี (x) และในแนวแกน (y)	50
3-6 ตัวประกอบอุณหภูมิ	51
3-7 พิกัดความเผื่อของเพลตันทรงกระบอก	52
3-8 พิกัดความเผื่อของเล็อดลับลูกปืน	53
3-9 กฎพีซีซีที่ใช้ในหุ่นยนต์	75
3-10 การกำหนดค่า G ในเซตพีซีซี แบบ 3 สมาชิก	75
3-11 การกำหนดค่า ΔG ในเซตพีซีซี แบบ 3 สมาชิก	76
3-12 การกำหนดค่าเอาต์พุตในเซตพีซีซี แบบ 3 สมาชิก	76
4-1 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบไม่มีโหลด	83
4-2 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบไม่มีโหลด	84
4-3 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 1 กิโลกรัม	85
4-4 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 1 กิโลกรัม	86
4-5 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 2 กิโลกรัม	87
4-6 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 2 กิโลกรัม	88
4-7 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 3 กิโลกรัม	89
4-8 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 2 กิโลกรัม	90
4-9 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีสิ่งกีดขวาง	91
4-10 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีสิ่งกีดขวาง	92

สารบัญภาพ

ภาพที่	หน้า
1-1 ตำแหน่งล้อของหุ่น	2
1-2 หน้าตัดของล้อ Omnidirectional Wheel	2
1-3 ล้อ Omnidirectional Wheel ที่สัมผัสกับพื้น	3
2-1 แสดงรูปการแตกความเร็ว V_x และ V_y และ v_x และ v_y	8
2-2 แสดงรูปเวกเตอร์ความเร็วเชิงมุมของแต่ละล้อ	8
2-3 ระบบ DC Servo Motor ที่มีการป้อนกลับตำแหน่งและความเร็ว	10
2-4 เอ็นโคเดอร์แบบอินคริเมนทอลและสัญญาณพัลส์	10
2-5 ตัวอย่างเอ็นโคตเตอร์	11
2-6 ตัวอย่างการเขียนสัญลักษณ์ของอัลตราโซนิกทรานสดิวเซอร์แบบต่างๆ กัน	13
2-7 หน้าจอขณะเตรียมพร้อมโปรแกรม Visual Studio 2012	13
2-8 หน้าจอขณะการใช้งานโปรแกรม Visual Studio 2012	14
2-9 แถบเครื่องมือ Toolbox	14
2-10 แท็บ Common Controls	15
2-11 แท็บ Containers	15
2-12 ตัวอย่างการใช้อุปกรณ์ในแท็บ Containers	16
2-13 แท็บ Menus & Toolbars	16
2-14 ตัวอย่างการใช้อุปกรณ์ในแท็บ Menus & Toolbars	16
2-15 แท็บ Data	17
2-16 แท็บ Components	17
2-17 แท็บ Printing	18
2-18 ตัวอย่างการใช้อุปกรณ์ในแท็บ Printing : PrintDialog	18
2-19 แท็บ Dialogs	19
2-20 ตัวอย่างการใช้อุปกรณ์ในแท็บ Dialogs : OpenFileDialog	19
2-21 ตัวอย่างการใช้อุปกรณ์ในแท็บ Dialogs : ColorDialog	20
2-22 แท็บ Reporting	20
2-23 แท็บ WPF Interoperability	20
2-24 หน้าต่าง Solution Explorer	21
2-25 หน้าต่าง Properties แสดงถึงคุณสมบัติของ Form	22
2-26 หน้าต่าง Properties แสดงถึงเหตุการณ์ของ Form	22
2-27 หน้าต่าง Editor แสดงตัวอย่างโค้ดโปรแกรม	22
2-28 โปรแกรม Android Studio ในส่วนของการออกแบบ User Interface	23

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2-29 บล็อกไดอะแกรมของมอเตอร์	24
2-30 รูปแบบการเชื่อมต่อแบบ Peer-to-Peer	25
2-31 รูปแบบการเชื่อมต่อแบบ Client/Server	26
2-32 รูปแบบการเชื่อมต่อแบบ Multiple access points and roaming	26
2-33 รูปแบบการเชื่อมต่อแบบ Use of an Extension Point	27
2-34 รูปแบบการเชื่อมต่อแบบมีการใช้ Antennas	27
2-35 ตรรกะแบบจริงเท็จ (Boolean Logic) กับตรรกะแบบฟัซซี (Fuzzy Logic)	29
2-36 ความไม่แน่นอน (Uncertainty)	30
2-37 ตัวอย่างเซตต้นฉบับ	30
2-38 ฟังก์ชันความเป็นสมาชิกในเซตผู้ที่ไม่แต่งงาน	30
2-39 การกำหนดค่าความเป็นสมาชิกของเซตวิทยุและเซตแบบฟัซซี	31
2-40 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบวิยัค A	32
2-41 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซีแบบต่อเนื่อง A	32
2-42 ตัวอย่างตัวแปรภาษา	33
2-43 ตัวอย่างปริภูมิรูปแบบการจัดกลุ่มด้วยกฎฟัซซี	34
2-44 โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซี	36
2-45 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก	36
2-46 ขั้นตอนที่ 1 ของการประมวลผลแบบฟัซซีลอจิก	37
2-47 ขั้นตอนที่ 2 ของการประมวลผลแบบฟัซซีลอจิก	37
2-48 ขั้นตอนที่ 3 ของการประมวลผลแบบฟัซซีลอจิก	37
2-49 ขั้นตอนที่ 4 ของการประมวลผลแบบฟัซซีลอจิก	38
3-1 โพลีชาร์ตการดำเนินงาน (1)	39
3-2 โพลีชาร์ตการดำเนินงาน (2)	40
3-3 โพลีชาร์ตการดำเนินงาน (3)	41
3-4 ชิ้นส่วนของหุ่นยนต์	43
3-5 สมมุติแรงที่เกิดขึ้นบนเพลลา	45
3-6 Free-body Diagram และ Bending Moment Diagram	46
3-7 แรงกดที่กระทำกับลูกปืน	46
3-8 ขนาดเพลลา	48
3-9 แรงกระทำที่เกิดขึ้นในแนวแกนของลูกปืน	48
3-10 ตารางมาตรฐานของลูกปืน	50

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-11 ขนาดของเพลาและล้อ	54
3-12 ไมโครคอนโทรลเลอร์ Arduino	55
3-13 เซ็นเซอร์ Ultrasonic	56
3-14 Encoder	56
3-15 ชุดแปลงไฟเลี้ยง	56
3-16 แบตเตอรี่	57
3-17 วงจรไฟฟ้าที่ใช้ในหุ่นยนต์	57
3-18 วงจร Interlock	58
3-19 การทำงานของวงจร Interlock (1)	58
3-20 การทำงานของวงจร Interlock (2)	59
3-21 การทำงานของวงจร Interlock (3)	59
3-22 วงจร Optocoupler	60
3-23 วงจร H-Bridge	60
3-24 แหล่งจ่ายไฟฟ้า	61
3-25 โพลีชาร์ตการทำงานบน C#/Android	62
3-26 โพลีชาร์ตการทำงานบนไมโครคอนโทรลเลอร์	62
3-27 ลักษณะการเคลื่อนที่	63
3-28 การกำหนดขนาดและทิศทางใน Quadrant ที่ 1	64
3-29 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 1	65
3-30 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 1	65
3-31 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 1	66
3-32 การกำหนดขนาดและทิศทางใน Quadrant ที่ 2	67
3-33 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 2	67
3-34 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 2	68
3-35 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 2	69
3-36 การกำหนดขนาดและทิศทางใน Quadrant ที่ 3	70
3-37 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 3	70
3-38 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 3	71
3-39 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 3	72
3-40 การกำหนดขนาดและทิศทางใน Quadrant ที่ 4	73
3-41 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 4	73

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-42 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 4	74
3-43 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 4	75
3-44 บล็อกไดอะแกรมของการควบคุมการทำงานของหุ่นยนต์	76
3-45 พลังงานอิสระของตัวแปร G , ΔG และเอนทัลปี	77
3-46 โพลีชาร์ตการทำงานบน C#/Android	78
3-47 โพลีชาร์ตการทำงานบนไมโครคอนโทรลเลอร์	79
3-48 โพลีชาร์ตการหลบสิ่งกีดขวางของหุ่นยนต์	80
4-1 หุ่นยนต์ที่ประกอบสมบูรณ์	81
4-2 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Visual Studio 2012	82
4-3 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Android Studio (1)	83
4-4 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Android Studio (2)	84
4-5 แผนภูมิผลการทดสอบวัดระยะแบบไม่มีโหลด	85
4-6 แผนภูมิผลการทดสอบวัดมุมแบบไม่มีโหลด	86
4-7 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 1 กิโลกรัม	87
4-8 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 1 กิโลกรัม	88
4-9 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 2 กิโลกรัม	89
4-10 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 2 กิโลกรัม	90
4-11 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 3 กิโลกรัม	91
4-12 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 3 กิโลกรัม	92
4-13 แผนภูมิผลการทดสอบวัดระยะแบบหลบสิ่งกีดขวาง	93
4-14 แผนภูมิผลการทดสอบวัดมุมแบบหลบสิ่งกีดขวาง	94
ก-1 ฐานของหุ่นยนต์	100
ก-2 โครงข้างนอกของหุ่นยนต์	101
ก-3 ชุดยี่ตวงจร	102
ก-4 ชุดยี่ตล้อ	103
ก-5 แกนล้อ	104
ก-6 ฐานรองรับโหลด	105
ก-7 ชิ้นส่วนทั้งหมดของหุ่นยนต์	106

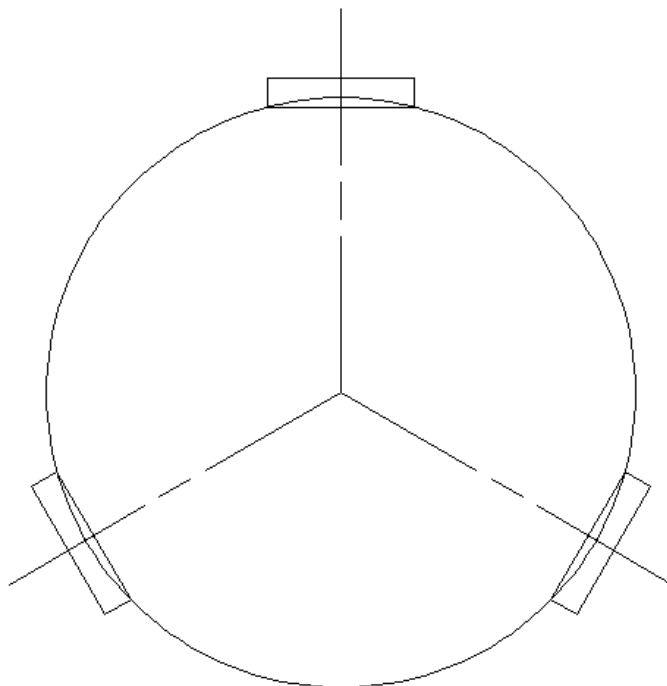
บทที่ 1

บทนำ

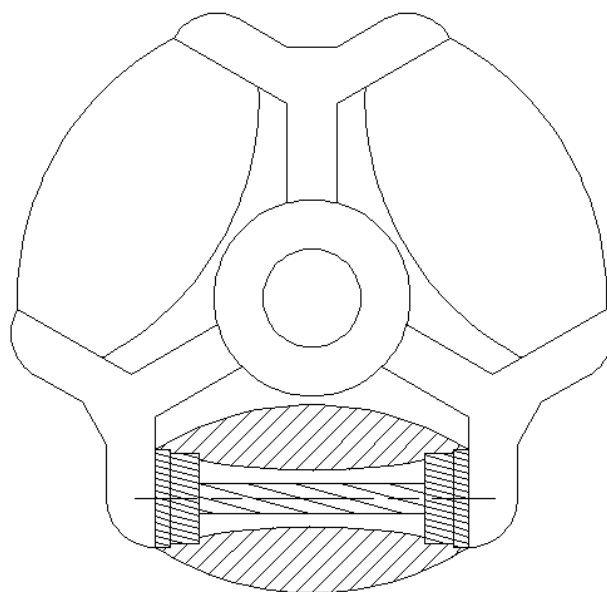
1.1 ความเป็นมาและความสำคัญของปัญหา

พลศาสตร์และการควบคุมเป็นงานวิจัยหลักแขนงหนึ่งที่มีความสำคัญในการพัฒนางานทางด้านวิทยาศาสตร์ นักวิจัยทั่วโลกได้ทำการศึกษาทางสาขานี้อย่างแพร่หลาย เช่น การควบคุมหุ่นยนต์อุตสาหกรรม การควบคุมการบินของเครื่องบิน ซึ่งโครงการนี้จำเป็นต้องใช้ความรู้ด้านวิชาพลศาสตร์เพื่อศึกษาพฤติกรรมการเคลื่อนที่ของระบบ และทฤษฎีการควบคุมเพื่อออกแบบระบบควบคุม ให้มีพฤติกรรมการเคลื่อนที่ ตามที่ต้องการ ยิ่งไปกว่านั้น ในการสร้างระบบจริง ทั้งยังต้องมีความรู้เกี่ยวกับอุปกรณ์วัดและอุปกรณ์ควบคุม ในเริ่มแรกนักวิจัยได้พัฒนาหุ่นยนต์และอุปกรณ์ควบคุมอื่นๆ เพื่อให้ การใช้ชีวิตประจำวันมีความสะดวกสบาย โดยเริ่มสร้างจากหุ่นยนต์แบบ Directional Wheel ถึงแม้ว่าหุ่นยนต์แบบ Directional Wheel นั้นสร้างง่ายและศึกษาพฤติกรรมการเคลื่อนที่ได้งานและสร้างระบบควบคุมได้ง่ายแต่ก็ยังมีข้อจำกัดมาก เช่น การเคลื่อนที่หุ่นยนต์ Directional Wheel จะเปลี่ยนเส้นทางโดยเคลื่อนที่ไปทางซ้ายหรือทางขวาก็ต้องหยุดอยู่กับที่ก่อน หรือต้องเคลื่อนที่โดยตีโค้งกว้างเพื่อที่จะได้ไปยังตำแหน่งที่ต้องการ ถ้าหากต้องการเคลื่อนที่ในพื้นที่แคบๆ ก็จะทำให้เคลื่อนที่ได้ลำบาก เมื่อเปลี่ยนเส้นทางเลี้ยวซ้ายหรือเลี้ยวขวาก็จะเกิดแรงเสียดทานขึ้นเนื่องจากหน้าสัมผัสของล้อกับพื้น จึงทำให้เกิดการสูญเสียพลังงานโดยเปล่าประโยชน์ และเวลาการทำงาน จะต้องใช้เวลานานเมื่อหุ่นยนต์ Direction Wheel เคลื่อนที่อยู่แล้วต้องการที่จะเคลื่อนที่ไปทางขวาหรือทางซ้าย ในแนวตั้งฉากก็ต้องสูญเสียเวลา เพื่อทำการเลี้ยวก่อนแล้วจึงเคลื่อนที่ต่อ

จากจุดด้อยต่างๆ ที่ได้เกิดขึ้นกับหุ่นยนต์ที่ใช้ล้อแบบ Directional Wheel จึงได้เริ่มมีการพัฒนาหุ่นยนต์ที่มีล้ออีกแบบหนึ่งที่รู้จักกันว่า Omnidirectional Wheel หุ่นยนต์ที่จะพัฒนาในโครงการนี้จะลักษณะตัวหุ่นดังภาพที่ 1.1 และมีลักษณะล้อดังภาพที่ 1.2

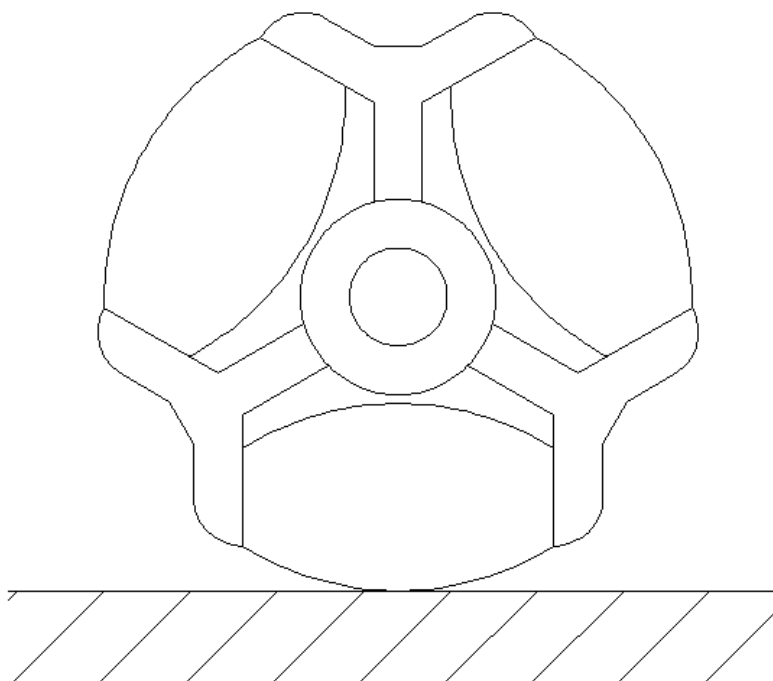


ภาพที่ 1-1 ตำแหน่งล้อของหุ่น



ภาพที่ 1-2 หน้าตัดของล้อ Omnidirectional Wheel

โดยจุดเด่นของหุ่นยนต์แบบ Omnidirectional Wheel จะช่วยลดข้อจำกัด ทางด้านการเคลื่อนที่ เพราะล้อสามารถทำให้ตัวหุ่นนั้นเคลื่อนที่ได้โดยอิสระ ทางด้านแรงเสียดทานก็ช่วยลดลงได้เพราะล้อของหุ่นนั้นจะสัมผัสกับพื้นแบบจุดเท่านั้นโดยจะสัมผัสกับพื้นดังภาพที่ 1-3 ทำให้เวลาที่หุ่นยนต์เคลื่อนที่จะใช้น้อยลง เพราะไม่จำเป็นต้องเสียเวลากลับลำของตัวหุ่นยนต์



ภาพที่ 1-3 ล้อ Omnidirectional Wheel ที่สัมผัสกับพื้น

ปัจจุบันงานวิจัยทางสาขานี้ในประเทศไทย ยังขาดการพัฒนาอย่างต่อเนื่อง อีกทั้งยังขาดผู้เชี่ยวชาญ โครงการนี้มุ่งหวังเพื่อเป็นการพัฒนางานวิจัยพื้นฐานในสาขานี้ โดยศึกษาการเคลื่อนตัวของหุ่นยนต์ Omnidirectional Wheel ซึ่งเป็นระบบที่สามารถนำไปประยุกต์ในหลายๆด้าน เช่น เป็นพาหนะสำหรับคนพิการหรือคนป่วย

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อสร้างหุ่นยนต์เคลื่อนที่โดยใช้ล้อหลายทิศทาง
- 1.2.2 เพื่อพัฒนาโปรแกรมสำหรับควบคุมการเคลื่อนที่กับหุ่นยนต์เคลื่อนที่โดยใช้ล้อหลายทิศทาง

1.3 ขอบเขตของโครงการ

- 1.3.1 หุ่นยนต์เคลื่อนที่หลายทิศทาง มีล้อหลายทิศทาง 3 ล้อ
- 1.3.2 วิ่งในพื้นเรียบ และรับโหลดได้สูงสุด 3 กิโลกรัม
- 1.3.3 ติดต่อกับหุ่นยนต์ Omni-Directional Wheel โดยผ่านทาง Wireless LAN
- 1.3.4 ควบคุมหุ่นยนต์โดยใช้ คอมพิวเตอร์ ที่ใช้ระบบปฏิบัติการ Microsoft Windows และโทรศัพท์ที่ใช้ระบบปฏิบัติการ Android

- 1.3.5 พัฒนาโปรแกรมควบคุมการเคลื่อนที่โดยใช้ Microsoft Visual Studio 2012 ในการพัฒนาโปรแกรมบน Microsoft Windows และ Android Studio ในการพัฒนาโปรแกรมบน Android
- 1.3.6 หุ่นยนต์ Omni-Directional Wheel เคลื่อนที่ไปตามตำแหน่งที่ต้องการ และหลบสิ่งกีดขวางได้
- 1.3.7 หุ่นยนต์มีความคลาดเคลื่อนในการใช้ระบบควบคุมอัตโนมัติในระยะ 10 เปอร์เซ็นต์

1.4 นิยามศัพท์เฉพาะ

- 1.4.1 Directional Wheel คือล้อที่มีสององศาอิสระ มีการเคลื่อนที่ตามแนวพื้นที่ผิวของล้อตามแนวการวางล้อ และการหมุนที่จุดสัมผัสระหว่างล้อกับพื้นโดยในการใช้งานจริงมีการติดตั้งกลไกการหมุน (Steering Mechanism) เพื่อเพิ่มองศาอิสระขึ้นเป็นสามองศาอิสระ^[1]
- 1.4.2 Omni-Directional Wheel คือล้อที่มีสามองศาอิสระ ประกอบด้วยองศาอิสระที่หนึ่ง คือ ทิศทางตามแนวการวางล้อ องศาอิสระที่สองเกิดจากการหมุนของลูกกลิ้งที่ติดอยู่รอบๆ ของล้อ ในทางทฤษฎีของมุมลูกกลิ้ง สามารถทำมุมเท่าไรก็ได้ นอกจากศูนย์ และองศาอิสระที่สามเกิดจากการหมุนที่จุดสัมผัสระหว่างลูกกลิ้งและพื้น^[1]
- 1.4.3 Microsoft Visual Studio คือ Integrated Development Environment (โปรแกรมประยุกต์ซอฟต์แวร์ ซึ่งอำนวยความสะดวกให้แก่ักเขียนโปรแกรมคอมพิวเตอร์ในการพัฒนาซอฟต์แวร์) พัฒนาขึ้นโดยไมโครซอฟท์ ซึ่งเป็นเครื่องมือที่ช่วยนักพัฒนาซอฟต์แวร์พัฒนาโปรแกรมคอมพิวเตอร์ เว็บไซต์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส ระบบที่รองรับการทำงานนั้นมีไมโครซอฟท์ วินโดวส์ พ็อคเก็ตพีซี สมาร์ทโฟน และ เว็บเบราว์เซอร์ ในปัจจุบัน Visual Studio นั้นสามารถใช้ภาษาโปรแกรมที่เป็นภาษา .NET ในโปรแกรมเดียวกัน เช่น VB.NET C++ C# J# เป็นต้น
- 1.4.4 Wireless LAN คือ เทคโนโลยีที่เชื่อมต่ออุปกรณ์ตั้งแต่สองตัวขึ้นไปเข้าด้วยกัน โดยใช้วิธีการกระจายแบบไร้สาย (ส่วนใหญ่แล้ว จะใช้ Spread-Spectrum หรือคลื่นวิทยุ OFDM) โดยปกติแล้ว จะมีการเชื่อมต่อผ่านทาง Access Point เพื่อเข้าไปยังโลกอินเทอร์เน็ต แลนไร้สายทำให้ผู้ใช้สามารถนำพาหรือเคลื่อนย้ายคอมพิวเตอร์ไปยังพื้นที่ใดก็ได้ที่มีสัญญาณของแลนไร้สาย และยังสามารถเชื่อมต่อกับเครือข่ายได้ตามปกติ

1.5 ประโยชน์ที่ได้รับจากโครงการ

ได้ความรู้จากการสร้างหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบที่ใช้ 3 ล้อที่ใช้ในการขนของเพื่อเป็นต้นแบบชิ้นงานสำหรับโครงงานอื่นๆ และความรู้จากการเขียนโปรแกรม C# และ Android

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในการทำโครงงานครั้งนี้ผู้ทำโครงงานได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง และนำเสนอตามหัวข้อต่อไปนี้

- 2.1 การเคลื่อนที่ของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel
- 2.2 แนวทางการควบคุมเคลื่อนตัวของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel
- 2.3 เอ็นโค้ดเดอร์ (Encoder)
- 2.4 ระบบอัลตราโซนิก (Ultrasonic)
- 2.5 โปรแกรม Microsoft Visual Studio
- 2.6 ส่วนประกอบของโปรแกรม Microsoft Visual Studio
- 2.7 โปรแกรม Android Studio
- 2.8 การหาความเร็วมอเตอร์
- 2.9 ระบบเครือข่ายไร้สาย (Wireless LAN Technology)
- 2.10 รูปแบบการเชื่อมต่อของระบบเครือข่ายไร้สาย
- 2.11 ฟัซซีลอจิก (Fuzzy Logic)
- 2.12 งานวิจัยที่เกี่ยวข้อง

2.1 การเคลื่อนที่ของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel

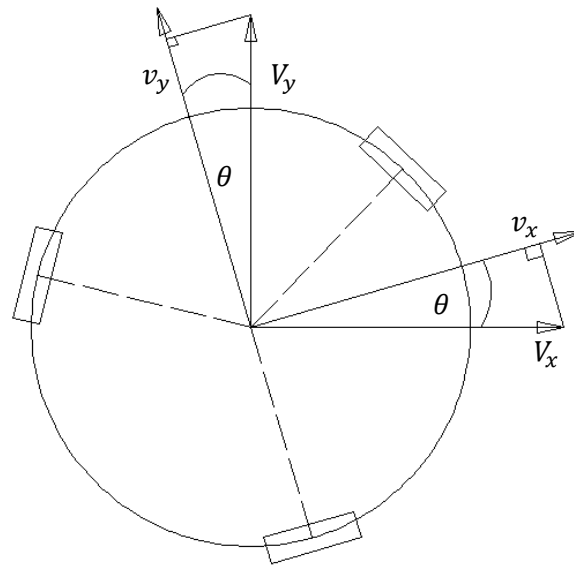
ในการควบคุมการเคลื่อนตัวของหุ่นยนต์ Omni-Directional Wheel ให้ความสะดวกแม่นยำนั้นจำเป็นต้องทราบการเคลื่อนที่ทางพลศาสตร์ของหุ่นยนต์ ดังภาพ 2-1 แสดงพารามิเตอร์และตัวแปรการเคลื่อนที่ต่างๆ ของหุ่นยนต์ Omni-Directional Wheel 3 ล้อ โดย X-Y คือแกนอ้างอิง (Reference Frame) และ X-Y คือแกนอ้างอิงซึ่งยึดติดกับตัวหุ่นยนต์ (Body Axis Coordinate Frame)

โดยกำหนดให้

v_x และ v_y คือ ความเร็วอ้างอิงของตัวหุ่น (เมตร/วินาที)

V_x และ V_y คือ ความเร็วสัมบูรณ์เชิงเส้นของตัวหุ่น (เมตร/วินาที)

เมื่อกำหนดความเร็วสัมบูรณ์เชิงเส้น V_x และ V_y ของตัวหุ่นยนต์ ก็จะได้ว่าความเร็วของแต่ละล้อมีความเร็วเท่าไร โดยก่อนอื่นจะต้องแตกความเร็วสัมบูรณ์ที่ V_x และ V_y เข้ามายังแกนความเร็วอ้างอิงของตัวหุ่นยนต์คือ v_x และ v_y



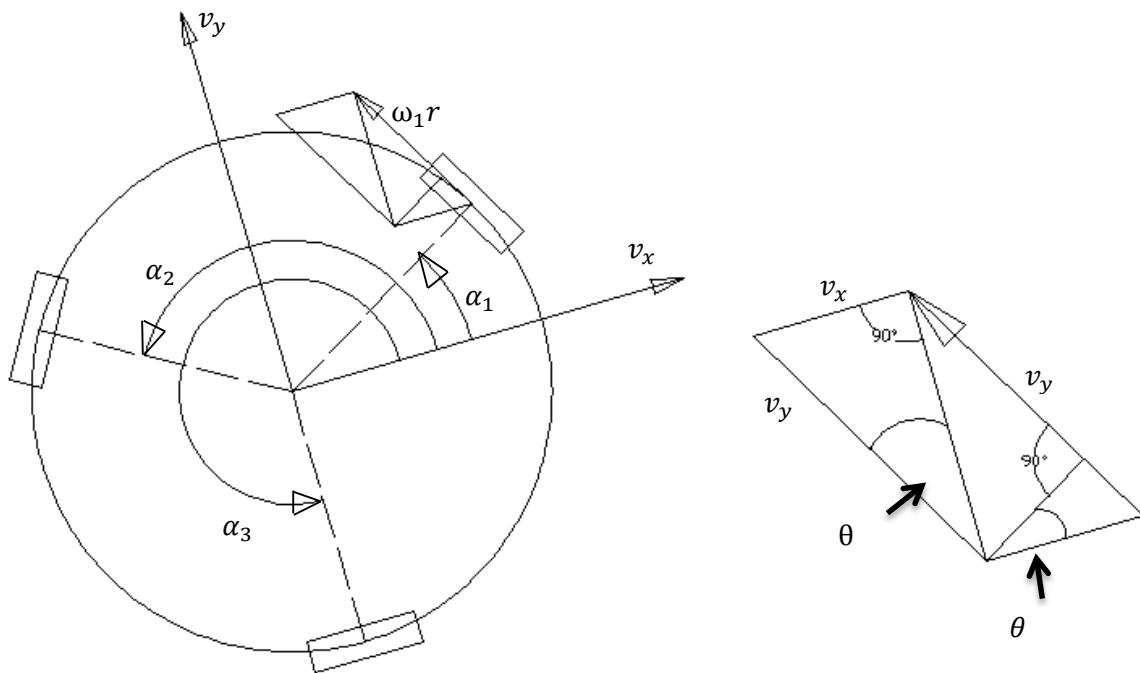
ภาพที่ 2-1 การแตกเวกเตอร์ความเร็ว V_x และ V_y และ v_x และ v_y

จากภาพเมื่อแตกเวกเตอร์ความเร็วสัมผัสที่ V_x และ V_y เข้ามายังแกนความเร็วอ้างอิงของตัวหุ่นยนต์ คือ v_x และ v_y ก็จะได้สมการดังนี้

$$v_x = V_x \cos \theta + V_y \sin \theta \quad (2.1)$$

$$v_y = -V_x \sin \theta + V_y \cos \theta \quad (2.2)$$

ความเร็วเชิงมุมแต่ละล้อ สัมพันธ์กับความเร็วของหุ่นยนต์ตามสมการ Inverse Kinematic



ภาพที่ 2-2 เวกเตอร์ความเร็วเชิงมุมของแต่ละล้อ

$$\text{ล้อที่ 1 : } \omega_1 = \frac{1}{r}(-v_x \sin \alpha_1 + v_y \cos \alpha_1 + \dot{\theta}R) \quad (2.3)$$

$$\text{ล้อที่ 2 : } \omega_2 = \frac{1}{r}(-v_x \sin \alpha_2 + v_y \cos \alpha_2 + \dot{\theta}R) \quad (2.4)$$

$$\text{ล้อที่ 3 : } \omega_3 = \frac{1}{r}(-v_x \sin \alpha_3 + v_y \cos \alpha_3 + \dot{\theta}R) \quad (2.5)$$

หากต้องการควบคุมหุ่นยนต์ให้มีความเร็วเทียบกับแกนอ้างอิง $(V_x, V_y, \dot{q})$ ตามที่ต้องการจะใช้สมการ (2.3) ถึง (2.5) เพื่อหาค่าความเร็วเชิงมุมของล้อ $(\omega_1, \omega_2, \omega_3)$ ที่ต้องการควบคุมในทางกลับกัน เราก็สามารถทราบค่าความเร็วของตัวหุ่นยนต์เมื่อกำหนดค่าความเร็วเชิงมุมของล้อจากสมการ Forward Kinematic

$$\begin{Bmatrix} v_x \\ v_y \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} -\sin \alpha_1 & \cos \alpha_1 & R \\ -\sin \alpha_2 & \cos \alpha_2 & R \\ -\sin \alpha_3 & \cos \alpha_3 & R \end{bmatrix}^{-1} \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{Bmatrix} \quad (2.6)$$

และ

$$v_x = V_x \cos \theta - V_y \sin \theta \quad (2.7)$$

$$v_y = V_x \sin \theta + V_y \cos \theta \quad (2.8)$$

โดย $\dot{\theta}$ หาได้จากบรรทัดที่ 3 ของสมการ (2.6)

2.2 แนวทางการควบคุมเคลื่อนตัวของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel

สำหรับการควบคุมการเคลื่อนตัวของหุ่นยนต์ที่ใช้ล้อแบบ Omni-Directional Wheel ให้เคลื่อนที่ในรูปแบบต่างๆ นั้น จะต้องส่งคำสั่งการเคลื่อนที่ไปควบคุมล้อให้ถูกต้องแม่นยำตามสมการ (2.1) ถึง (2.2) ซึ่งสำหรับระบบ Holonomic นั้น จะสามารถทำการอินทิเกรตความเร็วเพื่อหาระยะการเคลื่อนที่ได้ ดังนั้นจากสมการ (2.1) ถึง (2.5) เมื่อกำหนดค่า $(V_x, V_y, \dot{q})$ ก็จะทำให้ทราบค่า $(\omega_1, \omega_2, \omega_3)$ และเมื่อทำการอินทิเกรต จะได้

$$\phi_i(t) = \int \omega_i dt \quad (2.8)$$

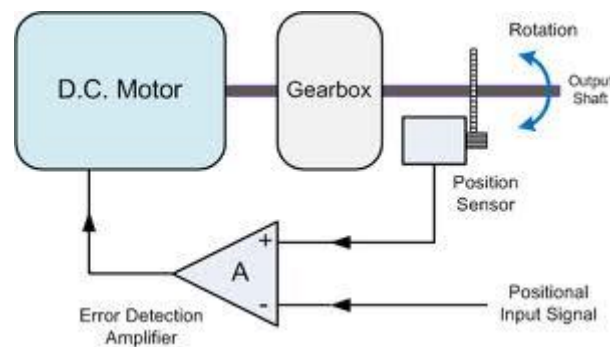
$$\phi_i(t) = \int \frac{1}{r}(-v_x \sin \alpha_1 + v_y \cos \alpha_1 + \dot{\theta}R) dt \quad (2.9)$$

$$\phi_i(t) = \int \frac{1}{r}(-v_x \sin \alpha_1 + v_y \cos \alpha_1) dt + R\theta(t) + \phi_i(0) \quad (2.10)$$

โดย $\phi_i(t)$ คือมุมการหมุนของล้อ Omni-Directional Wheel แต่ละล้อ มุมที่ได้นี้ถูกใช้เป็นคำสั่งการเคลื่อนที่ (Reference Command) เพื่อควบคุมการหมุนของล้อแต่ละล้อ

2.3 เอ็นโคดเดอร์ (Encoder)

ในการควบคุม DC Servo Motor ที่ดีต้องมีระบบคอนโทรลที่มีการป้อนกลับหรืออุปกรณ์ป้อนกลับ (Feedback Device) นั่นคือ เอ็นโคดเดอร์เพื่อทำหน้าที่วัดความเร็ว (Speed) วัดตำแหน่ง (Position) ตลอดจนทิศทางการหมุน (Direction of Rotation) ให้ถูกต้องและแม่นยำ



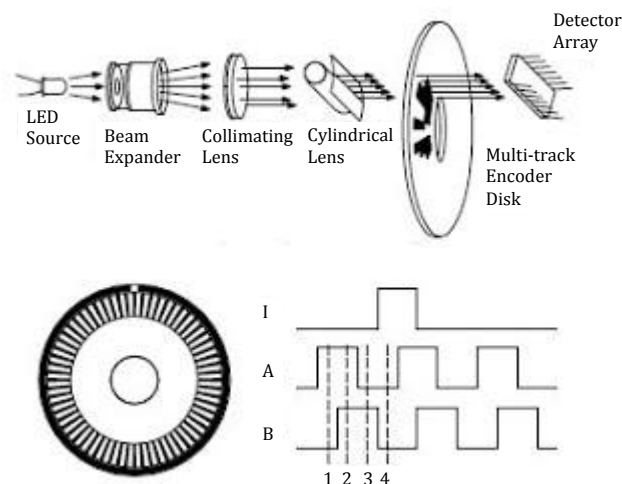
ภาพที่ 2-3 ระบบ DC Servo Motor ที่มีการป้อนกลับตำแหน่งและความเร็ว

เอ็นโค้ดเดอร์ที่ใช้อยู่ทั่วไปสามารถแบ่งออกเป็น 2 ชนิด ดังนี้

- a. แบบอินคริเมนทอล (Incremental Encoder)
- b. ชนิดแอ็บโซลูท (Absolute Encoder)

2.3.1 แบบอินคริเมนทอล

โดยทั่วไปเรียกว่าแบบโรตารี (Rotary Encoder) จะสร้างสัญญาณพัลส์ (Pulse) ที่แปรผันตรงกับการหมุน ของเฟลามาเตอร์ หรือจะหมุนด้วยความเร็วเท่ากับเฟลาของมอเตอร์นั่นเอง โดยแบบโรตารี จะประกอบด้วยจานหมุน (Rotary Disk) และอุปกรณ์ตรวจจับ (Sensor) โดยจานหมุนจะมีช่องเล็กๆ (Slit) เมื่อเฟลาของมอเตอร์หมุนจะทำให้จานหมุนไปตัดลำแสงของอุปกรณ์ตรวจจับ ทำให้ชุดรับแสงมีการรับสัญญาณเป็นช่วงๆ จึงทำให้สัญญาณเอาต์พุต (Output) มีลักษณะพัลส์โดยแสดงดังภาพที่ 2-4



ภาพที่ 2-4 เอ็นโค้ดเดอร์แบบอินคริเมนทอลและสัญญาณพัลส์

2.3.1 แบบอินคริเมนทอล

โดยทั่วไปเรียกว่า โพเทนชิโอมิเตอร์ เป็นเอ็นโคดเดอร์ อีกชนิดหนึ่งที่อาศัยหลักการของ **Optical** โดยทั่วไปแล้วการทำงานจะคล้ายๆกับ เอ็นโคดเดอร์แบบอินคริเมนทอล โดยการเคลื่อนที่ของ โพเทนชิโอมิเตอร์ จะแปรผันตรงกับความเร็ว (**Speed**) หรือระยะทาง (**Distance**) ของการเคลื่อนที่ เอ็นโคดเดอร์ชนิดนี้จะนิยม ใช้มากในระบบคอนโทรลจะให้ความเที่ยงตรงและสามารถบอกได้ทุกตำแหน่ง ของการเคลื่อนที่ ตลอดมีอายุการใช้งานที่ยาวนานกว่า



ภาพที่ 2-5 ตัวอย่างเอ็นโคดเดอร์

2.4 ระบบอัลตราโซนิก (Ultrasonic)

ระบบอัลตราโซนิก หมายถึง คลื่นเสียงที่มีความถี่สูงเกินกว่าที่มนุษย์จะได้ยิน โดยทั่วไปแล้วหูของมนุษย์โดยเฉลี่ยจะได้ยินเสียงสูงถึงเพียงแค่ประมาณ 15 กิโลเฮิร์ตซ์ เท่านั้น แต่พวกที่อายุยังน้อย ๆ อาจจะได้ยินเสียงที่มีความถี่สูงกว่านี้ได้ ดังนั้นโดยปกติแล้วคำว่า อัลตราโซนิก จึงมักจะหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20 กิโลเฮิร์ตซ์ ขึ้นไป จะสูงขึ้นจนถึงเท่าใดไม่ได้ระบุจำกัดเอาไว้

สาเหตุที่มีการนำเอาคลื่นย่านอัลตราโซนิกมาใช้ก็เพราะว่า เป็นคลื่นที่มีทิศทาง ทำให้เราสามารถเล็งคลื่นเสียงไปยังเป้าหมายที่ต้องการได้โดยเจาะจง เรื่องนี้เป็นคุณสมบัติของคลื่นอย่างหนึ่ง ยิ่งคลื่นมีความถี่ขึ้นความยาวคลื่นก็จะยิ่งสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิด (ที่ให้เสียงนั้นออกมา) ของตัวกำเนิดเสียงความถี่นั้นเช่น คลื่นความถี่ 300 เฮิร์ตซ์ ในอากาศจะมีความยาวถึงประมาณ 1 เมตรเศษ ๆ ซึ่งจะยาวกว่าช่องที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียงโดยทั่วไปมากมายคลื่นจะหักเบนที่ขอบด้านนอกของตัวกำเนิดเสียง ทำให้เกิดการกระจายทิศทางคลื่นแต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอัลตราโซนิก อย่างเช่น 40 กิโลเฮิร์ตซ์ จะมีความยาวคลื่นในอากาศเพียงประมาณ 8 มม. เท่านั้นซึ่งเล็กกว่ารูเปิดของตัวที่ให้กำเนิดเสียงความถี่นี้มากคลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบจึงพุ่งออกมาเป็นลำแคบ ๆ หรือที่เราเรียกว่า มีทิศทาง

การมีทิศทางของคลื่นเสียงย่านอัลตราโซนิก ทำให้เรานำไปใช้งานได้หลายอย่าง เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (**Ultrasonic Remote Control**) เครื่องล้างอุปกรณ์ (**Ultrasonic**

Cleaner) โดยให้น้ำล้นที่ความถี่สูง เครื่องวัดความหนาของวัตถุโดยสังเกตระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล ใช้ในเครื่องหาตำแหน่งอวัยวะบางส่วนในร่างกาย ใช้ทดสอบการรั่วไหลของท่อ เป็นต้น โดยความถี่ที่ใช้ขึ้นอยู่กับการใช้งาน เช่น คลื่นเสียงต้องเดินทางผ่านอากาศแล้ว ความถี่ที่ใช้ก็มักจะจำกัดอยู่เพียงไม่เกิน 50 กิโลเฮิร์ตซ์ เพราะที่ความถี่สูงขึ้นกว่านี้อากาศจะดูดกลืนคลื่นเสียงเพิ่มขึ้นมาก ทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ซึ่งต้องการวัดการสั่น ๆ ก็อาจใช้ความถี่ในช่วง 1 เมกะเฮิร์ตซ์ ถึง 10 เมกะเฮิร์ตซ์ ขณะที่ความถี่เป็น กิกะเฮิร์ตซ์ (10^9 เฮิร์ตซ์) ก็มีใช้กันหลาย ๆ การใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่อากาศ

อุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไปมา ซึ่งทำให้เกิดคลื่นเสียงย่านอัลตราโซนิก กระจายไปในอากาศได้หรือแปลงพลังงานทางกลให้มาเป็นพลังงานในรูปอื่นได้นั้น มีชื่อเรียกว่า อัลตราโซนิกทรานสดิวเซอร์ ในปัจจุบันมีหลายแบบขึ้นอยู่กับหลักการที่ใช้ แบบที่นิยมใช้กันมากได้แก่

- ทรานสดิวเซอร์ชนิดเพียโซอิเล็กทริก ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าและพลังงานทางกล โดยมีความถี่เรโซแนนซ์คงที่อย่างหนึ่ง
- ทรานสดิวเซอร์ชนิดแมกนีโตสตริกทิฟ ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่
- ทรานสดิวเซอร์ชนิดแมกนีโตสตริกทิฟ ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานทางกล

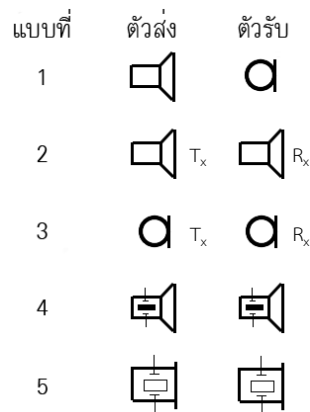
2.4.1 ส่วนประกอบของเซนเซอร์อัลตราโซนิก

2.4.1.1 ตัวส่ง

เป็นอัลตราโซนิกทรานสดิวเซอร์ที่ถูกออกแบบ เจาะจงมาให้แปลงสัญญาณไฟฟ้าที่ให้แก่วัสดุ ให้ออกมาเป็นคลื่นเสียงย่านอัลตราโซนิก หน้าที่ของตัวส่งจึงคล้าย ๆ กับเป็นลำโพง

2.4.1.2 ตัวรับ

เป็นอัลตราโซนิกทรานสดิวเซอร์ที่ถูกออกแบบเจาะจงมาให้ แปลงคลื่นเสียงย่านอัลตราโซนิก ที่มาตกกระทบตัวมันให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจึงคล้าย ๆ กับเป็นไมโครโฟน ด้วยเหตุนี้เวลาเขียนสัญลักษณ์ของทรานสดิวเซอร์ จึงนิยมเขียนตามหน้าที่ของมัน คือ ถ้าเป็นตัวส่ง จะเขียนสัญลักษณ์เป็นลำโพง หากเป็นตัวรับก็เขียนสัญลักษณ์เป็นไมโครโฟน ดังภาพที่



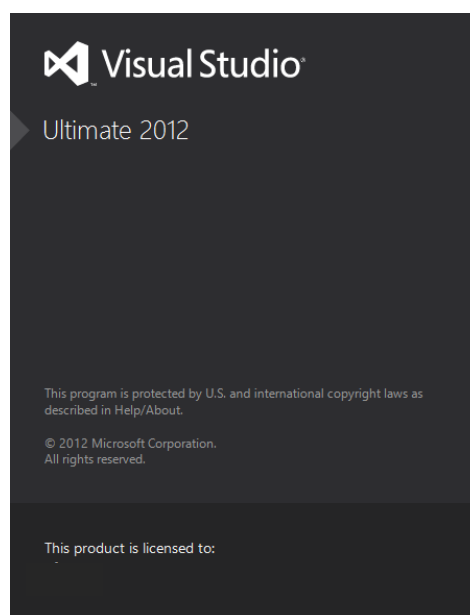
ภาพที่ 2-6 ตัวอย่างการเขียนสัญลักษณ์ของ อัลตราโซนิกทรานสดิวเซอร์แบบต่างๆ กัน

อัลตราโซนิกทรานสดิวเซอร์แบบเซรามิกที่มีจำหน่ายกัน จะมีค่าความถี่เรโซแนนซ์ให้เลือก ตั้งแต่ 23 กิโลเฮิร์ตซ์ขึ้นไปจนถึง 40 กิโลเฮิร์ตซ์ แต่ที่พบเห็นกันบ่อยก็มี 23 กิโลเฮิร์ตซ์, 25 กิโลเฮิร์ตซ์, และ 40 กิโลเฮิร์ตซ์ โดยความถี่ 40 กิโลเฮิร์ตซ์ เป็นรุ่นที่นิยมใช้กันมากที่สุด เพราะมีทิศทางดีกว่า

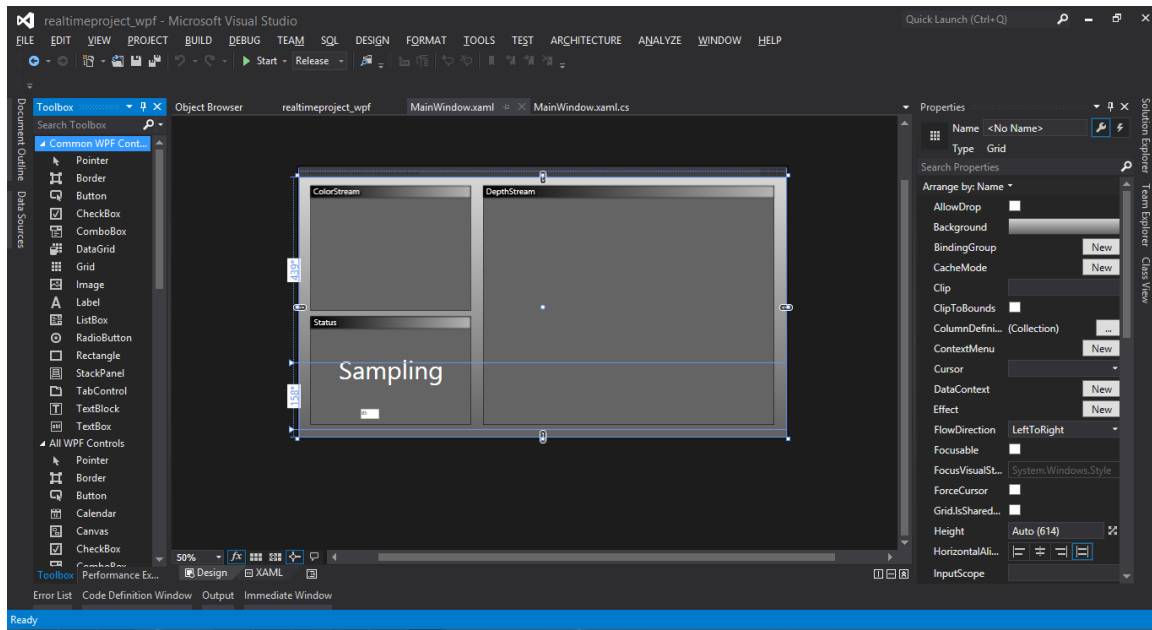
2.5 โปรแกรม Microsoft Visual Studio

Visual Studio ถือเป็นเครื่องมือพัฒนาแอปพลิเคชันที่ออกมารองรับตามแนวความคิดของ สถาปัตยกรรม .NET Framework ซึ่งในเครื่องมือนี้จะสามารถเขียนโปรแกรมได้หลายภาษา อาทิ เช่น BASIC, C#, C++ เป็นต้น

ในปัจจุบัน โปรแกรม Visual Studio ได้ออกมาถึงเวอร์ชัน 2012 แล้วซึ่งมีรูปร่างหน้าตาแบบใหม่ และรองรับกับการเขียนแอปพลิเคชันเข้าสู่ Windows Store (ในรูปแบบของ Metro App)



ภาพที่ 2-7 หน้าจอขณะเตรียมพร้อมโปรแกรม Visual Studio 2012

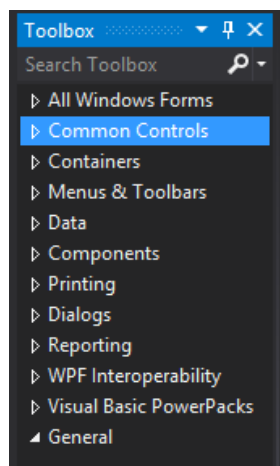


ภาพที่ 2-8 หน้าจอขณะการใช้งานโปรแกรม Visual Studio 2012

2.6 ส่วนประกอบของโปรแกรม Microsoft Visual Studio

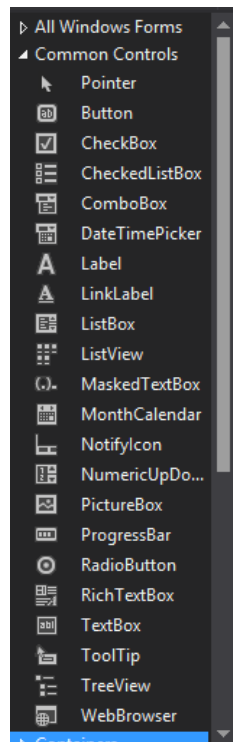
2.6.1 แถบเครื่องมือ Toolbox

เป็นแถบเครื่องมือที่ทำหน้าที่เก็บคอนโทรลต่างๆ ที่โปรแกรมสนับสนุน ประกอบไปด้วยกลุ่มคอนโทรล กลุ่มออบเจกต์และคอมโพเนนต์ ดังภาพที่ 2-9 แบ่งรายละเอียดตามภาพที่ 2-10 ถึง 2-28



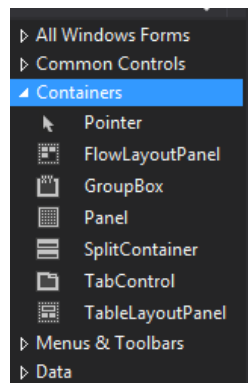
ภาพที่ 2-9 แถบเครื่องมือ Toolbox

- a. แท็บ Common Controls ทำหน้าที่เก็บคอนโทรลที่มีการใช้บ่อยๆ

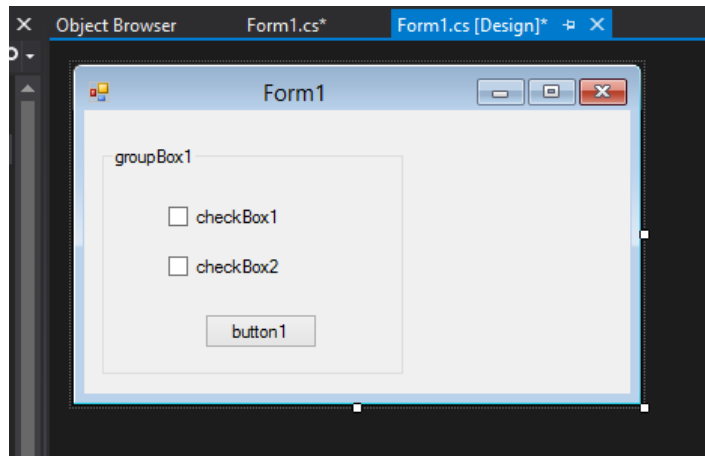


ภาพที่ 2-10 แท็บ Common Controls

- b. แท็บ Containers ทำหน้าที่เก็บคอนโทรลที่สามารถรับคอนโทรลตัวอื่นๆ ได้

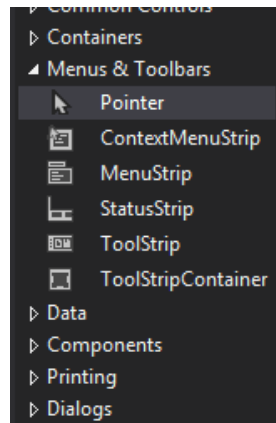


ภาพที่ 2-11 แท็บ Containers

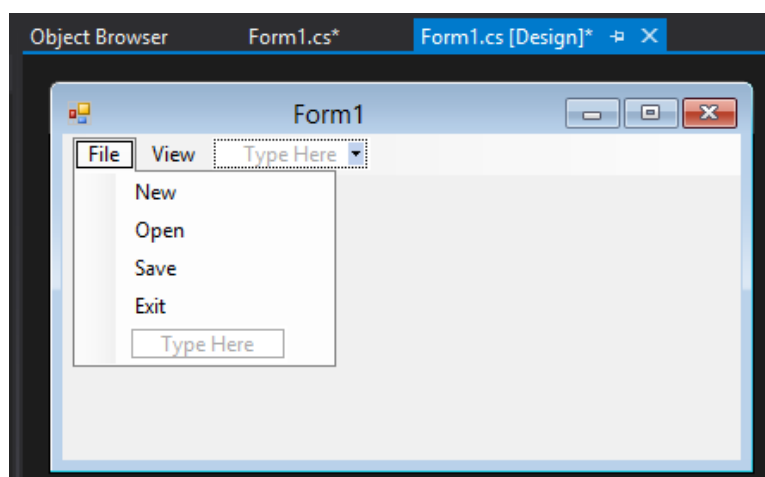


ภาพที่ 2-12 ตัวอย่างการใช้อุปกรณ์ในแท็บ Containers

c. แท็บ Menus & Toolbars ทำหน้าที่เก็บคอนโทรลที่ใช้ทำเมนูบาร์ หรือ สเตตัสบาร์

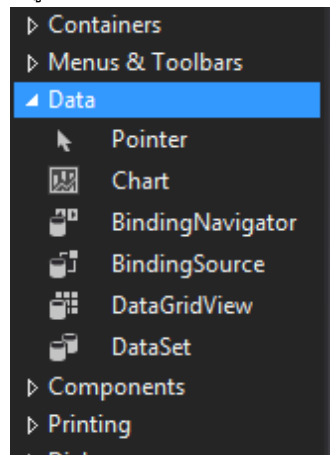


ภาพที่ 2-13 แท็บ Menus & Toolbars



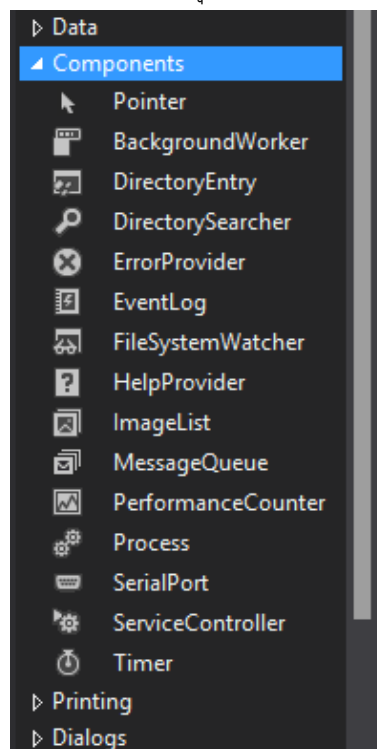
ภาพที่ 2-14 ตัวอย่างการใช้อุปกรณ์ในแท็บ Menus & Toolbars

- d. แท็บ Data ทำหน้าที่เก็บกลุ่มออบเจกต์ ADO.NET รับผิดชอบด้านการพัฒนาแอปพลิเคชันด้านฐานข้อมูล



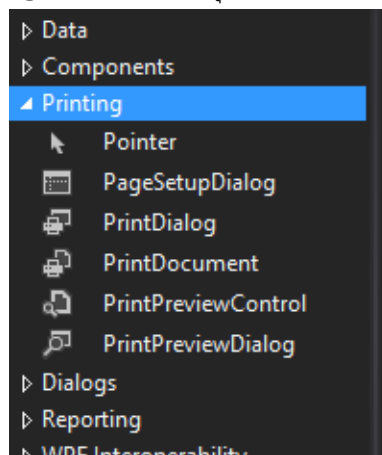
ภาพที่ 2-15 แท็บ Data

- e. แท็บ Components ทำหน้าที่เก็บกลุ่มออบเจกต์ด้านการติดต่อสื่อสารเครือข่ายต่างๆ

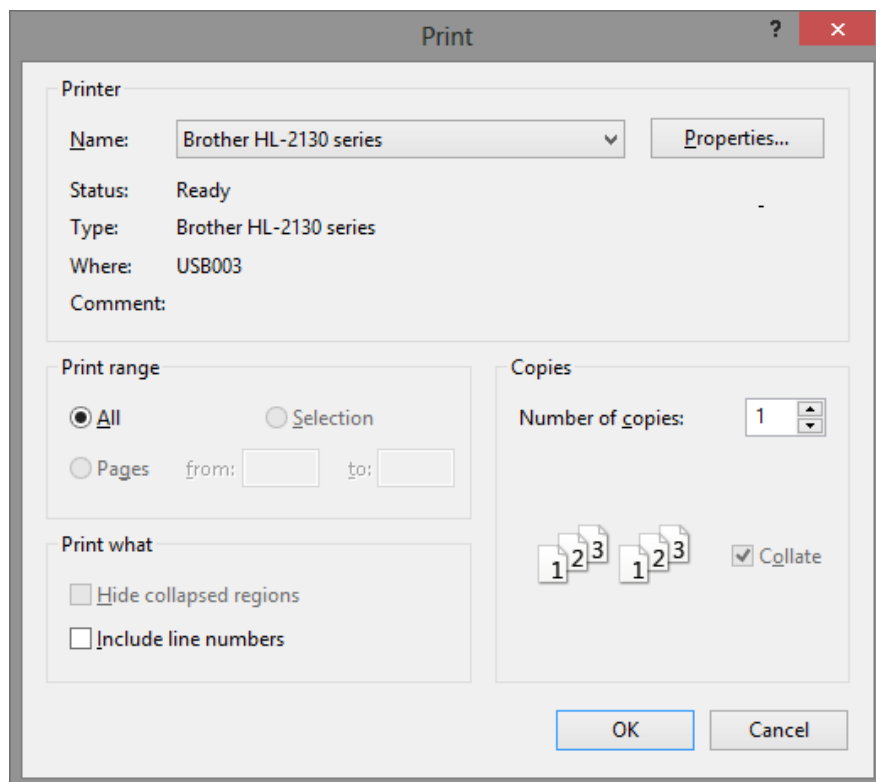


ภาพที่ 2-16 แท็บ Components

f. แท็บ **Printing** ทำหน้าที่เก็บกลุ่มออบเจกต์ด้านการที่เกี่ยวกับการพิมพ์

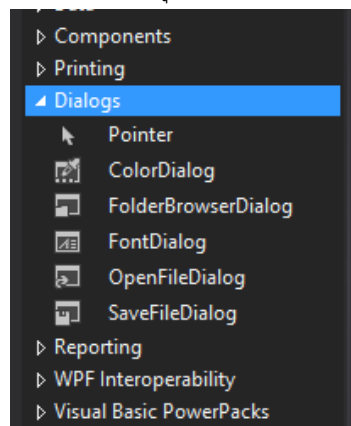


ภาพที่ 2-17 แท็บ Printing

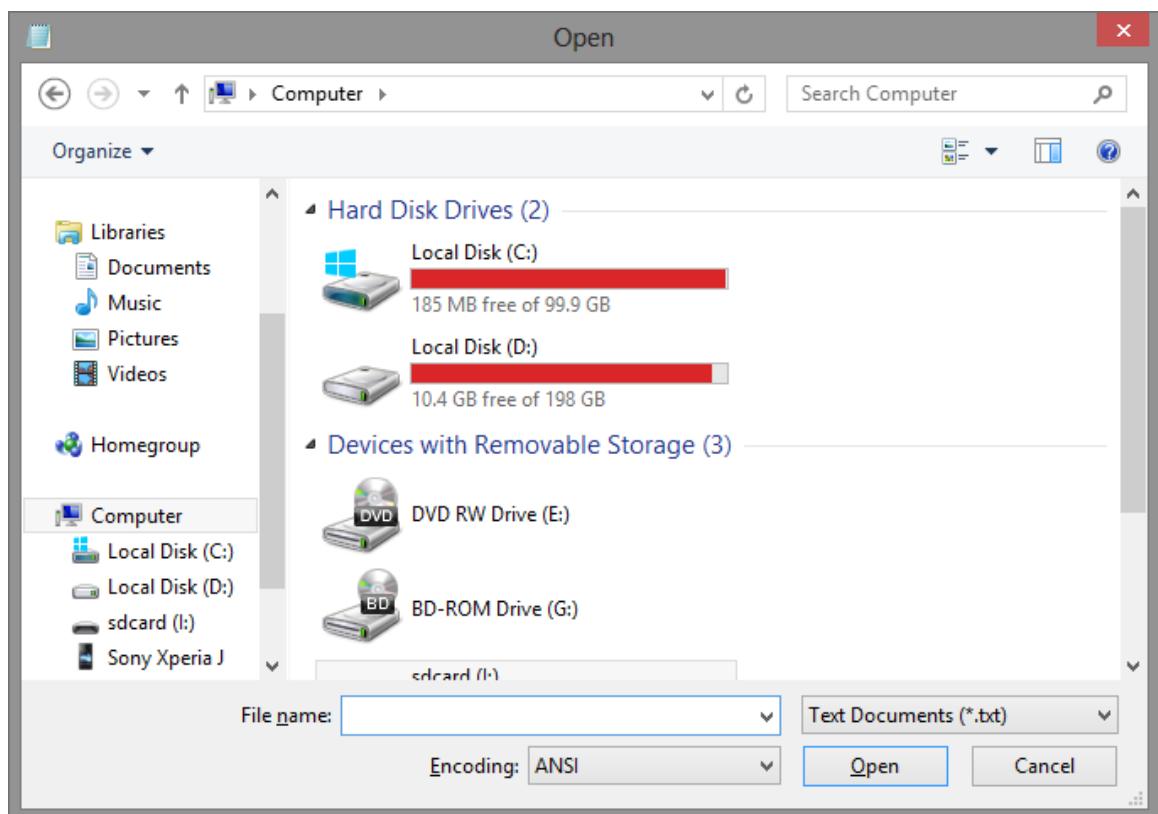


ภาพที่ 2-18 ตัวอย่างการใช้อุปกรณ์ในแท็บ Printing : PrintDialog

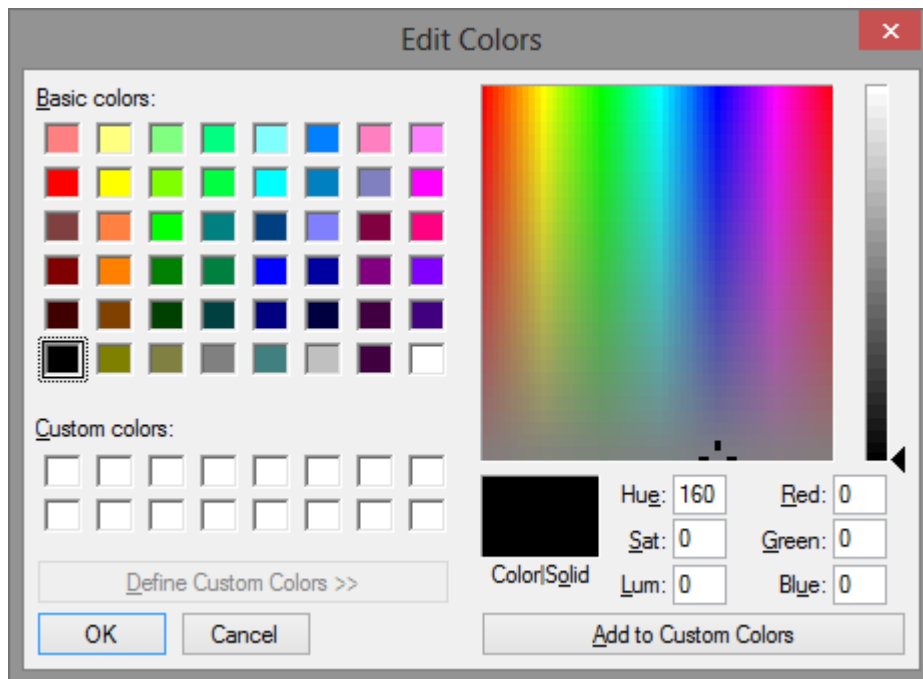
g. แท็บ Dialogs ทำหน้าที่เก็บกลุ่มออบเจกต์ที่ใช้สำหรับแสดงหน้าจอของระบบ



ภาพที่ 2-19 แท็บ Dialogs

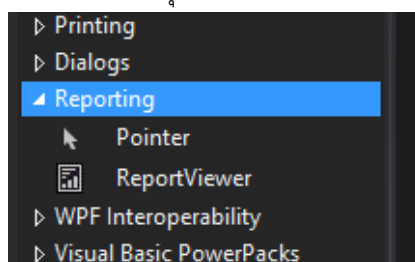


ภาพที่ 2-20 ตัวอย่างการใช้อุปกรณ์ในแท็บ Dialogs : OpenFileDialog



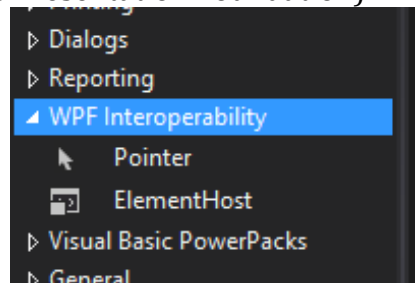
ภาพที่ 2-21 ตัวอย่างการใช้อุปกรณ์ในแท็บ Dialogs : ColorDialog

- h. แท็บ Reporting ทำหน้าที่เก็บกลุ่มออบเจกต์ที่ใช้สำหรับการแสดงตารางรายงาน



ภาพที่ 2-22 แท็บ Reporting

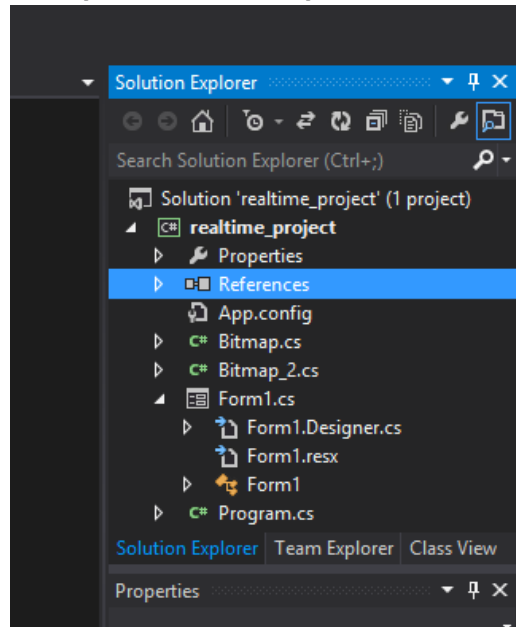
- i. แท็บ WPF Interoperability ทำหน้าที่เก็บกลุ่มออบเจกต์ที่ใช้แสดงแอปพลิเคชันแบบ WPF (Windows Presentation Foundation)



ภาพที่ 2-23 แท็บ WPF Interoperability

2.6.2 หน้าต่าง Solution Explorer

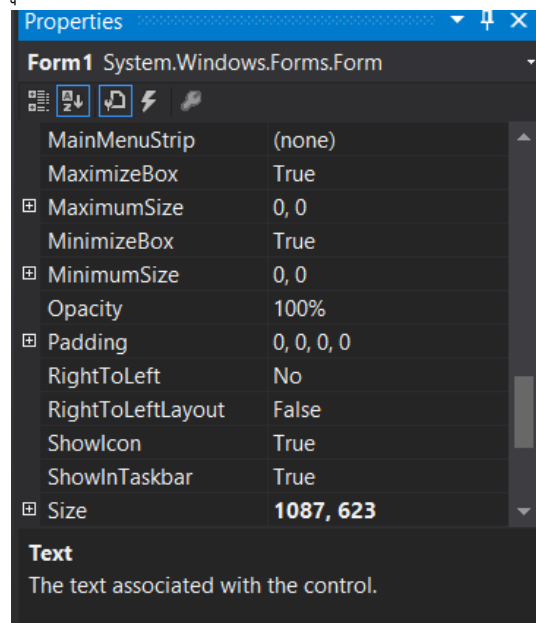
ทำหน้าที่แสดงรายการไฟล์ที่เกี่ยวข้องที่อยู่ในโปรเจกต์ เช่น ไฟล์รูปภาพ ไฟล์ของคลาสที่สร้างขึ้นมา ไฟล์แหล่งข้อมูล XML ฯลฯ ขึ้นอยู่กับว่าโปรเจกต์จะเรียกใช้อะไรบ้าง



ภาพที่ 2-24 หน้าต่าง Solution Explorer

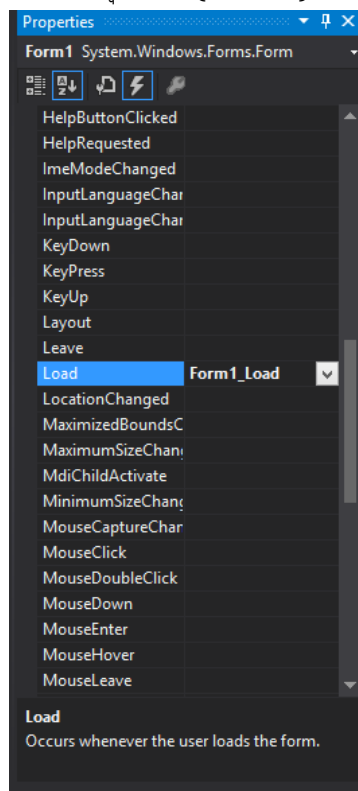
2.6.3 หน้าต่าง Properties และ Events

หน้าต่างคุณสมบัติทำหน้าที่แสดงคุณสมบัติ และเหตุการณ์ของคอนโทรลหรือออบเจกต์ตัวนั้นๆ สันนิษฐาน ดังภาพที่ 2-25



ภาพที่ 2-25 หน้าต่าง Properties แสดงถึงคุณสมบัติของ Form

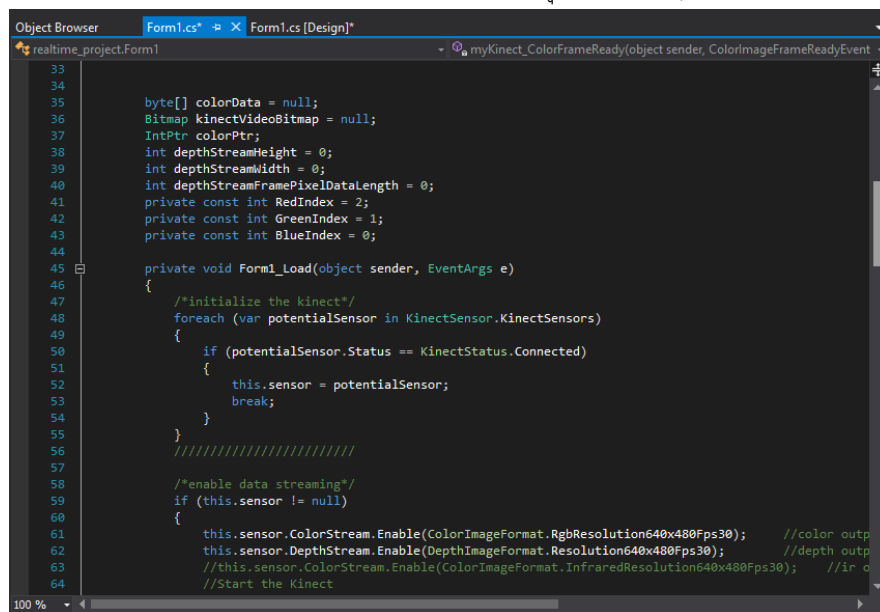
ในหน้าต่าง Properties จะมีส่วนของเหตุการณ์ (Events) ซึ่งเปิดได้โดยกดปุ่ม 



ภาพที่ 2-26 หน้าต่าง Properties แสดงถึงเหตุการณ์ของ Form

2.6.4 หน้าต่าง Editor

หน้าต่างสำหรับการเขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆ



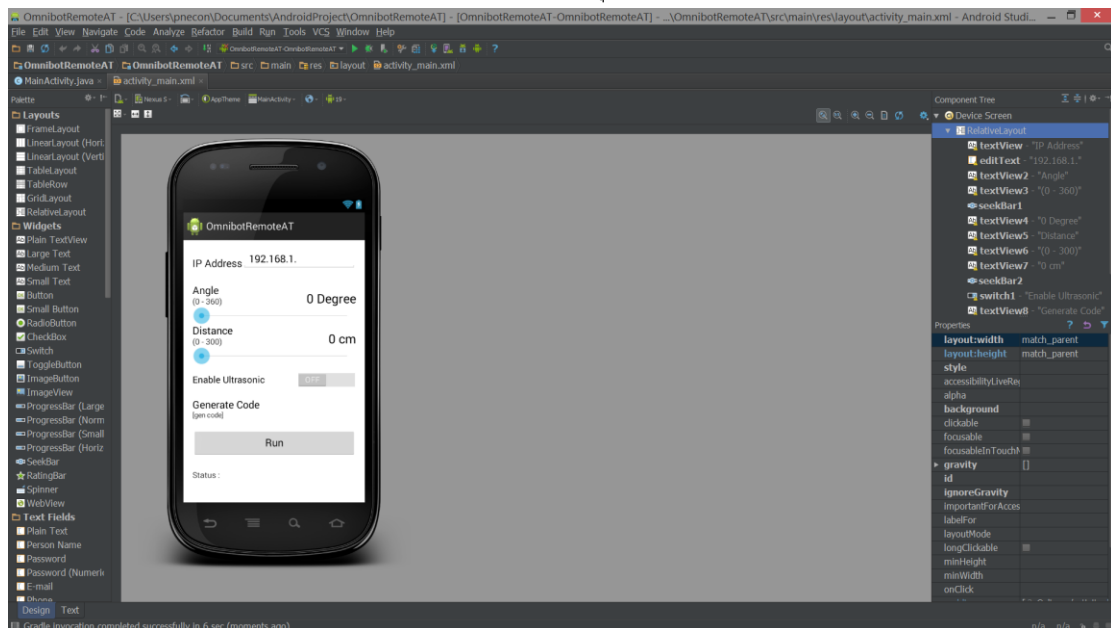
ภาพที่ 2-27 หน้าต่าง Editor แสดงตัวอย่างโค้ดโปรแกรม

2.7 โปรแกรม Android Studio

Android Studio ถือเป็นเครื่องมือพัฒนาแอปพลิเคชัน สำหรับพัฒนาโปรแกรมบนระบบปฏิบัติการ **Android** เปิดตัวครั้งแรกเมื่อวันที่ 16 พฤษภาคม 2556 ที่งานประชุม Google I/O โดยที่ตัวโปรแกรมสามารถใช้ได้บน **Windows, Mac OS X และ Linux**

ความสามารถโดยทั่วไปของ **Android Studio** อาทิเช่น

- สามารถลากวางตัวคอนโทรลต่างๆ ได้ (ลากปุ่มลงมาบนหน้าจออินเทอร์เน็ตเฟส)
- Refactoring** (ปรับโค้ดให้อ่านง่ายยิ่งขึ้น)
- แก้ไขหน้าจอได้หลายจอพร้อมกัน (ทำบนอุปกรณ์จอเล็ก และใหญ่พร้อมกัน)



ภาพที่ 2-28 โปรแกรม Android Studio ในส่วนของการออกแบบ User Interface

2.8 การหาความเร็วมอเตอร์

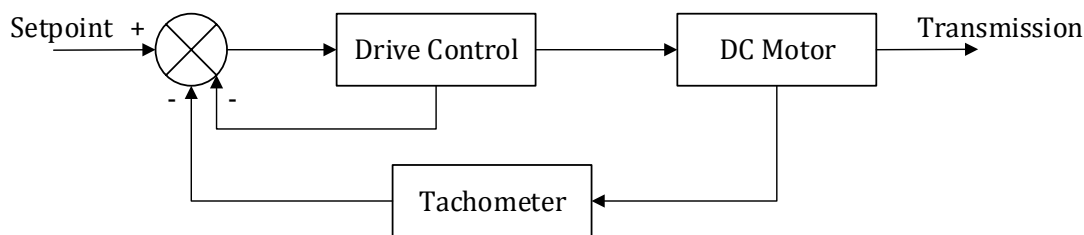
ในการทำงานของมอเตอร์จะมี 2 สภาวะคือ

2.8.1 Full Load คือ สภาวะกำลังที่มากที่สุดของมอเตอร์ที่สามารถทำได้ ในการเกิด **Full Load** นั้นจะทำให้มอเตอร์เกิดความร้อนในตัวแล้วจะทำให้มอเตอร์นั้นเกิดความเสียหาย หรือเรียกอาการเหล่านี้ว่า **Overload**

2.8.2 No Load คือ สภาวะที่มอเตอร์ทำงานโดยไม่มีผลกระทบทางกลภายนอก หรือ **Load** มากระทำขณะที่มอเตอร์กำลังทำงาน

Speed regulation คือการควบคุมระดับความเร็วของมอเตอร์โดยจะมีสมการดังนี้

$$\text{Speed regulation} = \frac{\text{No Load speed} - \text{Full Load Speed}}{\text{Full Load speed}} \times 100 \quad (2.11)$$



ภาพที่ 2-29 บล็อกไดอะแกรมของมอเตอร์

2.9 ระบบเครือข่ายไร้สาย (Wireless LAN Technology)

2.9.1 ประวัติความเป็นมา

ระบบเครือข่ายไร้สาย (Wireless LAN) ระบบเครือข่ายไร้สาย (Wireless LANs) เกิดขึ้นครั้งแรก ในปี ค.ศ. 1971 บนเกาะฮาวาย โดยโปรเจกต์ ของนักศึกษาของมหาวิทยาลัยฮาวาย ที่ชื่อว่า “ALOHNET” ขณะนั้นลักษณะการส่งข้อมูลเป็นแบบ Bi-directional ส่งไป-กลับง่ายๆ ผ่านคลื่นวิทยุ สื่อสารกันระหว่างคอมพิวเตอร์ 7 เครื่อง ซึ่งตั้งอยู่บนเกาะ 4 เกาะโดยรอบ และมีศูนย์กลางการเชื่อมต่ออยู่ที่เกาะหนึ่ง ที่ชื่อว่า Oahu

2.9.2 ความหมาย

ระบบเครือข่ายไร้สาย (WLAN = Wireless Local Area Network) คือ ระบบการสื่อสารข้อมูลที่มีรูปแบบในการสื่อสารแบบไม่ใช้สาย โดยการใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และคลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่อง ผ่านอากาศ, ทะลุกำแพง, เพดานหรือสิ่งก่อสร้างอื่นๆ โดยปราศจากความต้องการของการเดินสาย นอกจากนั้นระบบเครือข่ายไร้สายก็ยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบและแบบใช้สาย

ที่สำคัญก็คือ การที่ไม่ต้องใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก ไม่เหมือนระบบ LAN แบบใช้สาย ที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์

ปัจจุบันโลกของเราเป็นยุคแห่งการติดต่อสื่อสาร เทคโนโลยีต่างๆ เช่นโทรศัพท์มือถือ เป็นสิ่งจำเป็นต่อการดำเนินธุรกิจและการใช้ชีวิตประจำวัน ความต้องการข้อมูลและการบริการต่างๆ มีความจำเป็นสำหรับนักธุรกิจ เทคโนโลยีที่สนองต่อความต้องการเหล่านั้น มีมากมาย เช่น โทรศัพท์มือถือ เครื่องคอมพิวเตอร์โน้ตบุ๊ก เครื่องปาล์ม ได้ถูกนำมาใช้เป็นอย่างมากและ ผู้ที่น่าจะได้ประโยชน์จากการใช้ ระบบเครือข่ายไร้สาย มีมากมายไม่ว่าจะเป็น

- หมอหรือพยาบาลในโรงพยาบาล เพราะสามารถดึงข้อมูลมารักษาผู้ป่วยได้จาก เครื่องคอมพิวเตอร์โน้ตบุ๊ก ที่เชื่อมต่อกับ ระบบเครือข่ายไร้สายได้ทันที
- นักศึกษาในมหาวิทยาลัยก็สามารถใช้งานโน้ตบุ๊กเพื่อค้นคว้าข้อมูลในห้องสมุดของมหาวิทยาลัย หรือใช้อินเตอร์เน็ตจากสนามหญ้าในมหาวิทยาลัยได้

นักธุรกิจที่มีความจำเป็นต้องใช้งานเครื่องคอมพิวเตอร์นอกสถานที่ที่ทำงานปกติ ไม่ว่าจะเป็นการนำเสนองานยังบริษัทลูกค้า หรือการนำเครื่องคอมพิวเตอร์ติดตัวไปงานประชุมสัมมนาต่างๆ

บุคคลเหล่านี้มีความจำเป็นที่จะต้องเชื่อมต่อเข้ากับเครือข่ายคอมพิวเตอร์ ไม่ว่าจะเป็นเครือข่ายคอมพิวเตอร์ขององค์กรซึ่งอยู่ห่างออกไปหรือเครือข่ายคอมพิวเตอร์สาธารณะ เช่นเครือข่ายอินเทอร์เน็ต เทคโนโลยีเครือข่ายไร้สายจึงน่าจะอำนวยความสะดวกให้กับบุคคลเหล่านี้ได้ ซึ่งในปัจจุบันได้มีการเปิดให้บริการเชื่อมต่อเครือข่ายอินเทอร์เน็ตแบบไร้สาย ตามสนามบินใหญ่ทั่วโลก และนำมาใช้งานแพร่หลายในห้างสรรพสินค้า และโรงแรมต่างๆแล้ว

2.9.3 ประโยชน์ของระบบเครือข่ายไร้สาย

2.9.3.1 Mobility Improves Productivity and Service

มีความคล่องตัวสูง ดังนั้นไม่ว่าเราจะเคลื่อนที่ไปที่ไหน หรือเคลื่อนย้ายคอมพิวเตอร์ไปตำแหน่งใด ก็ยังมีการเชื่อมต่อ กับเครือข่ายตลอดเวลา ตราบใดที่ยังอยู่ในระยะการส่งข้อมูล

2.9.3.2 Installation Speed and Simplicity

สามารถติดตั้งได้ง่ายและรวดเร็ว เพราะ ไม่ต้องเสียเวลาติดตั้งสายเคเบิล และไม่รกรุงรัง

2.9.3.3 Installation Flexibility

สามารถขยายระบบเครือข่ายได้ง่าย เพราะเพียงแค่มียุติการเชื่อมต่อเข้ากับโน้ตบุ๊ก หรือพีซี ก็เข้าสู่เครือข่ายได้ทันที

2.9.3.4 Reduced Cost-of-Ownership

ลดค่าใช้จ่ายโดยรวม ที่ผู้ลงทุนต้องลงทุน ซึ่งมีราคาสูง เพราะในระยะยาวแล้ว ระบบเครือข่ายไร้สายไม่จำเป็นต้องเสียค่าบำรุงรักษาและการขยายเครือข่ายก็ลงทุนน้อยกว่าเดิมหลายเท่า เนื่องด้วยความสะดวกง่ายในการติดตั้ง

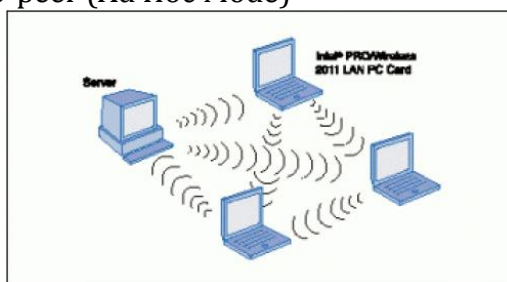
2.9.3.5 Scalability

เครือข่ายไร้สายทำให้องค์กรสามารถปรับขนาดและความเหมาะสมได้ง่ายไม่ยุ่งยาก เพราะสามารถโยกย้ายตำแหน่งการใช้งานโดยเฉพาะระบบที่มีการเชื่อมระหว่างจุดต่อจุด เช่น ระหว่างตึก

ระบบเครือข่ายไร้สาย เป็นระบบเครือข่ายคอมพิวเตอร์ขนาดเล็ก ที่ประกอบไปด้วยอุปกรณ์ไม่มากนัก และมักจำกัดอยู่ในอาคารหลังเดียวหรืออาคารในละแวกเดียวกัน การใช้งานที่น่าสนใจที่สุดของเครือข่ายไร้สายก็คือ ความสะดวกสบายที่ไม่ต้องติดอยู่กับที่ ผู้ใช้สามารถเคลื่อนที่ไปมาได้โดยที่ยังสื่อสารอยู่ในระบบเครือข่าย

2.10 รูปแบบการเชื่อมต่อของระบบเครือข่ายไร้สาย

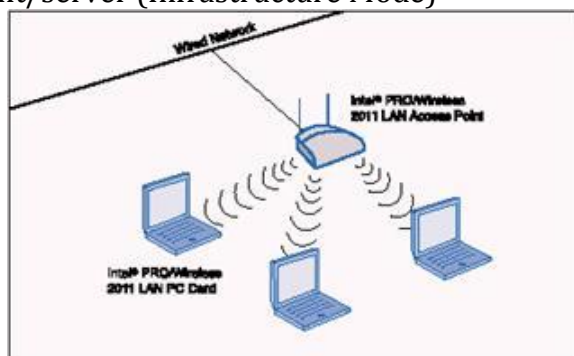
2.10.1 Peer-to-peer (Ad Hoc Mode)



ภาพที่ 2-30 รูปแบบการเชื่อมต่อแบบ Peer-to-Peer

รูปแบบการเชื่อมต่อระบบแลนไร้สายแบบ **Peer to Peer** เป็นลักษณะ การเชื่อมต่อแบบ โครงข่ายโดยตรงระหว่างเครื่องคอมพิวเตอร์ จำนวน 2 เครื่องหรือมากกว่านั้น เป็นการใช้งานร่วมกัน ของ **Wireless Adapter Cards** โดยไม่ได้มีการเชื่อมต่อกับเครือข่ายแบบใช้สายเลย โดยที่เครื่อง คอมพิวเตอร์แต่ละเครื่องจะมีความเท่าเทียมกัน สามารถทำงานของตนเองได้และขอใช้บริการเครื่อง อื่นได้เหมาะสำหรับการนำมาใช้งานเพื่อจุดประสงค์ในด้านความเร็วหรือติดตั้งได้โดยง่ายเมื่อไม่ มีโครงสร้างพื้นฐานที่จะรองรับ ยกตัวอย่างเช่น ในศูนย์ประชุม, หรือการประชุมที่จัดขึ้นนอกสถานที่

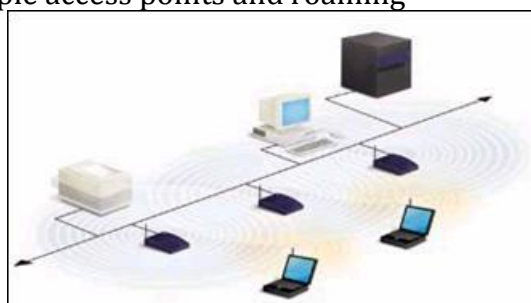
2.10.2 Client/server (Infrastructure Mode)



ภาพที่ 2-31 รูปแบบการเชื่อมต่อแบบ Client/Server

ระบบเครือข่ายไร้สายแบบ **Client / server** หรือ **Infrastructure mode** เป็นลักษณะการ รับส่งข้อมูลโดยอาศัย **Access Point (AP)** หรือเรียกว่า “Hot spot” ทำหน้าที่เป็นสะพานเชื่อมต่อ ระหว่างระบบเครือข่ายแบบใช้สายกับเครื่องคอมพิวเตอร์ลูกข่าย (**Client**) โดยจะกระจายสัญญาณ คลื่นวิทยุเพื่อ รับ-ส่งข้อมูลเป็นรัศมีโดยรอบ เครื่องคอมพิวเตอร์ที่อยู่ในรัศมีของ **AP** จะกลายเป็น เครือข่ายกลุ่มเดียวกันทันที โดยเครื่องคอมพิวเตอร์ จะสามารถติดต่อกัน หรือติดต่อกับ **Server** เพื่อแลกเปลี่ยนและค้นหาข้อมูลได้ โดยต้องติดต่อผ่าน **AP** เท่านั้น ซึ่ง **AP 1 จุด** สามารถให้บริการ เครื่องลูกข่ายได้ถึง 15-50 อุปกรณ์ ของเครื่องลูกข่าย เหมาะสำหรับการนำไปขยายเครือข่ายหรือใช้ ร่วมกับระบบเครือข่ายแบบใช้สายเดิมในออฟฟิต, ห้องสมุด หรือในห้องประชุม เพื่อเพิ่ม ประสิทธิภาพในการทำงานให้มากขึ้น

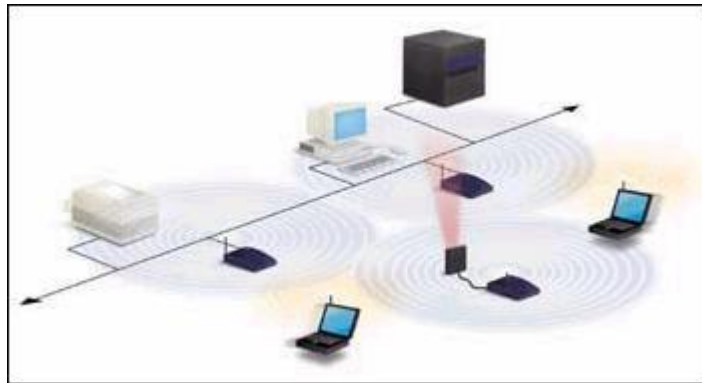
2.10.3 Multiple access points and roaming



ภาพที่ 2-32 รูปแบบการเชื่อมต่อแบบ Multiple access points and roaming

โดยทั่วไปแล้ว การเชื่อมต่อสัญญาณระหว่างเครื่องคอมพิวเตอร์ กับ Access Point ของเครือข่ายไร้สายจะอยู่ในรัศมีประมาณ 500 ฟุต ภายในอาคาร และ 500 ฟุต ภายนอกอาคาร หากสถานที่ที่ติดตั้งมีขนาดกว้าง มากๆ เช่นคลังสินค้า บริเวณภายในมหาวิทยาลัย สนามบิน จะต้องมีการเพิ่มจุดติดตั้ง AP ให้มากขึ้น เพื่อให้การรับส่งสัญญาณในบริเวณของเครือข่ายขนาดใหญ่ เป็นไปอย่างครอบคลุมทั่วถึง

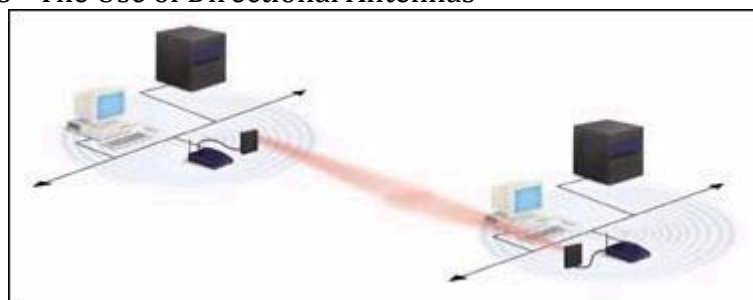
2.10.4 Use of an Extension Point



ภาพที่ 2-33 รูปแบบการเชื่อมต่อแบบ Use of an Extension Point

กรณีที่โครงสร้างของสถานที่ติดตั้งเครือข่ายแบบไร้สายมีปัญหาผู้ออกแบบระบบอาจจะใช้ Extension Points ที่มีคุณสมบัติเหมือนกับ Access Point แต่ไม่ต้องผูกติดไว้กับเครือข่ายไร้สาย เป็นส่วนที่ใช้เพิ่มเติมในการรับส่งสัญญาณ

2.10.5 The Use of Directional Antennas



ภาพที่ 2-34 รูปแบบการเชื่อมต่อแบบมีการใช้ Antennas

ระบบแลนไร้สายแบบนี้เป็นแบบใช้เสาอากาศในการรับส่งสัญญาณระหว่างอาคารที่อยู่ห่างกัน โดยการติดตั้งเสาอากาศที่แต่ละอาคาร เพื่อส่งและรับสัญญาณระหว่างกัน

2.11 ฟัซซีลอจิก (Fuzzy Logic)

ฟัซซีลอจิกพัฒนาจาก ทฤษฎีเซตที่ขอบเขตไม่เด่นชัด โดยเป็นการใช้เหตุผลแบบประมาณ ซึ่งแตกต่างจากการใช้เหตุผลแบบเด็ดขาดในลักษณะ ถูก/ผิด ใช่/ไม่ใช่ ของ ตรรกศาสตร์แบบฉบับ (Classical Logic) ฟัซซีลอจิกนั้นสามารถถือเป็นการประยุกต์ใช้งานเซตที่ขอบเขตไม่เด่นชัด เพื่อจำลองการตัดสินใจของผู้เชี่ยวชาญ ต่อปัญหาที่ซับซ้อน

ค่าระดับความจริง ในฟัซซีลอจิกนั้นมักจะสับสนกับ ค่าความน่าจะเป็น ซึ่งมีแนวความคิดที่แตกต่างกัน ค่าระดับความจริงคลุมเครือนั้นใช้ในการระบุ ค่าความเป็นสมาชิก ของเซต แต่ค่าความน่าจะเป็นนั้นระบุความเป็นไปได้ของสภาพการณ์แต่ละรูปแบบที่อาจจะเกิดขึ้น ตัวอย่างเช่น สมมติว่า นาย ก กำลังเดินเข้าบ้าน สถานะของนาย ก ตามตรรกศาสตร์แบบฉบับ คือ “อยู่ในบ้าน” หรือ “อยู่นอกบ้าน” แต่หากเขากำลังยืนอยู่ระหว่างช่องประตู เราอาจพิจารณาได้ว่าเขา “อยู่ในบ้านบางส่วน” ระดับของสถานะกึ่งนี้ จะระบุด้วยค่าความเป็นสมาชิกของเซตที่ขอบเขตไม่เด่นชัด สมมติเขาเพิ่งจะก้าวปลายนิ้วเท้าผ่านข้ามธรณีประตูเข้าบ้าน เราอาจกล่าวว่า นาย ก นั้น 0.99 “อยู่นอกบ้าน” ซึ่งต่างจากความน่าจะเป็นของเหตุการณ์สุ่ม (เช่น ความน่าจะเป็นระบุผลลัพธ์ของการโยนเหรียญ แต่ผลลัพธ์จะออก หัว หรือ ก้อย) หากพิจารณาความน่าจะเป็นที่นาย ก “อยู่นอกบ้าน” และ “อยู่ในบ้าน” จะออกผลลัพธ์เป็น นาย ก อยู่นอกบ้าน หรือ ในบ้าน ไม่ได้จำลองสถานะกึ่ง คือ กำลังยืนอยู่ที่ประตู เซตที่ขอบเขตไม่เด่นชัด นั้นมีหลักการพื้นฐานจากเซตที่มีขอบเขตคลุมเครือไม่ชัดเจน ไม่ได้มีพื้นฐานจากการสุ่ม

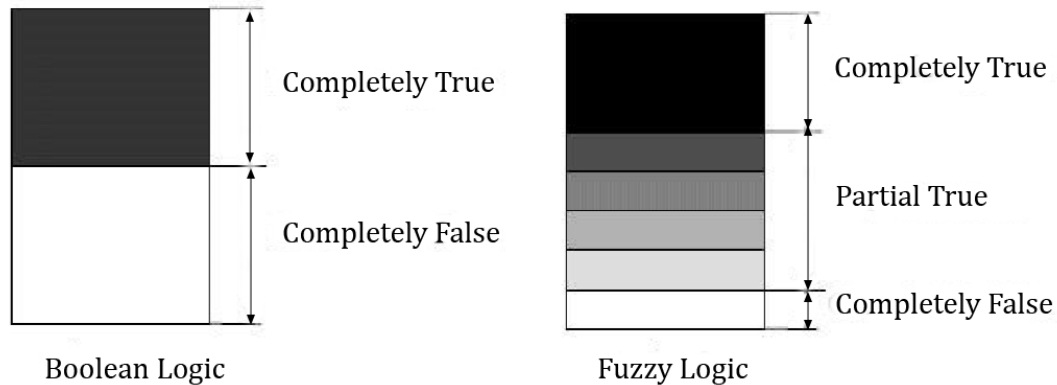
ฟัซซีลอจิกนั้นสามารถระบุค่าความเป็นสมาชิกของเซต (Set Membership Values) ด้วยค่าระหว่าง 0 และ 1 ทำให้เกิดระดับกึ่งในลักษณะของ สีเทา นอกจาก ขาว และ ดำ ซึ่งมีประโยชน์ในการจำลองระดับซึ่งสามารถระบุด้วยคำพูด “เล็กน้อย” “ค่อนข้าง” “มาก” โดยใช้ค่าความเป็นสมาชิกของเซตบางส่วน ฟัซซีลอจิกนี้มีความสัมพันธ์กับ เซตที่ขอบเขตไม่เด่นชัด และ ทฤษฎีความเป็นไปได้ (Possibility Theory) ซึ่งคิดค้นขึ้นในปี ค.ศ. 1965 โดยศาสตราจารย์ ลอตฟี ซาเดห์ แห่งมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เบิร์กลีย์

ฟัซซีลอจิก ถึงแม้ว่าจะได้รับการยอมรับค่อนข้างกว้างขวาง แต่ก็ยังถูกโต้แย้งจากบางกลุ่ม เช่น จากวิศวกรระบบควบคุม ในเรื่องของกรอบอธิบายพฤติกรรมต่างๆ และ จากนักสถิติบางกลุ่ม ซึ่งเชื่อมั่นว่าทฤษฎีความน่าจะเป็น เป็นวิธีทางคณิตศาสตร์ที่เคร่งครัดเพียงวิธีเดียว ในการจำลองความไม่แน่นอน (Uncertainty) นอกจากนั้นแล้ว ก็ยังมีการวิเคราะห์วิจารณ์ว่า เซตที่ขอบเขตไม่เด่นชัดนั้นไม่ได้เป็นซูเปอร์เซตของ ทฤษฎีเซตสามัญ เนื่องจาก ฟังก์ชันภาวะสมาชิก นั้นกำหนดในรูปของเซตแบบดั้งเดิม

2.11.1 พื้นฐานแนวคิดแบบฟัซซี

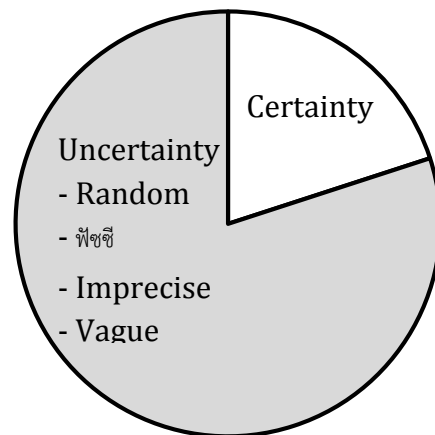
ฟัซซีลอจิก เป็นเครื่องมือที่ช่วยในการตัดสินใจภายใต้ความไม่แน่นอนของข้อมูล โดยยอมให้มีความยืดหยุ่นได้ ใช้หลักเหตุผลที่คล้ายการเลียนแบบวิถีความคิดที่ ซับซ้อนของมนุษย์ ฟัซซีลอจิก มีลักษณะที่พิเศษกว่าตรรกะแบบจริงแท้ (Boolean logic) เป็นแนวคิด ที่มีการต่อขยายในส่วน of ความจริง (Partial True) โดยค่าความจริงจะอยู่ในช่วงระหว่างจริง (Completely

true) กับเท็จ (Completely False) ส่วนตรรกศาสตร์เดิมจะมีค่าเป็นจริงกับเท็จเท่านั้น แสดงดังภาพที่ 2-35



ภาพที่ 2-35 ตรรกะแบบจริงเท็จ (Boolean Logic) กับตรรกะแบบฟัซซี

ความเป็นฟัซซี (Fuzziness) มีชื่อเรียกว่า มัลติวาลานซ์ (Multivalence) ซึ่งมีค่าที่เป็นสมาชิก มากกว่า 2 ค่า และแตกต่างกับไบวาลานซ์ (Bivalence) ที่มีความเป็นสมาชิกเพียง 2 ค่า ฟัซซีเซต (ฟัซซี Set) เป็นเครื่องมือทางคณิตศาสตร์ที่สื่อถึง “ความไม่แน่นอน (Uncertainty)” สามารถที่ไม่ใช่เพียง 2 กรณี ซึ่งหากกำหนดว่า คนที่อ้วนคือคนที่น้ำหนักมากกว่า 75 กิโลกรัม คอมพิวเตอร์จะ ให้ผลว่าคนที่น้ำหนัก 74.50 กิโลกรัม ไม่จัดเป็นคนที่อ้วน จะสร้างและกำหนดรูปแบบ (Modeling) ของลักษณะความไม่แน่นอนที่เป็นความคลุมเครือ ความไม่ตายตัว รวมถึงความขาดข้อมูลบางส่วน โดย ทฤษฎีของฟัซซีเซตจะใช้ลักษณะความหมายตัวแปร (Linguistic) มากกว่าปริมาณ (Quantitative) ของ ตัวแปร เช่น การหาความหมายของ “คนที่อ้วน” เราไม่สามารถนิยามค่าความอ้วนที่ตรงกันและระบุเป็น หนึ่งเดียว (Identical) สำหรับคนที่อ้วน นาย ก. จะให้ความหมายของ “คนอ้วน” หมายถึงคนที่มี น้ำหนักมากกว่า 70 กิโลกรัม นาย ข. ให้ความหมายว่าเป็นคนที่มีน้ำหนักมากกว่า 75 กิโลกรัม ซึ่งทั้งสองคนต่างแสดงความหมายของคำว่าคนที่อ้วนโดยเปรียบเทียบและในมุมมองของตัวเองตามน้ำหนักของตน ในการทำงานในมุมมองแบบฐานสอง (Binary Sense) จะได้ผลเป็น ใช่ หรือ ไม่ แต่จะเห็นว่า บุคคลนี้เป็นคนอ้วนน้ำหนักเกือบจะ 75 กิโลกรัม และถึงแม้ว่าบุคคลนี้จะน้ำหนัก 75 กิโลกรัม แต่หากพิจารณาจากกลุ่มคนที่มีน้ำหนักเฉลี่ย 90 กิโลกรัม บุคคลนี้ก็จะไม่จัดอยู่ในกลุ่มคนที่อ้วน แสดงให้เห็นว่า ความอ้วนไม่ได้มีลักษณะความไม่แน่นอนแบบสุ่ม จากการศึกษาค้นคว้าต่างๆ ไปจะแสดงถึง รูปแบบลักษณะการกระจายของปัญหาภาพที่ 2-36 เป็นการแสดงให้เห็นว่าแนวทางในการตัดสินใจของปัญหาทั้งหมดมีเพียงส่วนน้อย ที่เป็นสิ่งที่แน่นอน (Certainty) ที่เหลือคือสิ่งที่ไม่แน่นอนซึ่งประกอบด้วยความไม่แน่นอนที่มีลักษณะแบบสุ่ม และความไม่แน่นอนที่มีลักษณะเป็นฟัซซีหรือคลุมเครือ ซึ่งมีมากกว่าร้อยละ 40 เพราะปัญหา ส่วนมากเกี่ยวข้องกับการตัดสินใจของมนุษย์ซึ่งจะตัดสินใจตามพื้นฐานความคิดของตนเป็นหลัก

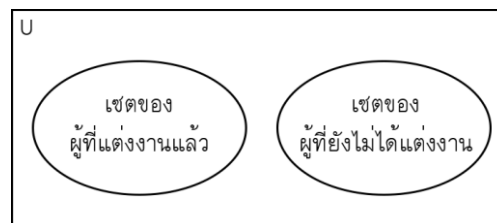


ภาพที่ 2-36 ความไม่แน่นอน (Uncertainty)

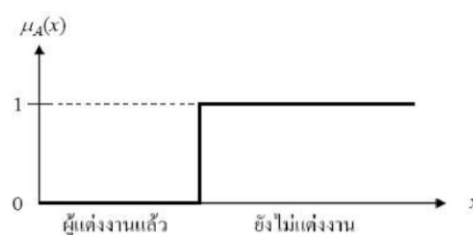
ฟัชซีจะสร้างวิธีทางคณิตศาสตร์ที่แสดงถึงความคลุมเครือ ความไม่แน่นอนของระบบที่ เกี่ยวข้องกับความคิดความรู้สึกของมนุษย์ เมื่อพิจารณาส่วนประกอบต่าง ๆ ในความไม่แน่นอนเพื่อ กำหนดเงื่อนไขในการตัดสินใจ (Decision Making) โดยอาศัยเซตของความไม่ เป็นสมาชิก (Set Membership)

2.11.2 เซตแบบฉบับ

ในเซตแบบฉบับ (Classical Set) หรือเซตทวินัย (Crisp Set) เป็นเซตที่มีค่าความ เป็นสมาชิก เป็น 0 หรือ 1 $\{0, 1\}$ เท่านั้น เซตในทฤษฎีเซตแบบฉบับจะมีขอบเขตแบบแข็ง (Sharp Boundary) ซึ่งเป็นขอบเขตที่ตัดขาดจากกันแบบทันทีทันใด เซตแบบฉบับมีการกำหนดค่า ความเป็นสมาชิกตาม แนวคิดเลขฐานสอง โดยที่ตัวแปรหนึ่ง ๆ จะมีค่าความเป็นสมาชิกเพียงสองค่า คือ 0 ไม่เป็นสมาชิก และ 1 เป็นสมาชิก ตัวอย่างเช่น เซตของคู่แต่งงาน จะสามารถบอกได้ว่าอย่าง แน่ชัดว่าเป็นกลุ่มผู้ แต่งงานหรือไม่แต่งงาน



ภาพที่ 2-37 ตัวอย่างเซตต้นฉบับ



ภาพที่ 2-38 ฟังก์ชันความเป็นสมาชิกในเซตผู้ที่ไม่แต่งงาน

ภาพที่ 2-37 แสดงตัวอย่างของเซตย่อยสองเซต คือเซตของผู้ที่แต่งงานและเซตของผู้ที่ไม่แต่งงาน จะเห็นได้ว่าคนหนึ่งคนจะเป็นสมาชิกภาพได้เพียงเซตเดียวเท่านั้น แต่งงานหรือไม่แต่งงาน ใน ภาพที่ 2-38 แสดงฟังก์ชันความเป็นสมาชิกของเซตผู้ที่ไม่แต่งงาน จากภาพจะเห็นได้ว่า ผู้ที่แต่งงานแล้ว จะมีค่าความเป็นสมาชิกในเซตของผู้ไม่แต่งงานเป็น 0 ส่วนผู้ที่ไม่แต่งงานมีค่าความเป็นสมาชิกภาพ ของเซตผู้ที่ไม่แต่งงานเป็น 1 ค่าความเป็นสมาชิกของทั้งสองเซตจะตัดขาดจากกัน อย่างทันทีทันใด รูปแบบคณิตศาสตร์ของเซตแบบฉบับมีรูปดังนี้

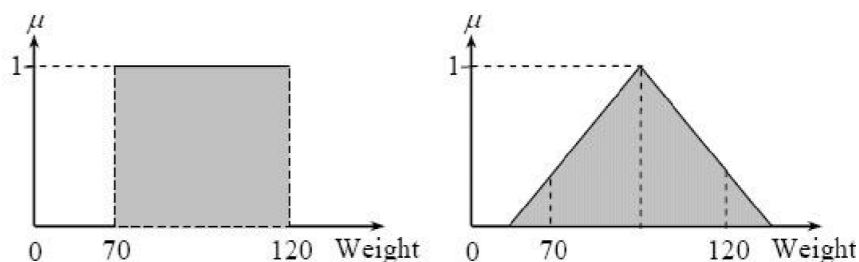
$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.12)$$

เมื่อ A เป็นเซตแบบฉบับหรือเซตแบบทวินัย x เป็นสมาชิกในเซต μ_A เป็นค่าความเป็นสมาชิกในเซต และ $\mu_A(x)$ เป็นฟังก์ชันความเป็นสมาชิกในเซต A

2.11.3 ฟัชซีเซต

เป็นเซตที่มีขอบเขตที่ราบเรียบ ทฤษฎีฟัชซีเซตจะครอบคลุมทฤษฎีเซต แบบฉบับ โดยฟัชซีเซตยอมให้มีค่าความเป็นสมาชิกของเซตระหว่าง 0 และ 1 ในโลกแห่งความเป็นจริงเซตไม่ใช่มีเฉพาะเซตแบบฉบับเท่านั้น จะมีเซตแบบฟัชซีด้วยฟัชซีเซต จะมีขอบเขตแบบฟัชซีไม่ใช่เปลี่ยนแปลงทันทีทันใดจากขาวเป็นดำ ตัวอย่างเช่น เซตของคู่แต่งงานที่มีความสุข จะเห็นได้ว่าสมาชิกในเซตนี้จะไม่มีเฉพาะคู่แต่งงานที่มีความสุขระดับเดียวกันหมด บางคู่จะมีความสุขมาก บางคู่มีความสุขน้อย แตกต่างกันไป การใช้เซตแบบดั้งเดิมจึงไม่เหมาะสม

ยกตัวอย่างเกี่ยวกับความอ้วน นิยามคำว่าคนอ้วนในเซตทวินัยอาจกำหนดเป็นคนที่ มีน้ำหนัก ตั้งแต่ 70 ถึง 120 กิโลกรัม โดยนิยามแบบฟัชซีSetอาจกำหนดเป็นคนที่มีความอ้วนประมาณ 80 กิโลกรัม ซึ่งเป็นการให้นิยามที่ไม่แสดงถึงขอบเขตที่แน่นอน



ภาพที่ 2-39 การกำหนดค่าความเป็นสมาชิกของเซตทวินัยและเซตแบบฟัชซี

นิยามของฟัชซีเซต : กำหนดให้ X เป็นเซตที่ไม่ว่างฟัชซีSet A สามารถแสดงลักษณะเฉพาะ ได้จากฟังก์ชันความเป็นสมาชิก

$$\mu_A(x) : X \rightarrow [0,1] \quad (2.13)$$

เมื่อ $\mu_A(x)$ สามารถตีความเป็นค่าของความเป็นสมาชิกภาพของตัวประกอบ x ใน ฟัซซีเซต $\underline{A}$ สำหรับแต่ละ (อ่านว่า “ x เป็นสมาชิกของ X ”) ฟัซซีเซต สามารถเขียนเป็นเซตของคู่ลำดับ (Tuples)

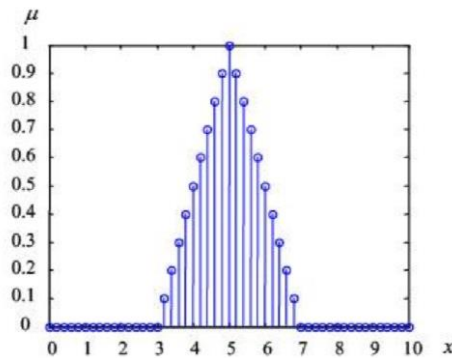
$$\underline{A} = \{(x, \mu_{\underline{A}}(x) \mid x \in X\} \quad (2.14)$$

เมื่อ $\underline{A}$ หมายถึงฟัซซีเซต $\underline{A}$ x หมายถึงสมาชิกของเซต (Set Membership) $\mu_A(x)$ หมายถึง ฟังก์ชัน ความเป็นสมาชิก (Membership Function) $\mu_A(x)$ บางครั้งแทนด้วย $\underline{A}(x)$ X หมายถึงเอกภพ ลัทธิพัทธ์ (Universe) หรือประชากร

ถ้า $X = \{x_1, x_2, x_3, \dots, x_n\}$ เป็นเซตจำกัด และ $\underline{A}$ เป็นฟัซซีเซตใน X ซึ่งเป็นชนิดวิฤต (Discrete) และจำกัดสัญกรณ์ (Notation) ของฟัซซีเซต เขียนได้เป็น

$$\underline{A} = \left\{ \frac{\mu_{\underline{A}}(x_1)}{x_1} + \frac{\mu_{\underline{A}}(x_2)}{x_2} + \dots + \frac{\mu_{\underline{A}}(x_n)}{x_n} \right\} = \left\{ \sum_{i=1}^n \frac{\mu_{\underline{A}}(x_i)}{x_i} \right\} \quad (2.15)$$

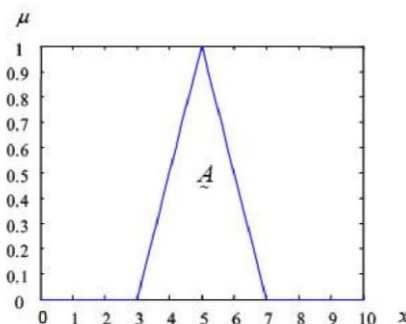
เมื่อพจน์ $\mu_{\underline{A}}(x_i)/x_i, i=1, 2, \dots, n$ หมายถึงค่าความเป็นสมาชิก $\mu_{\underline{A}}(x_i)$ ของ x_i ใน เซต $\underline{A}$ และเครื่องหมายบวก “+” หมายถึงยูเนียน (Union)



ภาพที่ 2-40 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซี แบบวิฤต $\underline{A}$

ถ้าเอกภพลัทธิพัทธ์ X เป็นต่อเนื่อง (Continuous) สัญกรณ์ (Notation) ของฟัซซี Set $\underline{A}$ เขียนได้เป็น

$$\underline{A} = \left\{ \int \frac{\mu_{\underline{A}}(x)}{x} \right\} \quad (2.16)$$



ภาพที่ 2-41 ฟังก์ชันความเป็นสมาชิกของเซตฟัซซี แบบต่อเนื่อง $\underline{A}$

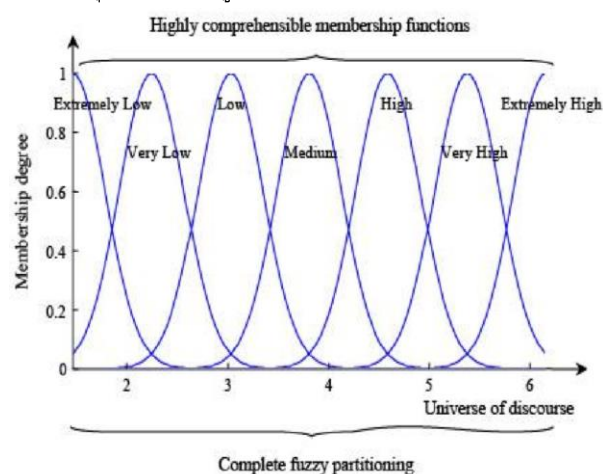
ทฤษฎีฟัซซีเซต สามารถแก้ปัญหาข้อจำกัดของเซตแบบดั้งเดิมได้ โดยฟัซซีเซต ยอมให้มีค่าหรือ ดีกรีของความเป็นสมาชิก (Degree of Membership) ซึ่งแสดงด้วยค่าตัวเลขระหว่าง 0 และ 1 หรือ เขียนเป็นสัญลักษณ์ $[0, 1]$, โดย 0 หมายถึง ไม่เป็นสมาชิกในเซต 1 หมายถึง เป็นสมาชิกในเซต และค่าระหว่าง 0 กับ 1 เป็นสมาชิกบางส่วนในเซต การทำเช่นนี้ ทำให้เกิดความราบเรียบในการเปลี่ยนจาก พื้นที่นอกเซตไปอยู่ในเซตของสมาชิกต่างๆ โดยมีฟังก์ชันสมาชิก (Membership Function) เป็นฟังก์ชันจัดเทียบ (Mapping Function) วัตถุในโดเมนใด ๆ ให้เป็นค่าความเป็นสมาชิกในฟัซซีSet ความเป็นสมาชิกสำหรับฟัซซีเซต มีจำนวนระดับความเป็นสมาชิกเป็นอนันต์ คือค่าต่อเนื่องในช่วง ตั้งแต่ 0 ถึง 1 ซึ่งครอบคลุมการกำหนดสมาชิกแบบฉบับ และเซตแบบฉบับหรือเซตทวินัย (Crisp Set) จะกำหนดตามดังสมการที่ (2.17)

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.17)$$

เมื่อ A เป็นเซตแบบฉบับหรือเซตแบบทวินัย x เป็นสมาชิกในเซต μ_A เป็นค่าความเป็นสมาชิกในเซตและ $\mu_A(x)$ เป็นฟังก์ชันความเป็นสมาชิกในเซต A

2.11.4 ตัวแปรภาษา (Linguistic Variable)

เซตแบบฟัซซีสามารถประยุกต์ใช้ในการอธิบายค่าของตัวแปรเช่นเดียวกับเซตแบบดั้งเดิม เช่น ประโยค “อุณหภูมิในห้องเย็น” คำว่า “เย็น” เป็นคำที่ใช้แสดงปริมาณอุณหภูมิในทางรูปนัย สามารถ เขียนได้เป็น ปริมาณอุณหภูมิในห้องเย็นหรือ Temperature Quantity is Cold ตัวแปร Temperature Quantity เป็นตัวแปรภาษา ซึ่งเป็นแนวคิดที่สำคัญมากในตรรกะแบบฟัซซี ตัวแปรภาษาช่วยกำหนดค่าของสิ่งที่จะอธิบายทั้งในรูปคุณภาพ โดยใช้พจน์ภาษา (Linguistic Term) และในรูปปริมาณ โดยใช้ฟังก์ชันความเป็นสมาชิกซึ่ง แสดงความของของเซตแบบฟัซซีพจน์ภาษาใช้สำหรับการแสดงแนวคิดและองค์ความรู้ในการสื่อสารของมนุษย์ส่วนฟังก์ชันความเป็นสมาชิก มีประโยชน์ในการจัดการกับอินพุตที่เป็นข้อมูลเชิงตัวเลข

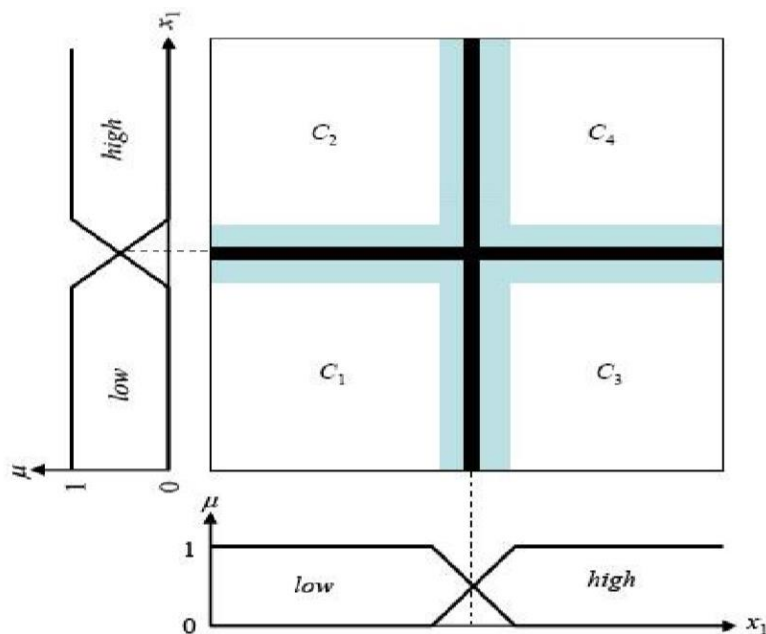


ภาพที่ 2-42 ตัวอย่างตัวแปรภาษา

ตัวแปรภาษาเป็นการประกอบกัน (Composition) ของตัวแปรสัญลักษณ์ (Symbolic Variable) และตัวแปรเชิงเลข (Numerical Variable) ตัวอย่างตัวแปรสัญลักษณ์ เช่น “รูปร่าง เป็น ทรงกระบอก” (Shape = Cylinder) คำว่า “รูปร่าง” เป็นตัวแปรที่บอกถึงรูปร่างของวัตถุ ตัวอย่างตัวแปรเชิงเลข เช่น “ความสูงเท่ากับ 4 ฟุต” (Height = 4') ตัวแปรเชิงเลขจะมีใช้กันในสาขาทางด้าน วิทยาศาสตร์ วิศวกรรมศาสตร์ คณิตศาสตร์ การแพทย์ และอื่นๆ ส่วนตัวแปรสัญลักษณ์มีความสำคัญ ในวิทยาการเกี่ยวกับปัญญาประดิษฐ์และการตัดสินใจ การใช้ตัวแปรภาษาเป็นการรวมตัวแปรเชิงเลข กับตัวแปรสัญลักษณ์เข้าด้วยกัน ภาพที่ 2-42 แสดงตัวอย่างเซตตัวแปรภาษาของเซตฟัซซี ได้แก่ Extremely Low, Very Low, Low, Medium, High, Very High และ Extremely High

2.11.5 กฎของฟัซซี

วิทยาการเกี่ยวกับฟัซซีลอจิก มีจำนวนมาก แต่ที่นิยมและการประยุกต์ใช้งานมากที่สุดคือแบบถ้า-แล้ว (Fuzzy If-Then Rule)



ภาพ 2-43 ตัวอย่างปริภูมิรูปแบบการจัดกลุ่มด้วยกฎฟัซซี

จากภาพที่ 2-43 สามารถเขียนเป็นกฎในรูปประโยคภาษาได้ดังนี้

กฎข้อ 1: ถ้า x_1 มีค่า low และ x_2 มีค่า low แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_1

กฎข้อ 2: ถ้า x_1 มีค่า low และ x_2 มีค่า high แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_2

กฎข้อ 3: ถ้า x_1 มีค่า high และ x_2 มีค่า low แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_3

กฎข้อ 4: ถ้า x_1 มีค่า high และ x_2 มีค่า high แล้ว ข้อมูล (x_1, x_2) เป็นกลุ่ม C_4

เมื่อ x_1 เป็นตัวแปรภาษาในมิติที่ 1, x_2 เป็นตัวแปรภาษาในมิติที่ 2, low และ high เป็นพจน์ภาษา, ข้อมูล (x_1, x_2) เป็นคู่ลำดับของวัตถุที่ต้องการจัดกลุ่ม และ C_1, C_2, C_3 และ C_4 เป็นกลุ่มข้อมูล 1, 2, 3 และ 4

สมมติให้กฎข้อ $l, l = 1, 2, \dots, L$ เป็นลำดับของกฎ ให้ข้อมูลเป็น $x = [x_1, x_2, \dots, x_n]$ เมื่อ n เป็น จำนวนมิติของข้อมูล ให้ A_{li} เป็นพจน์ภาษาในกฎข้อที่ l มิติที่ i และให้กลุ่มข้อมูลเป็น $C_k, k = 1, 2, \dots, K$ รูปแบบทั่วไปของกฎฟัซซีสามารถเขียนได้ดังนี้

กฎข้อ 1: ถ้า x_1 มีค่า A_{11} และ x_2 มีค่า A_{12} และ ... และ x_n มีค่า A_{1n} แล้วข้อมูล x เป็นกลุ่ม C_1

กฎข้อ 2: ถ้า x_1 มีค่า A_{21} และ x_2 มีค่า A_{22} และ ... และ x_n มีค่า A_{2n} แล้วข้อมูล x เป็นกลุ่ม C_2

กฎข้อ l : ถ้า x_1 มีค่า A_{l1} และ x_2 มีค่า A_{l2} และ ... และ x_n มีค่า A_{ln} แล้วข้อมูล x เป็นกลุ่ม C_k

2.11.5 ระบบกฎแบบฟัซซี

ในระบบปัญญาประดิษฐ์ (Artificial Intelligence) หรือเครื่องจักรอัจฉริยะ (Machine Intelligence) มีวิธีการหลายวิธีในการที่จะแสดงองค์ความรู้ของมนุษย์ในรูปแบบต่าง ๆ เช่น ตรรกะ (Logic) เฟรม (Frames) โครงข่ายความหมาย (Semantic Nets) ภาววิทยา (Ontology) และกฎ (Rules) ซึ่งแบบหลังสุดเป็นวิธีหนึ่งที่ยอมรับใช้ในระบบฟัซซี

2.11.6 รูปแบบกฎฟัซซี

ในระบบฟัซซีองค์ความรู้สามารถแสดงในรูปประโยค

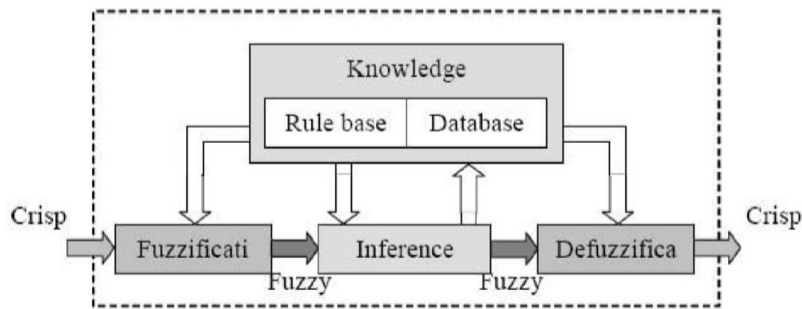
ถ้า ข้อตั้ง (ข้อนำ) ดังนั้น ข้อยุติ (ข้อตาม)

IF premise (antecedent), THEN conclusion (consequent)

ข้อความดังกล่าวรู้จักกันดีในนาม “รูปแบบฐานกฎถ้า-ดังนั้น” (IF-THEN Rule-Based Form) หรือ รูปแบบนิรนัย (Deductive Form) ในรูปแบบการแสดงอนุมาน หากเราทราบความจริง (ข้อ ตั้ง ข้อสมมุติฐาน หรือข้อนำ) แล้วเราสามารถอนุมาน หรือหาข้อสรุปความจริงอีกอย่างหนึ่งที่เรียกว่าข้อ ยุติหรือข้อตาม การแสดงรูปแบบองค์ความรู้นี้ เรียกว่า องค์ความรู้ตื้น (Shallow Knowledge) ซึ่ง ค่อนข้างมีความเหมาะสมในบริบทของภาษา เนื่องจากการแสดงประสิทธิภาพของมนุษย์และองค์ ความรู้เชิงศึกษาสำนึก (Heuristics) ในรูปแบบประโยคภาษามนุษย์ที่ใช้ในการสื่อสารทั่วไป แต่ไม่เป็น รูปแบบองค์ความรู้ที่ลึกล้ำ แบบที่เป็นการรู้เอง เป็นโครงสร้าง เป็นฟังก์ชัน หรือเป็นพฤติกรรมของวัตถุ รอบๆ ตัวเรา อย่างที่เรียกว่า อุปนัย (Inductive) ระบบกฎฟัซซีเป็น สิ่งที่มีประโยชน์ในการจัดรูปแบบของระบบ ที่ซับซ้อนที่สามารถสังเกตได้ โดยมนุษย์ เพราะระบบเหล่านี้สามารถแสดงด้วยตัวแปรภาษาในข้อนำและข้อตามของกฎได้ ตัวแปร ภาษาสามารถนำเสนอแสดงเชิงธรรมชาติด้วยฟัซซีเซตและตัวเชื่อมตรรกะของเซตเหล่านั้น

2.11.7 โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซีลอจิก

โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซีซึ่งประกอบด้วยส่วนที่สำคัญ 4 ส่วน
ดังนี้ ดัง ภาพที่ 2-44



ภาพ 2-44 โครงสร้างพื้นฐานของการประมวลผลแบบฟัซซี

ส่วนที่แปลงการอินพุตทั่วไปเปลี่ยนเป็นการอินพุตแบบตัวแปรฟัซซี (Fuzzification) หรือใน รูปแบบเซตฟัซซีหรือเรียกว่าเป็นตัวแปรภาษา (Linguistic Variable)

ฐานความรู้ (Knowledge base) เป็นส่วนที่จัดเก็บรวบรวมข้อมูลในการควบคุม ประกอบ 2 ส่วนคือ ฐานกฎ (Rule base) และฐานข้อมูล (Database)

ฐานกฎ (Rule base) ส่วนของการกำหนดวิธีการควบคุม ซึ่งได้จากผู้เชี่ยวชาญในรูปแบบของ ชุดข้อมูลแบบกฎของภาษา (Linguistic rule)

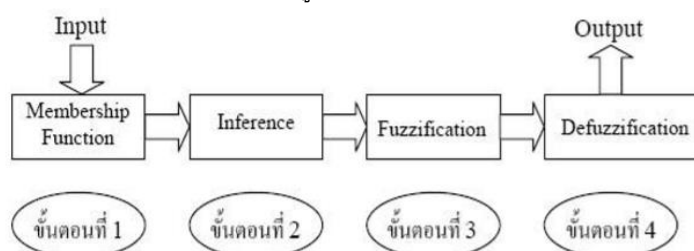
ฐานข้อมูล (Database) เป็นการเตรียมส่วนที่จำเป็นเพื่อที่จะใช้ในการกำหนด กฎการ ควบคุม และการจัดการข้อมูลของตรรกศาสตร์ฟัซซี

เครื่องอนุมานหรือการตีความ (Inference Engine) เป็นส่วนที่ทำหน้าที่ ตรวจสอบข้อเท็จจริง และกฎ เพื่อใช้ในการตีความหาเหตุผล เหมือนกลไกสำหรับควบคุมการใช้ ความรู้ในการแก้ไขปัญหา รวมทั้งการกำหนดวิธีการของการตีความเพื่อหาคำตอบ

ส่วนที่แปลงการเอาต์พุตให้อยู่ในช่วงที่เหมาะสม (Defuzzification) เป็นการทำการแปลง ข้อมูลที่อยู่ในรูปแบบฟัซซีให้เป็นค่าที่สรุปผลหรือค่าการควบคุมระบบ

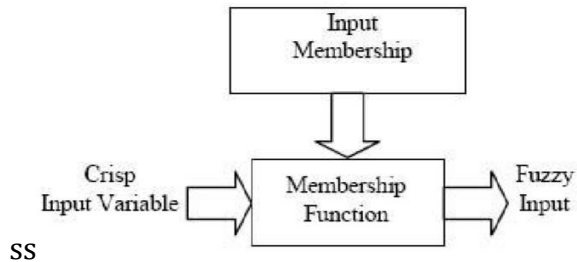
2.11.8 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนการประมวลผลจะมีรูปแบบการทำงานเป็น 4 ส่วนจะแสดงดังภาพที่ 2-45



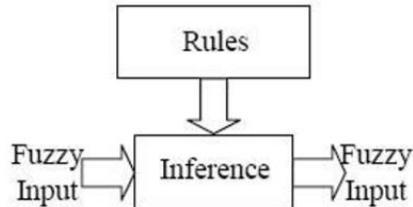
ภาพ 2-45 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก

ขั้นตอนที่ 1 เป็นการแปลงการอินพุตแบบทวินัยเปลี่ยนเป็นการอินพุตแบบตัวแปรฟัซซี โดยจะสร้างฟังก์ชันความเป็นสมาชิก โดยไม่จำเป็นต้องมีลักษณะเดียวกัน ขึ้นกับคุณลักษณะของแต่ละการ อินพุต (Input) และความสำคัญต่อการเอาต์พุต (Output) ที่น่าสนใจโดยฟังก์ชันจะมีลักษณะเป็นการ กำหนดภาษาสามัญ เพื่อให้เป็นฟัซซีการอินพุต ดังภาพที่ 2-46



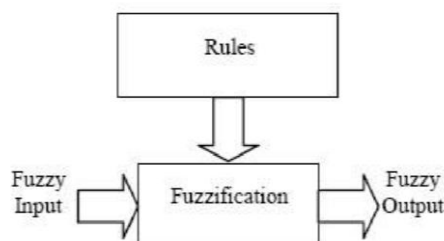
ภาพ 2-46 ขั้นตอนที่ 1 ของการประมวลผลแบบฟัซซีLogic

ขั้นตอนที่ 2 เป็นการสร้างความสัมพันธ์ระหว่างการอินพุตทั้งหมด ที่เกี่ยวข้องกับเอาต์พุตที่ อาศัยหลักการของการหาเหตุและผล อาจจะสร้างการเก็บข้อมูล การคาดการณ์จากการตัดสินใจของ มนุษย์ หรือค่าจากการทดลอง โดยเขียนเป็นกฎการควบคุมระบบ ซึ่งจะมีลักษณะอยู่ในรูปแบบ ถ้า (If) และ (And) หรือ (Or) ซึ่งเป็นภาษาสามัญ นำกฎทั้งหมดมาประมวลผลรวมกัน เพื่อการหาตัดสินใจที่ เหมาะสม ดังภาพที่ 2-47



ภาพ 2-47 ขั้นตอนที่ 2 ของการประมวลผลแบบฟัซซีลอจิก

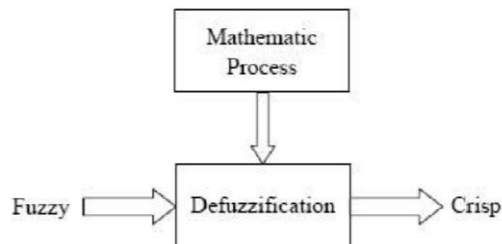
ขั้นตอนที่ 3 เป็นการหาฟัซซีเอาต์พุต โดยการนำกฎการควบคุมที่สร้างขึ้น ในขั้นตอนที่ 2 มา ประมวลผลกับฟัซซีอินพุต โดยใช้วิธีการทางคณิตศาสตร์ เพื่อนำค่าที่ได้ประมวลผล ดังภาพที่ 2-48



ภาพ 2-48 ขั้นตอนที่ 3 ของการประมวลผลแบบฟัซซีลอจิก

วิธีการทำเป็นค่าคลุมเครือ (Fuzzification) วิธีการที่นิยมใช้ในการตีความหาเหตุผลเลือกใช้ Max-Min method และ Max-Dot method

ขั้นตอนที่ 4 เป็นขั้นตอนสุดท้ายหรือขั้นตอนการสรุปเหตุผลฟัซซีโดยจะเปลี่ยนฟัซซีเอาต์พุต ให้เป็นทวินัยเอาต์พุตตามภาพที่ 2-49 และด้วยวิธีทางคณิตศาสตร์ เช่น วิธีการหาจุดศูนย์กลางถ่วง (Central of Gravity) เพื่อนำค่าที่ได้มาใช้ในการตัดสินใจเพื่อควบคุมระบบในสถานการณ์นั้นๆ



ภาพ 2-49 ขั้นตอนที่ 4 ของการประมวลผลแบบฟัซซีลอจิก

วิธีการหาค่าฟัซซีให้เป็นค่าปกติ (Defuzzification) วิธีการที่เป็นเทคนิคการเลือกค่าสูงสุด หรือสรุปหาเหตุผลจากหลาย ๆ เซตมาเพียงค่าเดียว ซึ่งเป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิก จากการกระทำหลายๆ แบบ และเลือกกระทำเพียงรูปแบบเดียว วิธีการหาจุดศูนย์กลางถ่วง (Central of Gravity: COG) เป็นวิธีการเฉลี่ยผลที่ได้จากการตีความหา เหตุที่นิยมใช้ในปัจจุบัน ค่าที่ได้จะคำนวณจุดศูนย์กลางถ่วงโดยรวมจะหาได้จากการประมาณค่าจากสมการ

$$COG = \frac{\sum_{i=1}^N a_i w_i}{\sum_{i=1}^N a_i} \quad (2.18)$$

โดยสมการ ได้กำหนดค่าของสมการดังนี้

COG แทน ค่าของจุดศูนย์กลางถ่วง (Central of Gravity)

N แทน ค่าตั้งแต่ตำแหน่งที่ 1 ถึงตำแหน่งที่ i

a_i แทน ค่าฟัซซีของเอาต์พุตในเซตฟัซซีตำแหน่งที่ i

w_i แทน พื้นที่ใต้โค้งของเซตฟัซซีตำแหน่งที่ i

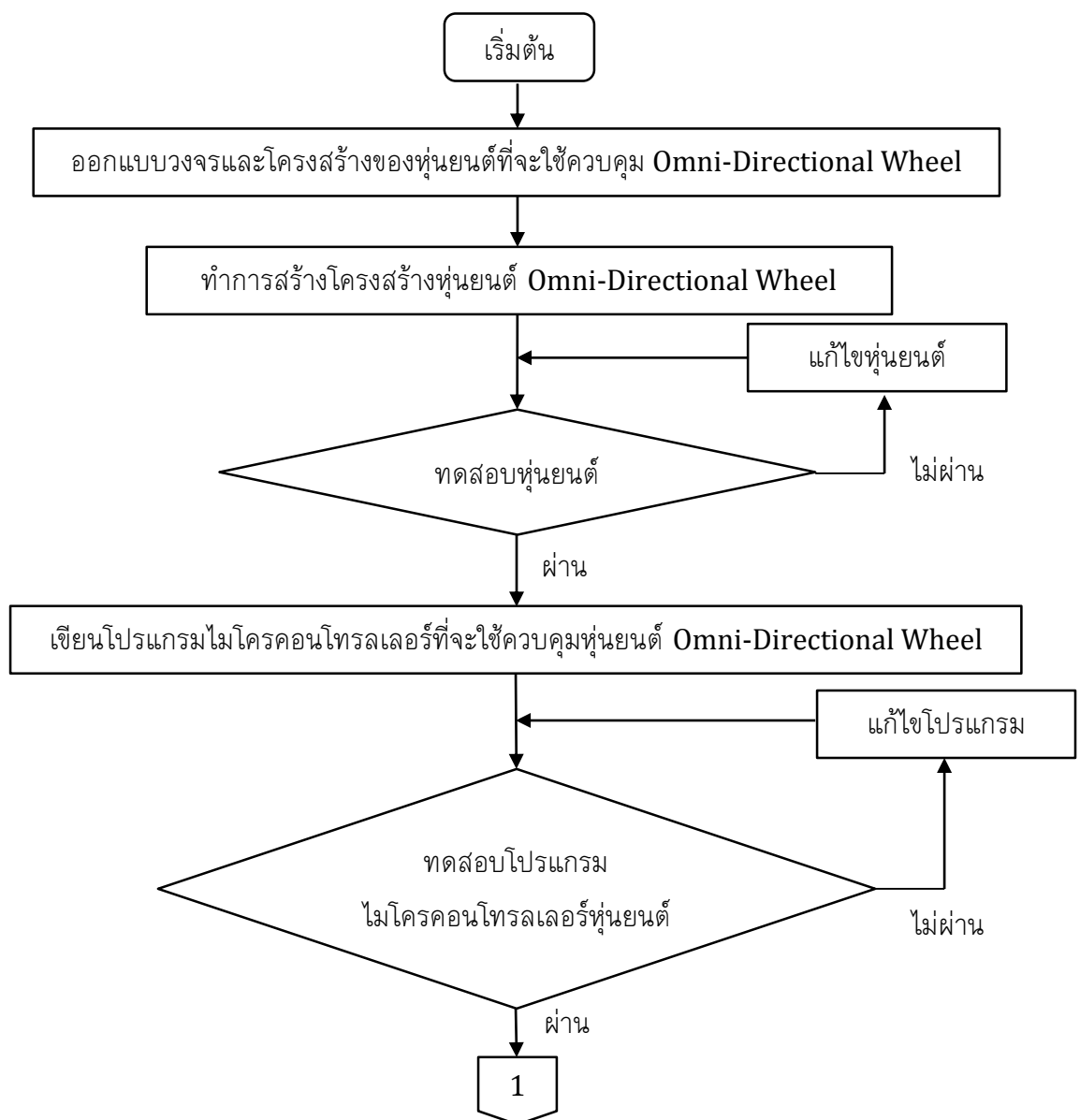
2.12 งานวิจัยที่เกี่ยวข้อง

2.12.1 การควบคุมการเคลื่อนตัวของหุ่นยนต์ Omni-Wheel ซึ่งทำการวิจัยโดย รศ. ดร.ธีระพงษ์ ว่องรัตนะไพศาล ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

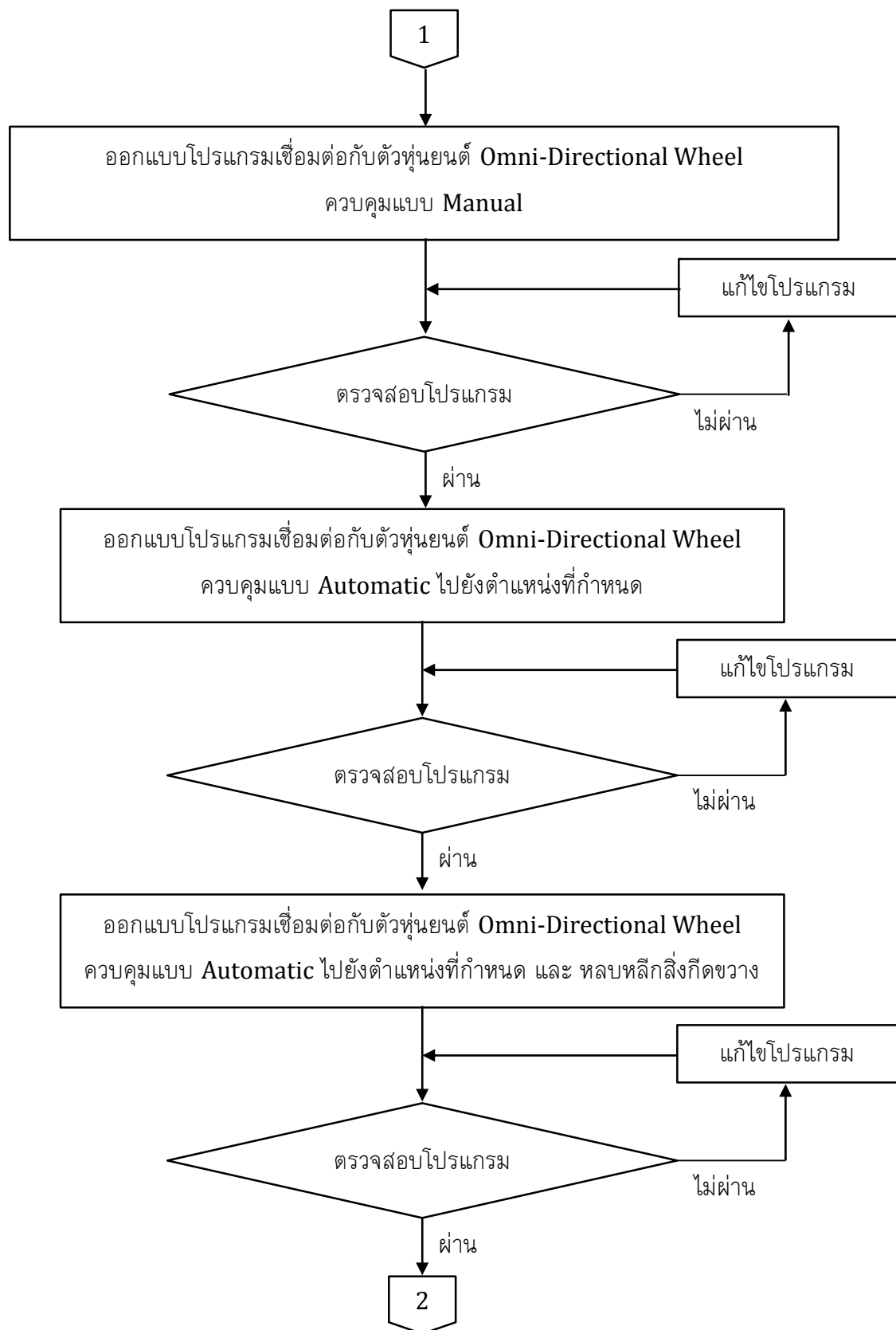
บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

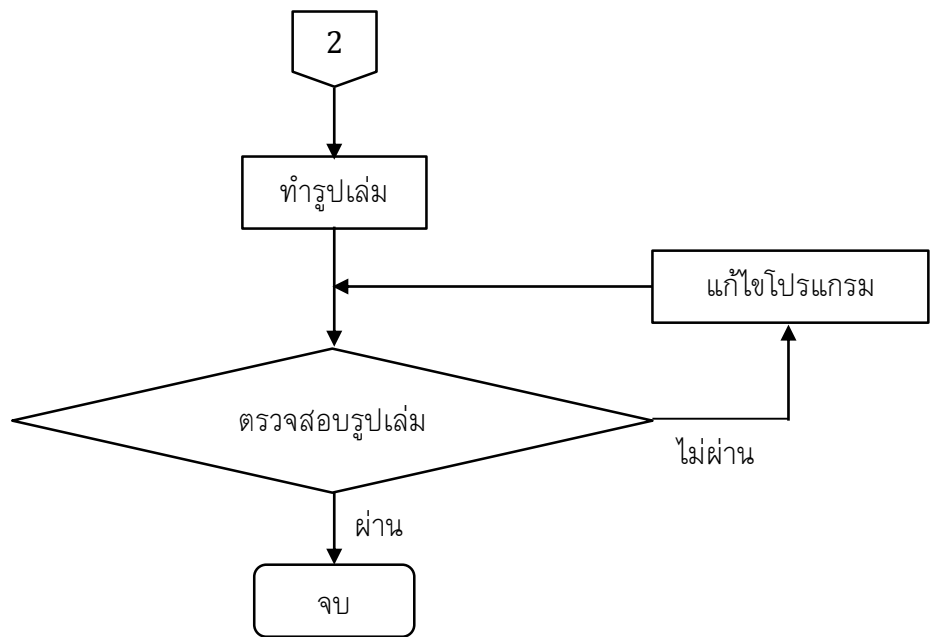
จากทฤษฎีและข้อมูลที่เกี่ยวข้อง ทางผู้จัดทำโครงการการควบคุมหุ่นยนต์เคลื่อนที่ด้วยล้อหลายทิศทาง ได้มีการวางแผนการดำเนินงานเพื่อที่จะเป็นแนวทางในการปฏิบัติงานเพื่อให้โครงการเสร็จตามวัตถุประสงค์ที่วางไว้ ดังแสดงในผังงาน



ภาพที่ 3-1 โฟลว์ชาร์ตการดำเนินงาน (1)



ภาพที่ 3-2 โฟลว์ชาร์ตการดำเนินงาน (2)



ภาพที่ 3-3 โฟลว์ชาร์ตการดำเนินงาน (3)

ขั้นตอนในการดำเนินงาน สามารถแบ่งขั้นตอนต่างๆ ได้ดังนี้

- ศึกษาและรวบรวมทฤษฎีที่เกี่ยวข้องกับปัญญานิพนธ์
- การออกแบบทางกลของหุ่นยนต์
- วงจรของหุ่นยนต์
- การประกอบหุ่นยนต์
- ออกแบบโปรแกรมควบคุมด้วยมือ (Manual)
- ออกแบบโปรแกรมควบคุมแบบอัตโนมัติ (Automatic)
- จัดทำปัญญานิพนธ์

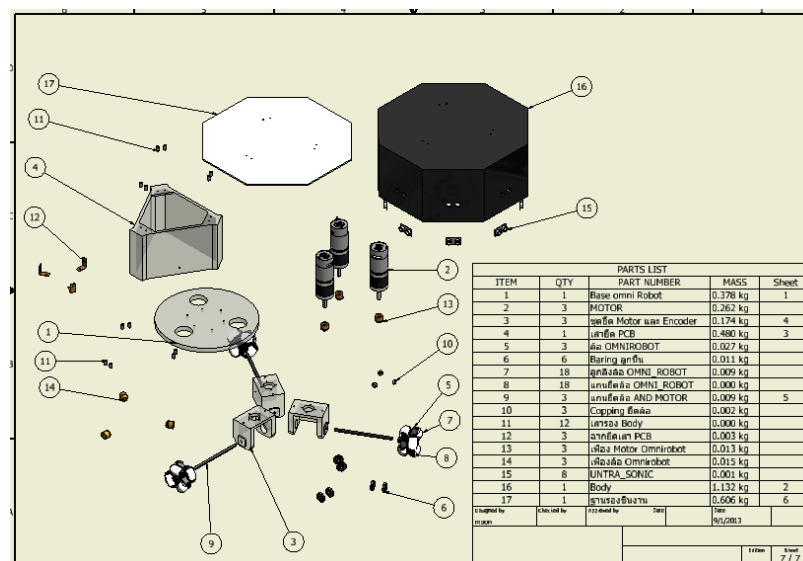
3.1 ศึกษาและรวบรวมทฤษฎีที่เกี่ยวข้องกับปัญญานิพนธ์

ผู้จัดทำโครงงานได้ทำการศึกษาข้อมูลต่าง ๆ เรียงตามลำดับ ดังนี้

- 3.1.1 การออกแบบทางกล
- 3.1.2 การออกแบบทางไฟฟ้า
- 3.1.3 การเคลื่อนที่ของหุ่นยนต์
- 3.1.4 การเขียนโปรแกรมควบคุมด้วยมือ
- 3.1.5 การเขียนโปรแกรมควบคุมแบบอัตโนมัติ

3.2 การออกแบบทางกลของหุ่นยนต์

ตัวหุ่นยนต์ที่ออกแบบนี้ เป็นหุ่นยนต์เคลื่อนที่หลายทิศทาง (Omnidirectional Wheel Robot) โดยที่ตัวหุ่นยนต์ จะประกอบไปด้วย 3 ล้อที่จะใช้ในการขับเคลื่อนตัวหุ่นยนต์ โดยที่ตัวหุ่นยนต์จะสามารถเคลื่อนที่ได้อย่างอิสระ และสามารถเคลื่อนในพื้นที่ที่จำกัดการเคลื่อนที่



ภาพที่ 3-4 ชิ้นส่วนของหุ่นยนต์

ดังภาพที่ภาคผนวก ก-7 มวลของตัวหุ่นยนต์อยู่ที่ 4.5 กิโลกรัมและจากขอบเขตของโครงการได้กำหนดไว้ว่า สามารถรับโหลดได้ 3 กิโลกรัม และมวลของแบตเตอรี่ 1.5 กิโลกรัม

ดังนั้นมวลรวมของหุ่นยนต์คือ

$$m_t = \text{มวลหุ่นยนต์} + \text{มวลโหลด} + \text{มวลแบตเตอรี่} \quad (3.1)$$

$$m_t = 4.5 + 3 + 1.5 \quad (3.2)$$

$$m_t = 9 \text{ kg} \quad (3.3)$$

โดยให้ m_t แทนมวลรวมทั้งหมดของหุ่นยนต์ (กิโลกรัม)

น้ำหนักเฉลี่ยของแต่ละล้อที่ต้องรับ

$$F = \frac{F_T}{3} \quad (3.4)$$

$$F = \frac{9}{3} \quad (3.5)$$

$$F = 3 \text{ kg} \quad (3.6)$$

$$F = 30 \text{ N} \quad (3.7)$$

โดยให้ F หมายถึง น้ำหนักเฉลี่ยลงแต่ละล้อ (นิวตัน)

การหาขนาดมอเตอร์

ถ้าต้องการใช้ความเร็วมอเตอร์ที่ความเร็วรอบ 300 รอบต่อนาที

กำลังของมอเตอร์จะหาได้จากสมการ (3.8)

$$P_{em} = F \times V \quad (3.8)$$

$$P_{em} = m \times g \left(\frac{n \times 2\pi}{60} \times r \right) [\text{kg} \cdot \text{m} / \text{s}^2][\text{rad} / \text{s} \cdot \text{m}] \quad (3.9)$$

$$P_{em} = 3 \times 10 \left(\frac{300 \times 2\pi}{60} \times 0.03 \right) \quad (3.10)$$

$$P_{em} = 28 \text{ W} \quad (3.11)$$

โดยให้ P_{em} หมายถึง กำลังของมอเตอร์ (วัตต์)

F หมายถึง น้ำหนักของวัตถุ (นิวตัน)

V หมายถึง น้ำหนักของหุ่นยนต์ทั้งหมด (กิโลกรัม)

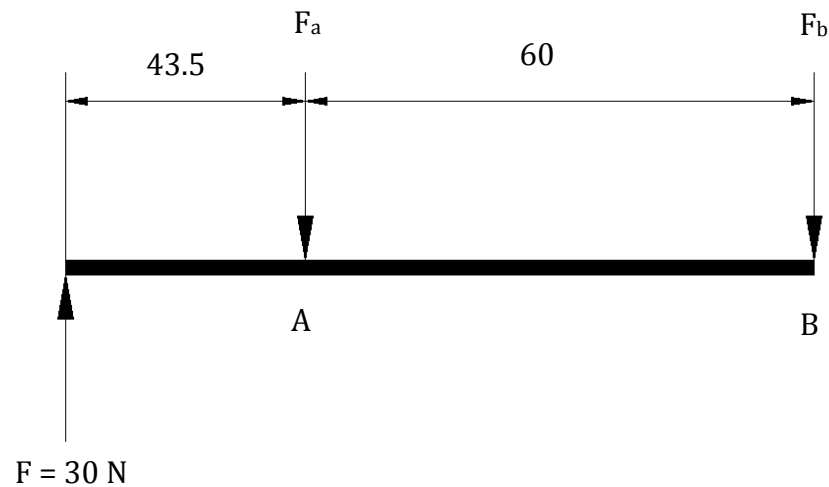
r หมายถึง รัศมีของล้อ (เมตร)

n หมายถึง ความเร็วรอบของมอเตอร์ (รอบ/นาที)

หาแรงบิดที่มอเตอร์กระทำกับเพลาในหน่วยนิวตันเมตร จะหาได้จากสมการ (3.12)

$$P_{em} = \frac{300 \times T}{9.55} \quad (3.12)$$

$$T = 0.89 \text{ นิวตันเมตร (Nm)} \quad (3.13)$$



ภาพที่ 3-5 แรงที่เกิดขึ้นบนเพลา

หาแรงที่เกิดขึ้นที่เพลาในแต่ละจุด

$$+ \circlearrowleft \sum M_A = 0 ;$$

$$0 = F_b \cdot 60 + F \cdot 43.5 \quad (3.14)$$

$$F_b = \frac{F \cdot 43.5}{60} \quad (3.15)$$

$$F_b = -21.75 \text{ นิวตัน (N)} \quad (3.16)$$

$$+ \circlearrowleft \sum M_B = 0 ;$$

$$0 = F_a \cdot 60 - F \cdot 103.5 \quad (3.17)$$

$$F_a = \frac{F \cdot 103.5}{60} \quad (3.18)$$

$$F_a = 51.75 \text{ นิวตัน (N)} \quad (3.19)$$

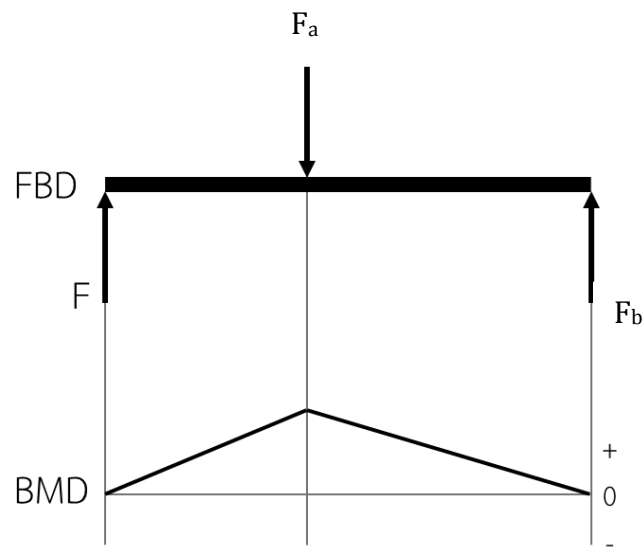
จากที่ได้สมมติแรงที่กระทำกับเพลาเมื่อคำนวณแล้วผลคือ

$$F_a = 51.75 \text{ นิวตัน (N)} \quad (3.20)$$

และ

$$F_b = -21.75 \text{ นิวตัน (N)} \quad (3.21)$$

F_a ได้กำหนดทิศทางของแรงที่กระทำได้ถูกต้อง แต่ F_b มีค่าเป็นลบจึงต้องทำให้เป็นบวกโดยการกลับทิศของแรงที่สมมุติขึ้นแล้วจะได้แรงที่กระทำกับเพลาแสดงดังรูปที่ 3-6



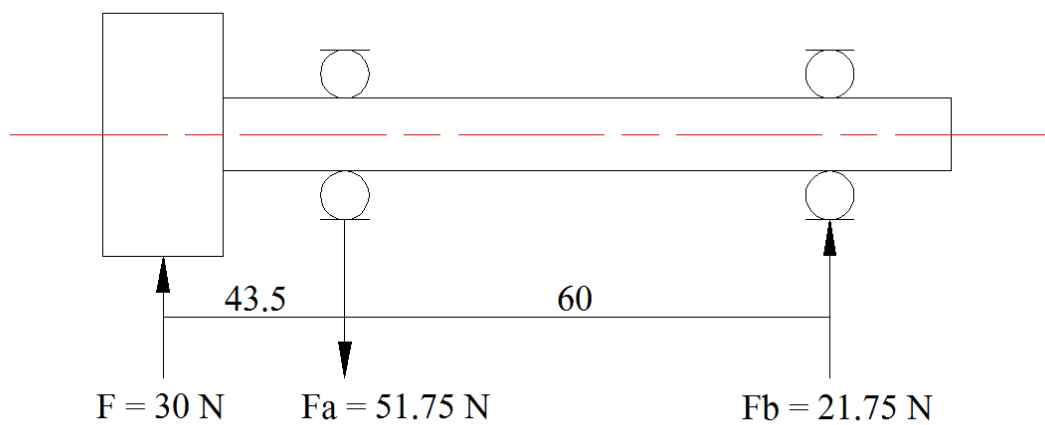
ภาพที่ 3-6 Free-body Diagram และ Bending Moment Diagram

ดังนั้นจึงได้แรงที่กระทำกับเพลลา

$$F_a = 51.75 \text{ นิวตัน (N)} \quad (3.22)$$

และ

$$F_b = 21.75 \text{ นิวตัน (N)} \quad (3.23)$$



ภาพที่ 3-7 แรงกดที่กระทำกับลูกปืน

การออกแบบเพลลา

จากภาพที่ 3-7 เมื่อมากระทำกับล้อทำให้การตัดที่เพลลาถ้าใช้เพลลาเหล็ก St50 และให้ค่า Safety Factor = 5, กำหนดให้ความเค้นเฉือนบิดออกแบบคือ $T_d = 0.18 \times \sigma_u$

ตารางที่ 3-2 คำนวณประกอบความล้า (Fatigue Factor)

ประเภทของภาระ	เนื่องจากการตัด (α_b)	เนื่องจากการบิด (α_t)
เพลลาอยู่นิ่ง แรงสั่นสะเทือนหรือเพิ่มขึ้นช้าๆ แรงกระตุก	1.00 1.50 – 2.00	1.00 1.50 – 2.00
เพลลาหมุน แรงสั่นสะเทือนหรือเพิ่มขึ้นช้าๆ แรงกระตุกเบา แรงกระตุกรุนแรง	1.50 1.50 – 2.00 2.00 – 3.00	1.00 1.00 – 1.50 1.50 – 3.00

เนื่องจากประเภทของภาระ เป็นแบบเพลลาหมุน และเป็นแบบกระตุกเบา จะได้ $\alpha_b = 1.5$ และ $\alpha_t = 1$

จากการเลือกใช้เหล็ก St50 จะได้

$$\sigma_u = 500 \text{ นิวตันต่อตารางเมตร (N/mm}^2\text{)} \quad (3.24)$$

ดังนั้นจึงทำให้

$$\tau_d = 0.18 \times \sigma_u \quad (3.25)$$

$$\tau_d = 90 \text{ นิวตันต่อตารางเมตร (N/mm}^2\text{)} \quad (3.26)$$

หาขนาดของเพลลาที่จุด a โดยใช้สมการดังนี้

$$d = \sqrt[3]{\frac{16}{\pi \times \tau_d} \times \left[M \times \alpha_b^2 + T \times \alpha_t^2 \right]^{\frac{1}{2}}} \quad (3.27)$$

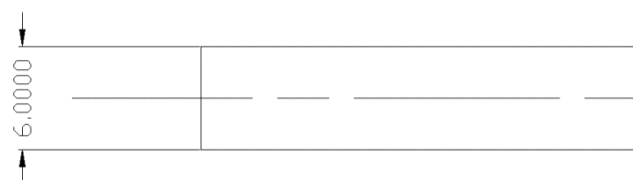
$$d = \sqrt[3]{\frac{16}{\pi \times 90} \times \left[51.75 \times 42.5 \times 1.5^2 + 0.89 \times 1^2 \right]^{\frac{1}{2}}} \quad (3.28)$$

$$d = 5.76 \text{ มิลลิเมตร (mm)} \quad (3.29)$$

ตารางที่ 3-3 ขนาดกำหนดของเพลาดำมาตรฐาน ISO/R 775-1969

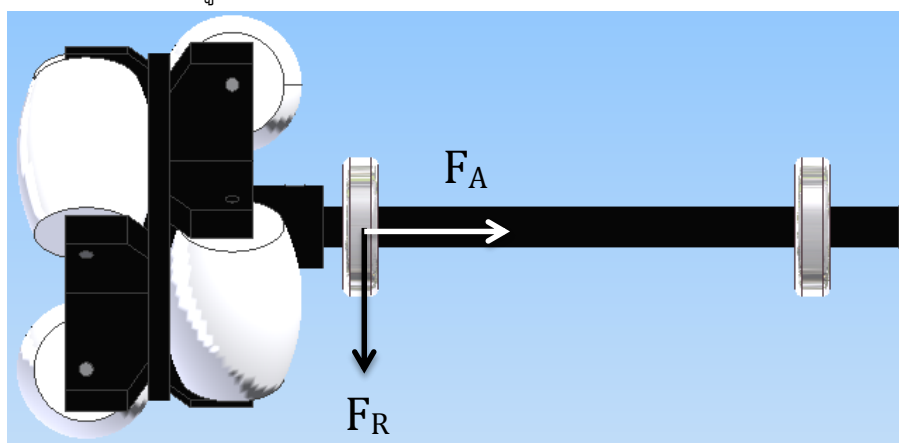
6	25	70	130	240
7	30	75	140	260
8	35	80	150	280
9	40	85	160	300
10	45	90	170	320
12	50	95	180	340
14	55	100	190	360
18	60	110	200	380
20	65	120	220	

จากการคำนวณจะได้ขนาดเพลาคือ 5.76 มิลลิเมตร ดังนั้นจึงเลือกใช้ขนาด 6 มิลลิเมตร เพราะมีขนาดใกล้เคียงกับมาตรฐานที่กำหนดไว้ดังตารางที่ 3-3



ภาพที่ 3-8 ขนาดเพล

การออกแบบแบร์ริงลูกกลิ้ง



ภาพที่ 3-9 แรงกระทำที่เกิดขึ้นในแนวแกนของลูกปืน

เนื่องจากมวลรวมของวัตถุมีน้ำหนักคือ 9 กิโลกรัม ถ้าหุ่นยนต์ สามารถวิ่งด้วยความเร็ว 1/30 เมตรต่อวินาที และถ้าแรงที่กระทำขณะชนเกิดขึ้นเป็นเวลา 0.001 วินาทีแล้วหุ่นยนต์เคลื่อนที่หลายทิศทางหยุด และกำหนดให้ความเร็วคือ 300 รอบต่อวินาที ขนาดเพลาคือ 6 มิลลิเมตร ค่าคงที่ $X = 0.56$ และ $Y = 1.7$ สามารถหาแรงที่กระทำกับลูกปืนโดยใช้สมการดังนี้

จากสมการการดล

$$p = m \times v \text{ [kg][m/ s]} \quad (3.30)$$

$$p = 9 \times \frac{1}{30} \text{ [kg][m/ s]} \quad (3.31)$$

$$p = 0.3 \text{ [kg][m/ s]}$$

ทำการหาแรงที่กระทำในแนวราบโดยใช้สมการดังนี้

$$F \times t = p \quad (3.32)$$

$$F(0.001) = 0.3 \text{ [kg][m/ s]} \quad (3.33)$$

$$F = 30 \text{ นิวตัน (N)} \quad (3.34)$$

ดังนั้น $F_A = F = 30$ นิวตัน

และ $F_R = F_a = 51.75$ นิวตัน

ตารางที่ 3-4 อายุงานลูกกลิ้งแบบรีริง

OPERATING CONDITIONS	BASIC RATING LIFE (Hours)
Infrequent operation	500
Short or intermittent operation. Failure has little effect on function	4000 – 8000
Intermittent operation Failure has significant effect on function	8000 – 12000
8 hours of non-continuous operation	12000 – 20000
8 hours of continuous operation	20000 – 30000
24 hours of continuous operation	40000 – 60000
24 hours of guaranteed trouble free operation	100000 – 200000

กำหนดให้พารามิเตอร์ที่ใช้มีดังนี้

- $L_{10req} = L_{10h} = 12000$ ชั่วโมง (อายุการใช้งาน)
- ความเร็ว 300 รอบต่อนาที
- อุณหภูมิ 25 องศาเซลเซียส
- ขนาดเพลา 6 มิลลิเมตร
- ค่าคงที่ $X = 0.56$ และ $Y = 1.7$

หาแรงในแนวรัศมีและแรงในแนวแกนที่กระทำกับลูกปืนโดยใช้สมการดังนี้

$$F = X \times F_R + Y \times F_A \quad (3.35)$$

$$F = 0.56 \times 51.75 + 1.7 \times 30 \quad (3.36)$$

$$F = 79.98 \text{ นิวตัน (N)} \quad (3.37)$$

ถ้า $F > F_R$; $79.98\text{N} > 51.75\text{N}$

$$F = 0.07998 \text{ กิโลนิวตัน (kN)} \quad (3.38)$$

หาอายุการใช้งานของลูกปืนที่ต้องการให้ทำงานได้โดยใช้สมการดังนี้

$$L_{10r} = \frac{L_{10h} \times n \times 60}{10^6} \quad (3.39)$$

$$L_{10r} = \frac{12000 \times 300 \times 60}{10^6} \quad (3.40)$$

$$L_{10r} = 216 \text{ ล้านรอบ} \quad (3.41)$$

ทำการหาแรงพลวัต

$$C = F L_{10r}^{\frac{1}{x}} \quad (3.42)$$

กำหนดให้ x คือตัวประกอบผันแปรอายุงาน (มีค่า = 3 เมื่อเป็นลูกปืนเม็ดกลม)

$$C = F L_{10r}^{\frac{1}{x}} \quad (3.43)$$

$$C = 0.07998 \cdot 180^{\frac{1}{3}} \quad (3.44)$$

$$C = 0.48 \text{ กิโลนิวตัน (kN)} \quad (3.45)$$

นำค่า Basic load ratings dynamic (C) ไปเทียบในภาพที่ 3-11

Deep groove ball bearings, single row
Product information

Principal dimensions			Basic load ratings		Fatigue load limit	Speed ratings	
d	D	B	dynamic	static	P _u	Reference speed	Limiting speed
mm			kN	C ₀	kN	r/min	
5	11	5	0,637	0,255	0,011	120000	60000
5	13	4	0,884	0,34	0,014	110000	67000
5	13	4	0,884	0,34	0,014	110000	53000
5	16	5	1,14	0,38	0,016	95000	60000
5	16	5	1,14	0,38	0,016	95000	48000
5	16	5	1,14	0,38	0,016	95000	60000
5	19	6	2,34	0,95	0,04	80000	50000
5	19	6	2,34	0,95	0,04	-	24000
5	19	6	2,34	0,95	0,04	80000	40000
5	19	6	2,34	0,95	0,04	80000	40000
5	19	6	2,34	0,95	0,04	-	24000
5	19	6	2,34	0,95	0,04	80000	50000
5	19	6	2,34	0,95	0,04	80000	50000
6	13	3,5	0,884	0,345	0,015	110000	67000
6	13	5	0,884	0,345	0,015	110000	53000
6	15	5	1,24	0,475	0,02	100000	63000
6	15	5	1,24	0,475	0,02	100000	50000
6	19	6	2,34	0,95	0,04	80000	50000
6	19	6	2,34	0,95	0,04	-	24000
6	19	6	2,34	0,95	0,04	80000	40000
6	19	6	2,34	0,95	0,04	80000	40000
6	19	6	2,34	0,95	0,04	-	24000
6	19	6	2,34	0,95	0,04	80000	50000
6	19	6	2,34	0,95	0,04	80000	50000
6,35	19,05	5,556	2,81	1,08	0,045	70000	50000

ภาพที่ 3-10 ตารางมาตรฐานของลูกปืน

เมื่อเทียบ C กับตารางจะได้ว่า เลือกใช้ลูกปืนเม็ดกลมร่องลึก เบอร์ 6186 โดยมีข้อมูล
รายละเอียดของลูกปืนเบอร์ต่อไปนี้
C = 0.884 กิโลนิวตัน, d = 6 มิลลิเมตร, D = 13 มิลลิเมตร, B = 3.5 มิลลิเมตร,
C_o = 0.345 กิโลนิวตัน

ตารางที่ 3-5 ค่าตัวประกอบแรงในแนวรัศมี (x) และในแนวแกน (y)

FA/C _o	0.014	0.028	0.056	0.084	0.110
e	0.19	0.22	0.26	0.28	0.30
เมื่อ F _a /F _r > e ค่า Y =	2.3	1.99	1.71	1.55	1.45

หมายเหตุ หากค่า FA/C_o, e และ Y ไม่ตรงตามตารางที่กำหนดให้ ต้องหาค่าโดยวิธี Interpolate

ทำการตรวจสอบดัลล์ลูกปืนที่เลือกโดยใช้สมการดังนี้

$$\frac{F_A}{C_o} = \frac{0.03}{0.345} = 0.087 \quad (3.46)$$

เมื่อ $\frac{F_A}{C_o}$ ไม่ตรงตามที่ตารางที่ 3-2 กำหนดไว้ ดังนั้นต้องทำการ Interpolate หาค่า e

$$i = \frac{0.11 - 0.084}{0.3 - 0.28} = 1.3 \quad (3.47)$$

$$1.3 = \frac{0.087 - 0.084}{e - 0.28} \quad (3.48)$$

$$e = \frac{0.087 - 0.084}{1.3} + 0.28 \quad (3.49)$$

$$e = 0.2823 \quad (3.50)$$

ถ้า $\frac{F_A}{F_R} > e$ ให้ใช้ค่า X = 0.42; ค่า Y ตามตารางที่ 3-2 แต่ถ้าค่า $\frac{F_A}{F_R} \leq e$ ให้ใช้ค่า X = 1;

Y = 0

$$\frac{F_A}{F_R} = \frac{0.03}{0.05175} = 0.58 \quad (3.51)$$

$$\therefore \frac{F_A}{F_R} > e ; 0.58 > 0.2823 ; X = 0.56 \quad (3.52)$$

หาค่าของ Y โดยใช้การ Interpolate เทียบหาค่า $\frac{F_A}{C_o}$

$$i = \frac{0.11 - 0.084}{1.45 - 1.55} = -0.26 \quad (3.53)$$

$$-0.26 = \frac{0.087 - 0.084}{Y - 1.55} \quad (3.54)$$

$$Y = \frac{0.087 - 0.084}{-0.26} + 1.55 \quad (3.55)$$

$$Y = 1.5385 \quad (3.56)$$

หาแรงที่ตลับลูกปืนสามารถรับได้ หาได้จากสมการดังต่อไปนี้

$$F = X \times F_R + Y \times F_A \quad (3.57)$$

$$F = 0.56 \times 51.75 + 1.5385 \times 30 \quad (3.58)$$

$$F = 75.135 \text{ N} = 0.075135 \text{ กิโลนิวตัน (kN)} \quad (3.59)$$

ตารางที่ 3-6 ตัวประกอบอุณหภูมิ

อุณหภูมิทำงาน	150°C	200°C	250°C	300°C
ตัวประกอบอุณหภูมิ(f_H)	1	0.9	0.75	0.6

จากตารางที่ 3-4 จะได้ค่าตัวประกอบอุณหภูมิ $f_H = 1$ เนื่องจากใช้งานที่อุณหภูมิที่ 25°C กำหนดให้ x คือตัวประกอบผันแปรอายุงาน (มีค่า = 3 เมื่อเป็นลูกปืนเม็ดกลม)

หาค่าอายุงานเป็นจำนวนรอบที่ตลับลูกปืนสามารถทำงานได้ หาได้จากสมการดังต่อไปนี้

$$L_{10r} = \left[f_H \times \frac{C}{F} \right]^x \quad (3.60)$$

$$L_{10r} = \left[1 \times \frac{0.884}{0.075135} \right]^3 \quad (3.61)$$

$$L_{10r} = 1628.658 \text{ ล้านรอบ} \quad (3.62)$$

หาค่าอายุงานเป็นชั่วโมงที่ตลับลูกปืนสามารถทำงานได้ หาได้จากสมการดังต่อไปนี้

$$L_{10h} = 10^6 \times \frac{L_{10r}}{n60} \quad (3.63)$$

$$L_{10h} = 10^6 \times \frac{1628.658}{300 \times 60} \quad (3.64)$$

$$L_{10h} = 90,481 \text{ ชั่วโมง} \quad (3.65)$$

สรุป ตลับลูกปืนที่เลือกมานี้ใช้งานได้เพราะลูกปืนที่ใช้มีอายุงานอยู่ที่ 90,481 ชั่วโมงซึ่งมากกว่าที่กำหนดไว้อยู่ที่ 12,000 ชั่วโมง

กำหนดพิภักตความเฝือของเพลลและเลือ

ตารางที่ 3-7 พิกัดความเฝือของเพลลตันทรงกระบอก

เงื่อนไข	ตัวอย่าง	ขนาดเส้นผ่าศูนย์กลางเพลลา (มิลลิเมตร)		พิกัด ความ เผื่อ
		Ball	Roller	
แหวนในรับภาระหมุนหรือภาระไม่คงที่				
ภาระเบาหรือภาระ เปลี่ยนแปลงเบาๆ (F<=0.06C)	ลูกปืนล้อยานสายพานลำเลียง ตั้งแรงตึงโซ่ สายพาน ห้อง เกียร์	18 ถึง 100 100 ถึง 140	<= 40 40 ถึง 100	j6 k6
ภาระปานกลาง (0.06C<F<=0.12 C)	ลูกปืนล้อยานของเครื่องจักรกล ส่งกำลัง ล้อรถยนต์ เครน	<=18 18 ถึง 100 100 ถึง 140 140 ถึง 200 200 ถึง 280	<=40 40 ถึง 100 100 ถึง 140 140 ถึง 200 200 ถึง 400	j5 k5 m5 m6 n6 p6
รับภาระหนักหรือ ภาระกระตุก (F>0.12C)	ลูกปืนล้อยานของเครื่องจักรกล หนัก เครื่องจักรกลเหมือง	<1 ถึง 140 140 ถึง 200 >280	50 ถึง 140 140 ถึง 200 >200	n6 p6 r6
ภาระเบาต้องการ ความเที่ยงตรง (F>0.6C)	มอเตอร์พัดลม แอร์ เครื่องใช้ ในบ้านหรือห้องทำงาน	<=18 18 ถึง 100 100 ถึง 200	<=40 40 ถึง 140 140 ถึง 200	h5 j5 k5 m5

ตารางที่ 3-8 พิกัดความเผื่อของเหล็กลดลูปป็น

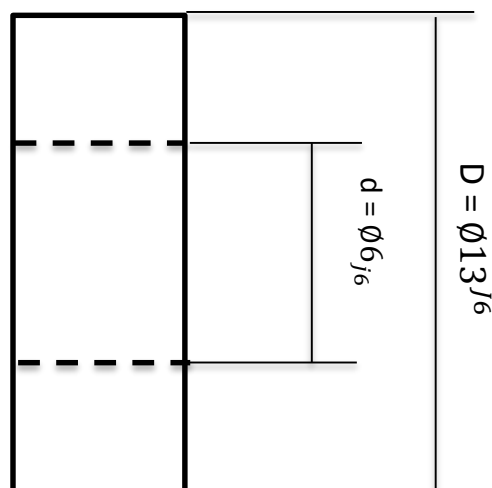
เงื่อนไข	ตัวอย่าง	พิกัดความเผื่อ	ลักษณะงานร่วม
แหวนนอกรับภาระไม่แน่นอน			
ภาระเบาดังต้องการความเที่ยงตรงและเสียงเงียบ ($F \leq 0.06C$)	มอเตอร์พัดลม แอร์ เครื่องใช้ในบ้านหรือห้องทำงาน	J6	สวมพอดี
ภาระปานกลาง ($0.06C < F \leq 0.12C$)	ลูกปืนมอเตอร์ไฟฟ้าทั่วไป มอเตอร์ปั้มน้ำ	K7	สวมอัด
รับภาระหนักหรือภาระกระตุก ($F > 0.12C$)	ลูกปืนมอเตอร์เครื่องกล เหมืองเจาะชุดหิน	M7	สวมอัด

รับภาระเบาหรือภาระเปลี่ยนแปลงเบาๆ ($F < 0.06C$) = ($0.075135 \text{ kN} < (0.06 \times 0.884) \text{ kN}$)

ดังนั้นจะได้

พิกัดความเผื่อของเพลาคือ $\emptyset 6_{j6}$ มิลลิเมตร

พิกัดความเผื่อของเหล็คือ $\emptyset 13_{l6}$ มิลลิเมตร



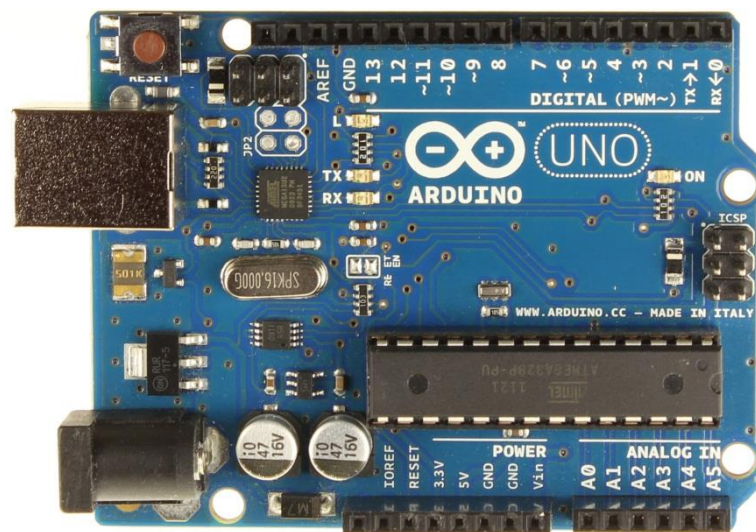
ภาพที่ 3-11 ขนาดของเพลาลและเหล็

3.3 วงจรของหุ่นยนต์

3.3.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

ใช้บอร์ด ATMEGA328 เป็นตัวควบคุมการทำงานโดยใช้ภาษา C และเป็นตัวรับสัญญาณจากเอ็นโค้ดเดอร์, อัลตราโซนิก และข้อมูลที่ส่งผ่านสัญญาณเครือข่ายไร้สาย เพื่อควบคุมการทำงานของหุ่นยนต์โดยที่ ไมโครคอนโทรลเลอร์จะมี PIN I/O ดังต่อไปนี้

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



ภาพที่ 3-12 ไมโครคอนโทรลเลอร์ Arduino

3.3.2 เซ็นเซอร์อัลตราโซนิก

เป็นเซ็นเซอร์ที่ใช้ตรวจสอบสิ่งกีดขวางที่เกิดขึ้นขณะที่หุ่นยนต์กำลังทำงาน โดยที่เซ็นเซอร์ตัวนี้สามารถจับสัญญาณที่สามารถใช้งานได้ตั้งแต่ 2 – 400 เซนติเมตรและใช้ไฟเลี้ยงที่ 5 โวลต์ ใช้ความถี่ (Frequency) ที่ 40 เฮิรตซ์



ภาพที่ 3-13 เซ็นเซอร์อัลตราโซนิก

3.3.3 เอ็นโค้ดเดอร์

เป็นเซ็นเซอร์ที่ใช้ตรวจสอบระยะที่หุ่นยนต์เคลื่อนที่ และมีหน้าทำการตรวจสอบความเร็วในแต่ละล้อ เพื่อควบคุมมุมที่จะกำหนดการเคลื่อนที่ของหุ่นยนต์โดยใช้ไฟเลี้ยงที่ 5 โวลต์ และการทำงานของเอ็นโค้ดเดอร์ทำงานแบบอินคริเมนทอล



ภาพที่ 3-14 เอ็นโค้ดเดอร์

3.3.4 ชุดแปลงไฟเลี้ยง (Regulator)

แปลงจากกระแสตรง ไป กระแสตรง มีแรงดันอินพุต 3.2 – 4.0 โวลต์และแรงดันเอาท์พุท 1.5 – 35 โวลต์ (ปรับค่าได้) และสามารถจ่ายกระแสสูงได้ 3 แอมป์ เพื่อนำไปเลี้ยงให้กับไมโครคอนโทรลเลอร์, เอ็นโค้ดเดอร์ และ เซ็นเซอร์อัลตราโซนิก



ภาพที่ 3-15 ชุดแปลงไฟเลี้ยง

3.3.5 แบตเตอรี่ (Battery)

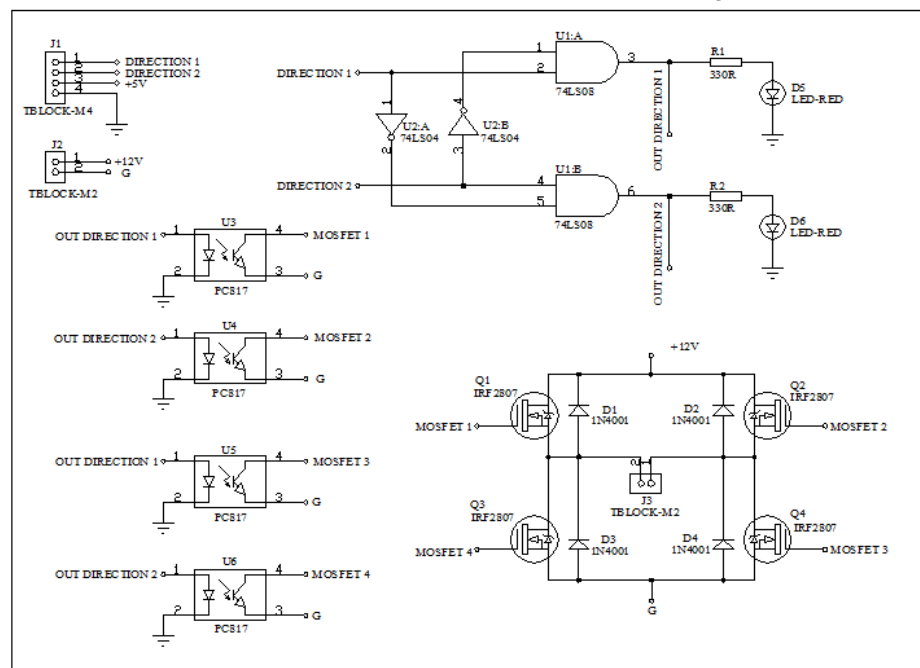
เป็นตัวจ่ายไฟให้กับ มอเตอร์ วงจรขับเคลื่อนมอเตอร์ ไมโครคอนโทรลเลอร์ เซ็นเซอร์ และ เราดเตอร์



ภาพที่ 3-16 แบตเตอรี่

3.3.6 วงจรขับเคลื่อนมอเตอร์

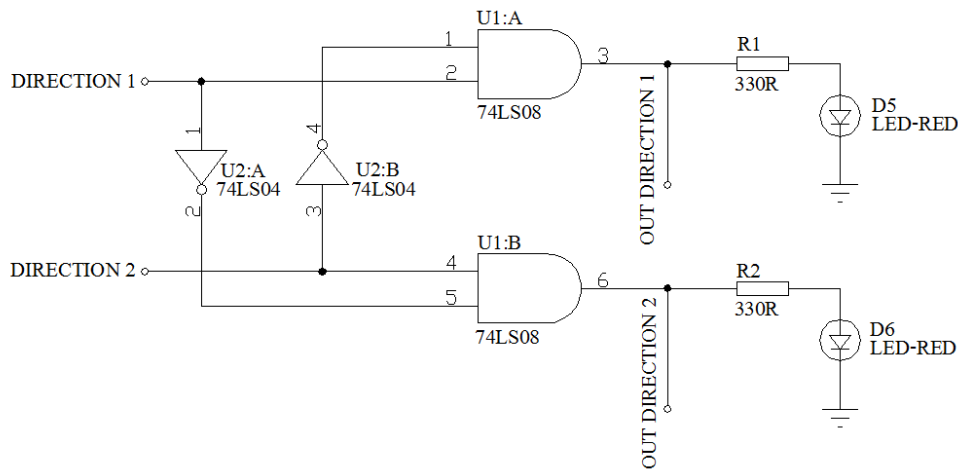
ในส่วนของวงจรขับเคลื่อนมอเตอร์ จะเป็นส่วนที่แปลงสัญญาณจากไมโครคอนโทรลเลอร์ เข้าไปควบคุมมอเตอร์ โดยที่ภายในระบบจะประกอบทั้งหมด 4 ส่วนด้วยกัน ตามรูปที่ 3-17



ภาพที่ 3-17 วงจรไฟฟ้าที่ใช้ในหุ่นยนต์

3.3.6.1 วงจร Interlock

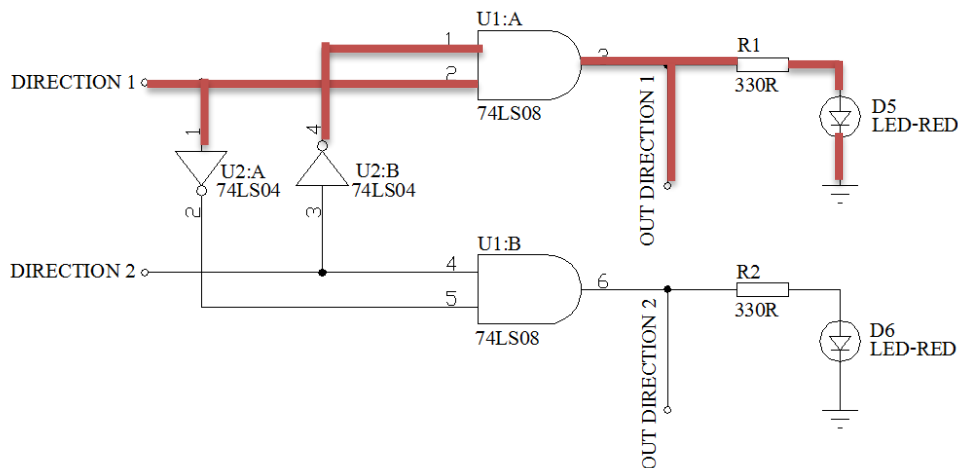
วงจร Interlock มีหน้าที่ป้องกันสัญญาณที่จะไปสั่งชุดขับเคลื่อนมอเตอร์ ไม่ให้เกิดการเสียหาย ตามรูปที่ 3-19



ภาพที่ 3-18 วงจร Interlock

การทำงานของวงจร Interlock

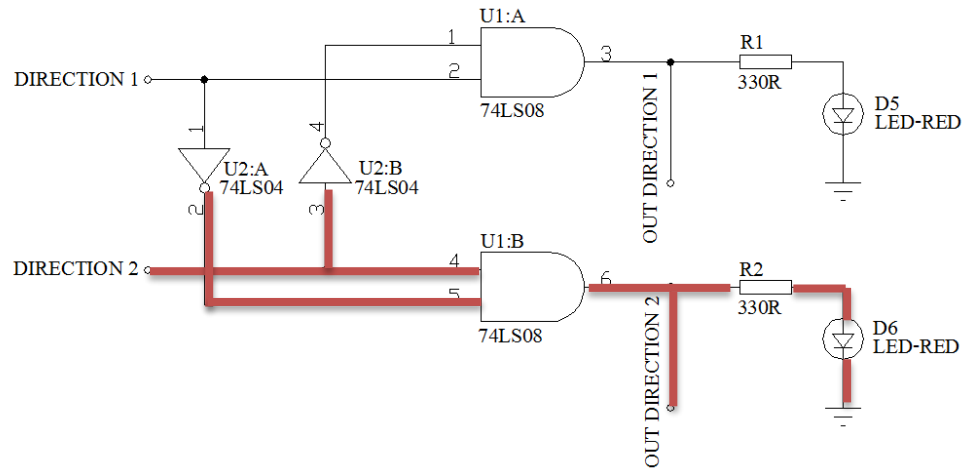
เมื่อส่งสัญญาณ PWM เข้ามายัง DIRECTION 1 สัญญาณจะส่งเข้าไปยัง IC 74LS08 (U1:A) ในกรณีที่ DIRECTION 2 ไม่ได้ส่งสัญญาณ PWM (OFF) เข้ามา ตัว IC 74LS04 (U2:B) จะทำการกลับสถานะให้ทำงาน (ON) เมื่อ 74LS08 มีสัญญาณ มากระทำที่ขา 1 และ ขา 2 จะทำให้ IC 74LS08 (U1:A) ส่งสัญญาณไปขับที่ OUT DIRECTION 1 และทำให้ LED (D5) เกิดการทำงาน แต่ OUT DIRECTION 2 จะไม่ทำงานเนื่องจาก IC 74LS08 (U1:B) ไม่ได้มีสัญญาณมาสั่งให้ทำงาน ตามรูปที่ 3-20



ภาพที่ 3-19 การทำงานของวงจร Interlock (1)

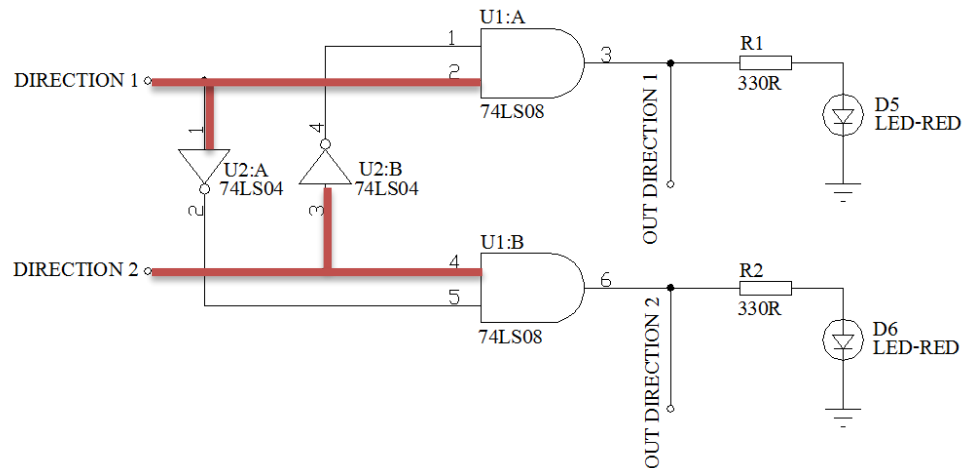
เมื่อส่งสัญญาณ PWM เข้ามายัง DIRECTION 2 สัญญาณจะส่งเข้าไปยัง IC 74LS08 (U1:B) ในกรณีที่ DIRECTION 1 ไม่ได้ส่งสัญญาณ PWM (OFF) เข้ามา ตัว IC 74LS04 (U2:A) จะทำการกลับสถานะให้ทำงาน (ON) เมื่อ 74LS08 มีสัญญาณ มากระทำที่ขา 4 และ ขา 5 จะทำให้ IC 74LS08 (U1:B) ส่งสัญญาณไปขับที่ OUT DIRECTION 2 และทำให้ LED (D6) เกิดการทำงาน

ทำงาน แต่ OUT DIRECTION 1 จะไม่ทำงานเนื่องจาก IC 74LS08 (U1:A) ไม่ได้มีสัญญาณมา
สั่งให้ทำงาน ตามรูปที่ 3-21



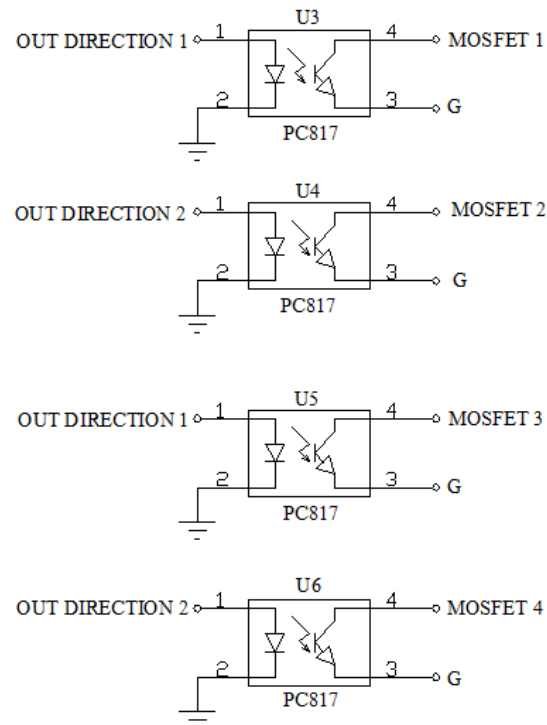
ภาพที่ 3-20 การทำงานของวงจร Interlock (2)

ในกรณีที่ DIRECTION 1 และ DIRECTION 2 ได้ส่งสัญญาณ PWM ทั้ง 2 ขา วงจรจะหยุด
การทำงานโดย IC 74LS04 (U2: A และ U2:B) เนื่องจาก IC 74LS04 (U2:A และ U2:B) จะ
กลับสถานะการทำงานจาก ON เป็น OFF ทำให้ IC 74LS08 (U1:A และ U1:B) ไม่ทำงานเพราะ
จะต้องมีสัญญาณส่งมาทั้ง 2 ขาที่ IC 74LS08 (U1:A และ U1:B) จึงจะทำงานตามรูปที่ 3-22



ภาพที่ 3-21 การทำงานของวงจร Interlock (3)

3.3.6.2 วงจร Optocoupler

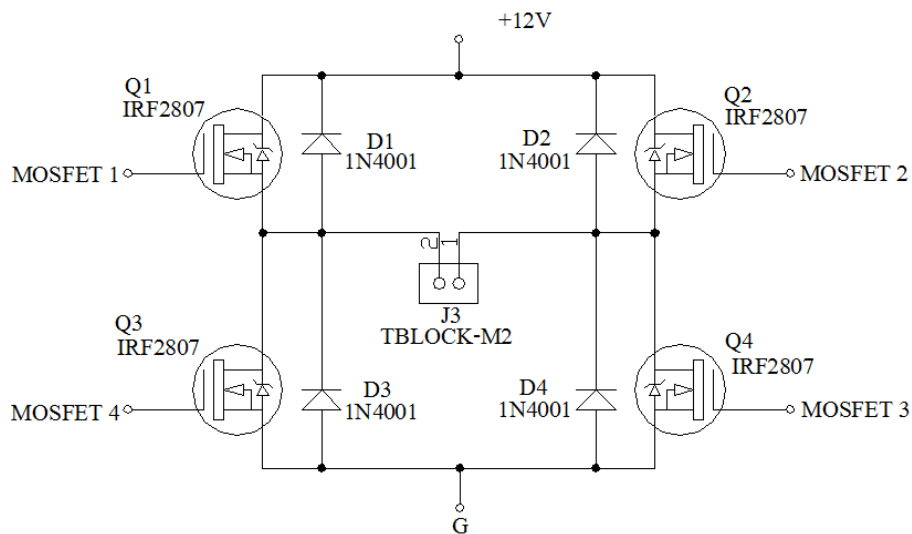


ภาพที่ 3-22 วงจร Optocoupler

การทำงานของวงจร Optocoupler

เมื่อมีสัญญาณเข้ามายัง OUT DIRECTION จะทำให้ MOSFET ทำงานและป้องกันไม่ให้เกิด วงจร Interlock และไม่ใคร่คอนโทรลเลอร์เสียหายเนื่องจาก วงจร H-Bridge เกิดปัญหา

3.3.6.3 วงจร H-Bridge

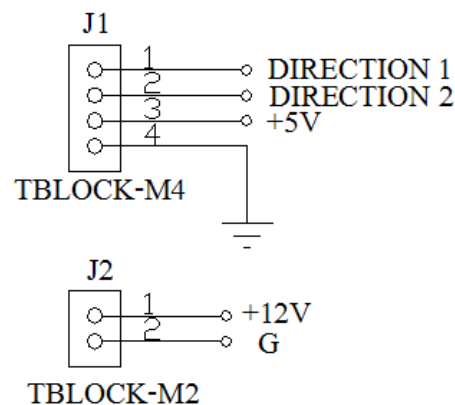


ภาพที่ 3-23 วงจร H-Bridge

การทำงานของวงจร H-Bridge

เมื่อ MOSFET Q1 และ MOSFET Q4 ทำงานจะทำให้มอเตอร์หมุนตามเข็มนาฬิกา ในทางตรงกันข้าม เมื่อ MOSFET Q2 และ MOSFET Q3 ทำงานจะทำให้มอเตอร์หมุนทวนเข็มนาฬิกา ในส่วนของ Diode มีหน้าที่ป้องกันการเกิด Back EMF (Electromotive Force) แต่ทิศทางการหมุนของมอเตอร์กับแรงดันที่ใช้ จะขึ้นอยู่กับการต่อและการใช้แรงดันของมอเตอร์ตัวนั้น

3.3.6.4 แหล่งจ่ายไฟฟ้า



ภาพที่ 3-24 แหล่งจ่ายไฟฟ้า

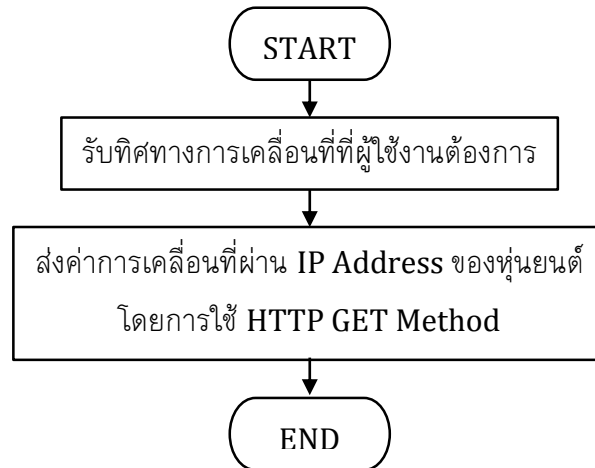
การทำงานของวงจร

จากภาพ J1 จะเป็นตัวส่งแหล่งจ่ายไฟฟ้าให้กับวงจร Interlock และเป็นส่วนที่รับการควบคุมจาก Microcontroller และ J2 เป็นส่วนที่ส่งแหล่งจ่ายไฟฟ้าให้กับ วงจร H-Bridge กับมอเตอร์

3.4 ออกแบบโปรแกรมควบคุมด้วยมือ

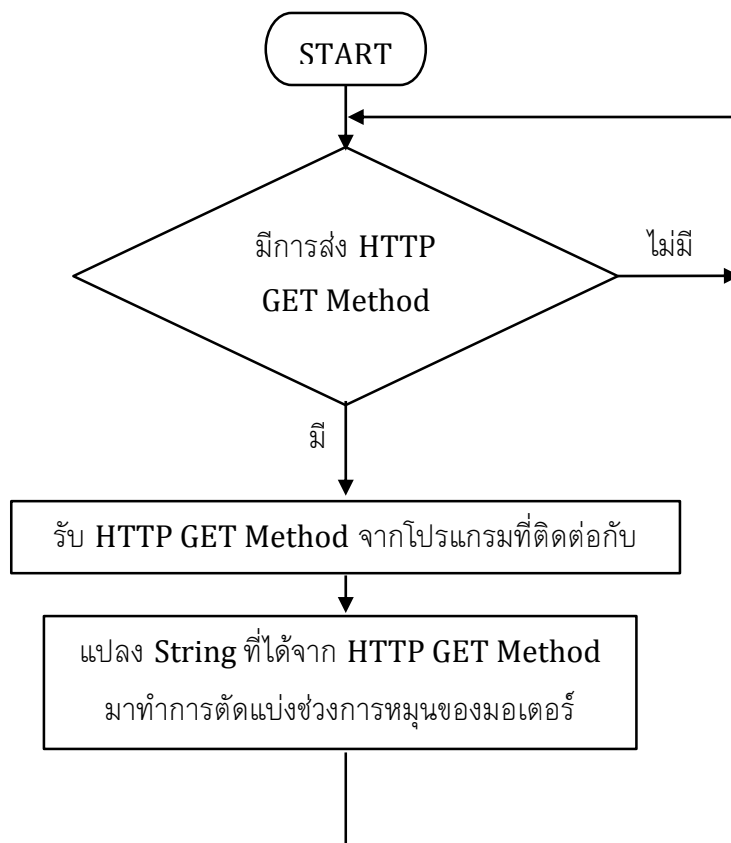
โปรแกรมจะแบ่งการทำงานเป็น 2 ส่วนคือ การทำงานบน C#/Android กับการทำงานบน ไมโครคอนโทรลเลอร์

3.4.1 โฟลว์ชาร์ตการทำงานบน C#/Android



ภาพที่ 3-25 โฟลว์ชาร์ตการทำงานบน C#/Android

3.4.2 โฟลว์ชาร์ตการทำงานบนไมโครคอนโทรลเลอร์



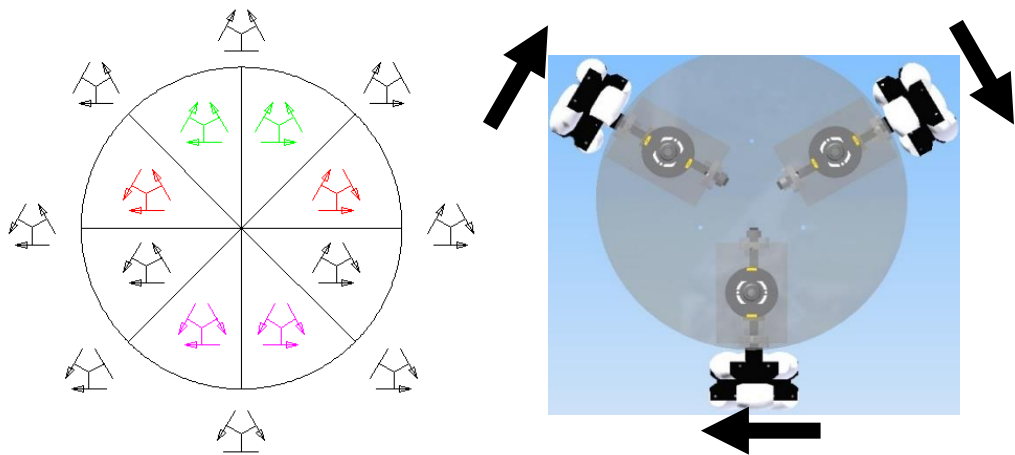
ภาพที่ 3-26 โฟลว์ชาร์ตการทำงานบนไมโครคอนโทรลเลอร์

3.5 ออกแบบโปรแกรมควบคุมแบบอัตโนมัติ

โปรแกรมจะแบ่งการทำงานเป็น 4 ส่วนคือ การหาทิศทางในการควบคุมหุ่นยนต์, การควบคุมการเคลื่อนที่ของหุ่นยนต์, การทำงานบน C#/Android และการทำงานบนไมโครคอนโทรลเลอร์

3.5.1 การหาทิศทางในการควบคุมหุ่นยนต์

รูปแบบการเคลื่อนที่ของหุ่นยนต์จะแบ่งการควบคุมการทำงานได้ลักษณะใหญ่ ได้ทั้งหมด 12 รูปแบบหลัก ตามภาพที่ 3-27 แต่ในการกำหนดรายละเอียดเพื่อให้หุ่นยนต์สามารถเคลื่อนที่ไปตามองศาจำเป็นจะต้องคำนวณหาความสัมพันธ์ในการกำหนดความเร็วของแต่ละล้อ โดยจะกำหนดให้ทิศทางของล้อที่ทำให้หุ่นยนต์หมุนตามเข็มนาฬิกาเป็นบวกดังภาพที่ 3-28



ภาพที่ 3-27 ลักษณะการเคลื่อนที่

มอเตอร์หมุนด้วยความเร็วกำหนดให้เป็น ω มีหน่วยเป็น รอบต่อวินาที (rpm)

เส้นผ่าศูนย์กลางของล้อกำหนดให้เป็น d มีหน่วยเป็น มิลลิเมตร

หาความเร็วของมอเตอร์

$$\omega_n = \frac{\omega \times 2\pi}{60} \text{ เรเดียนต่อวินาที (rad / s)} \quad (3.66)$$

$$\omega_n = \frac{\omega \pi}{30} \text{ เรเดียนต่อวินาที} \quad (3.67)$$

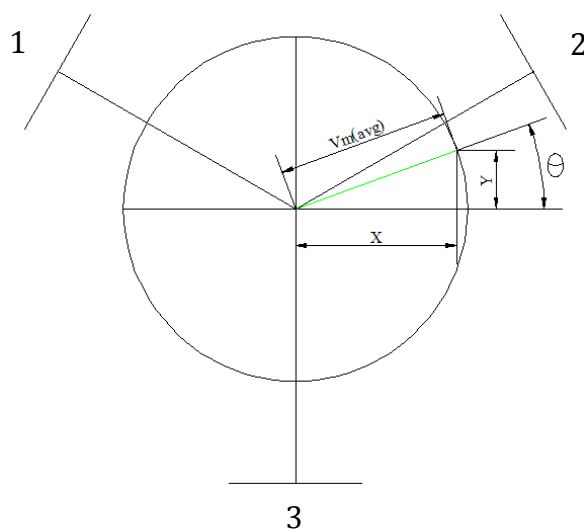
หาความเร็วของล้อหุ่นยนต์

$$V_m = \omega_n r \text{ เมตรต่อวินาที (m / s)} \quad (3.68)$$

ดังนั้นจะได้ความเร็วของตัวหุ่นยนต์

$$V_{avg} = V_m \cos 60 \quad (3.69)$$

การควบคุมความเร็วและตำแหน่งของหุ่นยนต์ใน Quadrant ที่ 1 คือ การทำงานของหุ่นยนต์จะอยู่ในช่วง $0 \leq \theta \leq 90$



ภาพที่ 3-28 การกำหนดขนาดและทิศทางใน Quadrant ที่ 1

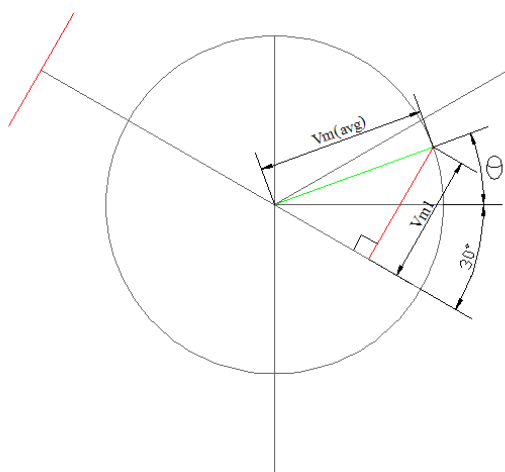
เมื่อต้องการควบคุมการทำงานในช่วง $0 - 90$ องศาและรู้ค่าของระยะของ X และ Y สามารถหามุม θ ได้โดยใช้สมการดังนี้

$$\theta = \tan^{-1} \frac{Y}{X} \quad (3.70)$$

หาระยะทางที่หุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งที่กำหนดโดยใช้สมการดังนี้

$$R = \sqrt{X^2 + Y^2} \quad (3.71)$$

ความเร็วของมอเตอร์ตัวที่ 1



ภาพที่ 3-29 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 1

จากภาพที่ 3-29 $V_{m(avg)}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่
จึงได้สมการการเคลื่อนที่

$$\frac{V_{m1}}{V_{m\ avg}} = \sin (30 + \theta) \quad (3.72)$$

เมื่อแก้สมการออกมาจะได้

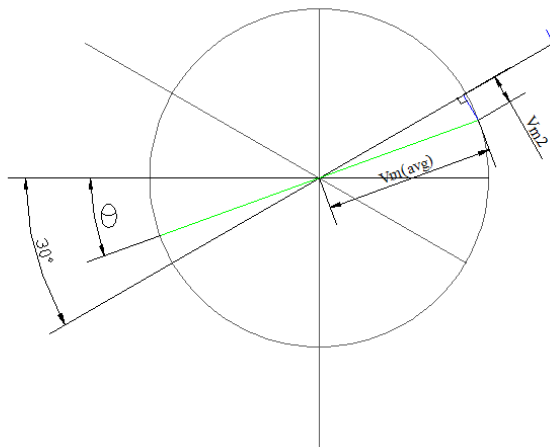
$$V_{m1} = V_{m\ avg} \sin(30 + \theta) \text{ เมตรต่อวินาที (m / s)} \quad (3.73)$$

จำนวนรอบของมอเตอร์ตัวที่ 1 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m1}}{R} = \sin (30 + \theta) \quad (3.74)$$

$$S_{m1} = R \sin (30 + \theta) \text{ มิลลิเมตร (mm)} \quad (3.75)$$

ความเร็วของมอเตอร์ตัวที่ 2



ภาพที่ 3-30 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 1

จากภาพที่ 3-31 $V_{m\ avg}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m2}}{V_{m\ avg}} = \sin 30 - \theta \quad (3.76)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

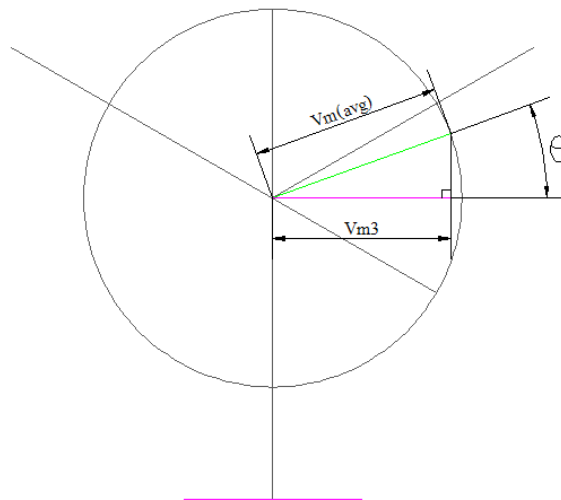
$$V_{m2} = V_{m\ avg} \sin 30 - \theta \text{ เมตรต่อวินาที (m / s)} \quad (3.77)$$

จำนวนรอบของมอเตอร์ตัวที่ 2 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m2}}{R} = \sin 30 - \theta \quad (3.78)$$

$$S_{m2} = R \sin 30 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.79)$$

ความเร็วของมอเตอร์ตัวที่ 3



ภาพที่ 3-31 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 1

จากภาพที่ 3-32 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m3}}{V_{m \text{ avg}}} = \cos \theta \quad (3.80)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

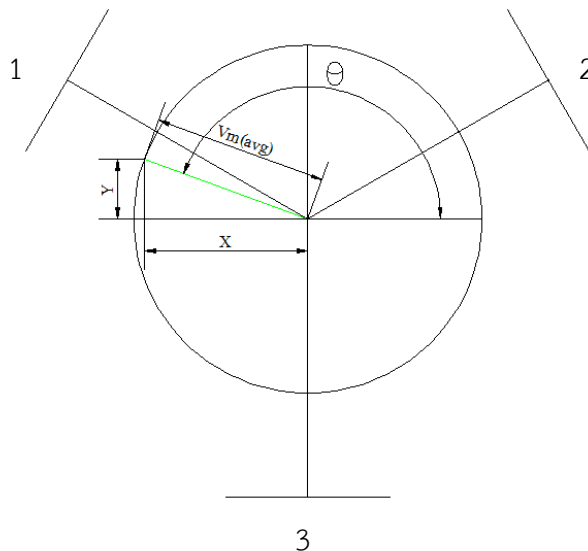
$$V_{m3} = V_{m \text{ avg}} \cos \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.81)$$

จำนวนรอบของมอเตอร์ตัวที่ 3 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m3}}{R} = \cos \theta \quad (3.82)$$

$$S_{m3} = R \cos \theta \text{ มิลลิเมตร (mm)} \quad (3.83)$$

การควบคุมความเร็วและตำแหน่งของหุ่นยนต์ใน Quadrant ที่ 2 คือ การทำงานของหุ่นยนต์จะอยู่ในช่วง $90 \leq \theta \leq 180$



ภาพที่ 3-32 การกำหนดขนาดและทิศทางใน Quadrant ที่ 2

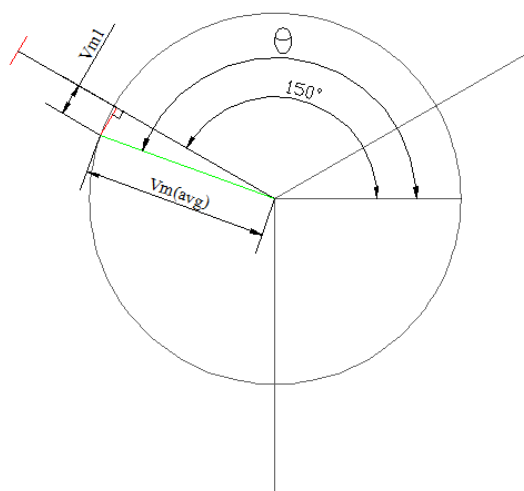
เมื่อต้องการควบคุมการทำงานในช่วง $90 - 180$ องศาและรู้ค่าของระยะของ X และ Y สามารถหามุม θ ได้โดยใช้สมการดังนี้

$$\theta = \tan^{-1} \frac{-x}{y} + 90 \quad (3.84)$$

หาระยะทางที่หุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งที่กำหนดโดยใช้สมการดังนี้

$$R = \sqrt{X^2 + Y^2} \quad (3.85)$$

ความเร็วของมอเตอร์ตัวที่ 1



ภาพที่ 3-33 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 2

จากภาพที่ 3-33 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m1}}{V_{m \text{ avg}}} = \sin 150 - \theta \quad (3.86)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

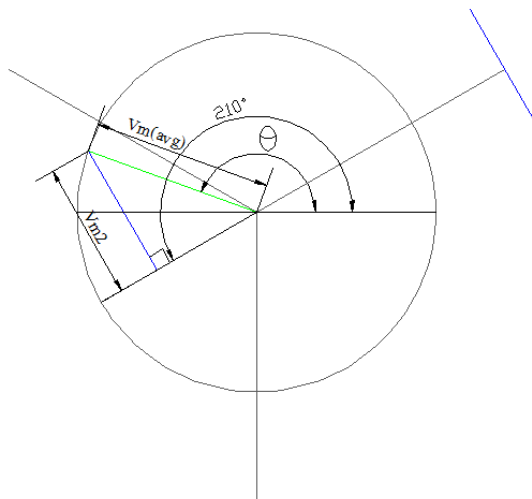
$$V_{m1} = V_{m \text{ avg}} \sin 150 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.87)$$

จำนวนรอบของมอเตอร์ตัวที่ 1 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m1}}{R} = \sin 150 - \theta \quad (3.88)$$

$$S_{m1} = R \sin 150 - \theta \text{ มิลลิเมตร (mm)} \quad (3.89)$$

ความเร็วของมอเตอร์ตัวที่ 2



ภาพที่ 3-34 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 2

จากภาพที่ 3-34 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m2}}{V_{m \text{ avg}}} = \sin 210 - \theta \quad (3.90)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

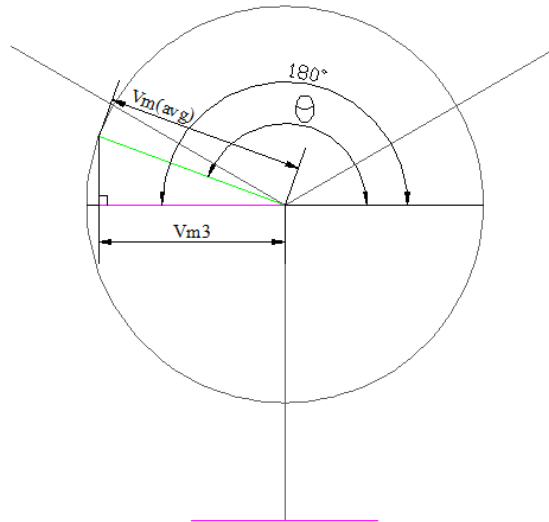
$$V_{m2} = V_{m \text{ avg}} \sin 210 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.91)$$

จำนวนรอบของมอเตอร์ตัวที่ 2 ที่ใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m2}}{R} = \sin 210 - \theta \quad (3.92)$$

$$S_{m2} = R \sin 210 - \theta \text{ มิลลิเมตร (mm)} \quad (3.93)$$

ความเร็วของมอเตอร์ตัวที่ 3



ภาพที่ 3-35 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 2

จากภาพที่ 3-35 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m3}}{-V_{m \text{ avg}}} = \cos (180 - \theta) \quad (3.94)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

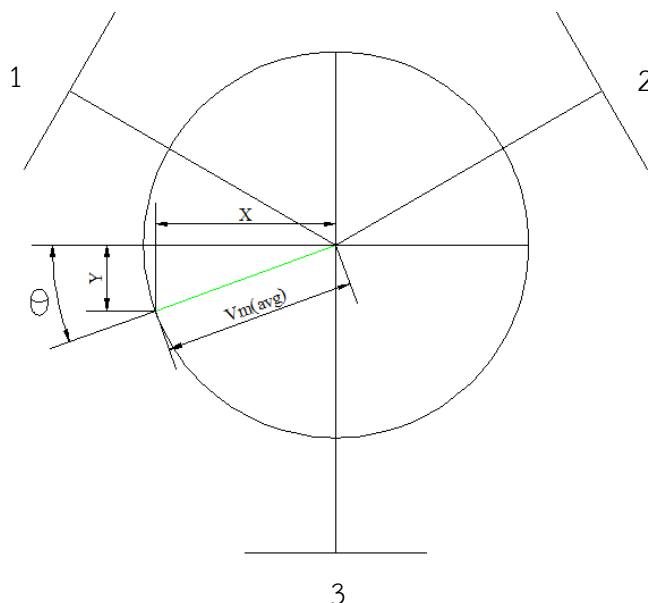
$$V_{m3} = -V_{m \text{ avg}} \cos 180 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.95)$$

จำนวนรอบของมอเตอร์ตัวที่ 3 ที่ใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m3}}{R} = \cos (180 - \theta) \quad (3.96)$$

$$S_{m3} = R \cos 180 - \theta \text{ มิลลิเมตร (mm)} \quad (3.97)$$

การควบคุมความเร็วและตำแหน่งของหุ่นยนต์ใน Quadrant ที่ 3 คือ การทำงานของหุ่นยนต์จะอยู่ในช่วง $180 \leq \theta \leq 270$



ภาพที่ 3-36 การกำหนดขนาดและทิศทางใน Quadrant ที่ 3

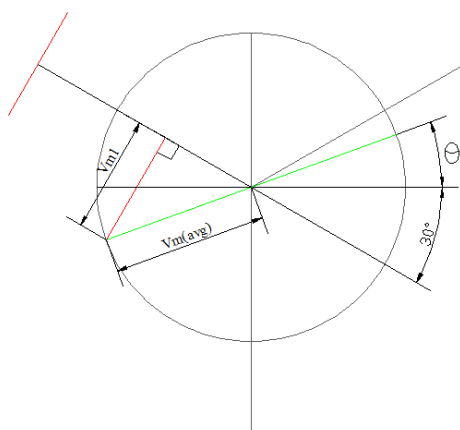
เมื่อต้องการควบคุมการทำงานในช่วง $180 - 270$ องศาและรู้ค่าของระยะของ X และ Y สามารถหามุม θ ได้โดยใช้สมการดังนี้

$$\theta = \tan^{-1} \frac{-X}{-Y} \quad (3.98)$$

หาระยะทางที่หุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งที่กำหนดโดยใช้สมการดังนี้

$$R = \sqrt{X^2 + Y^2} \quad (3.99)$$

ความเร็วของมอเตอร์ตัวที่ 1



ภาพที่ 3-37 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 3

จากภาพที่ 3-37 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่
จึงได้สมการการเคลื่อนที่

$$\frac{V_{m1}}{-V_{m \text{ avg}}} = \sin 30 + \theta \quad (3.100)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

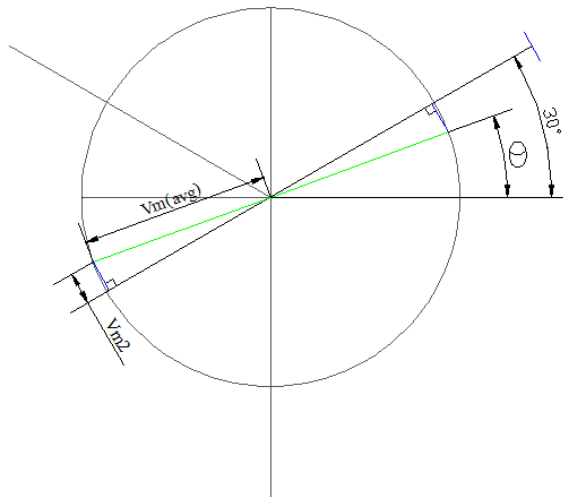
$$V_{m1} = -V_{m \text{ avg}} \sin 30 + \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.101)$$

จำนวนรอบของมอเตอร์ตัวที่ 1 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m1}}{R} = \sin 30 + \theta \quad (3.102)$$

$$S_{m1} = R \sin 30 + \theta \text{ มิลลิเมตร (mm)} \quad (3.103)$$

ความเร็วของมอเตอร์ตัวที่ 2



ภาพที่ 3-38 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 3

จากภาพที่ 3-38 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่
จึงได้สมการการเคลื่อนที่

$$\frac{V_{m2}}{-V_{m \text{ avg}}} = \sin 30 - \theta \quad (3.104)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

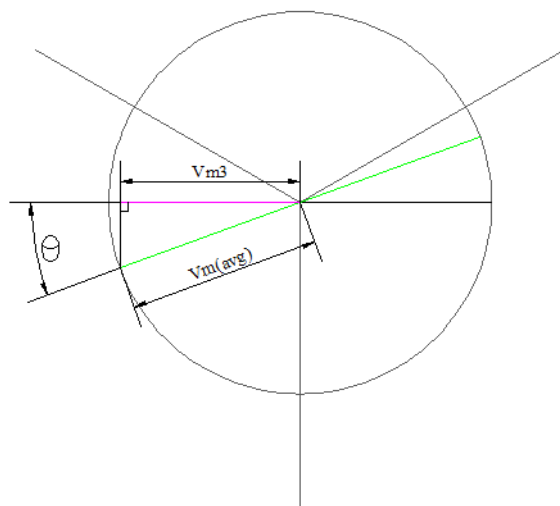
$$V_{m2} = -V_{m \text{ avg}} \sin 30 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.105)$$

จำนวนรอบของมอเตอร์ตัวที่ 2 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m2}}{R} = \sin 30 - \theta \quad (3.106)$$

$$S_{m2} = R \sin 30 - \theta \text{ มิลลิเมตร (mm)} \quad (3.107)$$

ความเร็วของมอเตอร์ตัวที่ 3



ภาพที่ 3-39 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 3

จากภาพที่ 3-39 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m3}}{-V_{m \text{ avg}}} = \cos \theta \quad (3.108)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

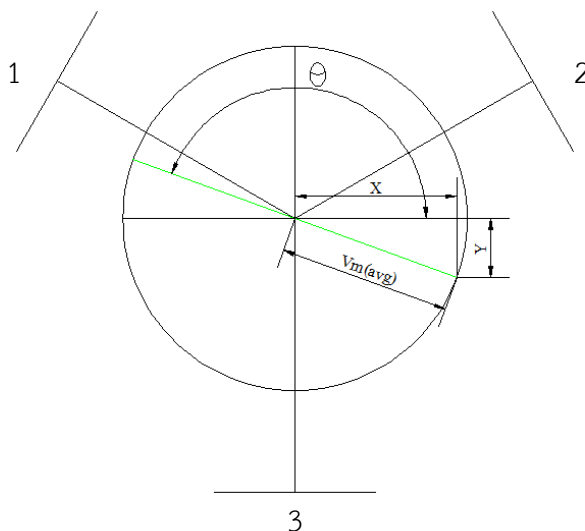
$$V_{m3} = -V_{m \text{ avg}} \cos \theta \text{ เมตรต่อวินาที (m / s)} \quad (3.109)$$

จำนวนรอบของมอเตอร์ตัวที่ 3 ที่จะใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m3}}{R} = \cos \theta \quad (3.110)$$

$$S_{m3} = R \cos \theta \text{ มิลลิเมตร (mm)} \quad (3.111)$$

การควบคุมความเร็วและตำแหน่งของหุ่นยนต์ใน Quadrant ที่ 4 คือ การทำงานของหุ่นยนต์จะอยู่ในช่วง $270 \leq \theta \leq 360$



ภาพที่ 3-40 การกำหนดขนาดและทิศทางใน Quadrant ที่ 4

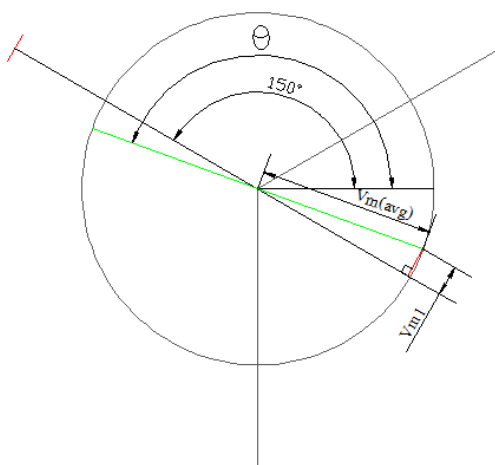
เมื่อต้องการควบคุมการทำงานในช่วง $270 - 360$ องศาและรู้ค่าของระยะของ X และ Y สามารถหามุม θ ได้โดยใช้สมการดังนี้

$$\theta = \tan^{-1} \frac{X}{-Y} + 90 \quad (3.112)$$

หาระยะทางที่หุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งที่กำหนดโดยใช้สมการดังนี้

$$R = \sqrt{X^2 + Y^2} \quad (3.113)$$

ความเร็วของมอเตอร์ตัวที่ 1



ภาพที่ 3-41 การกำหนดความเร็วล้อที่ 1 ใน Quadrant ที่ 4

จากภาพที่ 3-41 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m1}}{-V_{m \text{ avg}}} = \sin 150 - \theta \quad (3.114)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

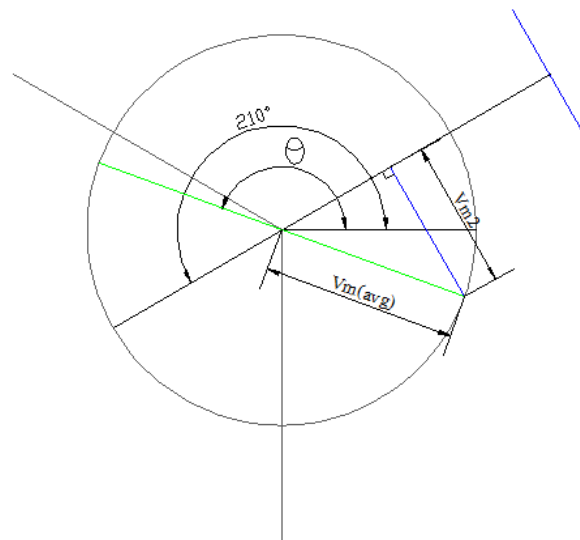
$$V_{m1} = -V_{m \text{ avg}} \sin 150 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.115)$$

จำนวนรอบของมอเตอร์ตัวที่ 1 ที่จะใช้ในการส่งระยะทางการ

$$\frac{S_{m1}}{R} = \sin 150 - \theta \quad (3.116)$$

$$S_{m1} = R \sin 150 - \theta \text{ มิลลิเมตร (mm)} \quad (3.117)$$

ความเร็วของมอเตอร์ตัวที่ 2



ภาพที่ 3-42 การกำหนดความเร็วล้อที่ 2 ใน Quadrant ที่ 4

จากภาพที่ 3-42 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m2}}{-V_{m \text{ avg}}} = \sin 210 - \theta \quad (3.118)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

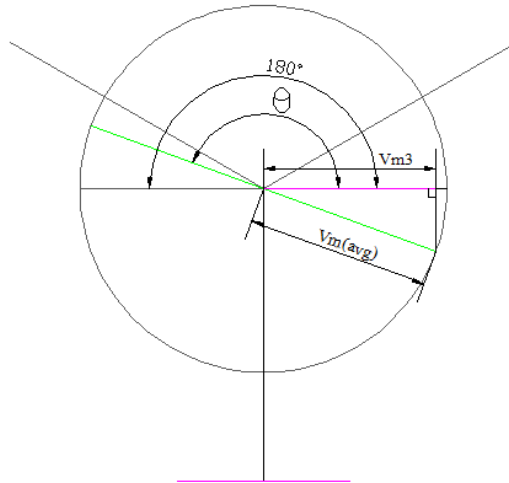
$$V_{m2} = -V_{m \text{ avg}} \sin 210 - \theta \text{ เมตรต่อวินาที (m/s)} \quad (3.119)$$

จำนวนรอบของมอเตอร์ตัวที่ 2 ที่ใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m2}}{R} = \sin 210 - \theta \quad (3.120)$$

$$S_{m2} = R \sin 210 - \theta \text{ มิลลิเมตร (mm)} \quad (3.121)$$

ความเร็วของมอเตอร์ตัวที่ 3



ภาพที่ 3-43 การกำหนดความเร็วล้อที่ 3 ใน Quadrant ที่ 4

จากภาพที่ 3-43 $V_{m \text{ avg}}$ จะเป็นทิศทางอ้างอิงที่จะให้หุ่นยนต์เคลื่อนที่

จึงได้สมการการเคลื่อนที่

$$\frac{V_{m3}}{-V_{m \text{ avg}}} = \cos (180 - \theta) \quad (3.122)$$

ดังนั้นเมื่อแก้สมการออกมาจะได้

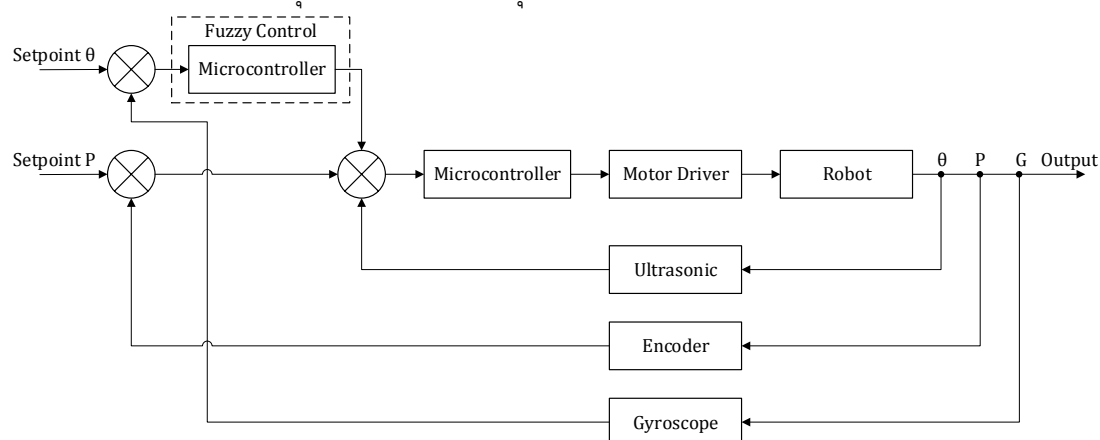
$$V_{m3} = -V_{m \text{ avg}} \cos (180 - \theta) \text{ เมตรต่อวินาที (m / s)} \quad (3.123)$$

จำนวนรอบของมอเตอร์ตัวที่ 3 ที่ใช้ในการส่งระยะทางการเคลื่อนที่

$$\frac{S_{m3}}{R} = \cos (180 - \theta) \quad (3.124)$$

$$S_{m3} = R \cos 180 - \theta \text{ มิลลิเมตร (mm)} \quad (3.125)$$

3.5.2 การควบคุมการเคลื่อนที่ของหุ่นยนต์



ภาพที่ 3-44 บล็อกไดอะแกรมของการควบคุมการทำงานหุ่นยนต์

ในการควบคุมการทำงานของหุ่นยนต์จะสั่งระยะทางและมุมที่ต้องการ โดยที่การทำงานของหุ่นยนต์จะมีเซ็นเซอร์ไจโรสโคป (Gyroscope) คอยควบคุมทิศทางในการทำงานเพื่อไม่ให้หุ่นยนต์เคลื่อนที่ออกจากมุมที่กำหนดไว้ โดยจะกำหนดให้ควบคุมแบบพีซี และเขียนไว้ในไมโครคอนโทรลเลอร์ และในการควบคุมระยะทางจะใช้เอ็นโค้ดเดอร์ ในการปรับเทียบการทำงานให้ไปยังตำแหน่งที่ต้องการ

ตารางที่ 3-9 กฎฟัซซีที่ใช้ในหุ่นยนต์

$G \backslash \Delta G$	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

จากตารางที่ 3-6 ตารางกฎฟัซซีแบบ 3 สมาชิก จะมีกฎทั้งหมด 9 กฎ ในการเรียกใช้กฎนั้นจะขึ้นอยู่กับค่าของเซ็นเซอร์ที่อ่านได้จากครั้งก่อน (G) และค่าของเซ็นเซอร์ที่อ่านได้ในครั้งต่อมา (ΔG) ในฟังก์ชันฟัซซีจะอยู่ที่โปรแกรมในไมโครคอนโทรลเลอร์ จะถูกเรียกใช้เมื่อเป็นไปตามกฎที่ตั้งไว้ เพื่อให้หุ่นยนต์สามารถเคลื่อนที่ไปยังตำแหน่งที่ต้องการ

ตารางที่ 3-10 การกำหนดค่า G ในเซตฟัซซี แบบ 3 สมาชิก

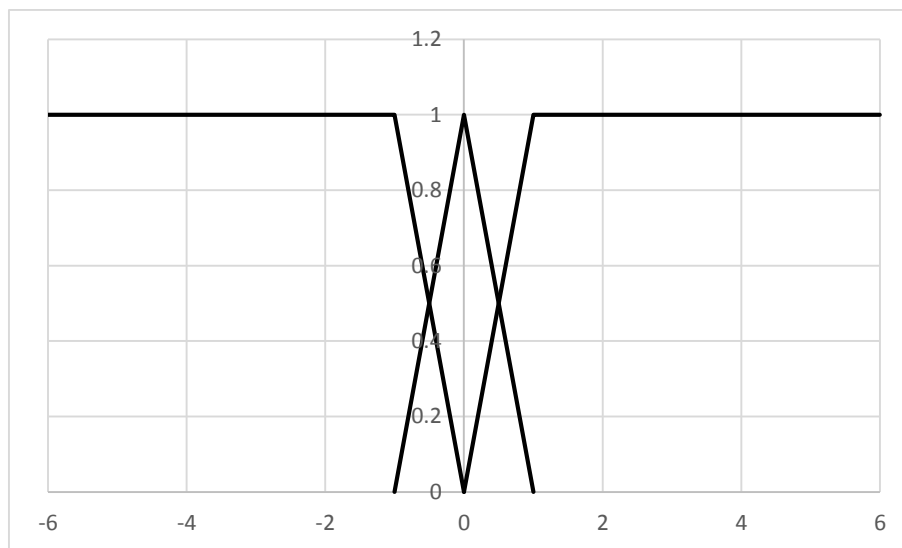
Membership function	Shape	Point
N	Trapezoid	-6;-1
Z	Triangle	-1;0;1
P	Trapezoid	1;6

ตารางที่ 3-11 การกำหนดค่า ΔG ในเซตฟัซซี แบบ 3 สมาชิก

Membership function	Shape	Point
N	Trapezoid	-6;-1
Z	Triangle	-1;0;1
P	Trapezoid	1;6

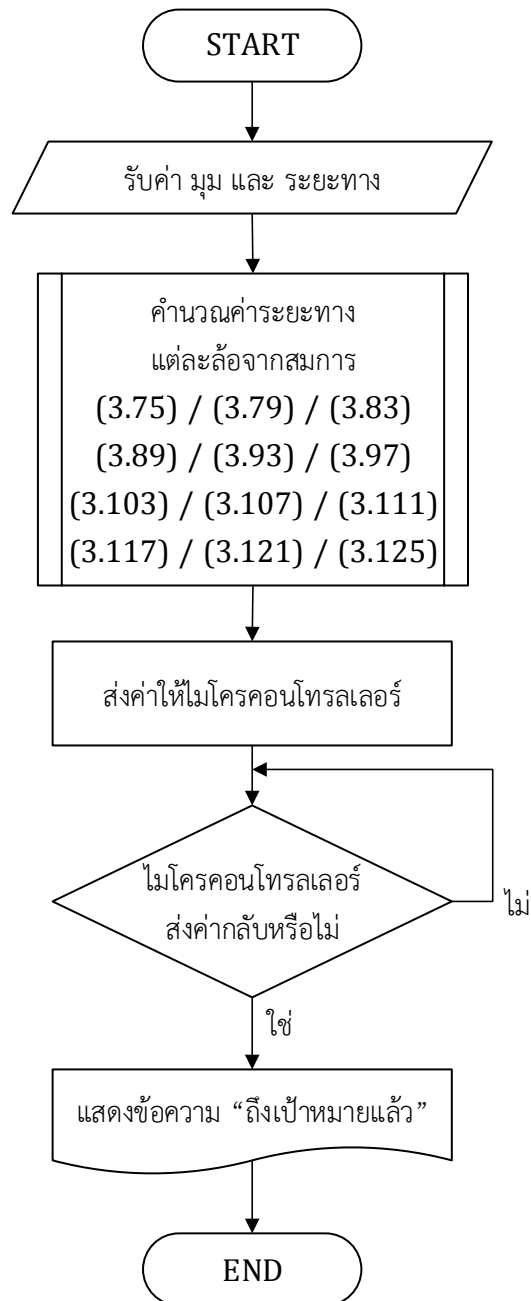
ตารางที่ 3-12 การกำหนดค่าเอาต์พุตในเซตฟัซซี แบบ 3 สมาชิก

Membership function	Shape	Point
N	Trapezoid	-6;-1
Z	Triangle	-1;0;1
P	Trapezoid	1;6



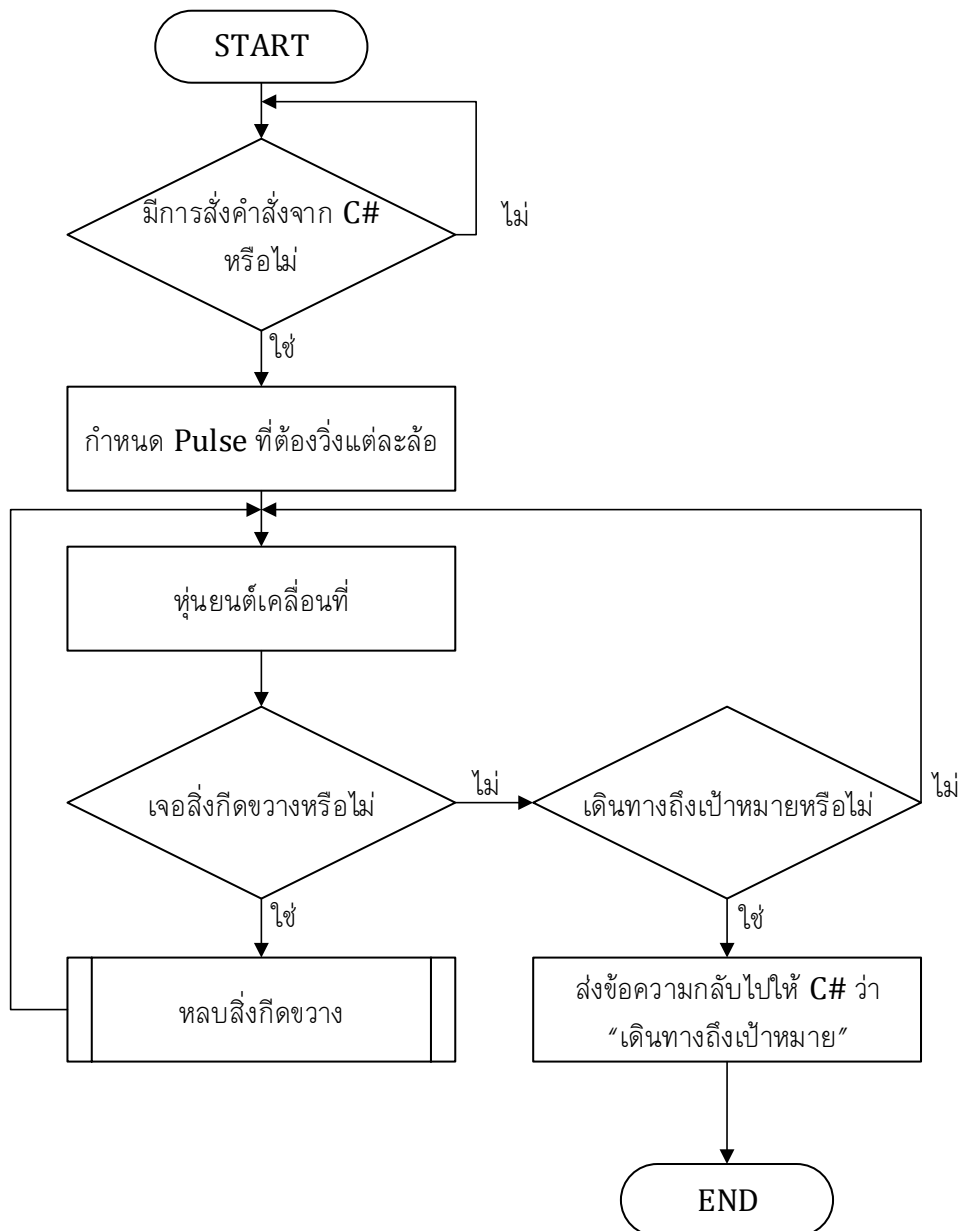
ภาพที่ 3-45 ฟังก์ชันสมาชิกของตัวแปร G , ΔG และเอาต์พุต

3.5.3 โฟลว์ชาร์ตการทำงานบน C#/Android



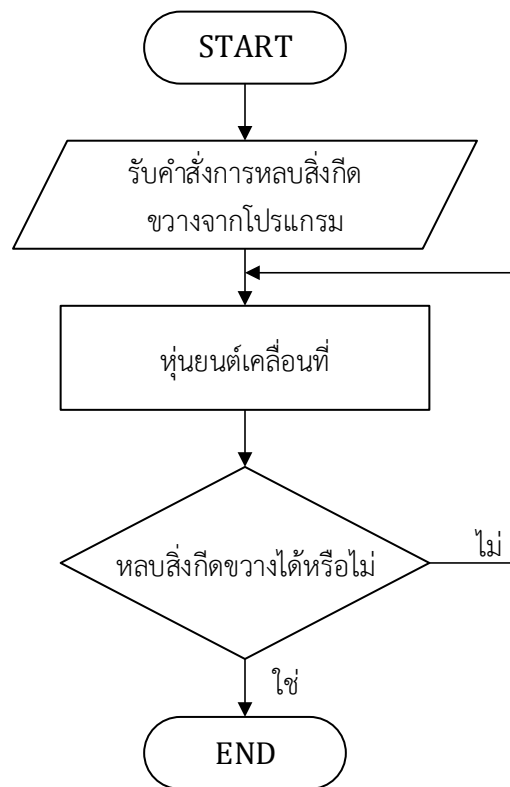
ภาพที่ 3-46 โฟลว์ชาร์ตการทำงานบน C#/Android

3.5.4 โฟลว์ชาร์ตการทำงานบนไมโครคอนโทรลเลอร์



ภาพที่ 3-47 โฟลว์ชาร์ตการทำงานบนไมโครคอนโทรลเลอร์

3.5.5 โฟลว์ชาร์ตการหลบสิ่งกีดขวางของหุ่นยนต์



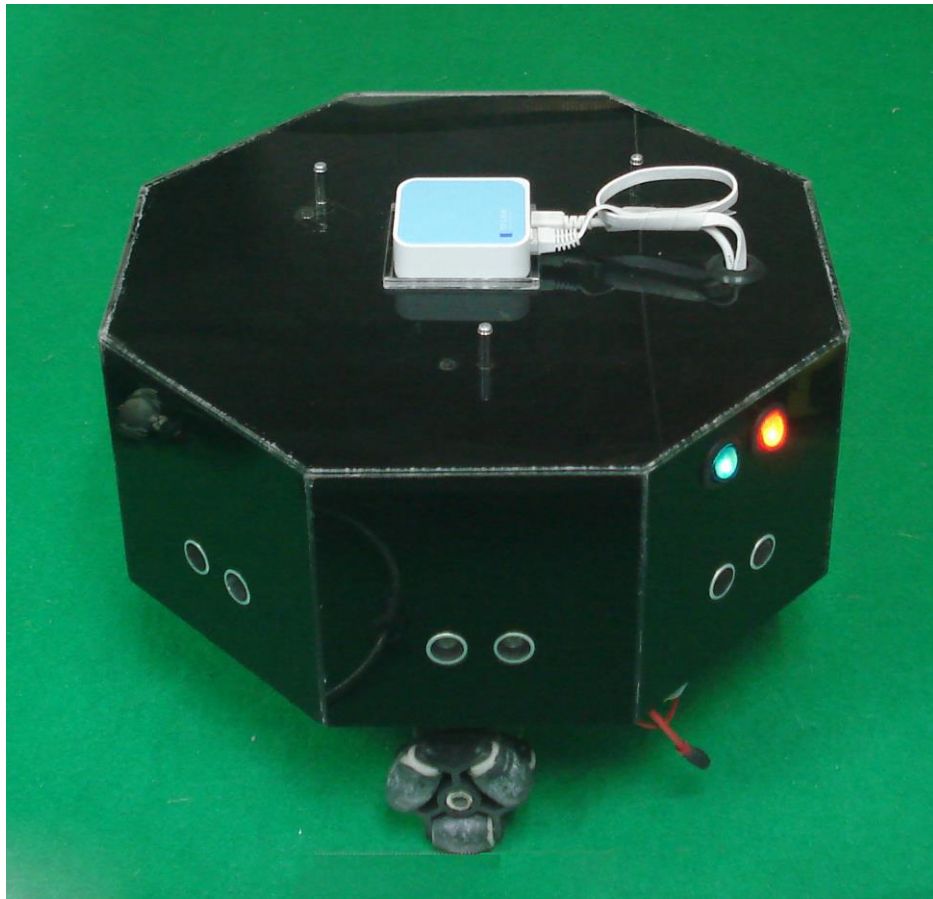
ภาพที่ 3-48 โฟลว์ชาร์ตการหลบสิ่งกีดขวางของหุ่นยนต์

บทที่ 4

ผลการดำเนินงาน

ในการดำเนินงานของโครงการการควบคุมหุ่นยนต์ล้อหลายทิศทาง ผลการดำเนินงานและการทดลองประกอบไปด้วย

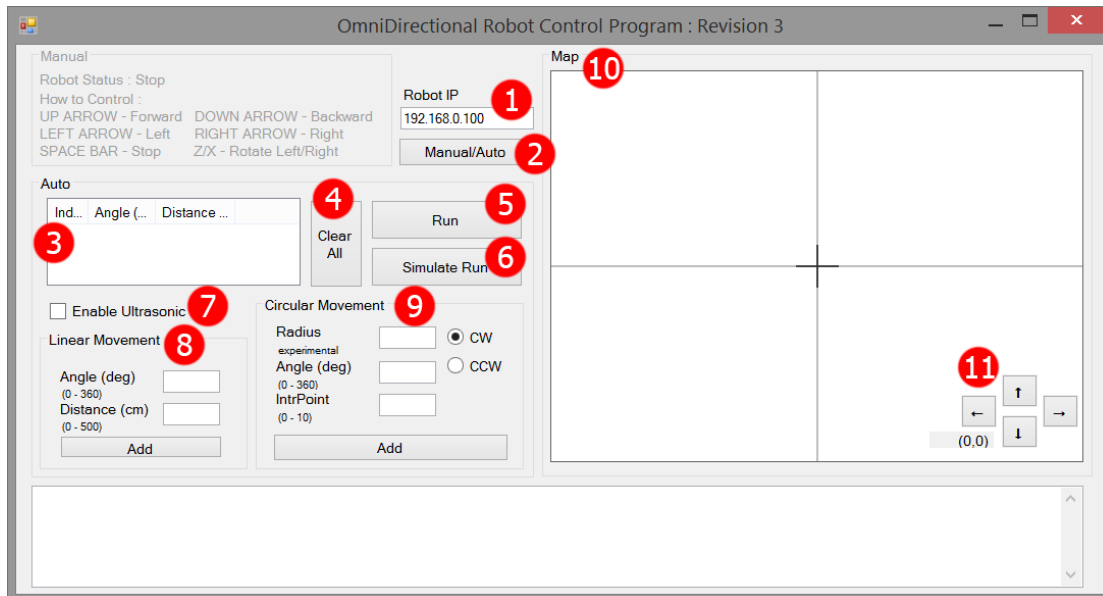
4.1 หุ่นยนต์ที่ประกอบสมบูรณ์



ภาพที่ 4-1 หุ่นยนต์ที่ประกอบสมบูรณ์

4.2 หน้าต่างโปรแกรมของการควบคุมหุ่นยนต์ล้อหลายทิศทาง

4.2.1 หน้าต่างโปรแกรมจากการพัฒนาโปรแกรมด้วย Visual Studio 2012



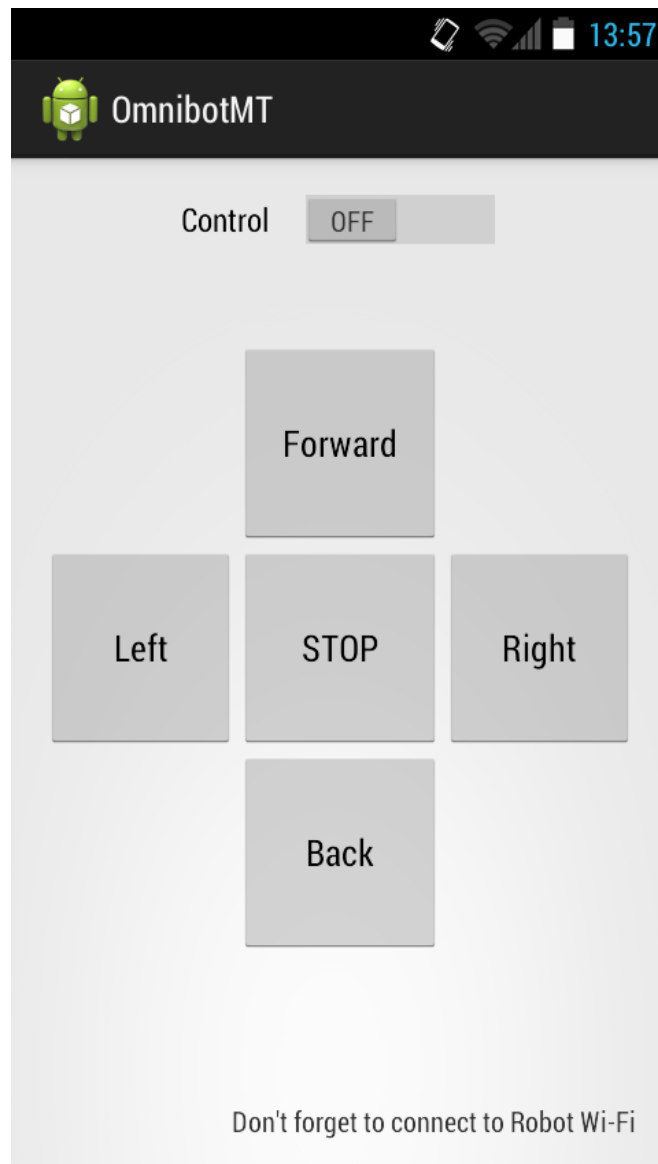
ภาพที่ 4-2 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Visual Studio 2012

ส่วนประกอบต่างๆ ของโปรแกรมมีดังนี้

- หมายเลข 1 : ช่องสำหรับใส่ IP Address ของหุ่นยนต์
- หมายเลข 2 : เป็นการปรับตัวเลือกการควบคุมระหว่างอัตโนมัติและควบคุมด้วยมือ
- หมายเลข 3 : แสดงรายการตำแหน่งของหุ่นยนต์ที่จะวิ่ง
- หมายเลข 4 : ปุ่มกดล้างค่ารายการตำแหน่งของหุ่นยนต์ที่จะวิ่ง
- หมายเลข 5 : ปุ่มกดสั่งให้หุ่นยนต์วิ่งจริง
- หมายเลข 6 : ปุ่มกดจำลองการวิ่งของหุ่นยนต์ ให้ออกเฉพาะแผนที่เท่านั้น
- หมายเลข 7 : ปุ่มที่ถูกใช้สำหรับการใช้เซ็นเซอร์ Ultrasonic ในการช่วยหลบสิ่งกีดขวาง
- หมายเลข 8 : ส่วนที่ให้ผู้ใช้ออกค่าการเคลื่อนที่ของหุ่นยนต์ โดยที่หุ่นยนต์วิ่งไปแนวเส้นตรง
- หมายเลข 9 : ส่วนที่ให้ผู้ใช้ออกค่าการเคลื่อนที่ของหุ่นยนต์ โดยที่หุ่นยนต์วิ่งไปแนวเส้นโค้ง
- หมายเลข 10 : แผนที่เส้นทางของหุ่นยนต์ที่เดินในระบบอัตโนมัติ
- หมายเลข 11 : ปุ่มกดสำหรับการเลื่อนภาพของแผนที่

4.2.2 หน้าต่างโปรแกรมจากการพัฒนาโปรแกรมด้วย Android Studio

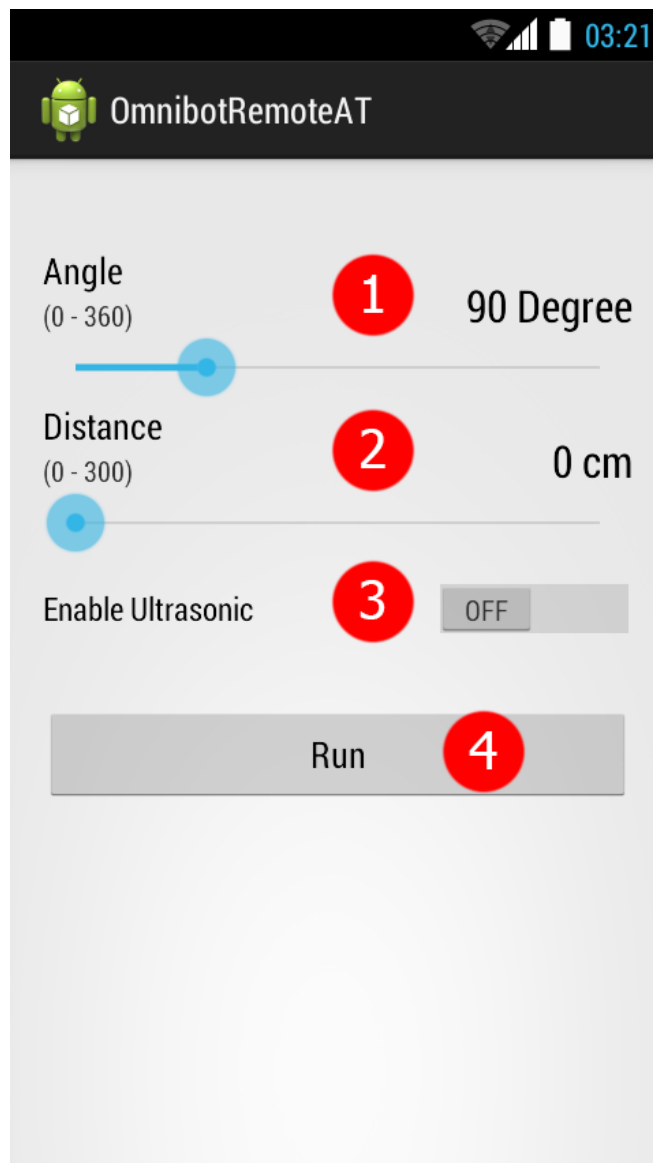
- แบบควบคุมด้วยมือ



ภาพที่ 4-3 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Android Studio (1)

โปรแกรมจะมีปุ่ม **Control** เพื่อเริ่มต้นควบคุมหุ่นยนต์ และปุ่มควบคุมเดินหน้า, ถอยหลัง, ซ้าย, ขวา, และหยุด

- แบบอัตโนมัติ



ภาพที่ 4-4 หน้าต่างโปรแกรมหุ่นยนต์จากการพัฒนาโปรแกรมด้วย Android Studio (2)

ส่วนประกอบต่างๆ ของโปรแกรมมีดังนี้

หมายเลข 1 : แถบปรับมุมที่จะส่งหุ่นยนต์

หมายเลข 2 : แถบปรับระยะทางที่จะส่งหุ่นยนต์

หมายเลข 3 : ปุ่มสวิตช์สำหรับการใช้เซ็นเซอร์ Ultrasonic ในการช่วยหลบสิ่งกีดขวาง

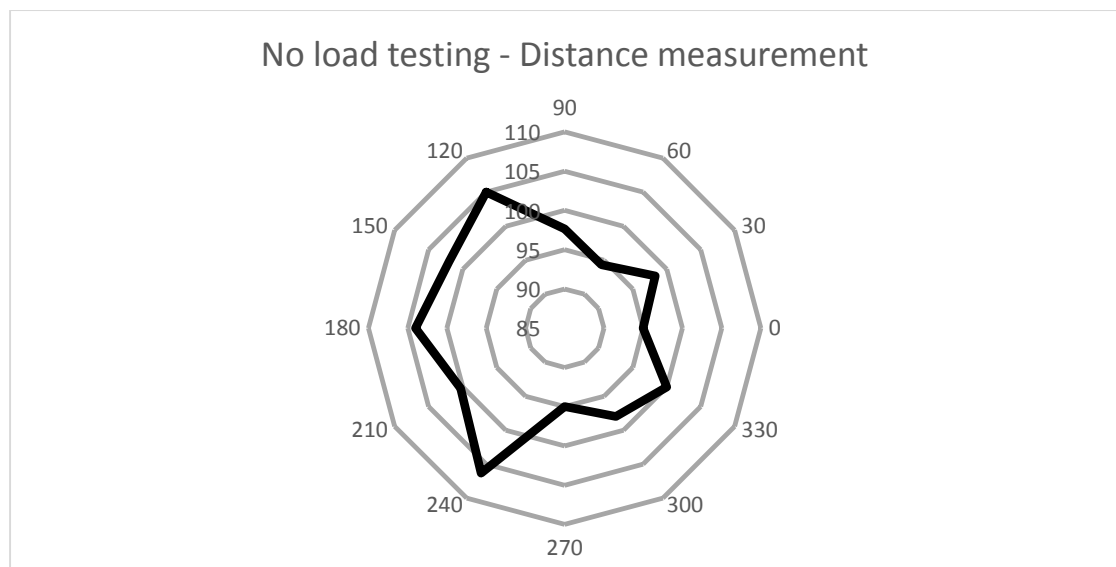
หมายเลข 4 : แสดงคำสั่งที่จะส่งหุ่นยนต์ไป

4.3 ผลการทดสอบการควบคุมหุ่นยนต์โดยไม่มีสิ่งกีดขวาง

การทดสอบความแม่นยำจะทดสอบโดยการสั่งหุ่นยนต์เคลื่อนที่บนพื้นผิวสี่เหลี่ยม ในระยะทาง 100 เซนติเมตร มุมเริ่มจาก 0 องศา เพิ่มขึ้นละ 30 องศาจนถึงมุม 330 องศา โดยแบ่งผลที่ได้จากการทดสอบออกเป็นระยะทางที่วัดได้ และ มุมที่วัดได้

ตารางที่ 4-1 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบไม่มีโหลด

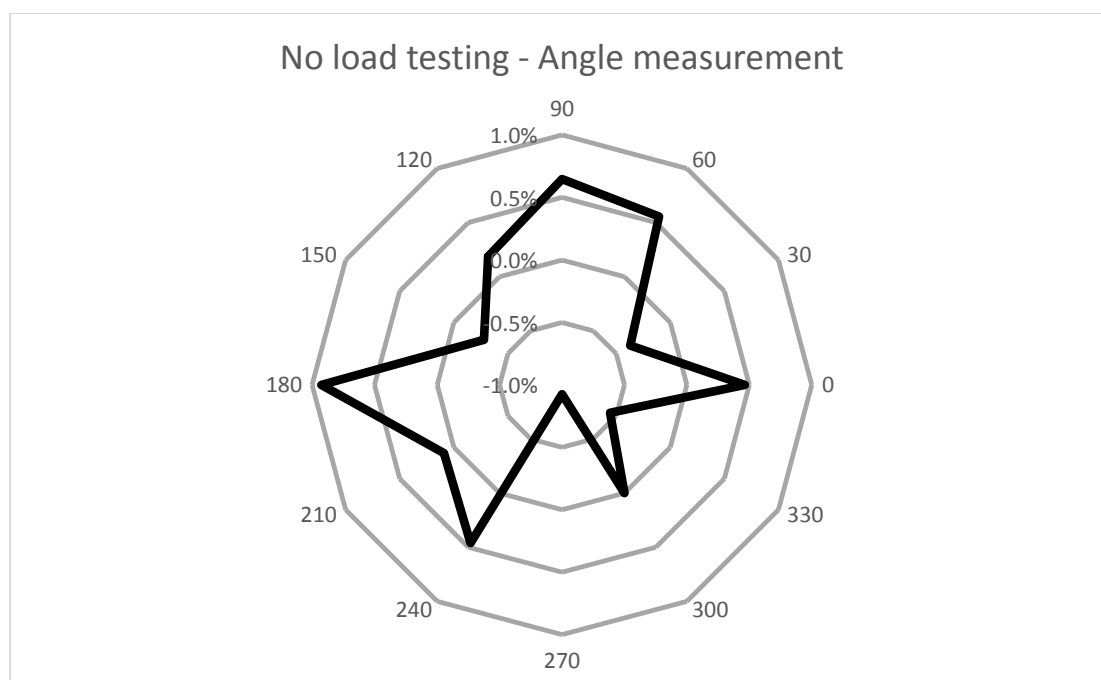
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	98	93	94	95	5.0%	2.64
30	102	94	99	98.33	1.7%	4.04
60	90	94	99	94.33	5.7%	4.50
90	95	101	97	97.67	2.3%	3.05
120	114	105	96	105	-5.0%	9.00
150	102	102	102	102	-2.0%	0.00
180	106	101	105	104	-4.0%	2.64
210	100	101	100	100.33	-0.3%	0.57
240	109	103	107	106.33	-6.3%	3.05
270	95	94	96	95	5.0%	1.00
300	102	96	96	98	2.0%	3.46
330	96	102	102	100	0.0%	3.46



ภาพที่ 4-5 แผนภูมิผลการทดสอบวัดระยะแบบไม่มีโหลด

ตารางที่ 4-2 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบไม่มีโหลด

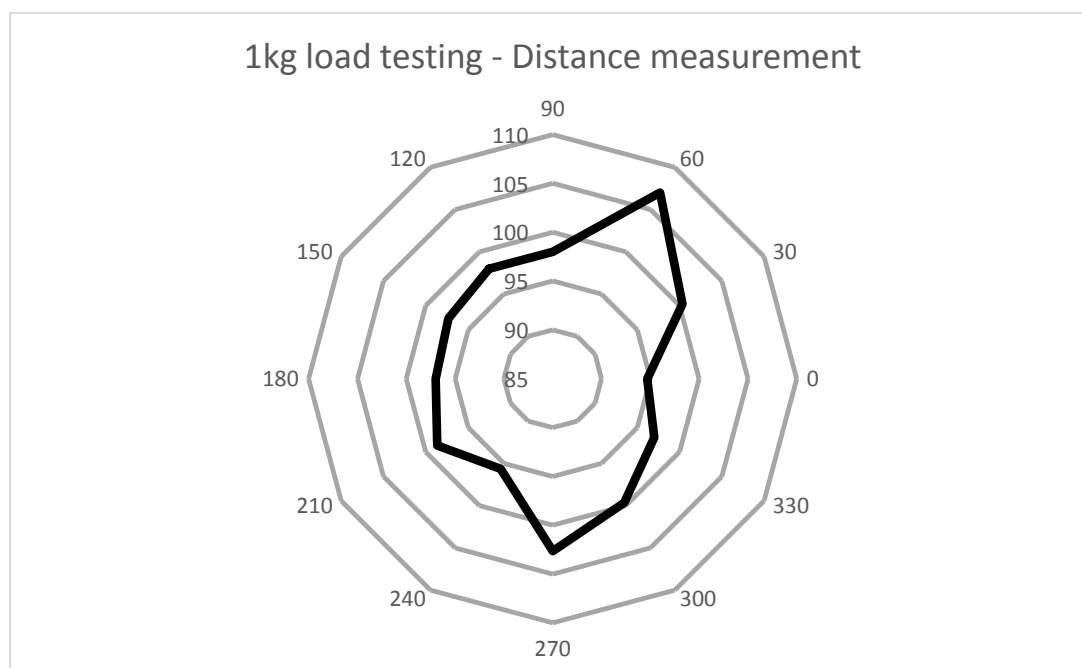
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	2	2	1	1.67	0.5%	0.57
30	31	27	28	28.67	-0.4%	2.08
60	70	59	57	62	0.6%	7.00
90	93	92	92	92.33	0.6%	0.57
120	120	121	121	120.67	0.2%	0.57
150	149	149	149	149	-0.3%	0.00
180	186	182	182	183.33	0.9%	2.30
210	211	207	213	210.33	0.1%	3.05
240	242	242	241	241.67	0.5%	0.57
270	265	267	268	266.67	-0.9%	1.52
300	303	298	299	300	0.0%	2.64
330	329	327	328	328	-0.6%	1.00



ภาพที่ 4-6 แผนภูมิผลการทดสอบวัดมุมแบบไม่มีโหลด

ตารางที่ 4-3 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 1 กิโลกรัม

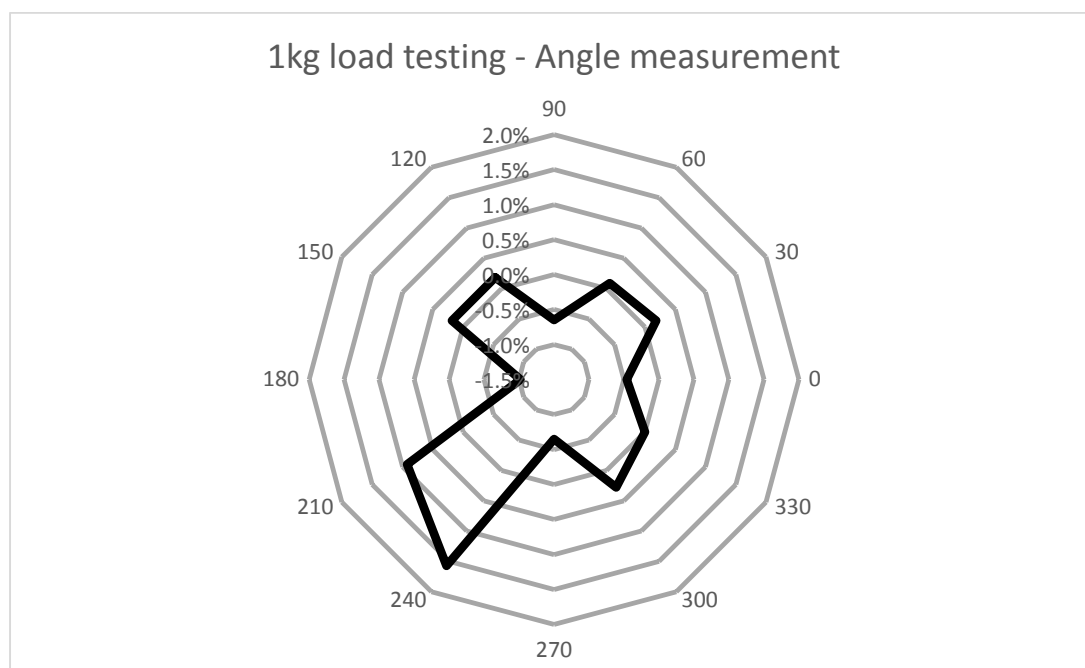
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	98	92	94	94.67	5.3%	3.05
30	100	104	97	100.33	-0.3%	3.51
60	107	104	110	107	-7.0%	3.00
90	103	90	101	98	2.0%	7.00
120	97	102	95	98	2.0%	3.60
150	100	93	99	97.33	2.7%	3.78
180	101	98	92	97	3.0%	4.58
210	98	97	101	98.67	1.3%	2.08
240	98	95	94	95.67	4.3%	2.08
270	103	112	93	102.67	-2.7%	9.50
300	104	94	101	99.67	0.3%	5.13
330	100	98	93	97	3.0%	3.60



ภาพที่ 4-7 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 1 กิโลกรัม

ตารางที่ 4-4 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 1 กิโลกรัม

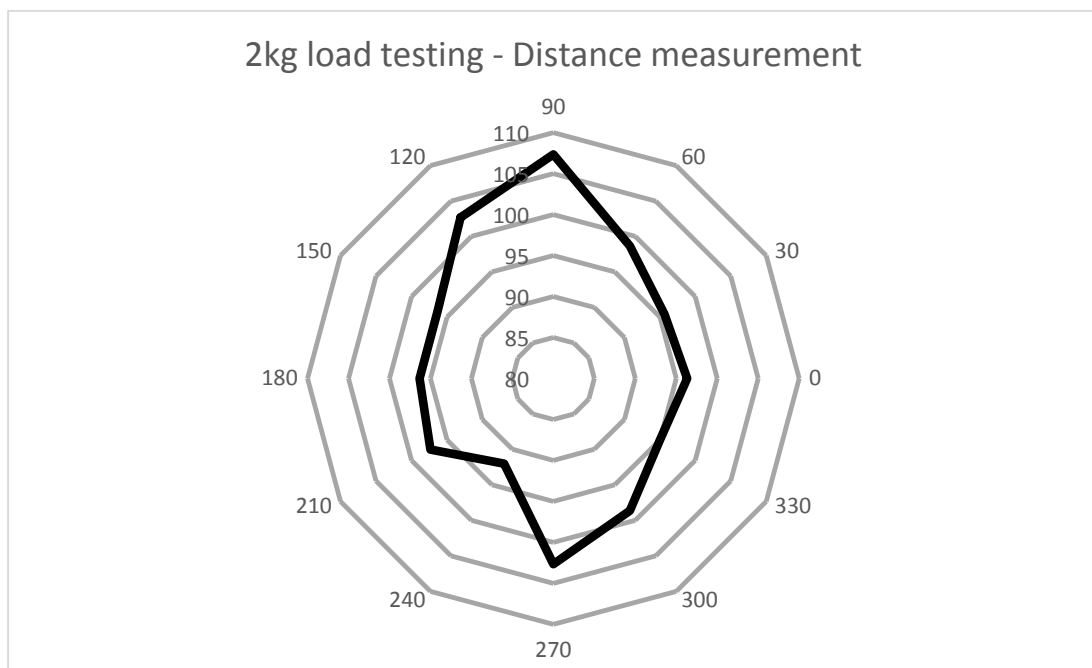
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	-2	0	-3	-1.67	1.6%	1.53
30	28	33	31	30.67	-2.2%	2.52
60	60	62	59	60.33	-0.6%	1.53
90	87	90	86	87.67	12.3%	2.08
120	117	122	123	120.67	-0.6%	3.21
150	149	153	150	150.67	-0.4%	2.08
180	176	175	178	176.33	2.0%	1.53
210	214	216	210	213.33	-1.6%	3.06
240	246	246	245	245.67	-2.4%	0.58
270	268	267	268	267.67	0.9%	0.58
300	300	300	303	301	-0.3%	1.73
330	332	328	330	330	0.0%	2.00



ภาพที่ 4-8 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 1 กิโลกรัม

ตารางที่ 4-5 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 2 กิโลกรัม

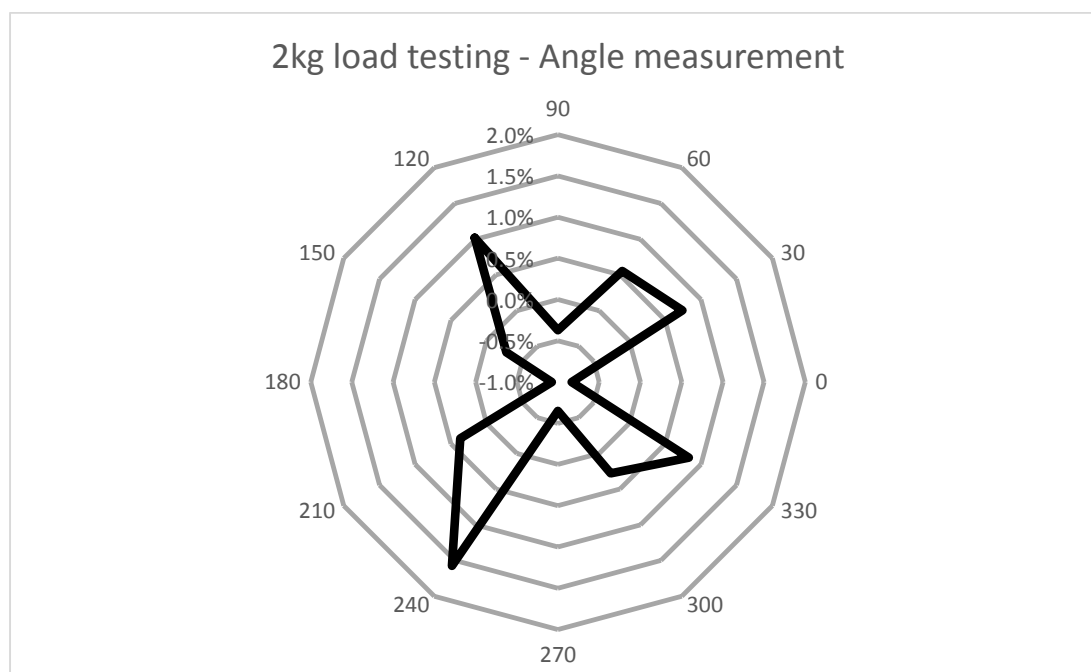
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	97	93	99	96.33	3.7%	3.05
30	91	95	101	95.67	4.3%	5.03
60	90	102	104	98.67	1.3%	7.57
90	105	107	110	107.33	-7.3%	2.51
120	107	100	101	102.67	-2.7%	3.78
150	99	95	95	96.33	3.7%	2.30
180	91	96	102	96.33	3.7%	5.50
210	97	99	96	97.33	2.7%	1.52
240	93	92	91	92	8.0%	1.00
270	104	94	98	102.67	-2.7%	9.50
300	107	90	99	98.67	1.3%	8.50
330	91	95	99	95	5.0%	4.00



ภาพที่ 4-9 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 2 กิโลกรัม

ตารางที่ 4-6 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 2 กิโลกรัม

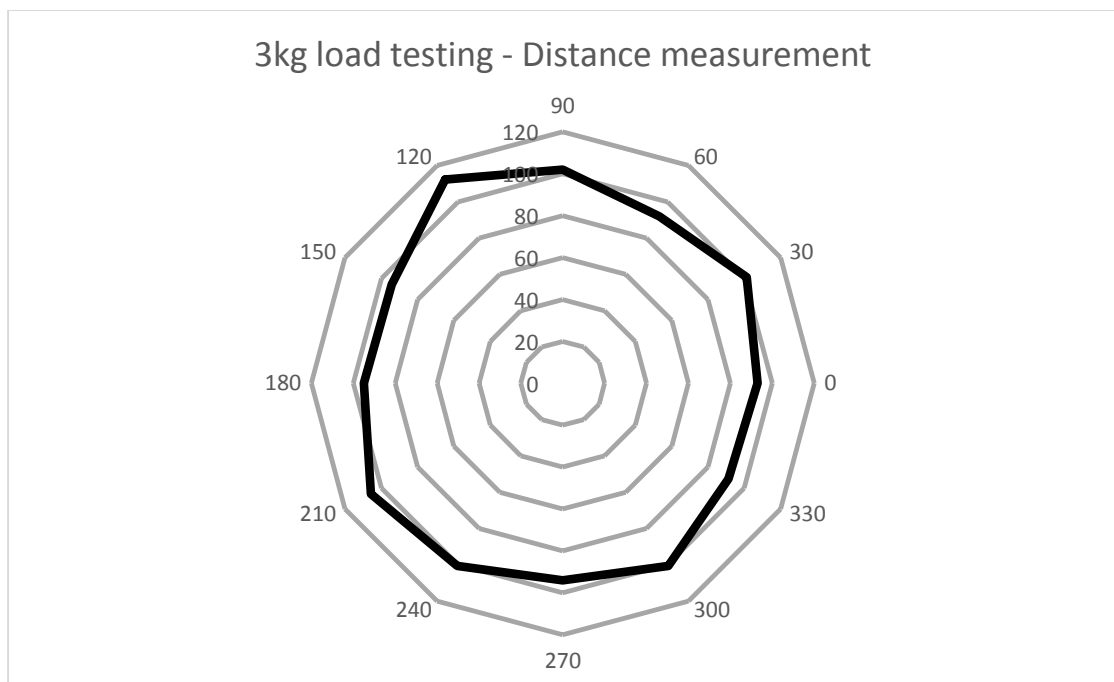
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	-3	-2	-4	-3.00	1.6%	1.00
30	31	33	34	32.67	-8.9%	1.52
60	56	65	65	62.00	-3.3%	5.19
90	87	90	89	88.67	1.5%	1.52
120	119	127	125	123.67	-3.1%	4.16
150	150	150	147	149.00	0.7%	1.73
180	177	177	176	176.67	1.9%	0.57
210	210	212	212	211.33	-0.6%	1.15
240	244	246	247	245.67	-2.4%	1.52
270	267	267	273	267.67	0.9%	0.57
300	302	301	300	301.00	-0.3%	1.00
330	333	333	333	333.00	-0.9%	0.00



ภาพที่ 4-10 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 2 กิโลกรัม

ตารางที่ 4-7 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 3 กิโลกรัม

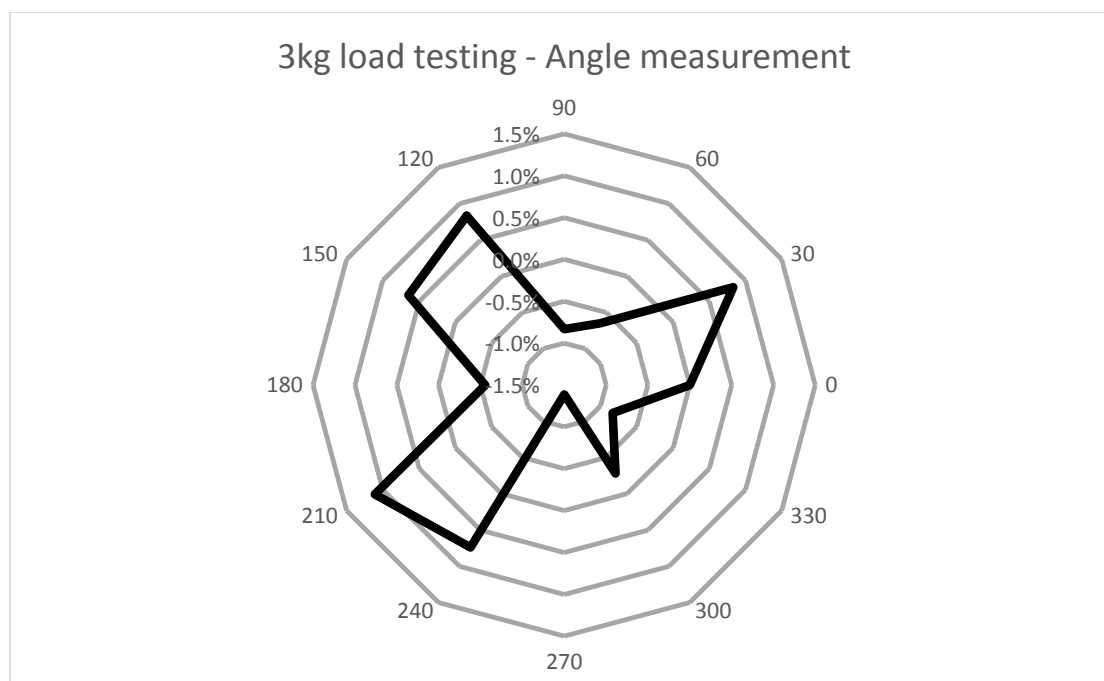
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	98	90	91	93	7.0%	4.36
30	98	104	102	101.33	-1.3%	3.05
60	91	92	93	92	8.0%	1.00
90	99	107	100	102	-2.0%	4.35
120	113	111	113	112.33	-12.3%	1.15
150	89	101	93	94.33	5.7%	6.11
180	94	95	96	95	5.0%	1.00
210	106	100	111	105.67	-5.7%	5.50
240	102	102	98	100.67	-0.7%	2.30
270	95	92	95	94	6.0%	1.73
300	104	99	99	100.67	-0.7%	2.88
330	98	84	92	91.33	8.7%	7.02



ภาพที่ 4-11 แผนภูมิผลการทดสอบวัดระยะแบบมีโหลด 3 กิโลกรัม

ตารางที่ 4-8 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีโหลด 3 กิโลกรัม

องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	-2	1	1	0	0.0%	1.73
30	34	32	33	33	0.8%	1.00
60	59	58	56	57.66667	-0.6%	1.53
90	87	86	88	87	-0.8%	1.00
120	121	123	125	123	0.8%	2.00
150	152	153	152	152.3333	0.6%	0.58
180	176	180	178	178	-0.6%	2.00
210	213	215	214	214	1.1%	1.00
240	244	243	241	242.6667	0.7%	1.53
270	265	265	265	265	-1.4%	0.00
300	299	300	298	299	-0.3%	1.00
330	328	322	331	327	-0.8%	4.58



ภาพที่ 4-12 แผนภูมิผลการทดสอบวัดมุมแบบมีโหลด 3 กิโลกรัม

4.4 ผลการทดสอบการควบคุมหุ่นยนต์โดยมีสิ่งกีดขวาง

การทดสอบความหลบสิ่งกีดขวาง จะทดสอบโดยการสั่งหุ่นยนต์เคลื่อนที่บนพื้นผิวล็กหาด ในระยะทาง 150 เซนติเมตร มีสิ่งกีดขวางที่ระยะห่างจากจุดเริ่มต้น 50 เซนติเมตร มุมเริ่มจาก 0 องศา เพิ่มขึ้นละ 30 องศาจนไปถึงมุม 330 องศา โดยแบ่งผลที่ได้จากการทดสอบออกเป็น ระยะทางที่วัดได้ และ มุมที่วัดได้

ตารางที่ 4-9 ผลการวัดระยะทางจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีสิ่งกีดขวาง

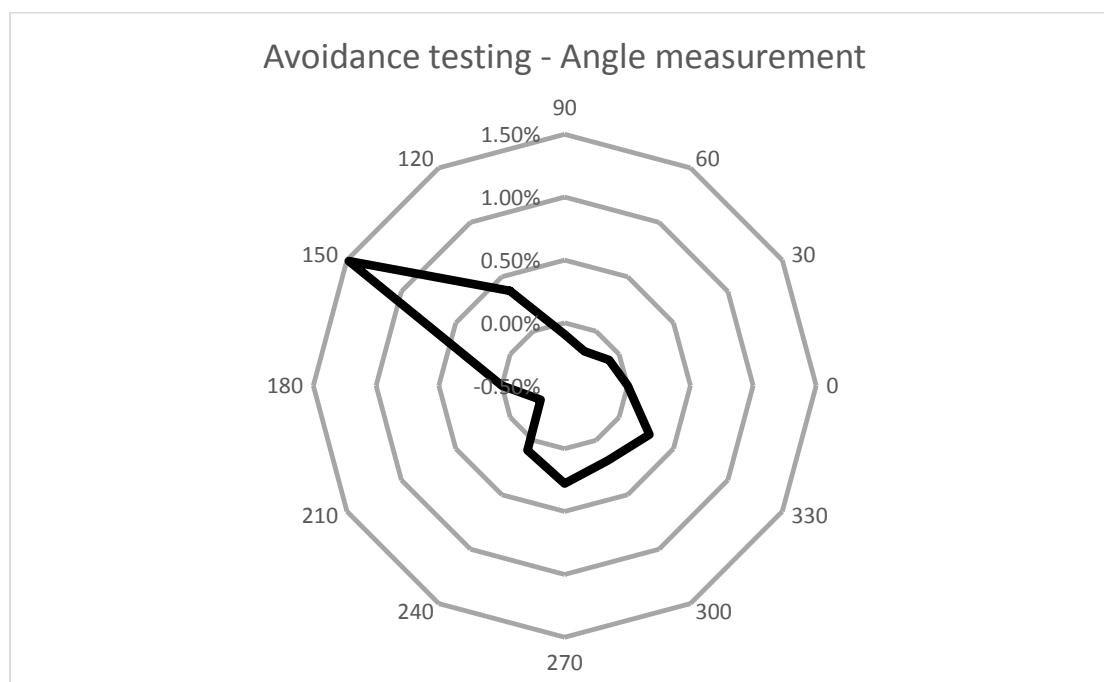
องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	155	170	145	156.67	4.4%	12.58
30	153	150	148	150.33	0.2%	2.52
60	145	137	141	141	-6.0%	4.00
90	142	147	152	147	-2.0%	5.00
120	141	141	147	143	-4.7%	3.46
150	165	146	166	159	6.0%	11.27
180	150	150	146	148.67	-0.9%	2.31
210	164	152	126	147.33	-1.8%	19.43
240	150	145	148	147.67	-1.6%	2.52
270	148	150	151	149.67	-0.2%	1.53
300	140	143	151	144.67	-3.6%	5.69
330	160	128	156	148	-1.3%	17.44



ภาพที่ 4-13 แผนภูมิผลการทดสอบวัดระยะแบบหลบสิ่งกีดขวาง

ตารางที่ 4-10 ผลการวัดมุมจากการทดสอบหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางแบบมีสิ่งกีดขวาง

องศาที่ทดสอบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความผิดพลาด	SD
0	155	170	145	156.67	4.4%	0.00
30	153	150	148	150.33	0.2%	3.51
60	145	137	141	141	-6.0%	1.15
90	142	147	152	147	-2.0%	2.08
120	141	141	147	143	-4.7%	0.58
150	165	146	166	159	6.0%	0.58
180	150	150	146	148.67	-0.9%	3.00
210	164	152	126	147.33	-1.8%	7.00
240	150	145	148	147.67	-1.6%	4.93
270	148	150	151	149.67	-0.2%	3.46
300	140	143	151	144.67	-3.6%	3.06
330	160	128	156	148	-1.3%	3.46



ภาพที่ 4-14 แผนภูมิผลการทดสอบวัดมุมแบบหลบสิ่งกีดขวาง

บทที่ 5

สรุปปัญหาและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

การศึกษาโครงการนี้ เพื่อพัฒนาและสร้างหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง สามารถรับโหลดได้สูงสุด 3 กิโลกรัม ติดต่อกับ หุ่นยนต์ล้อเคลื่อนที่หลายทิศทางโดยผ่านทาง Wireless LAN และควบคุมหุ่นยนต์โดยใช้คอมพิวเตอร์ ที่ใช้ระบบปฏิบัติการ Microsoft Windows และโทรศัพท์ที่ใช้ระบบปฏิบัติการ Android และเนื่องจากในการทำงานของหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางเป็นการทำงานแบบอิสระและเป็นการควบคุมที่คลุมเครือดังนั้นจึงได้ใช้ระบบฟัซซี เข้ามาช่วยเพื่อควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางในโครงการนี้ได้ทำการควบคุมฟัซซีเป็นแบบ 3 Membership และได้ทำการคิดค้นรูปแบบการเคลื่อนที่กับสมการที่จะใช้ในการควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทางโดยแบ่งออกเป็นหลักใหญ่ทั้งหมด 12 รูปแบบ โดยแต่ละรูปแบบจะขึ้นอยู่กับพิกัดและระยะทางที่ต้องการให้หุ่นยนต์ล้อเคลื่อนที่หลายทิศทางเคลื่อนที่ไปยังตำแหน่งกำหนด

ผลที่ได้ จากการทดลองสามารถควบคุมให้หุ่นยนต์เคลื่อนที่หลายทิศทางนั้นสามารถเคลื่อนที่ไปยังตำแหน่งที่กำหนดและสามารถหลบสิ่งกีดขวางได้โดยมีความผิดพลาดจากระยะไม่เกิน 10% แต่ค่าความผิดพลาดจะมากหรือน้อยนั้นขึ้นอยู่กับโหลดที่กระทำกับหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง โดยที่ในสภาวะไม่มีโหลดนั้นค่าความผิดพลาดของระยะทางที่เกิดขึ้นจะไม่ค่อยเกาะกลุ่มกัน และในส่วนของโหลด 1 กิโลกรัมนั้นค่าความผิดพลาดของระยะทางที่เกิดขึ้นเริ่มที่จะเกาะกลุ่มกันแต่ก็ยังมีส่วนของโหลด 1 กิโลกรัมที่สูงกว่าในส่วนอื่นๆ และในส่วนของโหลด 2 กิโลกรัมนั้นค่าความผิดพลาดของระยะทางที่เกิดขึ้นจะอยู่ในระดับที่ใกล้เคียงกัน และในส่วนของโหลด 3 กิโลกรัมนั้นค่าความผิดพลาดของระยะทางที่เกิดขึ้นน้อยมากเมื่อเทียบกับโหลดที่ 0, 1 และ 2 กิโลกรัม และเมื่อเป็นรูปแบบของการหลบสิ่งกีดขวางก็ยังมีระยะความผิดพลาดในบางส่วนนั้นมากกว่าในส่วนอื่นๆและความผิดพลาดในส่วนองมุมนั้นมีความผิดพลาดที่น้อยมาก

5.2 ปัญหาและแนวทางในการแก้ไขปัญหา

ปัญหาที่เกิดขึ้นขณะทำการควบคุมให้หุ่นยนต์เคลื่อนที่หลายทิศทางมีดังนี้

5.2.1 ในการควบคุมทิศทางของหุ่นยนต์เคลื่อนที่หลายทิศทาง ไม่สามารถไปยังตำแหน่งและมุมที่ต้องการได้ ซึ่งจะแก้ไขโดยใช้เซนเซอร์ไจโรสโคป ช่วยในการควบคุมทิศทาง การเคลื่อนที่ของหุ่นยนต์เคลื่อนที่หลายทิศทาง

5.2.2 ในการใช้งานครั้งแรก การเชื่อมต่อไวร์เลส กับไมโครคอนโทรลเลอร์ ถ้าเปิดใช้งานพร้อมจะไม่สามารถเชื่อมต่อกันได้ ซึ่งจะแก้ไขโดย การรีเซ็ตตัวไมโครคอนโทรลเลอร์ หลังจากการจ่ายไฟให้ตัวหุ่นยนต์แล้ว

- 5.2.3 ไม่สามารถส่งค่าการทำงานของ หุ่นยนต์เคลื่อนที่หลายทิศทาง ย้อนกลับมายัง **Microsoft Windows** และโทรศัพท์ที่ใช้ระบบปฏิบัติการ **Android** แบบต่อเนื่องได้ ซึ่งแก้ไขได้ โดยการทำให้โปรแกรมสามารถแสดงค่าได้โดยการส่งค่าควบคุมในครั้งถัดไป

5.3 ข้อเสนอแนะ

- 5.3.1 เปลี่ยนเอ็นโค้ดเดอร์ ที่มีความละเอียดเพื่อที่หุ่นยนต์เคลื่อนที่หลายทิศทางจะได้มีความแม่นยำมากยิ่งขึ้น
- 5.3.2 เปลี่ยนล้อเคลื่อนที่หลายทิศทาง เนื่องจากล้อที่ใช้เป็นแบบพลาสติกแข็งจึงทำให้ใช้กับพื้นที่ได้จำกัด
- 5.3.3 ทำช่องสำหรับชาร์จแบตเตอรี่
- 5.3.4 นำเซนเซอร์ที่อยู่ในโทรศัพท์หรือแท็บเล็ต มาใช้ควบคุมหุ่นยนต์ด้วยมือ
- 5.3.5 ใช้จอยสติ๊ก (Joystick) ในการควบคุมหุ่นยนต์ด้วยมือ
- 5.3.6 ทำส่วนติดต่อกับผู้ใช้ (User Interface) สำหรับแท็บเล็ต
- 5.3.7 ทำให้หุ่นยนต์ส่งค่าการทำงานในตัวกลับมาได้

บรรณานุกรม

ภาษาไทย

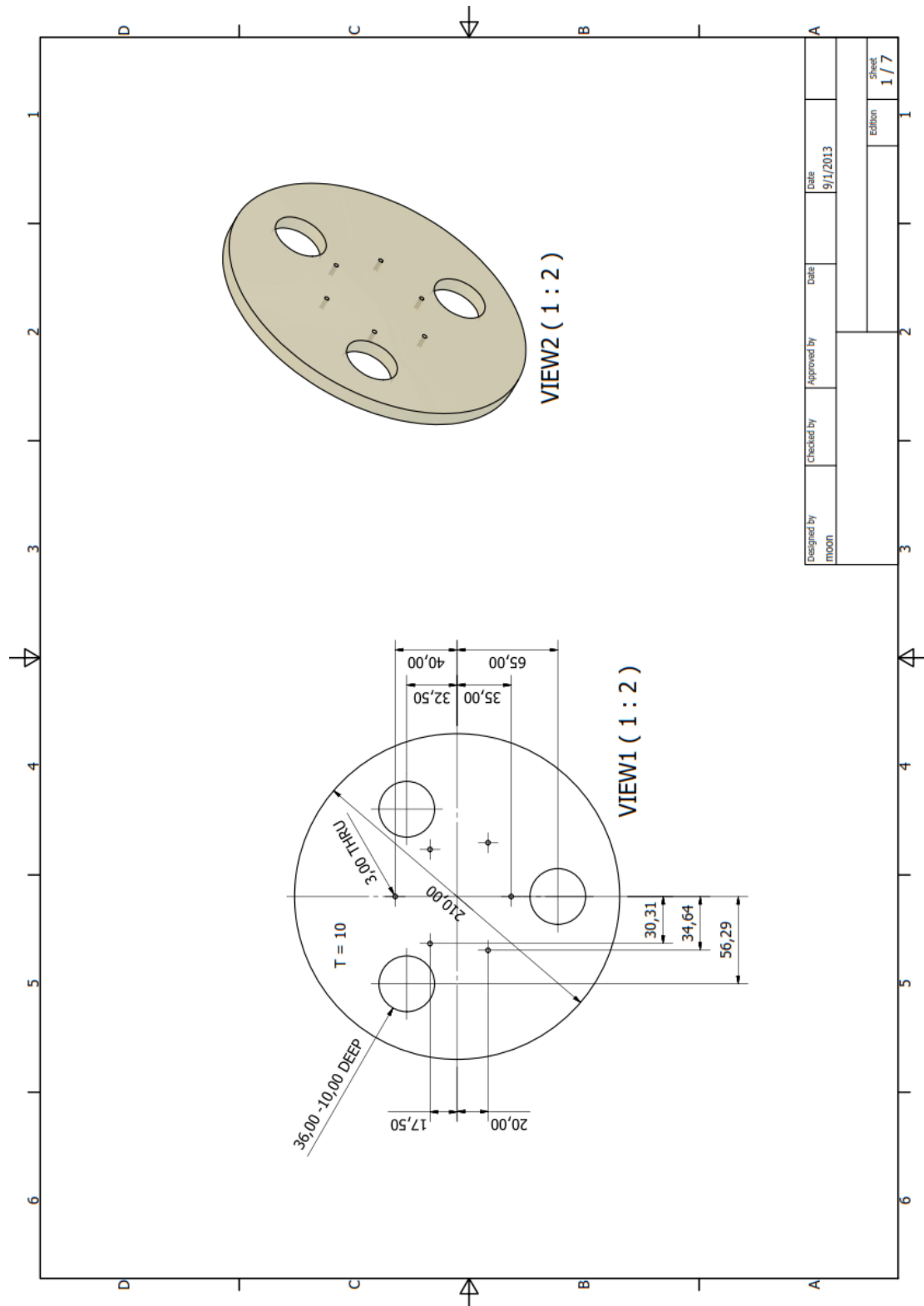
ประดิษฐ์ เหมือนคิด. วิชา 213314 การออกแบบการส่งกำลังทางกล. กรุงเทพฯ :
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2554
ศุภชัย สมพานิช. คู่มือการเขียนโปรแกรมและใช้งาน Visual C# .NET ฉบับสมบูรณ์.
นนทบุรี : อินโฟเเพรส, 2546

อินเทอร์เน็ต

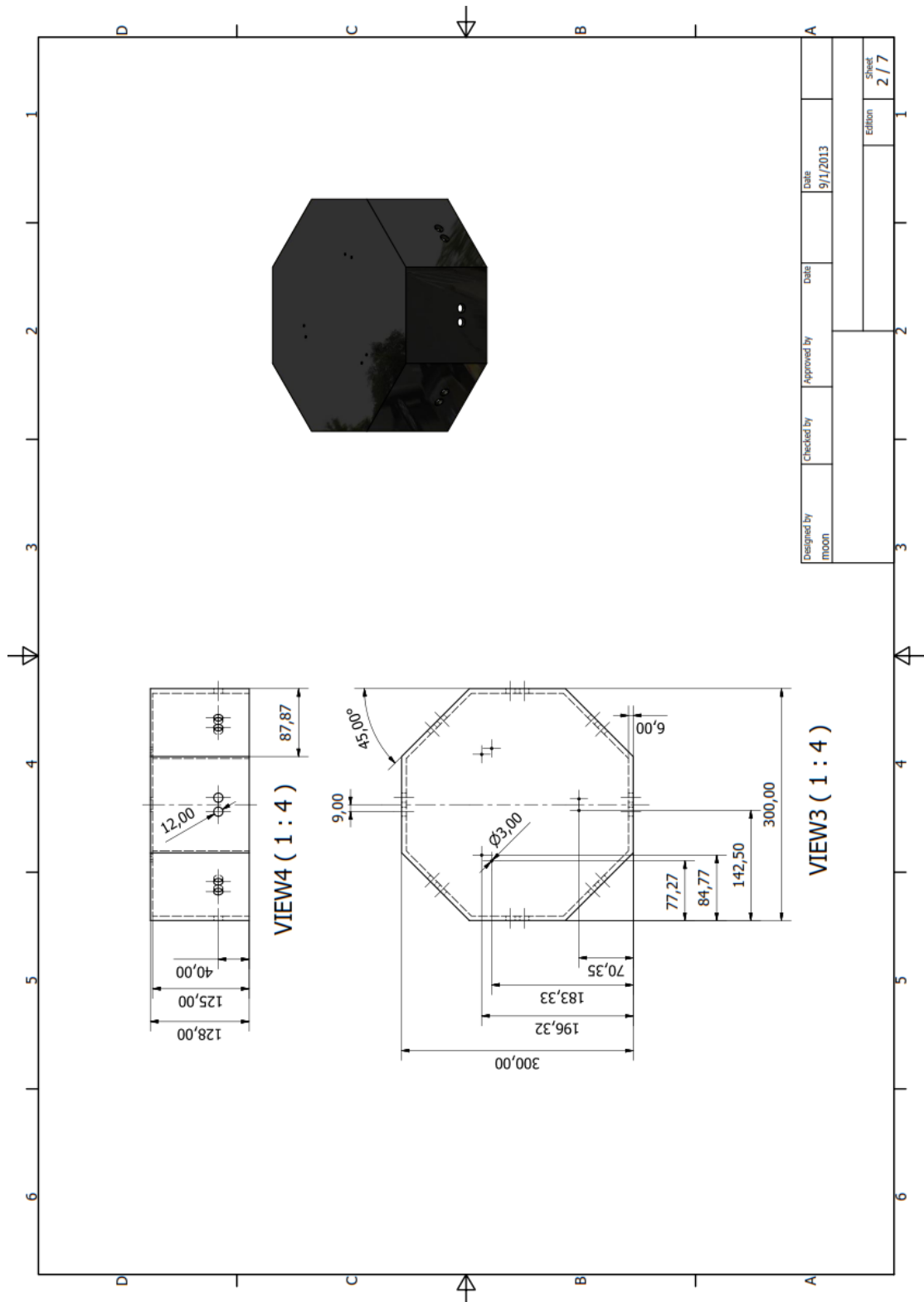
http://th.wikipedia.org/wiki/Integrated_Development_Environment
<http://th.wikipedia.org/wiki/แลนไร้สาย>
http://www.dol.go.th/it/index.php?option=com_content&task=view&id=115
<http://th.wikipedia.org/wiki/มือถือ>
http://en.wikipedia.org/wiki/Android_Studio
<http://th.wikipedia.org/wiki/ตรวจจับค่าสตรัคคลุมเครือ>
<http://bmesensor.blogspot.com/p/potentiometer-or-potentiometric.html>
<http://www.vcharkarn.com/uploads/252/252273.doc>
<http://arduino.cc/en/Main/ArduinoBoardUno>
<http://suanpalm3.kmitnb.ac.th/teacher/phayung/powerpoint.asp?pn=1>

ภาคผนวก

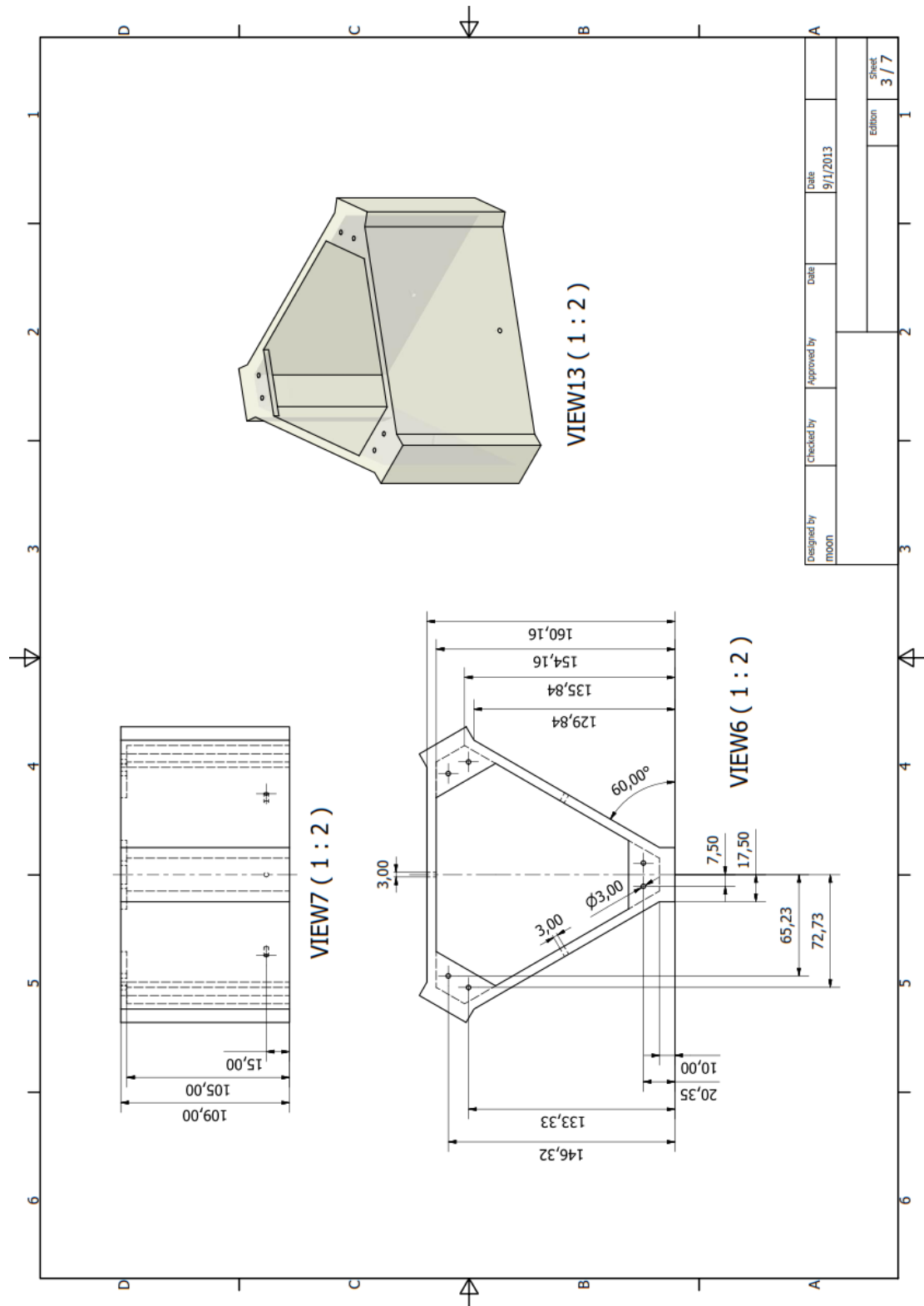
ภาคผนวก ก
ส่วนประกอบหุ่นยนต์



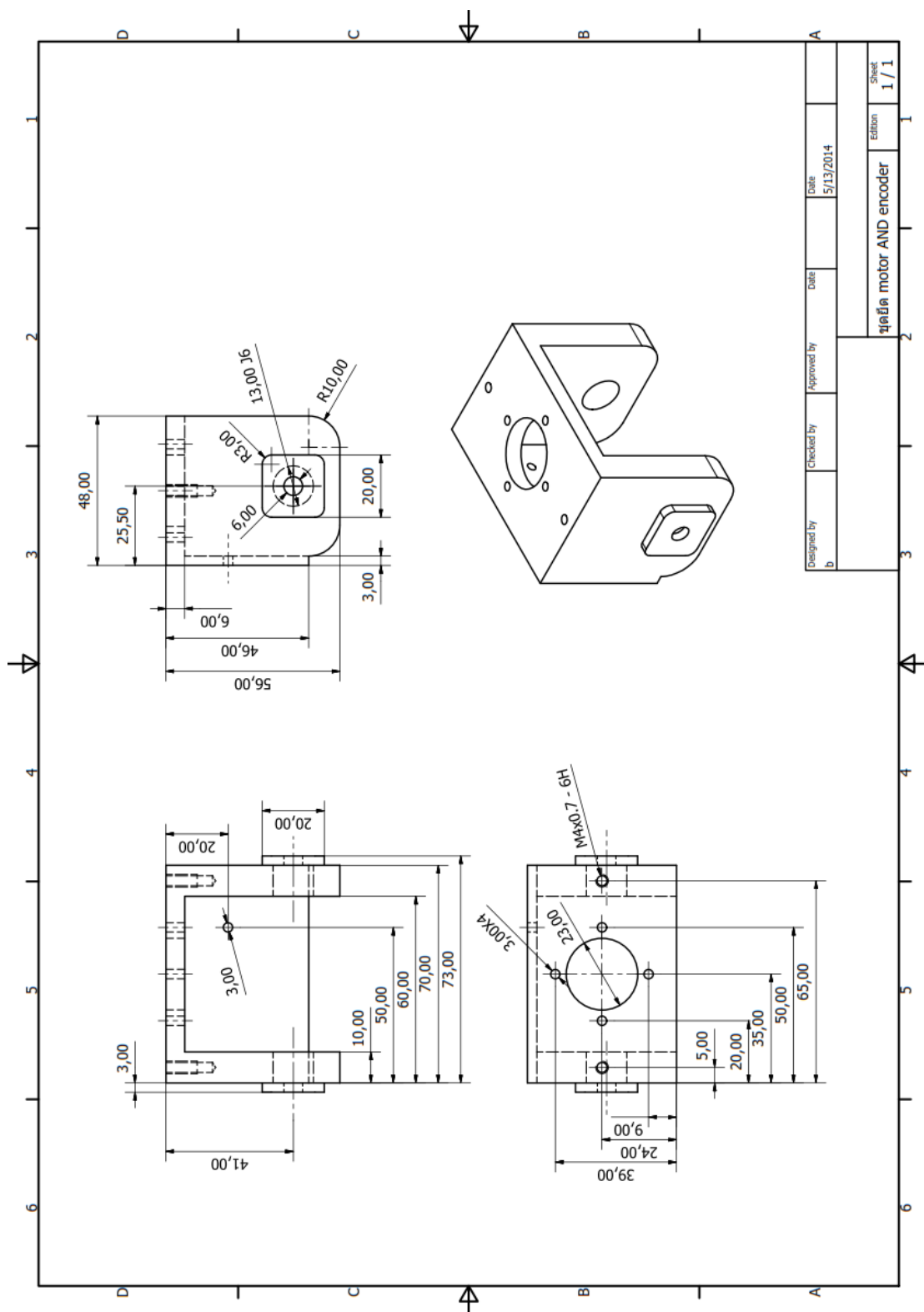
ภาพที่ ก-1 ฐานของหุ่นยนต์



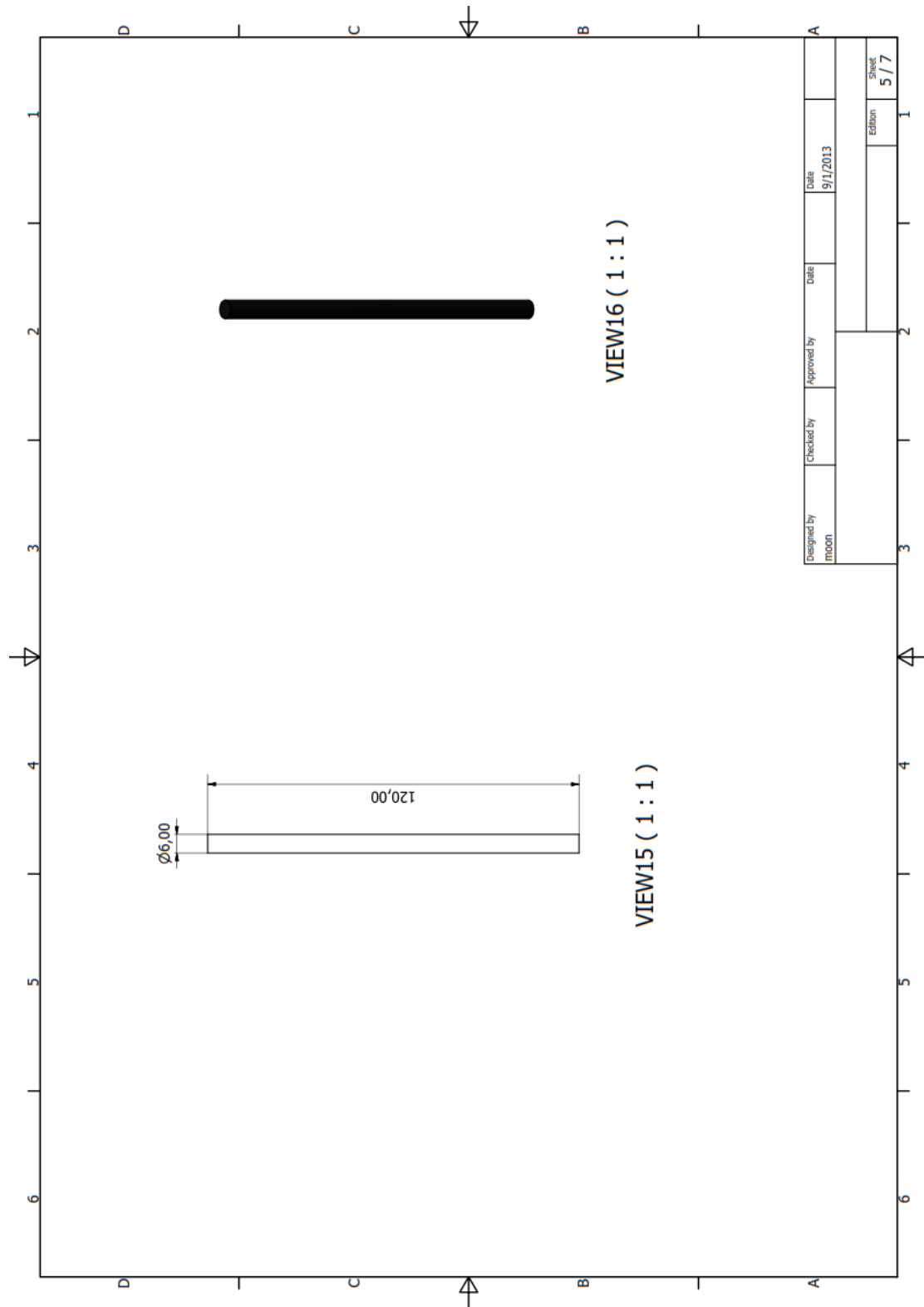
ภาพที่ ก-2 โครงข้างนอกของหุ่นยนต์



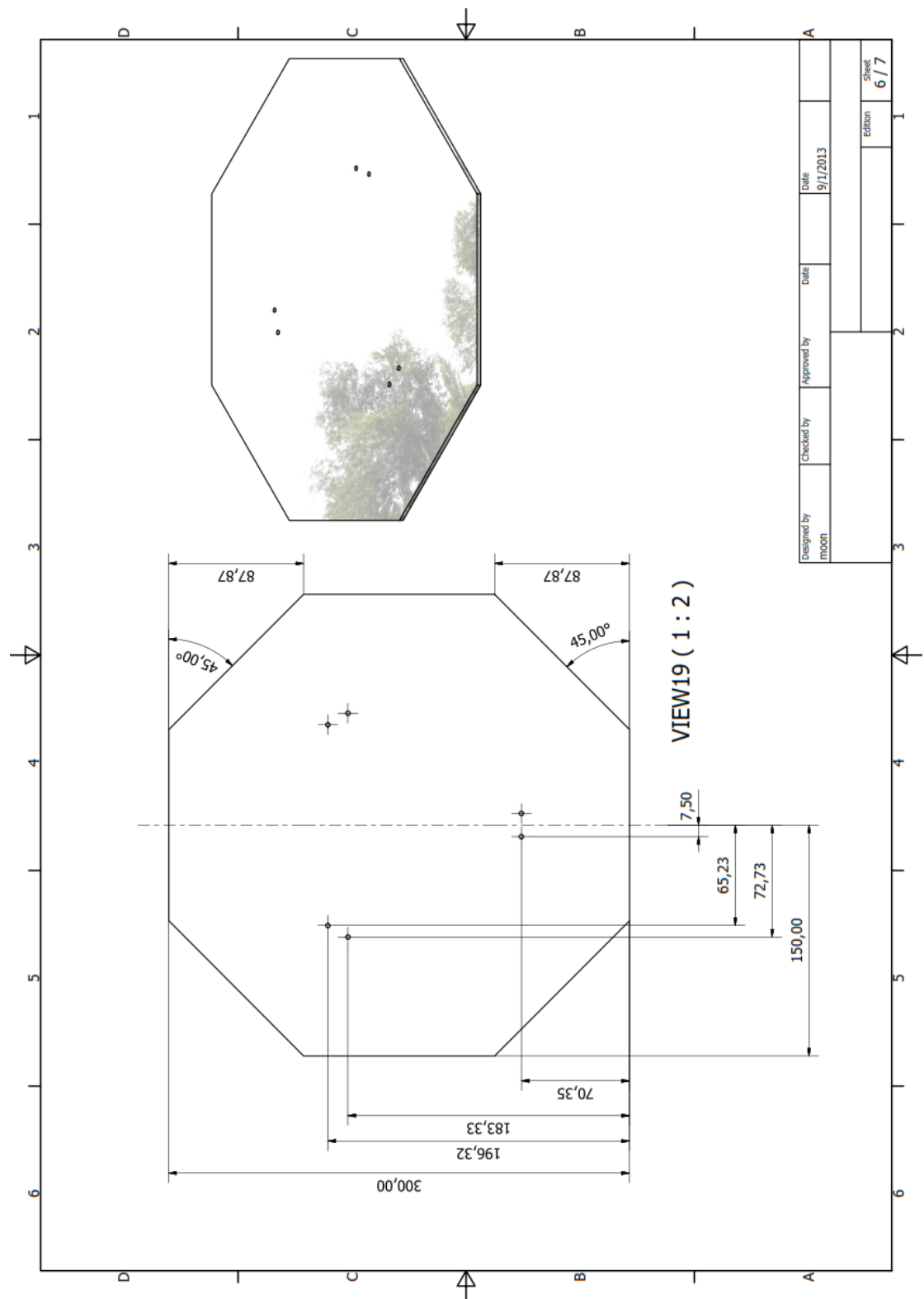
ภาพที่ ก-3 ชุดยึดวงจร



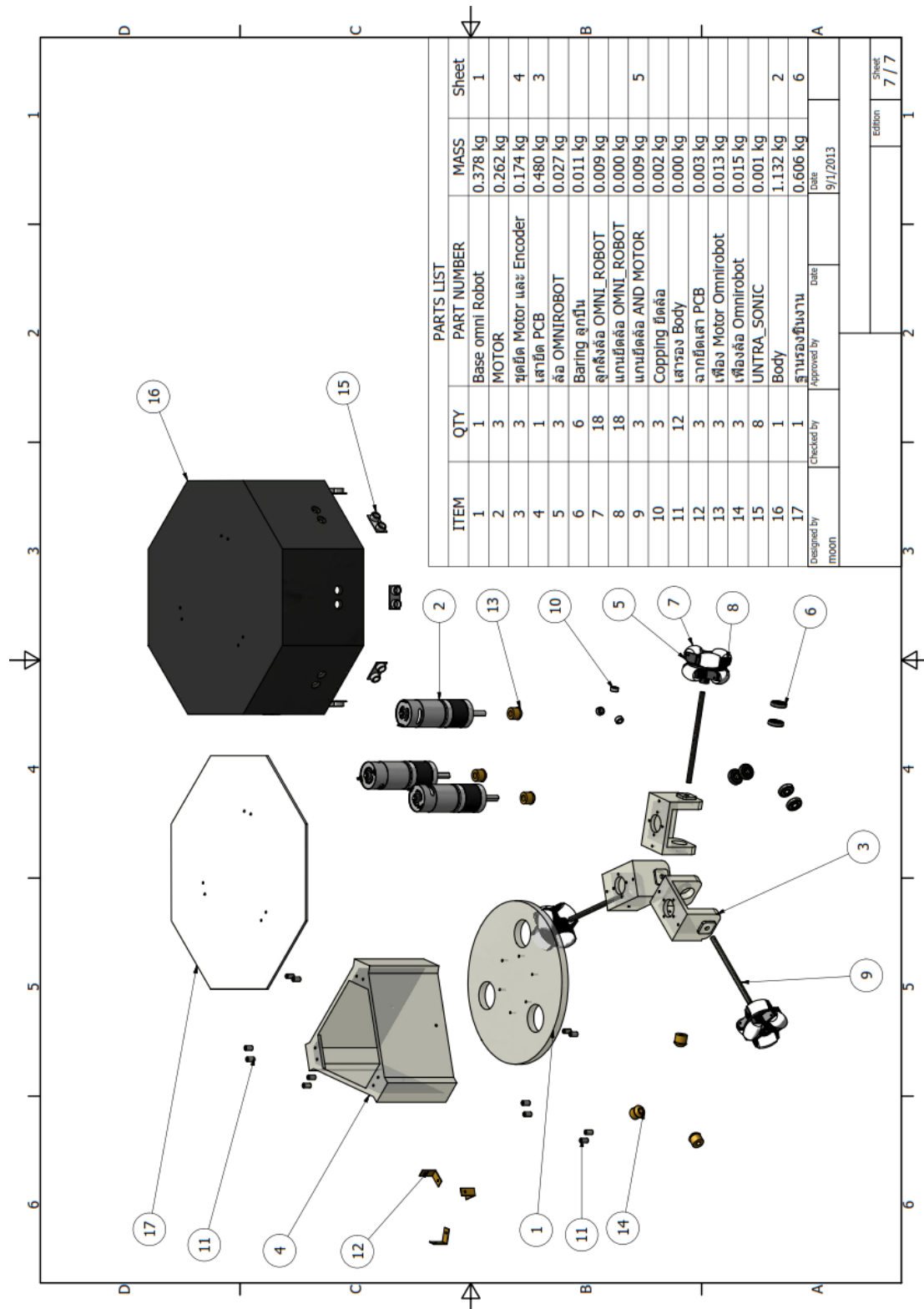
ภาพที่ ก-4 ชุดยึดล้อ



ภาพที่ ก-5 แกนล้อ



ภาพที่ ก-6 ฐานรองรับโหลด



ภาพที่ ก-7 ชิ้นส่วนทั้งหมดของหุ่นยนต์

ภาคผนวก ข

โปรแกรมไมโครคอนโทรลเลอร์

- ส่วนที่ติดต่อกับ **Ethernet Shield**

```
//+++++++ ดึง Library ที่จะใช้ติดต่อ Ethernet กับ I2C มาใช้งาน
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>

//+++++++ กำหนด IP ของ Microcontroller
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 0, 100 };
EthernetServer server(80);

//+++++++ เตรียมตัวแปรที่จะใช้รับส่งข้อมูล
String readString;
String dd;
int b_count = 0;
char write_from_readEthernet_1[42];
char write_from_readEthernet_2[42];
int countread = 0;

//+++++++ กำหนด Pin ที่จะใช้อ่านค่า Sensor
const int echoPin1 = 2;
const int echoPin2 = 3;
const int echoPin3 = 4;
const int echoPin4 = 5;
const int echoPin5 = 6;
const int echoPin6 = 7;
const int echoPin7 = 8;
const int echoPin8 = 9;
long duration1 = 0 , duration2 = 0 , duration3 = 0 , duration4 = 0 , duration5 = 0 , duration6 = 0 , duration7 = 0 , duration8 = 0;
long cm1 = 0 , cm2 = 0 , cm3 = 0 , cm4 = 0 , cm5 = 0 , cm6 = 0 , cm7 = 0 , cm8 = 0;
int i = 0;
int distane = 5;
int j;

//+++++++ เป็นส่วนในการกำหนดหน้าที่การทำงานของแต่ละ Pin
void setup()
{

//+++++++ ให้ Microcontroller เริ่มเชื่อมต่อ Ethernet
Ethernet.begin(mac);
server.begin();

//+++++++ กำหนด Baud Rate ของ RS232
Serial.begin(115200);
}

void loop()
{
//+++++++ อ่านข้อมูลที่ส่งผ่านมาจาก Ethernet
EthernetClient client = server.available();
```

```

if (client)
{
  boolean currentLineIsBlank = true;
  if (client.connected())
  {
    while (client.available())
    {
      char c = client.read();
      readString += c;
      countread++;
      //+++++++ นำข้อมูลที่ได้อ่านทั้งหมดส่งไปยัง RS 232 เพื่อใช้ในการควบคุม
      if (c == '\n')
      {
        client.println("HTTP/1.1 200 OK"); //send new page
        client.println("Content-Type: text/html");
        client.println();
        client.println("<HTML>");
        client.println("<HEAD>");
        client.println("<TITLE>Home Automation</TITLE>");
        client.println("</HEAD>");
        client.println("<BODY bgcolor='aqua'>");
        client.println("<H1>Home Automation</H1>");
        client.println("<hr />");
        client.println("<br />");
        client.println("</BODY>");
        client.println("</HTML>");
        readString.substring(5,47).toCharArray(write_from_readEthernet_1,42);
        Serial.println(write_from_readEthernet_1);
        readString="";
        client.stop();
      }
    }
  }
}

//+++++++ อ่านค่า Sensor
while(j<1)
{
  duration1 = pulseIn(echoPin1, HIGH);
  duration2 = pulseIn(echoPin2, HIGH);
  duration3 = pulseIn(echoPin3, HIGH);
  duration4 = pulseIn(echoPin4, HIGH);
  duration5 = pulseIn(echoPin5, HIGH);
  duration6 = pulseIn(echoPin6, HIGH);
  duration7 = pulseIn(echoPin7, HIGH);
  duration8 = pulseIn(echoPin8, HIGH);
  j++;
}
j = 0;
}

//+++++++ ส่งข้อมูลที่ได้จาก โดยผ่าน I2C
void requestEvent()

```

```

{
  if(duration1/3 <= 300)
  {
    Wire.write('a');
  }
  else if(duration3 <= 300)
  {
    Wire.write('c');
  }
  else if(duration5 <= 500)
  {
    Wire.write('e');
  }
  else if(duration7/2 <= 300)
  {
    Wire.write('g');
  }
  else if(duration8/3 <= 300)
  {
    Wire.write('h');
  }
  else if(duration4/2 <= 300)
  {
    Wire.write('d');
  }
  else if(duration6/2 <= 300)
  {
    Wire.write('f');
  }
  else if(duration2/3 <= 300)
  {
    Wire.write('b');
  }
  else
  {
  }
} {
  Wire.write('f'); // respond with message of 6 bytes
}
else if(duration2/3 <= 300)
{
  Wire.write('b'); // respond with message of 6 bytes
}
else
{
}

}
}

```

- ส่วนที่ติดต่อกับตัวหุ่นยนต์

```
//+++++ ดึง Library ที่จะใช้ควบคุมมอเตอร์กับ I2C มาใช้งาน

#include <Wire.h>
#include <Timer.h>
#include <Event.h>

//+++++ กำหนด Port เพื่ออ่านค่ามาใช้งานของ Gyroscope
#define CTRL_REG1 0x20
#define CTRL_REG2 0x21
#define CTRL_REG3 0x22
#define CTRL_REG4 0x23

//+++++ กำหนดขาที่จะใช้อ่าน Encoder
#define encoder0PinA 3
#define encoder0PinB 2

//+++++ กำหนด Address ของ Gyroscope และตัวแปรของ Gyroscope ที่ต้องใช้
int Addr = 105;
int x, y, z;
int g = 0, h = 0, j = 0, d = 0;
int selector = 0; // add

//+++++ ประกาศตัวแปร ในส่วนที่จะใช้ควบคุม หุ่นยนต์
int control[] = {4,13,6,7,8,9,10,11,12};
int i = 0, number_read = 0;
volatile int b_count = 0, l_count = 0, r_count = 0;
int loop_sensor = 0;
String summing_serial = "";
int cutting_serial = 0;
int mode, R_RUN = 0, L_RUN = 0, B_RUN = 0;
int MAX_R_RUN = 0, MAX_L_RUN = 0, MAX_B_RUN = 0;
int MIN_R_RUN = 0, MIN_L_RUN = 0, MIN_B_RUN = 0;
int RR_RUN = 0, LL_RUN = 0, BB_RUN = 0;
int step_control_first = 0, step_control_secound = 0;
int run_circule = 0;
int step_run = 0;

//+++++ กำหนด Pin ที่จะทำการส่งสัญญาณ ของเซ็นเซอร์ Ultrasonic
const int trigPin = 5;
int switch_Ultrasonic_on = 0;

//+++++ ประกาศตัวแปรที่ต้องใช้ในการ คำนวณของ Encoder
long B_Distance = 0, L_Distance = 0, R_Distance = 0;
long calculate_B_Distance = 0;
```

```

long calculate_L_Distance = 0;
long calculate_R_Distance = 0;
long l_in_loop_position = 0;
long b_in_loop_position = 0;
long r_in_loop_position = 0;
long l_out_1_loop_position = 0;
long b_out_1_loop_position = 0;
long r_out_1_loop_position = 0;
long l_out_2_loop_position = 0;
long b_out_2_loop_position = 0;
long r_out_2_loop_position = 0;
long l_out_3_loop_position = 0;
long b_out_3_loop_position = 0;
long r_out_3_loop_position = 0;
long l_out_4_loop_position = 0;
long b_out_4_loop_position = 0;
long r_out_4_loop_position = 0;
long l_out_5_loop_position = 0;
long b_out_5_loop_position = 0;
long r_out_5_loop_position = 0;
long l_out_6_loop_position = 0;
long b_out_6_loop_position = 0;
long r_out_6_loop_position = 0;
long l_out_7_loop_position = 0;
long b_out_7_loop_position = 0;
long r_out_7_loop_position = 0;
long l_out_8_loop_position = 0;
long b_out_8_loop_position = 0;
long r_out_8_loop_position = 0;

//+++++++ เป็นส่วนในการกำหนดหน้าที่การทำงาน
ของแต่ละ Pin

void setup()
{
  //+++++++ กำหนด Pin ที่จะใช้อ่าน Encoder
  pinMode(0, INPUT);
  pinMode(1, INPUT);
  pinMode(2, INPUT);

  //+++++++ ทำการกำหนดขา Output ใช้งาน
  for(i=0;i<=9;i++)
  {
    pinMode (control[i], OUTPUT);
  }

  //+++++++ ทำการประกาศตัวแปร Interrupt เพื่อใช้ในการอ่านค่า Encoder
  attachInterrupt (0, B_count, CHANGE);
  attachInterrupt (1, L_count, CHANGE);
  attachInterrupt (2, L_count, CHANGE);

  //+++++++ ทำการกำหนด Address ที่ใช้ในการอ่านค่า Gyroscope
  Wire.begin();

```

```

writeI2C(CTRL_REG1, 0x1F);
writeI2C(CTRL_REG3, 0x08);
writeI2C(CTRL_REG4, 0x80);

//+++++ กำหนด Baud Rate ของ RS232 และทำการเคลียค่าที่จะใช้อ่าน Encoder
Serial.begin(115200);
L_RUN = 0;
R_RUN = 0;
B_RUN = 0;
b_count = 0;
l_count = 0;
r_count = 0;
summing_serial = "";
pinMode(trigPin, OUTPUT);
}

void loop()
{
//+++++ ทำการกำหนดการส่งสัญญาณให้กับ Ultrasonic เพื่อใช้ในการตรวจสอบระยะที่
//ใช้ในการหลบลสิ่งกีดขวาง
analogWrite(trigPin,50);

//+++++ เรียกใช้ Function ย่อยของ Gyroscope
getGyroscopeValues();
g = g+((y/200));

//+++++ อ่านค่าข้อมูลที่ได้รับจากผู้ควบคุม
while(Serial.available())
{
int serial_read = Serial.read();
summing_serial += (char)serial_read;
if(serial_read == '/')
{
cutting_serial = 0;
summing_serial = "";
delay(50);
}
}

//+++++ ทำการถอดรหัสข้อมูลเพื่อนำมาใช้ในการควบคุมการทำงาน
selector = summing_serial.substring(0, 1).toInt();
switch_Ultrasonic_on = summing_serial.substring(1, 2).toInt();
mode = summing_serial.substring(2, 4).toInt();
R_RUN = summing_serial.substring(4, 7).toInt();
L_RUN = summing_serial.substring(7, 10).toInt();
B_RUN = summing_serial.substring(10, 13).toInt();
MAX_R_RUN = summing_serial.substring(13, 16).toInt();
MAX_L_RUN = summing_serial.substring(16, 19).toInt();
MAX_B_RUN = summing_serial.substring(19, 22).toInt();
MIN_R_RUN = summing_serial.substring(22, 25).toInt();
MIN_L_RUN = summing_serial.substring(25, 28).toInt();
MIN_B_RUN = summing_serial.substring(28, 31).toInt();

```

```

B_Distance = summing_serial.substring(31, 36).toInt();
L_Distance = summing_serial.substring(36, 41).toInt();
R_Distance = summing_serial.substring(41, 46).toInt();

//+++++++ เปิดใช้งานการควบคุมแบบ Manual
if(selector==0)
{
  switch(mode)
  {
    //+++++++ ทำการกำหนดให้เคล็อนที่ไปในมุม 90 องศา
    case 4:
      digitalWrite(13,HIGH);
      digitalWrite(7,LOW);
      digitalWrite(12,LOW);
      digitalWrite(4,LOW);
      digitalWrite(6,HIGH);
      digitalWrite(8,LOW);
      break;
    //+++++++ ทำการกำหนดให้เคล็อนที่ไปในมุม 270 องศา
    case 10:
      digitalWrite(13,LOW);
      digitalWrite(7,HIGH);
      digitalWrite(12,LOW);
      digitalWrite(4,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(8,LOW);
      break;
    //+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 30 องศา
    case 2:
      digitalWrite(13,HIGH);
      digitalWrite(7,LOW);
      digitalWrite(12,LOW);
      digitalWrite(4,LOW);
      digitalWrite(6,LOW);
      digitalWrite(8,HIGH);
      break;
    //+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 210 องศา
    case 8:
      digitalWrite(13,LOW);
      digitalWrite(7,LOW);
      digitalWrite(12,HIGH);
      digitalWrite(4,HIGH);
      digitalWrite(6,LOW);
      digitalWrite(8,LOW);
      break;
    //+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 150 องศา
    case 6:
      digitalWrite(13,LOW);
      digitalWrite(7,LOW);
      digitalWrite(12,HIGH);
      digitalWrite(4,LOW);
      digitalWrite(6,HIGH);

```

```

digitalWrite(8,LOW);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 330 องศา
case 12:
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 0 องศา
case 1:
digitalWrite(13,HIGH);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 180 องศา
case 7:
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);
digitalWrite(4,HIGH);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 60 องศา
case 3:
digitalWrite(13,HIGH);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,HIGH);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 240 องศา
case 9:
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,HIGH);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
break;
//+++++++ ทำการกำหนดให้เคล็อนที่ไปยังมุม 120 องศา
case 5:
digitalWrite(13,HIGH);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);

```

```

digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
break;
//+++++++ ทำการกำหนดให้เคลื่อนที่ไปยังมุม 300 องศา
case 11:
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
break;
}
//+++++++ ควบคุมความเร็วในแต่ละล้อ
analogWrite(9,R_RUN);
analogWrite(10,L_RUN);
analogWrite(11,B_RUN);
delay(250);
summing_serial = "";
}

//+++++++ เปิดใช้งาน Sensor Ultrasonic การควบคุมแบบ Auto
else if(selector==1)
{
if(switch_Ultrasonic_on == 1)
{
//+++++++ ทำการอ่านค่าแบบ I2C เพื่อใช้ในการหลบสิ่งกีดขวาง
Wire.requestFrom(2, 1);
if(Wire.available())
{
char c = Wire.read();

//+++++++ ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 90 องศา
if(mode == 4)
{
switch(c)
{

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 0 องศา
case('f'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_1_loop_position = l_out_1_loop_position + 1;
b_out_1_loop_position = b_out_1_loop_position + 1;

```

```

r_out_1_loop_position = r_out_1_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++++ เรียกค่า Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++++ ควบคุมทิศทางของล้อ
    digitalWrite(13,HIGH);
    digitalWrite(7,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,HIGH);
    //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
    if(g < 0)
    {
        LL_RUN = 130;
        RR_RUN = 130;
        BB_RUN = 255;
    }
    else if(g > 0)
    {
        LL_RUN = 80;
        RR_RUN = 80;
        BB_RUN = 255;
    }
    else
    {
        LL_RUN = 125;
        RR_RUN = 125;
        BB_RUN = 255;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;

```

```

r_count = 0;
b_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 0 องศา
case('e'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_1_loop_position = l_out_1_loop_position + 1;
b_out_1_loop_position = b_out_1_loop_position + 1;
r_out_1_loop_position = r_out_1_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่า Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ กำหนดทิศทางของล้อ
  digitalWrite(13,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,HIGH);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 130;
    RR_RUN = 130;
    BB_RUN = 255;
  }
  else if(g > 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 125;
    RR_RUN = 125;
    BB_RUN = 255;
  }
}

```

```

    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 180 องศา
case('g'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_5_loop_position = l_out_5_loop_position + 1;
b_out_5_loop_position = b_out_5_loop_position + 1;
r_out_5_loop_position = r_out_5_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    //chang mode 7
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++ เรียกค่าของ Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++ กำหนดทิศทางของล้อ
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(4,HIGH);
    digitalWrite(6,HIGH);
    digitalWrite(8,LOW);

```

```

//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 80;
  RR_RUN = 80;
  BB_RUN = 255;
}
else if(g > 0)
{
  LL_RUN = 130;
  RR_RUN = 130;
  BB_RUN = 255;
}
else
{
  LL_RUN = 125;
  RR_RUN = 125;
  BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;
}
}

//+++++++ ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 0 องศา
else if(mode == 1)
{
  switch(c)
  {

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 270 องศา
case('h'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);

```

```

digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_7_loop_position = l_out_7_loop_position + 1;
b_out_7_loop_position = b_out_7_loop_position + 1;
r_out_7_loop_position = r_out_7_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
    BB_RUN = 0;
  }
  else
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);

```

```

digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 270 องศา
case('g'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_7_loop_position = l_out_7_loop_position + 1;
b_out_7_loop_position = b_out_7_loop_position + 1;
r_out_7_loop_position = r_out_7_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++ เรียกค่า Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++ กำหนดทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  //+++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
  }
}

```

```

    BB_RUN = 0;
}
else
{
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 90 องศา
case('a'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_3_loop_position = l_out_3_loop_position + 1;
b_out_3_loop_position = b_out_3_loop_position + 1;
r_out_3_loop_position = r_out_3_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++++ เรียกค่า Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++++ ควบคุมทิศทางของล้อ

```

```

digitalWrite(13,HIGH);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 255;
  RR_RUN = 180;
  BB_RUN = 0;
}
else if(g > 0)
{
  LL_RUN = 255;
  RR_RUN = 230;
  BB_RUN = 0;
}
else
{
  LL_RUN = 255;
  RR_RUN = 230;
  BB_RUN = 0;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
break;
}
}

//+++++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 270 องศา)
else if(mode == 10)
{
  switch(c)
  {

    //+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 180 องศา

```

```

case('b'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_5_loop_position = l_out_5_loop_position + 1;
b_out_5_loop_position = b_out_5_loop_position + 1;
r_out_5_loop_position = r_out_5_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่า Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;
  }
  else if(g > 0)
  {
    LL_RUN = 130;
    RR_RUN = 130;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 125;
    RR_RUN = 125;
    BB_RUN = 255;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}

```

```

digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

```

```

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 180 องศา
case('a'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_5_loop_position = l_out_5_loop_position + 1;
b_out_5_loop_position = b_out_5_loop_position + 1;
r_out_5_loop_position = r_out_5_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++ เรียกค่า Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(8,LOW);
  //+++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;

```

```

    }
    else if(g > 0)
    {
        LL_RUN = 130;
        RR_RUN = 130;
        BB_RUN = 255;
    }
    else
    {
        LL_RUN = 125;
        RR_RUN = 125;
        BB_RUN = 255;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 0 องศา
case('c'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_1_loop_position = l_out_1_loop_position + 1;
b_out_1_loop_position = b_out_1_loop_position + 1;
r_out_1_loop_position = r_out_1_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;

```

```

r_in_loop_position = r_in_loop_position;
//+++++++ เรียกค่า Gyroscope มาใช้งาน
getGyroscopeValues();
g = g+((y/200));
//+++++++ ควบคุมทิศทางของล้อ
digitalWrite(13,HIGH);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 130;
  RR_RUN = 130;
  BB_RUN = 255;
}
else if(g > 0)
{
  LL_RUN = 80;
  RR_RUN = 80;
  BB_RUN = 255;
}
else
{
  LL_RUN = 125;
  RR_RUN = 125;
  BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;
}
}
//+++++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 180 องศา)

```

```

else if(mode == 7)
{
    switch(c)
    {

        //+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปทาง 90 องศา
        case('d'):
            digitalWrite(13,LOW);
            digitalWrite(7,LOW);
            digitalWrite(12,LOW);
            digitalWrite(4,LOW);
            digitalWrite(6,LOW);
            digitalWrite(8,LOW);
            analogWrite(9,0);
            analogWrite(10,0);
            analogWrite(11,0);
            l_out_3_loop_position = l_out_3_loop_position + 1;
            b_out_3_loop_position = b_out_3_loop_position + 1;
            r_out_3_loop_position = r_out_3_loop_position + 1;
            delay(500);
            g = 0;
            for(int o=0;o<=350;o++)
            {
                l_in_loop_position = l_in_loop_position;
                b_in_loop_position = b_in_loop_position;
                //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
                getGyroscopeValues();
                g = g+((y/200));
                //+++++++ ควบคุมทิศทางของล้อ
                digitalWrite(13,HIGH);
                digitalWrite(7,LOW);
                digitalWrite(12,LOW);
                digitalWrite(4,LOW);
                digitalWrite(6,HIGH);
                digitalWrite(8,LOW);
                //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
                if(g < 0)
                {
                    LL_RUN = 255;
                    RR_RUN = 180;
                    BB_RUN = 0;
                }
                else if(g > 0)
                {
                    LL_RUN = 255;
                    RR_RUN = 230;
                    BB_RUN = 0;
                }
                else
                {
                    LL_RUN = 255;
                    RR_RUN = 230;
                    BB_RUN = 0;
                }
            }
        }
    }
}

```

```

    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 90 องศา
case('c'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_3_loop_position = l_out_3_loop_position + 1;
b_out_3_loop_position = b_out_3_loop_position + 1;
r_out_3_loop_position = r_out_3_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++++ กำหนดทิศทางของล้อ
    digitalWrite(13,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(8,LOW);
    //+++++++ ควบคุมให้ไปตามองศาที่กำหนด

```

```

if(g < 0)
{
  LL_RUN = 255;
  RR_RUN = 180;
  BB_RUN = 0;
}
else if(g > 0)
{
  LL_RUN = 255;
  RR_RUN = 230;
  BB_RUN = 0;
}
else
{
  LL_RUN = 255;
  RR_RUN = 230;
  BB_RUN = 0;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 270 องศา
case('e'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_7_loop_position = l_out_7_loop_position + 1;
b_out_7_loop_position = b_out_7_loop_position + 1;
r_out_7_loop_position = r_out_7_loop_position + 1;
delay(500);

```

```

g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ กำหนดทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
    BB_RUN = 0;
  }
  else
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;

```

```

    break;
}
}

//+++++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 330 องศา และ 300 องศา)
else if((mode == 12)|| (mode == 11))
{
    switch(c)
    {

        //+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 210 องศา
        case('a'):
            digitalWrite(13,LOW);
            digitalWrite(7,LOW);
            digitalWrite(12,LOW);
            digitalWrite(4,LOW);
            digitalWrite(6,LOW);
            digitalWrite(8,LOW);
            analogWrite(9,0);
            analogWrite(10,0);
            analogWrite(11,0);
            l_out_6_loop_position = l_out_6_loop_position + 1;
            b_out_6_loop_position = b_out_6_loop_position + 1;
            r_out_6_loop_position = r_out_6_loop_position + 1;
            delay(500);
            g = 0;
            for(int o=0;o<=420;o++)
            {
                l_in_loop_position = l_in_loop_position;
                b_in_loop_position = b_in_loop_position;
                r_in_loop_position = r_in_loop_position;
                //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
                getGyroscopeValues();
                g = g+((y/200));
                //+++++++ กำหนดทิศทางของล้อ
                digitalWrite(13,LOW);
                digitalWrite(7,LOW);
                digitalWrite(12,HIGH);
                digitalWrite(4,HIGH);
                digitalWrite(6,LOW);
                digitalWrite(8,LOW);
                //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
                if(g < 0)
                {
                    LL_RUN = 230;
                    RR_RUN = 0;
                    BB_RUN = 255;
                }
                else if(g > 0)
                {
                    LL_RUN = 230;
                    RR_RUN = 0;

```

```

    BB_RUN = 240;
}
else
{
    LL_RUN = 230;
    RR_RUN = 0;
    BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 0 องศา
case('b'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_1_loop_position = l_out_1_loop_position + 1;
b_out_1_loop_position = b_out_1_loop_position + 1;
r_out_1_loop_position = r_out_1_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=420;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++++ ควบคุมทิศทางของล้อ

```

```

digitalWrite(13,HIGH);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 130;
  RR_RUN = 130;
  BB_RUN = 255;
}
else if(g > 0)
{
  LL_RUN = 80;
  RR_RUN = 80;
  BB_RUN = 255;
}
else
{
  LL_RUN = 125;
  RR_RUN = 125;
  BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 270 องศา
case('h'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);

```

```

analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_7_loop_position = l_out_7_loop_position + 1;
b_out_7_loop_position = b_out_7_loop_position + 1;
r_out_7_loop_position = r_out_7_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=420;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
    BB_RUN = 0;
  }
  else
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);

```

```

    analogWrite(10,0);
    analogWrite(11,0);
    delay(500);
    g = 0;
    l_count = 0;
    b_count = 0;
    r_count = 0;
    break;
}
}

//+++++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 210 องศา และ 240 องศา)
else if((mode == 8)|| (mode == 9))
{
    switch(c)
    {
        //+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 150 องศา
        case('c'):
            digitalWrite(13,LOW);
            digitalWrite(7,LOW);
            digitalWrite(12,LOW);
            digitalWrite(4,LOW);
            digitalWrite(6,LOW);
            digitalWrite(8,LOW);
            analogWrite(9,0);
            analogWrite(10,0);
            analogWrite(11,0);
            l_out_4_loop_position = l_out_4_loop_position + 1;
            b_out_4_loop_position = b_out_4_loop_position + 1;
            r_out_4_loop_position = r_out_4_loop_position + 1;
            delay(500);
            g = 0;
            for(int o=0;o<=350;o++)
            {
                l_in_loop_position = l_in_loop_position;
                b_in_loop_position = b_in_loop_position;
                r_in_loop_position = r_in_loop_position;
                //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
                getGyroscopeValues();
                g = g+((y/200));
                //+++++++ ควบคุมทิศทางของล้อ
                digitalWrite(13,LOW);
                digitalWrite(7,LOW);
                digitalWrite(12,HIGH);
                digitalWrite(4,LOW);
                digitalWrite(6,HIGH);
                digitalWrite(8,LOW);
                //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
                if(g < 0)
                {
                    LL_RUN = 0;
                    RR_RUN = 230;

```

```

    BB_RUN = 255;
}
else if(g > 0)
{
    LL_RUN = 0;
    RR_RUN = 255;
    BB_RUN = 250;
}
else
{
    LL_RUN = 0;
    RR_RUN = 255;
    BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 180 องศา
case('b'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_5_loop_position = l_out_5_loop_position + 1;
b_out_5_loop_position = b_out_5_loop_position + 1;
r_out_5_loop_position = r_out_5_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;

```

```

b_in_loop_position = b_in_loop_position;
r_in_loop_position = r_in_loop_position;
//+++++++ เรียกค่าของ Gyroscope มาใช้งาน
getGyroscopeValues();
g = g+((y/200));
//+++++++ ควบคุมทิศทางของล้อ
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);
digitalWrite(4,HIGH);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 80;
  RR_RUN = 80;
  BB_RUN = 255;
}
else if(g > 0)
{
  LL_RUN = 130;
  RR_RUN = 130;
  BB_RUN = 255;
}
else
{
  LL_RUN = 125;
  RR_RUN = 125;
  BB_RUN = 255;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 270 องศา
case('d'):

```

```

digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_7_loop_position = l_out_7_loop_position + 1;
b_out_7_loop_position = b_out_7_loop_position + 1;
r_out_7_loop_position = r_out_7_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
    BB_RUN = 0;
  }
  else
  {
    LL_RUN = 255;
    RR_RUN = 255;
    BB_RUN = 0;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);

```

```

digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;
}
}

//+++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 120 องศา และ 150 องศา)
else if((mode == 5)|| (mode == 6))
{
  switch(c)
  {

    //+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 60 องศา
    case('e'):
      digitalWrite(13,LOW);
      digitalWrite(7,LOW);
      digitalWrite(12,LOW);
      digitalWrite(4,LOW);
      digitalWrite(6,LOW);
      digitalWrite(8,LOW);
      analogWrite(9,0);
      analogWrite(10,0);
      analogWrite(11,0);
      l_out_2_loop_position = l_out_2_loop_position + 1;
      b_out_2_loop_position = b_out_2_loop_position + 1;
      r_out_2_loop_position = r_out_2_loop_position + 1;
      delay(500);
      g = 0;
      for(int o=0;o<=350;o++)
      {
        l_in_loop_position = l_in_loop_position;
        b_in_loop_position = b_in_loop_position;
        r_in_loop_position = r_in_loop_position;
        //+++++ เรียกค่าของ Gyroscope มาใช้งาน
        getGyroscopeValues();
        g = g+((y/200));
        //+++++ ควบคุมทิศทางของล้อ
        digitalWrite(13,HIGH);
        digitalWrite(7,LOW);
        digitalWrite(12,LOW);
        digitalWrite(4,LOW);
        digitalWrite(6,HIGH);

```

```

digitalWrite(8,HIGH);
//+++++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 255;
  RR_RUN = 113;
  BB_RUN = 83;
}
else if(g > 0)
{
  LL_RUN = 255;
  RR_RUN = 126;
  BB_RUN = 88;
}
else
{
  LL_RUN = 255;
  RR_RUN = 126;
  BB_RUN = 88;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 90 องศา
case('d'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_3_loop_position = l_out_3_loop_position + 1;
b_out_3_loop_position = b_out_3_loop_position + 1;

```

```

r_out_3_loop_position = r_out_3_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
    l_in_loop_position = l_in_loop_position;
    b_in_loop_position = b_in_loop_position;
    r_in_loop_position = r_in_loop_position;
    //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
    getGyroscopeValues();
    g = g+((y/200));
    //+++++++ ควบคุมทิศทางของล้อ
    digitalWrite(13,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(8,LOW);
    //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
    if(g < 0)
    {
        LL_RUN = 255;
        RR_RUN = 180;
        BB_RUN = 0;
    }
    else if(g > 0)
    {
        LL_RUN = 255;
        RR_RUN = 230;
        BB_RUN = 0;
    }
    else
    {
        LL_RUN = 255;
        RR_RUN = 230;
        BB_RUN = 0;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;

```

```

b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 180 องศา
case('f'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_5_loop_position = l_out_5_loop_position + 1;
b_out_5_loop_position = b_out_5_loop_position + 1;
r_out_5_loop_position = r_out_5_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;
  }
  else if(g > 0)
  {
    LL_RUN = 130;
    RR_RUN = 130;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 125;
    RR_RUN = 125;
    BB_RUN = 255;
  }
}

```

```

    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
  }
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  analogWrite(9,0);
  analogWrite(10,0);
  analogWrite(11,0);
  delay(500);
  g = 0;
  l_count = 0;
  b_count = 0;
  r_count = 0;
  break;
}
}

//+++++ (ทำการตรวจสอบสิ่งกีดขวางในการเคลื่อนที่มุม 30 องศา และ 60 องศา)
else if((mode == 2)|| (mode == 3))
{
  switch(c)
  {

    //+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปทาง 300 องศา
    case('g'):
      digitalWrite(13,LOW);
      digitalWrite(7,LOW);
      digitalWrite(12,LOW);
      digitalWrite(4,LOW);
      digitalWrite(6,LOW);
      digitalWrite(8,LOW);
      analogWrite(9,0);
      analogWrite(10,0);
      analogWrite(11,0);
      l_out_8_loop_position = l_out_8_loop_position + 1;
      b_out_8_loop_position = b_out_8_loop_position + 1;
      r_out_8_loop_position = r_out_8_loop_position + 1;
      delay(500);
      g = 0;
      for(int o=0;o<=350;o++)
      {
        l_in_loop_position = l_in_loop_position;
        b_in_loop_position = b_in_loop_position;
        r_in_loop_position = r_in_loop_position;
        //+++++ เรียกค่าของ Gyroscope มาใช้งาน
        getGyroscopeValues();
        g = g+((y/200));
      }
    }
  }
}

```

```

//+++++ ควบคุมทิศทางของล้อ
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
//+++++ ควบคุมให้ไปตามองศาที่กำหนด
if(g < 0)
{
  LL_RUN = 75;
  RR_RUN = 255;
  BB_RUN = 108;
}
else if(g > 0)
{
  LL_RUN = 102;
  RR_RUN = 200;
  BB_RUN = 144;
}
else
{
  LL_RUN = 100;
  RR_RUN = 246;
  BB_RUN = 143;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 0 องศา
case('f'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);

```

```

digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_1_loop_position = l_out_1_loop_position + 1;
b_out_1_loop_position = b_out_1_loop_position + 1;
r_out_1_loop_position = r_out_1_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,HIGH);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 130;
    RR_RUN = 130;
    BB_RUN = 255;
  }
  else if(g > 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 125;
    RR_RUN = 125;
    BB_RUN = 255;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);

```

```

analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;

//+++++++ เปลี่ยนรูปแบบการควบคุมให้เคลื่อนที่ไปในทาง 90 องศา
case('h'):
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
l_out_3_loop_position = l_out_3_loop_position + 1;
b_out_3_loop_position = b_out_3_loop_position + 1;
r_out_3_loop_position = r_out_3_loop_position + 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  l_in_loop_position = l_in_loop_position;
  b_in_loop_position = b_in_loop_position;
  r_in_loop_position = r_in_loop_position;
  //+++++++ เรียกค่าของ Gyroscope มาใช้งาน
  getGyroscopeValues();
  g = g+((y/200));
  //+++++++ ควบคุมทิศทางของล้อ
  digitalWrite(13,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(8,LOW);
  //+++++++ ควบคุมให้ไปตามองศาที่กำหนด
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 180;
    BB_RUN = 0;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 230;
    BB_RUN = 0;
  }
}

```

```

    }
    else
    {
        LL_RUN = 255;
        RR_RUN = 230;
        BB_RUN = 0;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
analogWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
delay(500);
g = 0;
l_count = 0;
b_count = 0;
r_count = 0;
break;
}
}
else
{
}
}
}
else
{
}

//+++++ RUN Robot Control (การควบคุมหลักในการควบคุมหุ่นยนต์)
switch(mode)
{
    //+++++ การควบคุมการทำงานในมุม 90 องศา
    case(4):
        calculate_L_Distance = L_Distance;
        calculate_B_Distance = B_Distance;
        calculate_R_Distance = R_Distance;
        l_in_loop_position = l_in_loop_position + 1;
        b_in_loop_position = b_in_loop_position + 1;
        r_in_loop_position = r_in_loop_position + 1;
        l_count = 0;
        b_count = 0;
        r_count = 0;
        digitalWrite(13,HIGH);
        digitalWrite(7,LOW);

```

```

digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
//+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN,L_RUN,0,0,6);
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
else if(g > 0)
{
  LL_RUN = L_RUN;
  RR_RUN = map(RR_RUN,R_RUN,0,0,6);
  BB_RUN = B_RUN;
}
else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++ ควบคุมระยะทาง
if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++ การควบคุมการทำงานในมุม 270 องศา
case(10):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;

```

```

b_count = 0;
r_count = 0;
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
//+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN, L_RUN ,0,0,6);
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
else if(g > 0)
{
  LL_RUN = L_RUN;
  RR_RUN = map(RR_RUN, R_RUN ,0,0,6);
  BB_RUN = B_RUN;
}
else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++ ควบคุมระยะทาง
if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++ การควบคุมการทำงานในมุม 60 องศา
case(2):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;

```

```

l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,HIGH);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
//+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN, L_RUN ,0,0,6);
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
else if(g > 0)
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = map(BB_RUN, B_RUN ,0,0,6);
}
else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++++ ควบคุมระยะทาง
if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++++ การควบคุมการทำงานในมุม 300 องศา

```

```

case(8):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
//+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
LL_RUN = map(LL_RUN, L_RUN ,0,0,6);
RR_RUN = R_RUN;
BB_RUN = B_RUN;
}
else if(g > 0)
{
LL_RUN = L_RUN;
RR_RUN = R_RUN;
BB_RUN = map(BB_RUN, B_RUN ,0,0,6);
}
else
{
LL_RUN = L_RUN;
RR_RUN = R_RUN;
BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++++ ควบคุมระยะทาง
if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
{
Serial.println(b_in_loop_position);
Serial.println(l_in_loop_position);
summing_serial = "";
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = 0;
b_in_loop_position = 0;
r_in_loop_position = 0;
delay(30);

```

```

}
break;

//+++++++ การควบคุมการทำงานในมุม 150 องศา
case(6):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
//+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
LL_RUN = L_RUN;
RR_RUN = R_RUN;
BB_RUN = map(BB_RUN, B_RUN ,0 ,0,6);
}
else if(g > 0)
{
LL_RUN = L_RUN;
RR_RUN = map(RR_RUN, R_RUN ,0 ,0,6);
BB_RUN = B_RUN;
}
else
{
LL_RUN = L_RUN;
RR_RUN = R_RUN;
BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++++ ควบคุมระยะทาง
if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
{
Serial.println(b_in_loop_position);
Serial.println(l_in_loop_position);
summing_serial = "";
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;

```

```

    l_in_loop_position = 0;
    b_in_loop_position = 0;
    r_in_loop_position = 0;
    delay(30);
}
break;
//+++++ การควบคุมการทำงานในมุม 330 องศา
case(12):
    calculate_L_Distance = L_Distance;
    calculate_B_Distance = B_Distance;
    calculate_R_Distance = R_Distance;
    l_in_loop_position = l_in_loop_position + 1;
    b_in_loop_position = b_in_loop_position + 1;
    r_in_loop_position = r_in_loop_position + 1;
    l_count = 0;
    b_count = 0;
    r_count = 0;
    digitalWrite(13,LOW);
    digitalWrite(7,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,HIGH);
    //+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
    if(g < 0)
    {
        LL_RUN = L_RUN;
        RR_RUN = map(RR_RUN, R_RUN ,0 ,0,6);
        BB_RUN = B_RUN;
    }
    else if(g > 0)
    {
        LL_RUN = L_RUN;
        RR_RUN = R_RUN;
        BB_RUN = map(BB_RUN, B_RUN ,0 ,0,6);
    }
    else
    {
        LL_RUN = L_RUN;
        RR_RUN = R_RUN;
        BB_RUN = B_RUN;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
    //+++++ ควบคุมระยะทาง
    if((calculate_L_Distance < l_in_loop_position)|| (calculate_B_Distance <
b_in_loop_position))
    {
        Serial.println(b_in_loop_position);
        Serial.println(l_in_loop_position);
        summing_serial = "";

```

```

    calculate_L_Distance = L_Distance;
    calculate_B_Distance = B_Distance;
    calculate_R_Distance = R_Distance;
    l_in_loop_position = 0;
    b_in_loop_position = 0;
    r_in_loop_position = 0;
    delay(30);
}
break;

//+++++ การควบคุมการทำงานในมุม 0 องศา
case(1):
    calculate_L_Distance = L_Distance;
    calculate_B_Distance = B_Distance;
    calculate_R_Distance = R_Distance;
    l_in_loop_position = l_in_loop_position + 1;
    b_in_loop_position = b_in_loop_position + 1;
    r_in_loop_position = r_in_loop_position + 1;
    l_count = 0;
    b_count = 0;
    r_count = 0;
    digitalWrite(13,HIGH);
    digitalWrite(7,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,HIGH);
//+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
    if(g < 0)
    {
        LL_RUN = map(LL_RUN, MAX_L_RUN ,0 ,0,6);
        RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
        BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
    }
    else if(g > 0)
    {
        LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
        RR_RUN = map(RR_RUN, MIN_R_RUN ,255 ,0,6);
        BB_RUN = map(BB_RUN, MAX_B_RUN ,0 ,0,6);
    }
    else
    {
        LL_RUN = L_RUN;
        RR_RUN = R_RUN;
        BB_RUN = B_RUN;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
//+++++ ควบคุมระยะทาง
    if((calculate_L_Distance <= l_in_loop_position)||
    (calculate_B_Distance <= b_in_loop_position))

```

```

{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++++ การควบคุมการทำงานในมุม 180 องศา
case(7):
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = l_in_loop_position + 1;
  b_in_loop_position = b_in_loop_position + 1;
  r_in_loop_position = r_in_loop_position + 1;
  l_count = 0;
  b_count = 0;
  r_count = 0;
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(8,LOW);
  //+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
  if(g < 0)
  {
    LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
    RR_RUN = map(RR_RUN, MIN_R_RUN ,255 ,0,6);
    BB_RUN = map(BB_RUN, MAX_B_RUN ,0 ,0,6);
  }
  else if(g > 0)
  {
    LL_RUN = map(LL_RUN, MAX_L_RUN ,0 ,0,6);
    RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
    BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
  }
  else
  {
    LL_RUN = L_RUN;
    RR_RUN = R_RUN;
    BB_RUN = B_RUN;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);

```

```

analogWrite(11,BB_RUN);
//+++++ ควบคุมระยะทาง
if((calculate_L_Distance <= l_in_loop_position)|| (calculate_B_Distance <=
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++ การควบคุมการทำงานในมุม 60 องศา
case(3):
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = l_in_loop_position + 1;
  b_in_loop_position = b_in_loop_position + 1;
  r_in_loop_position = r_in_loop_position + 1;
  l_count = 0;
  b_count = 0;
  r_count = 0;
  digitalWrite(13,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,HIGH);
  digitalWrite(8,HIGH);
//+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN, MAX_L_RUN ,0 ,0,6);
  RR_RUN = map(RR_RUN, MIN_R_RUN ,255 ,0,6);
  BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
}
else if(g > 0)
{
  LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
  RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
  BB_RUN = map(BB_RUN, MAX_B_RUN ,0 ,0,6);
}
else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;

```

```

    BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++ ควบคุมระยะทาง
if((calculate_L_Distance <= l_in_loop_position)|| (calculate_B_Distance <=
b_in_loop_position))
{
    Serial.println(b_in_loop_position);
    Serial.println(l_in_loop_position);
    summing_serial = "";
    calculate_L_Distance = L_Distance;
    calculate_B_Distance = B_Distance;
    calculate_R_Distance = R_Distance;
    l_in_loop_position = 0;
    b_in_loop_position = 0;
    r_in_loop_position = 0;
    delay(30);
}
break;

//+++++ การควบคุมการทำงานในมุม 240 องศา
case(9):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,HIGH);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
//+++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
    LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
    RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
    BB_RUN = map(BB_RUN, MAX_B_RUN ,0 ,0,6);
}
else if(g > 0)
{
    LL_RUN = map(LL_RUN, MAX_L_RUN ,0 ,0,6);
    RR_RUN = map(RR_RUN, MIN_R_RUN ,255 ,0,6);
    BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
}

```

```

else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++++ ควบคุมระยะทาง
if((calculate_L_Distance <= l_in_loop_position)|| (calculate_B_Distance <=
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++++ การควบคุมการทำงานในมุม 120 องศา
case(5):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,HIGH);
digitalWrite(7,LOW);
digitalWrite(12,HIGH);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(8,LOW);
//+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN, MAX_L_RUN ,0 ,0,6);
  RR_RUN = map(RR_RUN, MIN_R_RUN ,255 ,0,6);
  BB_RUN = map(BB_RUN, MAX_B_RUN ,0 ,0,6);
}
else if(g > 0)
{

```

```

LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
}
else
{
  LL_RUN = L_RUN;
  RR_RUN = R_RUN;
  BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++++ ควบคุมระยะทาง
if((calculate_L_Distance <= l_in_loop_position)|| (calculate_B_Distance <=
b_in_loop_position))
{
  Serial.println(b_in_loop_position);
  Serial.println(l_in_loop_position);
  summing_serial = "";
  calculate_L_Distance = L_Distance;
  calculate_B_Distance = B_Distance;
  calculate_R_Distance = R_Distance;
  l_in_loop_position = 0;
  b_in_loop_position = 0;
  r_in_loop_position = 0;
  delay(30);
}
break;

//+++++++ การควบคุมการทำงานในมุม 300 องศา
case(11):
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = l_in_loop_position + 1;
b_in_loop_position = b_in_loop_position + 1;
r_in_loop_position = r_in_loop_position + 1;
l_count = 0;
b_count = 0;
r_count = 0;
digitalWrite(13,LOW);
digitalWrite(7,HIGH);
digitalWrite(12,LOW);
digitalWrite(4,HIGH);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
//+++++++ ทำการตรวจสอบการเคลื่อนที่ไม่ให้ออกไปจากทิศทางที่
กำหนดโดยปรับเทียบสัดส่วนแบบ Fuzzy
if(g < 0)
{
  LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
  RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);

```

```

    BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
}
else if(g > 0)
{
    LL_RUN = map(LL_RUN, MIN_L_RUN ,255 ,0,6);
    RR_RUN = map(RR_RUN, MAX_R_RUN ,0 ,0,6);
    BB_RUN = map(BB_RUN, MIN_B_RUN ,255 ,0,6);
}
else
{
    LL_RUN = L_RUN;
    RR_RUN = R_RUN;
    BB_RUN = B_RUN;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
//+++++ ควณคุมระยะทาง
if((calculate_L_Distance <= l_in_loop_position)|| (calculate_B_Distance <=
b_in_loop_position))
{
    Serial.println(b_in_loop_position);
    Serial.println(l_in_loop_position);
    summing_serial = "";
    calculate_L_Distance = L_Distance;
    calculate_B_Distance = B_Distance;
    calculate_R_Distance = R_Distance;
    l_in_loop_position = 0;
    b_in_loop_position = 0;
    r_in_loop_position = 0;
    delay(30);
}
break;

//+++++ ทำการหยุดการควบคุมการทำงาน
case(0):// STOP MODE
calculate_L_Distance = L_Distance;
calculate_B_Distance = B_Distance;
calculate_R_Distance = R_Distance;
l_in_loop_position = 0;
b_in_loop_position = 0;
r_in_loop_position = 0;
summing_serial = "";
L_RUN = 0;
R_RUN = 0;
B_RUN = 0;
LL_RUN = 0;
RR_RUN = 0;
BB_RUN = 0;
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
delay(50);

```

```

digitalWrite(13,LOW);
digitalWrite(7,LOW);
digitalWrite(12,LOW);
digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
break;
}

//+++++++ ส่วนของการเคลื่อนที่กลับไปยังตำแหน่งที่กำหนดเมื่อมีการหลบสิ่งกีดขวาง
if(mode == 0)
{

    //+++++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 90 , 270 , 300 , 330 , 30 , 60
องศา โดยกลับไปทางมุม 180 องศา
    if((l_out_1_loop_position > 0)|| (b_out_1_loop_position > 0))
    {
        digitalWrite(13,LOW);
        digitalWrite(7,LOW);
        digitalWrite(12,LOW);
        digitalWrite(4,LOW);
        digitalWrite(6,LOW);
        digitalWrite(8,LOW);
        l_out_1_loop_position = l_out_1_loop_position - 1;
        b_out_1_loop_position = b_out_1_loop_position - 1;
        delay(500);
        g = 0;
        for(int o=0;o<=350;o++)
        {
            getGyroscopeValues();
            g = g+((y/200));
            digitalWrite(13,LOW);
            digitalWrite(7,LOW);
            digitalWrite(12,HIGH);
            digitalWrite(4,HIGH);
            digitalWrite(6,HIGH);
            digitalWrite(8,LOW);
            if(g < 0)
            {
                LL_RUN = 80;
                RR_RUN = 80;
                BB_RUN = 255;
            }
            else if(g > 0)
            {
                LL_RUN = 130;
                RR_RUN = 130;
                BB_RUN = 255;
            }
            else
            {
                LL_RUN = 125;
                RR_RUN = 125;
            }
        }
    }
}

```

```

        BB_RUN = 255;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
}

//+++++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 120 , 150 องศา โดยกลับไปทางมุม
240
else if((l_out_2_loop_position > 0)|| (b_out_2_loop_position > 0))
{
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,LOW);
    l_out_2_loop_position = l_out_2_loop_position - 1;
    b_out_2_loop_position = b_out_2_loop_position - 1;
    delay(500);
    g = 0;
    for(int o=0;o<=350;o++)
    {
        getGyroscopeValues();
        g = g+((y/200));
        digitalWrite(13,LOW);
        digitalWrite(7,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(4,HIGH);
        digitalWrite(6,LOW);
        digitalWrite(8,LOW);
        if(g < 0)
        {
            LL_RUN = 234;
            RR_RUN = 147;
            BB_RUN = 104;
        }
        else if(g > 0)
        {
            LL_RUN = 255;
            RR_RUN = 122;
            BB_RUN = 92;
        }
    }
    else
    {
        LL_RUN = 255;
        RR_RUN = 147;
        BB_RUN = 104;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}

```

```

    }
}

//+++++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 0 , 180 , 120 , 150 , 30 , 60
องศาโดยกลับไปทางมุม 270 องศา
else if((l_out_3_loop_position > 0)|| (b_out_3_loop_position > 0))
{
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,LOW);
    l_out_3_loop_position = l_out_3_loop_position - 1;
    b_out_3_loop_position = b_out_3_loop_position - 1;
    delay(500);
    g = 0;
    for(int o=0;o<=350;o++)
    {
        getGyroscopeValues();
        g = g+((y/200));
        digitalWrite(13,LOW);
        digitalWrite(7,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(4,HIGH);
        digitalWrite(6,LOW);
        digitalWrite(8,LOW);
        if(g < 0)
        {
            LL_RUN = 255;
            RR_RUN = 255;
            BB_RUN = 0;
        }
        else if(g > 0)
        {
            LL_RUN = 255;
            RR_RUN = 180;
            BB_RUN = 0;
        }
        else
        {
            LL_RUN = 255;
            RR_RUN = 255;
            BB_RUN = 0;
        }
        analogWrite(9,LL_RUN);
        analogWrite(10,RR_RUN);
        analogWrite(11,BB_RUN);
    }
}

//+++++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 210 , 240 องศาโดยกลับไปทางมุม
330

```

```

else if((l_out_4_loop_position > 0)|| (b_out_4_loop_position > 0))
{
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,LOW);
    l_out_4_loop_position = l_out_4_loop_position - 1;
    b_out_4_loop_position = b_out_4_loop_position - 1;
    delay(500);
    g = 0;
    for(int o=0;o<=350;o++)
    {
        getGyroscopeValues();
        g = g+((y/200));
        digitalWrite(13,LOW);
        digitalWrite(7,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(4,LOW);
        digitalWrite(6,LOW);
        digitalWrite(8,HIGH);
        if(g < 0)
        {
            LL_RUN = 0;
            RR_RUN = 255;
            BB_RUN = 200;
        }
        else if(g > 0)
        {
            LL_RUN = 0;
            RR_RUN = 200;
            BB_RUN = 255;
        }
        else
        {
            LL_RUN = 0;
            RR_RUN = 200;
            BB_RUN = 255;
        }
        analogWrite(9,LL_RUN);
        analogWrite(10,RR_RUN);
        analogWrite(11,BB_RUN);
    }
}

```

//+++++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 90 , 270 , 210 , 240 , 120 , 150 องศาโดยกลับไปทางมุม 0 องศา

```

else if((l_out_5_loop_position > 0)|| (b_out_5_loop_position > 0))
{
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);

```

```

digitalWrite(4,LOW);
digitalWrite(6,LOW);
digitalWrite(8,LOW);
l_out_5_loop_position = l_out_5_loop_position - 1;
b_out_5_loop_position = b_out_5_loop_position - 1;
delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  getGyroscopeValues();
  g = g+((y/200));
  digitalWrite(13,HIGH);
  digitalWrite(7,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,HIGH);
  if(g < 0)
  {
    LL_RUN = 130;
    RR_RUN = 130;
    BB_RUN = 255;
  }
  else if(g > 0)
  {
    LL_RUN = 80;
    RR_RUN = 80;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 125;
    RR_RUN = 125;
    BB_RUN = 255;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
}

//+++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 300 , 330 องศาโดยกลับทางมุม 30
องศา
else if((l_out_6_loop_position > 0)|| (b_out_6_loop_position > 0))
{
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  l_out_6_loop_position = l_out_6_loop_position - 1;
  b_out_6_loop_position = b_out_6_loop_position - 1;

```

```

delay(500);
g = 0;
for(int o=0;o<=350;o++)
{
  getGyroscopeValues();
  g = g+((y/200));
  digitalWrite(13,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,HIGH);
  if(g < 0)
  {
    LL_RUN = 255;
    RR_RUN = 0;
    BB_RUN = 240;
  }
  else if(g > 0)
  {
    LL_RUN = 255;
    RR_RUN = 0;
    BB_RUN = 255;
  }
  else
  {
    LL_RUN = 255;
    RR_RUN = 0;
    BB_RUN = 255;
  }
  analogWrite(9,LL_RUN);
  analogWrite(10,RR_RUN);
  analogWrite(11,BB_RUN);
}
}

```

//+++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 0 , 180 , 300 , 330 , 210 , 240 องศาโดยกลับไปทางมุม 90 องศา

```

else if((l_out_7_loop_position > 0)|| (b_out_7_loop_position > 0))
{
  digitalWrite(13,LOW);
  digitalWrite(7,LOW);
  digitalWrite(12,LOW);
  digitalWrite(4,LOW);
  digitalWrite(6,LOW);
  digitalWrite(8,LOW);
  l_out_7_loop_position = l_out_7_loop_position - 1;
  b_out_7_loop_position = b_out_7_loop_position - 1;
  delay(500);
  g = 0;
  for(int o=0;o<=350;o++)
  {
    getGyroscopeValues();

```

```

    g = g+((y/200));
    digitalWrite(13,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(8,LOW);
    if(g < 0)
    {
        LL_RUN = 255;
        RR_RUN = 180;
        BB_RUN = 0;
    }
    else if(g > 0)
    {
        LL_RUN = 255;
        RR_RUN = 230;
        BB_RUN = 0;
    }
    else
    {
        LL_RUN = 255;
        RR_RUN = 230;
        BB_RUN = 0;
    }
    analogWrite(9,LL_RUN);
    analogWrite(10,RR_RUN);
    analogWrite(11,BB_RUN);
}
}

//+++++ ในการเกิดการหลบสิ่งกีดขวางในมุม 30 , 60 องศาโดยกลับไปทางมุม
120 องศา
else if((l_out_8_loop_position > 0)|| (b_out_8_loop_position > 0))
{
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    digitalWrite(6,LOW);
    digitalWrite(8,LOW);
    l_out_8_loop_position = l_out_8_loop_position - 1;
    b_out_8_loop_position = b_out_8_loop_position - 1;
    delay(500);
    g = 0;
    for(int o=0;o<=350;o++)
    {
        getGyroscopeValues();
        g = g+((y/200));
        digitalWrite(13,HIGH);
        digitalWrite(7,LOW);
        digitalWrite(12,HIGH);
        digitalWrite(4,LOW);

```

```

digitalWrite(6,HIGH);
digitalWrite(8,LOW);
if(g < 0)
{
  LL_RUN = 144;
  RR_RUN = 243;
  BB_RUN = 102;
}
else if(g > 0)
{
  LL_RUN = 126;
  RR_RUN = 251;
  BB_RUN = 83;
}
else
{
  LL_RUN = 143;
  RR_RUN = 251;
  BB_RUN = 100;
}
analogWrite(9,LL_RUN);
analogWrite(10,RR_RUN);
analogWrite(11,BB_RUN);
}
}

```

```

//+++++ เมื่อกลับมายังตำแหน่งที่กำหนดแล้วก็ทำการเคลียค่าที่อ่านมาทั้งหมด
else
{
  b_count = 0;
  l_count = 0;
  r_count = 0;
  l_out_1_loop_position = 0;
  b_out_1_loop_position = 0;
  r_out_1_loop_position = 0;
  l_out_2_loop_position = 0;
  b_out_2_loop_position = 0;
  r_out_2_loop_position = 0;
  l_out_3_loop_position = 0;
  b_out_3_loop_position = 0;
  r_out_3_loop_position = 0;
  l_out_4_loop_position = 0;
  b_out_4_loop_position = 0;
  r_out_4_loop_position = 0;
  l_out_5_loop_position = 0;
  b_out_5_loop_position = 0;
  r_out_5_loop_position = 0;
  l_out_6_loop_position = 0;
  b_out_6_loop_position = 0;
  r_out_6_loop_position = 0;
  l_out_7_loop_position = 0;
  b_out_7_loop_position = 0;
  r_out_7_loop_position = 0;
}

```

```

    l_out_8_loop_position = 0;
    b_out_8_loop_position = 0;
    r_out_8_loop_position = 0;
    g=0;
  }
}
}
}

```

//+++++++ Function ย่อยของ Encoder เพื่อใช้ในการตรวจสอบตำแหน่งการทำงาน

```

void B_count()
{
  if(digitalRead(2) == LOW)
  {
    j = 1;
  }
  if ((digitalRead(2) == HIGH)&&(j == 1))
  {
    b_count++;
    j = 0;
  }
}

void L_count()
{
  if(digitalRead(3) == LOW)
  {
    h = 1;
  }
  if ((digitalRead(3) == HIGH)&&(h == 1))
  {
    l_count++;
    h = 0;
  }
}

void R_count()
{
  if(digitalRead(3) == LOW)
  {
    d = 1;
  }
  if ((digitalRead(3) == HIGH)&&(d == 1))
  {
    r_count++;
    d = 0;
  }
}

```

//+++++++ Function ย่อย ของ Gyroscope เพื่อใช้ในการแปลงค่า I2C เป็นมุมที่
เกิดขึ้นขณะเคลื่อนที่

```

void getGyroscopeValues ()
{
  byte MSB, LSB;

```

```

MSB = readI2C(0x29);
LSB = readI2C(0x28);
x = ((MSB << 8) | LSB);

```

```

MSB = readI2C(0x2B);
LSB = readI2C(0x2A);
y = ((MSB << 8) | LSB);

```

```

MSB = readI2C(0x2D);
LSB = readI2C(0x2C);
z = ((MSB << 8) | LSB);
}

```

```

int readI2C (byte regAddr)
{
  Wire.beginTransaction(Addr);
  Wire.write(regAddr);
  Wire.endTransmission();
  Wire.requestFrom(Addr, 1);
  while(!Wire.available()) {
  };    // Wait for receipt
  return(Wire.read());
}

```

```

void writeI2C (byte regAddr, byte val)
{
  Wire.beginTransaction(Addr);
  Wire.write(regAddr);
  Wire.write(val);
  Wire.endTransmission();
}

```

ภาคผนวก ค

โปรแกรม C#

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using System.Net;
using System.IO;

namespace rbt_ctrl
{
    public partial class Form1 : Form
    {
        //ชุดตัวแปรค่าคงที่ของหุ่นยนต์
        const int langle = 60;
        const int rangle = -60;
        const int bangle = 90;
        const double RAD_TO_DEG = Math.PI / 180;
        int mode = 1;
        const int MANUAL = 0;
        const int AUTO = 1;
        int k = 0;

        //ชุดตัวแปรสำหรับการส่งข้อมูลไร้สาย
        HttpWebRequest request;
        HttpWebResponse response;
        StreamReader sr;
        string srespond;

        //ชุดตัวแปรสำหรับการสร้างแผนที่
        Bitmap b;
        int xpanacc = 0, ypanacc = 0;
        double rtnx, rtny;
        Point starting_point;
        Point newpoint;
        Point oldpoint;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e) //กำหนดค่าเริ่มต้นของระบบแผนที่
        {
            b = new Bitmap(pictureBox1.Image);
            starting_point = new Point(b.Width / 2, b.Height / 2);
            oldpoint = starting_point;
        }
    }
}

```

//เพิ่มสำหรับการส่งข้อมูลไร้สาย
 //เพิ่มสำหรับการเขียนแผนที่
 //มุมที่ล้อซ้าย
 //มุมที่ล้อขวา
 //มุมที่ล้อหลัง
 //ค่าคงที่ในการแปลงหน่วยเรเดียนเป็นองศา
 //สำหรับการเลือกโหมดของผู้ใช้
 //กำหนดให้ค่าว่า MANUAL = 0
 //กำหนดให้ค่าว่า AUTO = 1
 //สำหรับการจัดลำดับการทำงาน

```

private void button2_Click(object sender, EventArgs e)
//ตรวจสอบว่าค่าที่ผู้ใช้ป้อนมารับได้หรือไม่
{
    try
    {
        //มุมจะรับได้อยู่ที่ 0-360 องศา
        if ((Convert.ToInt16(txtangle.Text) > 360) || Convert.ToInt16(txtangle.Text) < 0)
        {
            MessageBox.Show("Angle Out of Range");
        }
        //ระยะทางจะรับได้อยู่ที่ 0-500 เซนติเมตร (ปรับให้รับได้มากกว่านี้ได้)
        if ((Convert.ToInt16(txtdistance.Text) > 500) || Convert.ToInt16(txtdistance.Text) <
0)
        {
            MessageBox.Show("Distance Out of Range");
        }
        else
        {
            AddItemAuto(Convert.ToInt16(txtangle.Text),
Convert.ToInt16(txtdistance.Text));
        }
    }
    catch (FormatException) //ในกรณีที่ป้อนค่าที่ไม่ใช่ตัวเลข จะมีค่าเตือนขึ้นมา
    {
        MessageBox.Show("Input value isn't integer.");
    }
}

private void btnclr_Click(object sender, EventArgs e) //ลบลำดับการทำงานของหุ่นยนต์
{
    listView1.Items.Clear();
    k = 0;
}

private void btnrun_Click(object sender, EventArgs e)
//สั่งให้หุ่นยนต์ทำงานตามลำดับการทำงาน (แสดงทั้งบนแผนที่ และลจจจริง)
{
    AutoSys_SortList();
    txtoutput.SelectionStart = txtoutput.Text.Length;
    txtoutput.ScrollToCaret();
    k = 0;
}

private void button3_Click(object sender, EventArgs e)
//เป็นการสมมุติให้หุ่นยนต์ทำงานตามลำดับการทำงาน (แสดงแค่บนแผนที่เท่านั้น)
{
    AutoSys_SortList_SimulateRun();
    txtoutput.SelectionStart = txtoutput.Text.Length;
    txtoutput.ScrollToCaret();
    k = 0;
}

```

```
private void button1_Click(object sender, EventArgs e)
//สำหรับการเปลี่ยนไปมาระหว่าง Auto/Manual
{
    if (mode == 1) { man_group.Enabled = true; auto_group.Enabled = false; mode = 0; }
    else if (mode == 0)
    {
        man_group.Enabled = false; auto_group.Enabled = true; mode = 1;
    }
}
```

```
protected override bool ProcessCmdKey(ref Message msg, Keys keyData)
//สำหรับการใช้ควบคุมแบบ Manual ซึ่งจะควบคุมโดยการตรวจจับปุ่มบนคีย์บอร์ด
{
    if (mode == MANUAL)
    {
        if (keyData == Keys.Up) //กดปุ่มลูกศรบน ให้หุ่นยนต์เดินไปข้างหน้า
        {
            label2.Text = "Robot Status : Forward";
            ExecutionGateway("000423025500000000000000000000000000000000");
            return true;
        }
        else if (keyData == Keys.Down) //กดปุ่มลูกศรล่าง ให้หุ่นยนต์ถอยหลัง
        {
            label2.Text = "Robot Status : Down";
            ExecutionGateway("001025525500000000000000000000000000000000");
            return true;
        }
        else if (keyData == Keys.Left) //กดปุ่มลูกศรซ้าย ให้หุ่นยนต์เดินไปทางซ้าย
        {
            label2.Text = "Robot Status : Left";
            ExecutionGateway("000712512525500000000000000000000000000000");
            return true;
        }
        else if (keyData == Keys.Right) //กดปุ่มลูกศรขวา ให้หุ่นยนต์เดินไปทางขวา
        {
            label2.Text = "Robot Status : Right";
            ExecutionGateway("000112512525500000000000000000000000000000");
            return true;
        }
        else if (keyData == Keys.Space) //กดปุ่ม Space Bar ให้หุ่นยนต์หยุด
        {
            label2.Text = "Robot Status : Stop";
            ExecutionGateway("000000000000000000000000000000000000000000");
            return true;
        }
    }
}
return base.ProcessCmdKey(ref msg, keyData);
}
```

```

private void btnaddcir_Click(object sender, EventArgs e)
/*
คำนวณการเคลื่อนที่เป็นวงกลม โดยจะเคลื่อนที่เป็นแบบ Interpolate
โดยที่จะหารระยะที่ต้องใช้แต่ละจุดได้โดย “ระยะทางแต่ละจุด = ระยะทางเชิงมุม / จำนวนจุดที่ใช้ Interpolate”
และมุมที่ต้องใช้เพิ่มในแต่ละจุดโดย “มุมที่ต้องใช้เพิ่มในแต่ละจุด = มุมที่ต้องการให้หมุนหยุด / จำนวนจุดที่ใช้ Interpolate”
*/
{
    double r = Convert.ToDouble(txtradius.Text);
    double angle = Convert.ToDouble(txtanglecir.Text);
    double interpolatepoint = Convert.ToDouble(txtinrpts.Text);

    double curveangle = angle / (interpolatepoint + 1);
    double distance = (2 * Math.PI * r) * (curveangle / 360);

    if (rb_cw.Checked)
    {
        for (double i = 1; i <= interpolatepoint; i++)
        {
            AddItemAuto(Convert.ToInt16((curveangle * i) + 90), (int)distance);
        }
    }
    else if (rb_ccw.Checked)
    {
        for (double i = 1; i <= interpolatepoint; i++)
        {
            AddItemAuto(-Convert.ToInt16((curveangle * i) + 90), (int)distance);
        }
    }
    if (angle == 360)
    {
        AddItemAuto(0, (int)distance);
    }
}

```

```

private void AddItemAuto(int angle, int dist)
//เพิ่มลำดับการทำงานของคำสั่งหุ่นยนต์แบบ Auto
{
    k++;
    ListViewItem item = new ListViewItem(Convert.ToString(k), 0);
    item.SubItems.Add(Convert.ToString(angle));
    item.SubItems.Add(Convert.ToString(dist));
    listView1.Items.Add(item);
}

```

```

private void AutoSys_SortList()
{
    //ฟังก์ชันสำหรับการส่งข้อมูลการเคลื่อนที่แบบไร้สายของหุ่นยนต์ตามลำดับการทำงาน
    for (; )
    {
        if (listView1.Items.Count >= 1)
        {

```

```

        string gw_str = "";
        int deg = Convert.ToInt16(listView1.Items[0].SubItems[1].Text);
        int dist = Convert.ToInt16(listView1.Items[0].SubItems[2].Text);
        MapDrawer(deg, dist);
        gw_str += "1";
        gw_str += Convert.ToInt16(cb_ultsn.Checked);
        gw_str += directionstring(deg, 1, AUTO);
        gw_str += distancestring(deg, dist);
        ExecutionGateway(gw_str);
        System.Threading.Thread.Sleep(40 * dist);
        txtoutput.Text += Environment.NewLine + "C# : Point Complete";
        listView1.Items[0].Remove();
    }
    else
    {
        txtoutput.Text += Environment.NewLine + "C# : Execution Complete";
        break;
    }
}
HttpRequest request = (HttpRequest)WebRequest.Create("http://" +
txtip.Text);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream(),
Encoding.UTF8);
srespond = sr.ReadToEnd();
using (StringReader reader = new StringReader(srespond))
{
    string line;
    while ((line = reader.ReadLine()) != null)
    {
        if (line.Contains("*"))
        {
        }
    }
}
}

private void AutoSys_SortList_SimulateRun()
{
    //ฟังก์ชันการจำลองการส่งข้อมูลการเคลื่อนที่แบบไร้สายของหุ่นยนต์ตามลำดับการทำงาน (จะมีเพียงเส้นทางขึ้นบนแผนที่)
    for (; )
    {
        if (listView1.Items.Count >= 1)
        {
            string gw_str = "";
            int deg = Convert.ToInt16(listView1.Items[0].SubItems[1].Text);
            int dist = Convert.ToInt16(listView1.Items[0].SubItems[2].Text);
            MapDrawer(deg, dist);
            gw_str += Convert.ToInt16(cb_ultsn.Checked) + directionstring(deg, 1, AUTO) +
            distancestring(deg, dist);
            txtoutput.Text += Environment.NewLine + "C# : SimulateRun at " + deg;

```

```

        txtoutput.Text += Environment.NewLine + "C# : Point " + gw_str + " Complete";
        listView1.Items[0].Remove();
    }
    else
    {
        txtoutput.Text += Environment.NewLine + "C# : Execution Complete";
        break;
    }
}
}

private void ExecutionGateway(string str)
{
    //ฟังก์ชันสำหรับการส่งข้อมูลไร้สาย ซึ่งส่งโดยใช้วิธี HTTPGET
    try
    {
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://" +
txtip.Text + "/" + str);
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();
        StreamReader sr = new StreamReader(response.GetResponseStream(),
Encoding.UTF8);
        srespond = sr.ReadToEnd();
    }
    catch { MessageBox.Show("Only 1 Execution can rec."); }
}

private string directionstring(int deg, double speed, int mode)
{
    //ฟังก์ชันสำหรับการกำหนดทิศทางการหมุนของล้อ
    deg = -deg;
    deg += 90;
    if (deg < -180) deg += 360;
    string rtnstr = "";
    if (deg == 0) { rtnstr += "04"; }
    else if (deg >= 1 && deg <= 59) { rtnstr += "03"; }
    else if (deg == 60) { rtnstr += "02"; }
    else if (deg >= 61 && deg <= 119) { rtnstr += "01"; }
    else if (deg == 120) { rtnstr += "12"; }
    else if (deg >= 121 && deg <= 179) { rtnstr += "11"; }
    else if (deg == 180) { rtnstr += "10"; }

    else if (deg >= -59 && deg <= -1) { rtnstr += "05"; }
    else if (deg == -60) { rtnstr += "06"; }
    else if (deg >= -119 && deg <= -61) { rtnstr += "07"; }
    else if (deg == -120) { rtnstr += "08"; }
    else if (deg >= -179 && deg <= -121) { rtnstr += "09"; }
    else if (deg == -180) { rtnstr += "10"; }

    int LOOPCOUNT = 0;
    if (mode == MANUAL) LOOPCOUNT = 0;
    if (mode == AUTO) LOOPCOUNT = 2;
}

```

```

/*ฟังก์ชันสำหรับการกำหนดความเร็วของล้อ
สามารถแบ่งได้เป็นมุม
-180 - -150 / -150 - -120 / -120 - -90 / -90 - -60 / -60 - -30 / -30 - 0
0 - 30 / 30 - 60 / 60 - 90 / 90 - 120 / 120 - 150 / 150 - 180
ความเร็วของล้อจะส่งค่า PWM ซึ่งแบ่งได้เป็น 3 ค่า ซึ่งจะใส่ในรูปแบบอาร์เรย์
ค่าแรกจะเป็นความเร็วกลาง ค่าที่สองจะเป็นความเร็วสูงสุด ค่าที่สามจะเป็นความเร็วต่ำสุด
การใช้งานค่าที่สองและค่าที่สามจะขึ้นอยู่กับ Gyroscope (ตรวจสอบการเบี่ยงของหุ่นยนต์)
*/
if (deg >= 0 && deg <= 30)
{
    double[] rspds = { 230, 230, 180 }; double[] rspde = { 126, 126, 113 };
    double[] bspds = { 0, 0, 0 }; double[] bspde = { 88, 88, 83 };
    double[] rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * (deg / 30));
        bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * (deg / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (255 * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= 30 && deg <= 60)
{
    double[] rspds = { 126, 126, 113 }; double[] rspde = { 0, 0, 0 };
    double[] bspds = { 88, 88, 83 }; double[] bspde = { 255, 255, 240 };
    double[] rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg - 30) / 30));
        bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * ((deg - 30) / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (255 * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= 60 && deg <= 90)
{
    double[] lspds = { 255, 255, 255 }; double[] lspde = { 125, 130, 80 };
    double[] rspds = { 0, 0, 0 }; double[] rspde = { 125, 130, 80 };
    double[] bspds = { 255, 255, 240 }; double[] bspde = { 255, 255, 255 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] + (Math.Abs(rspds[i] - rspde[i]) * ((deg - 60) / 30));
        lspdr[i] = lspds[i] - (Math.Abs(lspds[i] - lspde[i]) * ((deg - 60) / 30));
        bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * ((deg - 60) / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= 90 && deg <= 120)

```

```

{
    double[] lspds = { 125, 130, 80 }; double[] lspde = { 0, 0, 0 };
    double[] rspds = { 125, 130, 80 }; double[] rspde = { 200, 255, 200 };
    double[] bspds = { 255, 255, 255 }; double[] bspde = { 255, 255, 200 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] + (Math.Abs(rspds[i] - rspde[i]) * ((deg - 90) / 30));
        lspdr[i] = lspds[i] - (Math.Abs(lspds[i] - lspde[i]) * ((deg - 90) / 30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg - 90) / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= 120 && deg <= 150)
{
    double[] lspds = { 0, 0, 0 }; double[] lspde = { 100, 102, 75 };
    double[] rspds = { 200, 255, 200 }; double[] rspde = { 246, 255, 200 };
    double[] bspds = { 255, 255, 200 }; double[] bspde = { 143, 144, 108 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] + (Math.Abs(rspds[i] - rspde[i]) * ((deg - 120) / 30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg - 120) / 30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg - 120) / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= 150 && deg <= 180)
{
    double[] lspds = { 120, 122, 90 }; double[] lspde = { 255, 255, 255 };
    double[] rspds = { 255, 255, 200 }; double[] rspde = { 255, 255, 180 };
    double[] bspds = { 120, 122, 90 }; double[] bspde = { 0, 0, 0 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg - 150) / 30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg - 150) / 30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg - 150) / 30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= -30 && deg <= 0)
{
    double[] rspds = { 230, 230, 180 }; double[] rspde = { 251, 251, 243 };
    double[] lspds = { 255, 255, 255 }; double[] lspde = { 143, 144, 126 };
    double[] bspds = { 0, 0, 0 }; double[] bspde = { 100, 102, 83 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];

```

```

for (int i = 0; i <= LOOPCOUNT; i++)
{
    rspdr[i] = rspds[i] + (Math.Abs(rspds[i] - rspde[i]) * (deg / -30));
    lspdr[i] = lspds[i] - (Math.Abs(lspds[i] - lspde[i]) * (deg / -30));
    bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * (deg / -30));
    rtnstr += (rspdr[i] * speed).ToString("000");
    rtnstr += (lspdr[i] * speed).ToString("000");
    rtnstr += (bspdr[i] * speed).ToString("000");
}
}
else if (deg >= -60 && deg <= -30)
{
    double[] lspds = { 120, 122, 100 }; double[] lspde = { 0, 0, 0 };
    double[] rspds = { 255, 255, 255 }; double[] rspde = { 255, 255, 230 };
    double[] bspds = { 120, 122, 100 }; double[] bspde = { 250, 255, 255 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg + 30) / -30));
        lspdr[i] = lspds[i] - (Math.Abs(lspds[i] - lspde[i]) * ((deg + 30) / -30));
        bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * ((deg + 30) / -30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= -90 && deg <= -60)
{
    double[] lspds = { 0, 0, 0 }; double[] lspde = { 125, 130, 80 };
    double[] rspds = { 255, 255, 230 }; double[] rspde = { 125, 130, 80 };
    double[] bspds = { 250, 255, 255 }; double[] bspde = { 255, 255, 255 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg + 60) / -30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg + 60) / -30));
        bspdr[i] = bspds[i] + (Math.Abs(bspds[i] - bspde[i]) * ((deg + 60) / -30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= -120 && deg <= -90)
{
    double[] lspds = { 125, 130, 80 }; double[] lspde = { 230, 230, 230 };
    double[] rspds = { 125, 130, 80 }; double[] rspde = { 0, 0, 0 };
    double[] bspds = { 255, 255, 255 }; double[] bspde = { 255, 255, 240 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg + 90) / -30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg + 90) / -30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg + 90) / -30));
    }
}

```

```

        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= -150 && deg <= -120)
{
    double[] rspds = { 0, 0, 0 }; double[] rspde = { 147, 147, 122 };
    double[] lspds = { 230, 230, 230 }; double[] lspde = { 255, 255, 234 };
    double[] bspds = { 255, 255, 245 }; double[] bspde = { 104, 104, 92 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] + (Math.Abs(rspds[i] - rspde[i]) * ((deg + 120) / -30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg + 120) / -30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg + 120) / -30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
else if (deg >= -180 && deg <= -150)
{
    double[] lspds = { 147, 147, 122 }; double[] lspde = { 255, 255, 255 };
    double[] rspds = { 255, 255, 234 }; double[] rspde = { 255, 255, 180 };
    double[] bspds = { 124, 134, 102 }; double[] bspde = { 0, 0, 0 };
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= LOOPCOUNT; i++)
    {
        rspdr[i] = rspds[i] - (Math.Abs(rspds[i] - rspde[i]) * ((deg + 150) / -30));
        lspdr[i] = lspds[i] + (Math.Abs(lspds[i] - lspde[i]) * ((deg + 150) / -30));
        bspdr[i] = bspds[i] - (Math.Abs(bspds[i] - bspde[i]) * ((deg + 150) / -30));
        rtnstr += (rspdr[i] * speed).ToString("000");
        rtnstr += (lspdr[i] * speed).ToString("000");
        rtnstr += (bspdr[i] * speed).ToString("000");
    }
}
return rtnstr;
}

private string distancestring(int deg, double dist)
{
    //ฟังก์ชันหาระยะทางของหุ่นยนต์ที่ต้องเคลื่อนที่ โดยที่จะแปลงค่าจากหน่วยเซนติเมตร เป็น pulse ของ Encoder
    deg = -deg;
    deg += 90;
    if (deg < -180) deg += 360;
    string rtnstr = "";
    double lr = (deg + langle) * RAD_TO_DEG;
    double rr = (deg + rangle) * RAD_TO_DEG;
    double br = (deg + bangle) * RAD_TO_DEG;
    lr = Math.Sin(lr);
    rr = -Math.Sin(rr);

```

```

br = -Math.Cos(br);
double ldistance;
double rdistance;
double bdistance;
if (deg >= 0 && deg <= 30)
{
    double dists = 5.04, diste = 15;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg) / 30)));
    dist *= distr;
}
else if (deg >= 30 && deg <= 60)
{
    double dists = 19, diste = 20;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg - 30) / 30)));
    dist *= distr;
}
else if (deg >= 60 && deg <= 90)
{
    double dists = 20, diste = 18;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg - 60) / 30)));
    dist *= distr;
}
else if (deg >= 90 && deg <= 120)
{
    double dists = 18, diste = 15;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg - 90) / 30)));
    dist *= distr;
}
else if (deg >= 120 && deg <= 150)
{
    double dists = 15, diste = 18;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg - 120) / 30)));
    dist *= distr;
}
else if (deg >= 150 && deg <= 180)
{
    double dists = 18, diste = 18.5;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg - 150) / 30)));
    dist *= distr;
}
else if (deg >= -30 && deg <= 0)
{
    double dists = 5.04, diste = 20.5;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg) / -30)));
    dist *= distr;
}
else if (deg >= -60 && deg <= -30)
{
    double dists = 20.5, diste = 14.5;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg + 30) / -30)));
    dist *= distr;
}
else if (deg >= -90 && deg <= -60)

```

```

{
    double dists = 14.5, diste = 20;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg + 60) / -30)));
    dist *= distr;
}
else if (deg >= -120 && deg <= -90)
{
    double dists = 20, diste = 22;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg + 90) / -30)));
    dist *= distr;
}
else if (deg >= -150 && deg <= -120)
{
    double dists = 22, diste = 18;
    double distr = dists - ((Math.Abs(dists - diste) * ((deg + 120) / -30)));
    dist *= distr;
}
else if (deg >= -180 && deg <= -150)
{
    double dists = 18, diste = 18.5;
    double distr = dists + ((Math.Abs(dists - diste) * ((deg + 150) / -30)));
    dist *= distr;
}
dist *= .55;
if (lr == 0) { }
else
{
    ldistance = Math.Abs(dist * lr);
    if (ldistance.ToString("00000") == "00000")
    { }
    else
    {
        rtnstr += ldistance.ToString("00000");
    }
}
if (br == 0) { }
else
{
    bdistance = Math.Abs(dist * br);
    if (bdistance.ToString("00000") == "00000") { }
    else
    {
        rtnstr += bdistance.ToString("00000");
    }
}
if (rr == 0) { }
else
{
    rdistance = Math.Abs(dist * rr);
    if (rdistance.ToString("00000") == "00000")
    { }
    else
    {

```

```

        rtnstr += rdistance.ToString("00000");
    }
}
return rtnstr;
}

#region MapSystem
private void MapDrawer(int deg, int dist)
{
    /*ฟังก์ชันการวาดแผนที่การเดินทางของหุ่นยนต์
    ภาพในระบบคอมพิวเตอร์จะใช้การระบบตำแหน่งไม่เหมือนกับระบบคาร์ทีเซียน
    ภาพจะใช้เฉพาะ Quadrant ที่ 4 ของระบบคาร์ทีเซียนและไม่มีค่าติดลบ
    วิธีการวาดแผนที่นั้นจะกำหนดตัวแปรการวาด (Graphics)
    เพื่อนำมาใช้กับตัวแปรภาพ (Bitmap) เพื่อที่ให้อ้างอิงกับคอนโทรล PictureBox ได้
    ตัวแปรภาพจะต้องเตรียมให้มีขนาดใหญ่ (ในงานนี้จะใช้ 1000×1000 pixels)
    เมื่อจะวาดครั้งแรกให้ตั้งจุดกึ่งกลางของภาพเพื่อป้องกันไม่ให้เส้นทางนั้นหลุดขอบไป (ซึ่งอาจจะทำให้การวาดมีปัญหา)
    เมื่อวาดแล้วให้จำคูลำดับล่าสุดเอาไว้
    */
    rtnx = dist * (Math.Cos(deg * RAD_TO_DEG));
    rtny = -dist * (Math.Sin(deg * RAD_TO_DEG));
    b = new Bitmap(pictureBox1.Image);
    Pen pen = new Pen(Color.Black, 5);
    Pen pospen = new Pen(Color.Green, 6);
    using (Graphics g = Graphics.FromImage(b))
    {
        newpoint = new Point(oldpoint.X + (int)rtnx, oldpoint.Y + (int)rtny);
        g.DrawLine(pen, oldpoint, newpoint);
        g.DrawEllipse(pospen, newpoint.X - 3, newpoint.Y - 3, 6, 6);
        oldpoint = newpoint;
    }
    pictureBox1.Image = b;
}

private void btnpanup_Click(object sender, EventArgs e)
{
    //สำหรับปุ่มการเลื่อนภาพไปข้างบน
    b = new Bitmap(pictureBox1.Image);
    Bitmap bp = new Bitmap(pictureBox1.Image);
    using (Graphics g = Graphics.FromImage(bp))
    {
        ypanacc -= 10;
        g.Clear(Color.White);
        g.DrawImage(pictureBox1.Image, 0, -10, b.Width, b.Height);
        pictureBox1.Image = bp;
    }
    oldpoint = new Point(oldpoint.X, oldpoint.Y - 10);
    lbl_pancoord.Text = "(" + xpanacc + "," + ypanacc + ")";
}

private void btnpandown_Click(object sender, EventArgs e)
{
    //สำหรับปุ่มการเลื่อนภาพแผนที่ไปข้างล่าง
    b = new Bitmap(pictureBox1.Image);
    Bitmap bp = new Bitmap(pictureBox1.Image);

```

```

using (Graphics g = Graphics.FromImage(bp))
{
    ypanacc += 10;
    g.Clear(Color.White);
    g.DrawImage(pictureBox1.Image, 0, 10, b.Width, b.Height);
    pictureBox1.Image = bp;
}
oldpoint = new Point(oldpoint.X, oldpoint.Y + 10);
lbl_pancoor.Text = "(" + xpanacc + "," + ypanacc + ")";
}
private void btnpanleft_Click(object sender, EventArgs e)
{
    //สำหรับปุ่มการเลื่อนภาพแผนที่ไปทางซ้าย
    b = new Bitmap(pictureBox1.Image);
    Bitmap bp = new Bitmap(pictureBox1.Image);
    using (Graphics g = Graphics.FromImage(bp))
    {
        xpanacc -= 10;
        g.Clear(Color.White);
        g.DrawImage(pictureBox1.Image, -10, 0, b.Width, b.Height);
        pictureBox1.Image = bp;
    }
    oldpoint = new Point(oldpoint.X - 10, oldpoint.Y);
    lbl_pancoor.Text = "(" + xpanacc + "," + ypanacc + ")";
}
private void btnpanright_Click(object sender, EventArgs e)
{
    //สำหรับปุ่มการเลื่อนภาพแผนที่ไปทางขวา
    b = new Bitmap(pictureBox1.Image);
    Bitmap bp = new Bitmap(pictureBox1.Image);
    using (Graphics g = Graphics.FromImage(bp))
    {
        xpanacc += 10;
        g.Clear(Color.White);
        g.DrawImage(pictureBox1.Image, 10, 0, b.Width, b.Height);
        pictureBox1.Image = bp;
    }
    oldpoint = new Point(oldpoint.X + 10, oldpoint.Y);
    lbl_pancoor.Text = "(" + xpanacc + "," + ypanacc + ")";
}
}
#endregion
}
}

```


ภาคผนวก ง
โปรแกรม Android

```
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;

public class MainActivity extends ActionBarActivity {

    final int AUTO = 1;
    final int MANUAL = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //กำหนดตัวแปรแทนอุปกรณ์บนหน้าจอ
        Button btn_forward = (Button) findViewById(R.id.button);
        Button btn_stop = (Button) findViewById(R.id.button2);
        Button btn_right = (Button) findViewById(R.id.button3);
        Button btn_left = (Button) findViewById(R.id.button4);
        Button btn_back = (Button) findViewById(R.id.button5);

        //หากมีการกดปุ่มขึ้น ให้สั่งหุ่นยนต์เดินหน้า
        btn_forward.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ExecuteGateway("0004230255000000000000000000000000000000000000");
            }
        });

        //หากมีการกดปุ่มหยุด ให้สั่งหุ่นยนต์หยุดทำงาน
        btn_stop.setOnClickListener(new View.OnClickListener() {
            @Override
```



```

    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

// ฟังก์ชันสำหรับการส่งข้อมูลไร้สาย
private void ExecuteGateway(final String gw) {
    // สร้าง Thread สำหรับการส่งข้อมูล หากไม่สร้างจะทำให้โปรแกรมค้าง
    Thread background = new Thread(new Runnable() {

        private final HttpClient Client = new DefaultHttpClient();
        private String URL = "http://192.168.0.100/" + gw; // IP ตั้งที่เร้าท์เตอร์
        // ฟังก์ชันสำหรับการส่งข้อมูล
        public void run() {
            try {
                String SetServerString = "";
                HttpGet httpget = new HttpGet(URL);
                ResponseHandler<String> responseHandler = new BasicResponseHandler();
                SetServerString = Client.execute(httpget, responseHandler);
                threadMsg(SetServerString);
            } catch (Throwable t) {
                Log.i("Animation", "Thread exception " + t);
            }
        }
    });
    // ฟังก์ชันสำหรับการรับผลกลับมา
    private void threadMsg(String msg) {

        if (!msg.equals(null) && !msg.equals("")) {
            Message msgObj = handler.obtainMessage();
            Bundle b = new Bundle();
            b.putString("message", msg);
            msgObj.setData(b);
            handler.sendMessage(msgObj);
        }
    }

    private final Handler handler = new Handler() {

        public void handleMessage(Message msg) {
            String aResponse = msg.getData().getString("message");
            if ((null != aResponse)) {
            } else {
                Toast.makeText(
                    getBaseContext(),

```

```

        "Not Got Response From Server.",
        Toast.LENGTH_SHORT).show();
    }
}
};
});
background.start();
}
}

```

- โปรแกรมควบคุมอัตโนมัติ

```
package tt.shuusei.omnibotctrlAT;
```

```
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.SeekBar;
import android.widget.Toast;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;
```

```
public class MainActivity extends ActionBarActivity {
```

//ชุดตัวแปรค่าคงที่ของหุ่นยนต์

```
final int langle = 60;
```

//มมที่ล้อซ้าย

```
final int rangle = -60;
```

//มมที่ล่อขวา

```
final int bangle = 90;
```

//มมที่ล้อหลัง

```
final double RAD_TO_DEG = Math.PI / 180;
```

//ค่าคงที่ในการแปลงหน่วยเรเดียนเป็นองศา

```
final int MANUAL = 0;
```

```
//กำหนดให้คำว่า MANUAL = 0
```

```
final int AUTO = 1;
```

```
//กำหนดให้คำว่า AUTO = 1
```

//ตัวแปรที่ใช้สำหรับการคำนวณ

```
public int angle = 0;
```

```
public int distance = 0;
```

```
public String gwstr = "000000000000000000000000";
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //กำหนดตัวแปรแทนอุปกรณ์บนหน้าจอ
    final TextView lb_angle = (TextView) findViewById(R.id.textView4);
    final TextView lb_distance = (TextView) findViewById(R.id.textView7);
    final SeekBar sb_angle = (SeekBar) findViewById(R.id.seekBar1);
    final SeekBar sb_distance = (SeekBar) findViewById(R.id.seekBar2);
    final Switch sw_ultrasn = (Switch) findViewById(R.id.switch1);
    final Button btn_run = (Button) findViewById(R.id.button);

    //เมื่อมีการเลื่อน SeekBar ให้ทำการเปลี่ยนค่า : สำหรับมุม
    sb_angle.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
            angle = i * 10;
            lb_angle.setText(angle + " Degree");
            stringBuilder();
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });

    //เมื่อมีการเลื่อน SeekBar ให้ทำการเปลี่ยนค่า : สำหรับระยะทาง
    sb_distance.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
            distance = i * 10;
            lb_distance.setText(distance + " cm");
            stringBuilder();
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    });
}

```

```

//เมื่อมีการเลื่อนติ๊กให้ใช้อัลตราโซนิกหาลบสิ่งกีดขวาง ให้ส่งค่าเตรียมไว้
sw_ultrasn.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        stringBuilding();
    }
});

//เมื่อมีการสั่งให้หุ่นยนต์วิ่ง ให้เริ่มส่งข้อมูล
btn_run.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        stringBuilding();
        ExecuteGateway(gwstr);
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

//ฟังก์ชันสำหรับการสร้างความพร้อมส่งข้อมูล
private void stringBuilding() {
    final Switch sw_ultrasn = (Switch) findViewById(R.id.switch1);
    int sw = (sw_ultrasn.isChecked()) ? 1 : 0;
    gwstr = Integer.toString(sw);
    gwstr += directionstring(angle, 1, AUTO);
    //gwstr += distancestring(angle, distance);
    gwstr += distancestring_r2(angle, distance);
}

//ฟังก์ชันสำหรับการส่งข้อมูล
private void ExecuteGateway(final String gw) {
    final Button btn_run = (Button) findViewById(R.id.button);
    btn_run.setEnabled(false);
    Toast.makeText(getApplicationContext(),

```

```

        "Please wait, connecting to server.",
        Toast.LENGTH_SHORT).show();

Thread background = new Thread(new Runnable() {

    private final HttpClient Client = new DefaultHttpClient();
    private String URL = "http://192.168.0.100/" + gw;
    public void run() {
        try {
            String SetServerString = "";
            HttpGet httpget = new HttpGet(URL);
            ResponseHandler<String> responseHandler = new BasicResponseHandler();
            SetServerString = Client.execute(httpget, responseHandler);
            threadMsg(SetServerString);

        } catch (Throwable t) {
            Log.i("Animation", "Thread exception " + t);
        }
    }

    private void threadMsg(String msg) {

        if (!msg.equals(null) && !msg.equals("")) {
            Message msgObj = handler.obtainMessage();
            Bundle b = new Bundle();
            b.putString("message", msg);
            msgObj.setData(b);
            handler.sendMessage(msgObj);
        }
    }

    private final Handler handler = new Handler() {

        public void handleMessage(Message msg) {

            String aResponse = msg.getData().getString("message");

            if ((null != aResponse)) {

                Toast.makeText(
                    getBaseContext(),
                    "Server Response: " + aResponse,
                    Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(
                    getBaseContext(),
                    "Not Got Response From Server.",
                    Toast.LENGTH_SHORT).show();
            }
            btn_run.setEnabled(true);
        }
    };
};

```

```

});

background.start();
}

private String directionstring(double deg, double speed, int mode
{
    //ฟังก์ชันสำหรับการกำหนดทิศทางการหมุนของล้อ
    deg = -deg;
    deg += 90;
    if (deg < -180) deg += 360;
    String rtnstr = "";
    if (deg == 0) {
        rtnstr += "04";
    } else if (deg >= 1 && deg <= 59) {
        rtnstr += "03";
    } else if (deg == 60) {
        rtnstr += "02";
    } else if (deg >= 61 && deg <= 119) {
        rtnstr += "01";
    } else if (deg == 120) {
        rtnstr += "12";
    } else if (deg >= 121 && deg <= 179) {
        rtnstr += "11";
    } else if (deg == 180) {
        rtnstr += "10";
    } else if (deg >= -59 && deg <= -1) {
        rtnstr += "05";
    } else if (deg == -60) {
        rtnstr += "06";
    } else if (deg >= -119 && deg <= -61) {
        rtnstr += "07";
    } else if (deg == -120) {
        rtnstr += "08";
    } else if (deg >= -179 && deg <= -121) {
        rtnstr += "09";
    } else if (deg == -180) {
        rtnstr += "10";
    }
}

/*ฟังก์ชันสำหรับการกำหนดความเร็วของล้อ
สามารถแบ่งได้เป็นมุม
-180 - -150 / -150 - -120 / -120 - -90 / -90 - -60 / -60 - -30 / -30 - 0
0 - 30 / 30 - 60 / 60 - 90 / 90 - 120 / 120 - 150 / 150 - 180
ความเร็วของล้อจะส่งค่า PWM ซึ่งแบ่งได้เป็น 3 ค่า ซึ่งจะใส่ในรูปแบบอาร์เรย์
ค่าแรกจะเป็นความเร็วกลาง ค่าที่สองจะเป็นความเร็วสูงสุด ค่าที่สามจะเป็นความเร็วต่ำสุด
การใช้งานค่าที่สองและค่าที่สามจะขึ้นอยู่กับ Gyroscope (ตรวจสอบการเบี่ยงของหุ่นยนต์)
*/

if (deg >= 0 && deg <= 30) {
    double[] rspds = {230, 230, 180};
    double[] rspde = {126, 126, 113};

```

```

double[] bspds = {0, 0, 0};
double[] bspde = {88, 88, 83};
double[] bspecial = {75, 75, 50};
double[] rspdr = new double[3], bspdr = new double[3];
for (int i = 0; i <= 2; i++) {
    rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * (deg / 30));
    bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * (deg / 30));
    rtnstr += String.format("%03.0f", rspdr[i] * speed);
    rtnstr += "255";
    if (bspdr[i] * speed > bspecial[i])
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    else
        rtnstr += String.format("%03.0f", bspecial[i]);
}
} else if (deg >= 30 && deg <= 60) {
    double[] rspds = {126, 126, 113};
    double[] rspde = {0, 0, 0};
    double[] bspds = {88, 88, 83};
    double[] bspde = {255, 255, 240};
    double[] rspecial = {50, 50, 50};
    double[] rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * ((deg - 30) / 30));
        bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * ((deg - 30) / 30));
        if (rspdr[i] * speed > rspecial[i])
            rtnstr += String.format("%03.0f", rspdr[i] * speed);
        else
            rtnstr += String.format("%03.0f", rspecial[i]);
        rtnstr += "255";
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= 60 && deg <= 90) {
    double[] lspds = {255, 255, 255};
    double[] lspde = {125, 130, 80};
    double[] rspds = {0, 0, 0};
    double[] rspde = {125, 130, 80};
    double[] bspds = {255, 255, 240};
    double[] bspde = {255, 255, 255};
    double[] rspecial = {50, 50, 40};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * ((deg - 60) / 30));
        lspdr[i] = lspds[i] - (Math.abs(lspds[i] - lspde[i]) * ((deg - 60) / 30));
        bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * ((deg - 60) / 30));
        if (rspdr[i] * speed > rspecial[i])
            rtnstr += String.format("%03.0f", rspdr[i] * speed);
        else
            rtnstr += String.format("%03.0f", rspecial[i]);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= 90 && deg <= 120) {
    double[] lspds = {125, 130, 80};

```

```

double[] lspde = {0, 0, 0};
double[] rspds = {125, 130, 80};
double[] rspde = {200, 255, 200};
double[] bspds = {255, 255, 255};
double[] bspde = {255, 255, 200};
double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
for (int i = 0; i <= 2; i++) {
    rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * ((deg - 90) / 30));
    lspdr[i] = lspds[i] - (Math.abs(lspds[i] - lspde[i]) * ((deg - 90) / 30));
    bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg - 90) / 30));
    rtnstr += String.format("%03.0f", rspdr[i] * speed);
    rtnstr += String.format("%03.0f", lspdr[i] * speed);
    rtnstr += String.format("%03.0f", bspdr[i] * speed);
}
} else if (deg >= 120 && deg <= 150) {
    double[] lspds = {0, 0, 0};
    double[] lspde = {100, 102, 75};
    double[] rspds = {200, 255, 200};
    double[] rspde = {246, 255, 200};
    double[] bspds = {255, 255, 200};
    double[] bspde = {143, 144, 108};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * ((deg - 120) / 30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg - 120) / 30));
        bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg - 120) / 30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= 150 && deg <= 180) {
    double[] lspds = {120, 122, 90};
    double[] lspde = {255, 255, 255};
    double[] rspds = {255, 255, 200};
    double[] rspde = {255, 255, 180};
    double[] bspds = {120, 122, 90};
    double[] bspde = {0, 0, 0};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * ((deg - 150) / 30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg - 150) / 30));
        bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg - 150) / 30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= -30 && deg <= 0) {
    double[] rspds = {230, 230, 180};
    double[] rspde = {251, 251, 243};
    double[] lspds = {255, 255, 255};
    double[] lspde = {143, 144, 126};
    double[] bspds = {0, 0, 0};
    double[] bspde = {100, 102, 83};

```

```

double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
for (int i = 0; i <= 2; i++) {
    rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * (deg / -30));
    lspdr[i] = lspds[i] - (Math.abs(lspds[i] - lspde[i]) * (deg / -30));
    bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * (deg / -30));
    rtnstr += String.format("%03.0f", rspdr[i] * speed);
    rtnstr += String.format("%03.0f", lspdr[i] * speed);
    rtnstr += String.format("%03.0f", bspdr[i] * speed);
}
} else if (deg >= -60 && deg <= -30) {
    double[] lspds = {120, 122, 100};
    double[] lspde = {0, 0, 0};
    double[] rspds = {255, 255, 255};
    double[] rspde = {255, 255, 230};
    double[] bspds = {120, 122, 100};
    double[] bspde = {250, 255, 255};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * ((deg + 30) / -30));
        lspdr[i] = lspds[i] - (Math.abs(lspds[i] - lspde[i]) * ((deg + 30) / -30));
        bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * ((deg + 30) / -30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= -90 && deg <= -60) {
    double[] lspds = {0, 0, 0};
    double[] lspde = {125, 130, 80};
    double[] rspds = {255, 255, 230};
    double[] rspde = {125, 130, 80};
    double[] bspds = {250, 255, 255};
    double[] bspde = {255, 255, 255};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * ((deg + 60) / -30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg + 60) / -30));
        bspdr[i] = bspds[i] + (Math.abs(bspds[i] - bspde[i]) * ((deg + 60) / -30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= -120 && deg <= -90) {
    double[] lspds = {125, 130, 80};
    double[] lspde = {230, 230, 230};
    double[] rspds = {125, 130, 80};
    double[] rspde = {0, 0, 0};
    double[] bspds = {255, 255, 255};
    double[] bspde = {255, 255, 240};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] - (Math.abs(rspds[i] - rspde[i]) * ((deg + 90) / -30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg + 90) / -30));
        bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg + 90) / -30));
    }
}

```

```

        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= -150 && deg <= -120) {

    double[] rspds = {0, 0, 0};
    double[] rspde = {147, 147, 122};
    double[] lspds = {230, 230, 230};
    double[] lspde = {255, 255, 234};
    double[] bspds = {255, 255, 245};
    double[] bspde = {104, 104, 92};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * ((deg + 120) / -30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg + 120) / -30));
        bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg + 120) / -30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
} else if (deg >= -180 && deg <= -150) {
    double[] lspds = {147, 147, 122};
    double[] lspde = {255, 255, 255};
    double[] rspds = {255, 255, 234};
    double[] rspde = {255, 255, 180};
    double[] bspds = {104, 104, 92};
    double[] bspde = {0, 0, 0};
    double[] lspdr = new double[3], rspdr = new double[3], bspdr = new double[3];
    for (int i = 0; i <= 2; i++) {
        rspdr[i] = rspds[i] + (Math.abs(rspds[i] - rspde[i]) * ((deg + 150) / -30));
        lspdr[i] = lspds[i] + (Math.abs(lspds[i] - lspde[i]) * ((deg + 150) / -30));
        bspdr[i] = bspds[i] - (Math.abs(bspds[i] - bspde[i]) * ((deg + 150) / -30));
        rtnstr += String.format("%03.0f", rspdr[i] * speed);
        rtnstr += String.format("%03.0f", lspdr[i] * speed);
        rtnstr += String.format("%03.0f", bspdr[i] * speed);
    }
}

return rtnstr;
}

private String distancestring(int deg, double dist)
{
    //ฟังก์ชันหาระยะทางของหุ่นยนต์ที่ต้องเคลื่อนที่ โดยที่จะแปลงค่าจากหน่วยเซนติเมตร เป็น pulse ของ Encoder
    deg = -deg;
    deg += 90;
    String rtnstr = "";
    double lr = (deg + langle) * RAD_TO_DEG;
    double rr = (deg + rangle) * RAD_TO_DEG;
    double br = (deg + bangle) * RAD_TO_DEG;
    lr = Math.sin(lr);
    rr = -Math.sin(rr);

```

```

br = -Math.cos(br);
double ldistance;
double rdistance;
double bdistance;
if (deg >= 0 && deg <= 30) {
    double dists = 20.5, diste = 19;
    double distr = dists - ((Math.abs(dists - diste) * ((deg) / 30)));
    dist *= distr;
} else if (deg >= 30 && deg <= 60) {
    double dists = 19, diste = 20;
    double distr = dists + ((Math.abs(dists - diste) * ((deg - 30) / 30)));
    dist *= distr;
} else if (deg >= 60 && deg <= 90) {
    double dists = 20, diste = 24;
    double distr = dists + ((Math.abs(dists - diste) * ((deg - 60) / 30)));
    dist *= distr;
} else if (deg >= 90 && deg <= 120) {
    double dists = 24, diste = 14;
    double distr = dists - ((Math.abs(dists - diste) * ((deg - 90) / 30)));
    dist *= distr;
} else if (deg >= 120 && deg <= 150) {
    double dists = 14, diste = 17.5;
    double distr = dists + ((Math.abs(dists - diste) * ((deg - 120) / 30)));
    dist *= distr;
} else if (deg >= 150 && deg <= 180) {
    double dists = 17.5, diste = 18.5;
    double distr = dists + ((Math.abs(dists - diste) * ((deg - 150) / 30)));
    dist *= distr;
} else if (deg >= -30 && deg <= 0) {
    double dists = 20.5, diste = 13;
    double distr = dists - ((Math.abs(dists - diste) * ((deg) / -30)));
    dist *= distr;
} else if (deg >= -60 && deg <= -30) {
    double dists = 13, diste = 13.88;
    double distr = dists + ((Math.abs(dists - diste) * ((deg + 30) / -30)));
    dist *= distr;
} else if (deg >= -90 && deg <= -60) {
    double dists = 13.88, diste = 24;
    double distr = dists + ((Math.abs(dists - diste) * ((deg + 60) / -30)));
    dist *= distr;
} else if (deg >= -120 && deg <= -90) {
    double dists = 23, diste = 21;
    double distr = dists - ((Math.abs(dists - diste) * ((deg + 90) / -30)));
    dist *= distr;
} else if (deg >= -150 && deg <= -120) {
    double dists = 21, diste = 37;
    double distr = dists + ((Math.abs(dists - diste) * ((deg + 120) / -30)));
    dist *= distr;
} else if (deg >= -180 && deg <= -150) {
    double dists = 37, diste = 18.5;
    double distr = dists - ((Math.abs(dists - diste) * ((deg + 150) / -30)));
    dist *= distr;
}

```

```
dist *= 1.1;
if (lr == 0) {
} else {
    ldistance = Math.abs(dist * lr);
    if (String.format("%.05f", ldistance) == "00000") {
    } else {
        rtnstr += String.format("%.05f", ldistance);
    }
}
if (br == 0) {
} else {
    bdistance = Math.abs(dist * br);
    if (String.format("%.05f", bdistance) == "00000") {
    } else {
        rtnstr += String.format("%.05f", bdistance);
    }
}
if (rr == 0) {
} else {
    rdistance = Math.abs(dist * rr);
    if (String.format("%.05f", rdistance) == "00000") {
    } else {
        rtnstr += String.format("%.05f", rdistance);
    }
}
return rtnstr;
}
```


ประวัติผู้จัดทำ

ชื่อ : นายอุดมศักดิ์ สมภักดี
 ชื่อปริญญาบัตร : การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
 สาขาวิชา : วิศวกรรมแมคคาทรอนิกส์

ประวัติ

บิดาชื่อ : นายสมชัย สมภักดี
 มารดาชื่อ : นางจามจุรี สมภักดี
 ที่อยู่ : 21 หมู่ที่ 2 ตำบลคลองตะเคียน อำเภอพระนครศรีอยุธยา จังหวัด
 พระนครศรีอยุธยา รหัสไปรษณีย์ 13000
 ประวัติการศึกษา : จบการศึกษาระดับประกาศนียบัตรวิชาชีพ สาขาเครื่องมือกลและซ่อมบำรุง
 วิทยาลัยเทคโนโลยีและอุตสาหกรรมการต่อพระนครศรีอยุธยา
 ปีการศึกษา 2553
 ศึกษาต่อระดับปริญญาตรี สาขาวิศวกรรมแมคคาทรอนิกส์
 ภาควิชาครุศาสตร์เครื่องกล คณะครุศาสตร์อุตสาหกรรม
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประวัติผู้จัดทำ

ชื่อ : นายณัฐชัย ชุ่มประดิษฐ์
 ชื่อปริญญาบัตร : การควบคุมหุ่นยนต์ล้อเคลื่อนที่หลายทิศทาง
 สาขาวิชา : วิศวกรรมแมคคาทรอนิกส์

ประวัติ

บิดาชื่อ : นายสุธี ชุ่มประดิษฐ์
 มารดาชื่อ : นางอุษณีย์ ชุ่มประดิษฐ์
 ที่อยู่ : 25/474 หมู่ 4 ซอยรามคำแหง 174 มีนบุรี กรุงเทพฯ รหัสไปรษณีย์ 10510
 ประวัติการศึกษา : จบการศึกษาระดับมัธยมศึกษาตอนปลาย สายวิทย์-คณิต
 โรงเรียนนวมิทธาโชติศึกษา เตรียมอุดมศึกษาน้อมเกล้า ปีการศึกษา 2553
 ศึกษาต่อระดับปริญญาตรี สาขาวิศวกรรมแมคคาทรอนิกส์
 ภาควิชาครุศาสตร์เครื่องกล คณะครุศาสตร์อุตสาหกรรม
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ