การพัฒนาหุ่นยนต์กู้ภัยโดยระบบกึ่งอัตโนมัติ

DEVELOPMENT OF SEMI-AUTO RESCUE ROBOT

นายนนท์ จูเทศ รหัส 48370891

นายมฆวาล ผ่องใส รหัส 49371262

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่ลง 19 มี.ค. 2555
เลขทะเบียน 15756856
เลขเรียกหนังสือ ว6.
มหาวิทยาลัยนเรศวร 4154 ก
2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การพัฒนาหุ่นยนต์กู้ภัยโดยระบบกึ่งอัตโนมัติ		
ผู้ดำเนินโครงงาน	นายนนท์	ขูเทศ	รหัส 48370891
	นายมฆวาล	ผ่องใส	รหัส 49371262
อาจารย์ที่ปรึกษา	อาจารย์เศรษฐา	ตั้งค้าวานิช	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2552		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

.....กรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

.....กรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

หัวข้อโครงงาน	การพัฒนาหุ่นยนต์กู้ภัยโดยระบบกึ่งอัตโนมัติ		
ผู้ดำเนินโครงงาน	นายนนท์	จูเทศ	รหัส 48370891
	นายมฆวาล	ผ่องใส	รหัส 49371262
อาจารย์ที่ปรึกษา	อาจารย์เศรษฐา	ตั้งค้าวานิช	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2552		

.....

บทคัดย่อ

วัตถุประสงค์ของโครงงานเรื่อง หุ่นยนต์กู้ภัย เป็นการออกแบบและสร้างหุ่นยนต์เพื่อช่วยค้นหาผู้ประสบภัย เนื่องจากผู้ประสบภัยจะอยู่ในสภาพแวดล้อมที่เข้าถึงยากและเป็นอันตรายต่อมนุษย์ หากมีหุ่นยนต์ที่สามารถทำงานในสภาพแวดล้อมดังกล่าวได้ จะช่วยเหลือได้อย่างทันท่วงที โดยหุ่นยนต์จะเคลื่อนที่ด้วยล้อตีนตะขาบที่มีความคงทนและแข็งแรง โดยการควบคุมบนคอมพิวเตอร์ ผ่านอุปกรณ์แบบไร้สายแล้วส่งไปยังหุ่นยนต์ผ่านชุดไมโครคอนโทลเลอร์เพื่อควบคุมหุ่นยนต์และควบคุมการหมุนของมอเตอร์กระแสตรงและเซนเซอร์ต่างๆ จากนั้นหุ่นยนต์จะทำการประมวลผลแล้วทำการส่งข้อมูลกลับมายังคอมพิวเตอร์แบบไร้สาย เพื่อที่จะระบุข้อมูลของสภาพแวดล้อมที่จำเป็นได้ โครงงานนี้จะอธิบายถึงการทำงานของหุ่นยนต์ที่พัฒนาขึ้น อุปกรณ์สำคัญที่ใช้ เช่น กล้อง โมดูลวัดระยะทางด้วย Infrared โมดูลการวัดอุณหภูมิ รวมไปถึงอุปกรณ์ต่างๆ ที่ใช้ควบคุมการทำงานของหุ่นยนต์สิ่งที่ได้รับจากโครงงานนี้คือ ทำให้มีหุ่นยนต์ต้นแบบที่สามารถพัฒนาให้ทำงานร่วมกับมนุษย์ เพื่อช่วยเหลือผู้ประสบภัยจากเหตุการณ์ต่างๆ

Project Title	Development of semi-auto rescue robot		
Name	Mr. Non	Juted	ID. 48370891
	Mr. Makawal	Pongsai	ID. 49371262
Project Advisor	Mr. Settha	Thangkawanit	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2009		

.....

ABSTRACT

The objective of the project on. Rescue robots. A design and build robots to help find victims. Because victims are difficult to access in the environment and harm people. If a robot can work in such environments. Will help instantly. The robots will be mobile by the caterpillar wheels are durable and strong. The controls on the computer. Through a wireless device and then sent to the robot through a set of Microsoft Content Master dealer to control the robot and control the rotation of the direct current motor. Then the robot will perform data processing and then returned to the computer wirelessly. To provide the necessary environment. This project will describe the work of robots developed. Critical equipment such as camera modules used with the Infrared distance measuring module to measure temperature. Include devices that control the operation of the robot is obtained from this project. Make a robot prototype that can be developed to work with people. To assist victims of events.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์อย่างยิ่งในการให้คำปรึกษาและคำแนะนำเกี่ยวกับกับการสร้างหุ่นยนต์กู้ภัย ตลอดทั้งให้ความเอื้อเฟื้อสถานที่ในการทำงานและอุปกรณ์เครื่องมือต่าง ๆ จากอาจารย์เศรษฐา ตั้งค้าวานิช อาจารย์ที่ปรึกษาโครงงาน อีกทั้งคณาจารย์และภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ให้เงินทุนในการสนับสนุนการทำโครงงานนี้ ทำให้การทำโครงงานนี้สำเร็จไปได้ด้วยดี ผู้จัดทำโครงงานขอกราบขอบพระคุณทุกท่านเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นายนนท์ จูเทศ รหัส 48370891

นายมฆวาล ผ่องใส รหัส 49371262

สารบัญ

หน้า

ใบรับรองโครงงานวิศวกรรม ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบข่ายของโครงงาน 1
1.4 ผลที่คาดว่าจะได้รับ 2
1.5 แผนการดำเนินงาน 2
1.6 งบประมาณที่ใช้ 3

บทที่ 2 ทฤษฎีและหลักการทำงาน 4
2.1 มอเตอร์ไฟฟ้ากระแสตรง 4
2.2 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM) 6
2.3 วงจรไดรเวอร์มอเตอร์ 7
2.4 หลักการควบคุมป้อนกลับ 10
2.5 เซ็นเซอร์ 12
2.6 ไมโครคอนโทรลเลอร์ MCS-51 18
2.7 การสื่อสารแบบอนุกรม 20
2.8 เครือข่ายไร้สาย Wireless LAN 22

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและชุดควบคุมการทำงาน....25

3.1 หลักการทำงาน....25

3.2 การออแบบชุดควบคุมหุ่นยนต์....26

3.3 การออกแบบหุ่นยนต์....33

3.4 ชุดเซ็นเซอร์ของหุ่นยนต์....46

3.5 กล้อง....49

3.6 การประกอบชุดควบคุมและกล้อง....51

3.7 การติดตั้งอุปกรณ์ติดต่อสื่อสาร....52

3.8 การออกแบบการเขียนโปรแกรมติดต่อสื่อสารผ่านระบบ RS-232 Wireless....54

3.9 การติดต่อสื่อสารผ่านระบบ RS-232 Wireless....54

3.10 การเขียนโปรแกรมในการรับภาพจากกล้อง IP Camera....56

บทที่ 4 ผลการทดลอง....57

4.1 ขั้นตอนการทดลองหุ่นยนต์....57

4.2 การทดสอบโปรแกรมที่พัฒนาขึ้น....58

4.3 การทดสอบตัวหุ่นยนต์....60

4.4 การทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพแวดล้อมต่างๆ....61

4.5 การทดสอบเซนเซอร์....64

4.6 การทดสอบระยะที่สามารถควบคุมได้....67

4.7 ปัญหาและการแก้ไขปัญหาในการทดลอง....68

บทที่ 5 บทสรุปและข้อเสนอแนะ....71

5.1 สรุปผล....71

5.2 ปัญหาที่พบ....71

5.3 แนวทางการแก้ปัญหาและข้อเสนอแนะ....72

5.4 แนวทางในการพัฒนาเพิ่มเติม....72

เอกสารอ้างอิง....74

ภาคผนวก ก โปรแกรม 75
ภาคผนวก ข การเข้าร่วมการแข่งขัน Thailand Rescue Robot 2009..... 118

มหาวิทยาลัยนเรศวร

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนดำเนินงาน....2

2.1 ลักษณะการหมุนของมอเตอร์เมื่อเปลี่ยนอินพุท....9

2.2 Wireless networking standards.... 23

3.1 การเคลื่อนที่ของหุ่นยนต์....31

3.2 การกำหนดค่าของอุปกรณ์ RF แบบ Full Duplex....55

4.1 ผลการทดสอบเซนเซอร์อุณหภูมิ.... 64

4.2 ผลการทดลองการวัดระยะทางโดยใช้ Encoder.... 66

สารบัญรูป

รูปที่ หน้า

2.1 H-Bridge Switching....5
2.2 H-Bridge Switching เมื่อ S1 และ S3 On....5
2.3 H-Bridge Switching เมื่อ S2 และ S4 On....6
2.4 ความกว้างของพัลส์ขนาดต่างๆ และค่าดิวตี้ไซเคิลของช่วงพัลส์ที่มีความถี่คงที่....6
2.5 โครงสร้างของ MOSFET....7
2.6 โครงสร้างภายในของรีเลย์....8
2.7 หลักการ H-Bridge ของไอซีเบอร์ L298....9
2.8 หลักการทำงานแบบป้อยกลับ....10
2.9 ระบบควบคุมแบบ PID....10
2.10 ระบบควบคุมแบบดิจิตอล....11
2.11 Incremental Encoder....13
2.12 ตัวตรวจจับรังสีอินฟราเรด....14
2.13 Fresnel lens....14
2.14 การทำงานของ PIR....14
2.15 การทำงานของ PIR เมื่อมีสิ่งมีชีวิตเดินผ่าน....15
2.16 TYPICAL CONFIGURATION OF PYROSENSOR....15
2.17 Ultrasonic sensor....16
2.18 วงจรชุดรับของ Ultrasonic sensor....16
2.19 วงจรชุดส่งของ Ultrasonic....17
2.20 ตัววัดอุณหภูมิแบบอินฟราเรด....18
2.21 วงจรไมโครคอนโทรลเลอร์ P89V51RD2....19
2.22 รูปแบบการส่งข้อมูลแบบอนุกรม....20
2.23 การสื่อสารแบบอะซิงโคนัสที่ไม่มีพาริตี้บิต....21
2.24 การสื่อสารแบบอะซิงโคนัสที่มีพาริตี้บิต....21
2.25 Serial port (Com Port) ใช้ในการเชื่อมต่อการส่งสัญญาณ....22
2.26 การต่อสายสัญญาณตามมาตรฐาน RS-232....22
2.27 อุปกรณ์ Wireless Router Access Point....24
3.1 หลักการทำงานของชุดควบคุมหุ่นยนต์....25

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.2 การออกแบบวงจรไมโครคอนโทรลเลอร์ (Microcontroller)....26
3.3 วงจรไมโครคอนโทรลเลอร์แสดงพอร์ทต่างๆ....26
3.4 บอร์ดไมโครคอนโทรลเลอร์....27
3.5 วงจรติดต่อผ่านพอร์ทสื่อสารอนุกรม RS-232....27
3.6 การจัดการขาของไอซี L298....28
3.7 การต่อวงจรใช้งานแบบ 1 ช่องของ L298....29
3.8 บอร์ดควบคุมมอเตอร์ H-Bridge ขนาด 3A....29
3.9 บอร์ดควบคุมมอเตอร์ H-Bridge ขนาด 40A....29
3.10 วงจรขยายกระแสด้วย IC ULN2003A และ IC Not Gate 7404....32
3.11 เหล็กกระบอกสี่เหลี่ยมสำหรับทำโครงล้อ....33
3.12 แผ่นเหล็กสำหรับทำที่ประกบล้อหน้า....33
3.13 แผ่นเหล็กสำหรับทำที่ประกบล้อหลัง....34
3.14 แผ่นเหล็กไว้ยึดแผ่นประกบล้อ....34
3.15 การประบอกชิ้นส่วนของฐานล้อ....34
3.16 รูปฐานล้อเมื่อประกอบแล้ว....35
3.17 เหล็กฉากที่จะนำไปยึดเป็นโครง....35
3.18 โครงสร้างฐานล้อของหุ่นยนต์....35
3.19 ประกบแขนล้อด้านนอก....36
3.20 ประกบแขนล้อด้านในที่เจาะรูแล้ว....36
3.21 แขนล้อของหุ่นยนต์....36
3.22 สเตอร์รถจักรยานยนต์ขนาด 34 ฟัน....37
3.23 รูปจริงของสเตอร์รถจักรยานยนต์ขนาด 34 ฟัน....37
3.24 ชุดส่วนประกอบของล้อ....37
3.25 เพลาขับเคลื่อนและลูกปืนของหุ่นยนต์....38
3.26 โครงหุ่นยนต์ที่ใส่เพลาขับเคลื่อนและเพลายกแขน....38
3.27 ชุดของตัวขับเคลื่อนตัวแขนล้อ....39
3.28 โครงหุ่นยนต์ที่ใส่สเตอร์ขับเคลื่อนแล้ว....39

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.29 โครงฐานของหุ่นยนต์เมื่อใส่ล้อทั้งหมด....39

3.30 เฟืองโซ่ขนาด 9 ฟัน....40

3.31 มอเตอร์กระจกไฟฟ้า....40

3.32 โครงของหุ่นยนต์เมื่อติดตั้งมอเตอร์เข้าไปแล้ว....40

3.33 การติดตั้งชุดของแขนล้อ....41

3.34 การติดตั้งของแขนล้อ....41

3.35 การติดตั้งชุดแขนล้อของหุ่นยนต์....42

3.36 การออกแบบส่วนประกอบของล้อตีนตะขาบ....42

3.37 การออกแบบส่วนประกอบของล้อยกตีนตะขาบ....43

3.38 แผ่นเหล็กที่เชื่อมกับโซ่....43

3.39 ชิ้นส่วนของโซ่ตีนตะขาบ....43

3.40 สายพานรถจักรยานยนต์ที่ตัดแล้ว....44

3.41 ส่วนประกอบของล้อตีนตะขาบ....44

3.42 ล้อเมื่อใส่โซ่ตีนตะขาบแล้ว....44

3.43 การออกแบบท่อนแขน....45

3.44 การออกแบบฐานหมุนกล้อง....45

3.45 บอร์ดวงจร PIR sensor....46

3.46 โมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด GP2Y0A21YK0F....46

3.47 โมดูล Analog to Digital เบอร์ PCF8591 8-bit A/D....47

3.48 การติดตั้งโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด....48

3.49 ตัวตรวจจับอุณหภูมิแบบอินฟราเรด....48

3.50 การติดตั้งตัวตรวจจับอุณหภูมิ....49

3.51 การติดตั้ง Encoder....49

3.52 การเชื่อมต่อกล้องเว็บแคมเข้า Wireless Router....50

3.53 ภาพแสดงกล้อง IP....50

3.54 การประกอบอุปกรณ์การเชื่อมต่อสายไฟ....51

3.55 การประกอบกล่องควบคุมเข้ากับหุ่นยนต์....51

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.56 การติดตั้งกล้อง IP เข้ากับตัวหุ่นยนต์....52

3.57 การติดตั้ง Router Wireless เข้ากับหุ่นยนต์....52

3.58 RS232 Wireless....53

3.59 รูปหุ่นยนต์กู้ภัย....53

3.60 การรับส่งข้อมูลแบบ Full Duplex....55

4.1 สนามที่ใช้ในการทดลองหุ่นยนต์กู้ภัย....57

4.2 โปรแกรมติดต่อ Serial Port....58

4.3 การแสดงผลการวัดระยะทาง....59

4.4 แสดงผลอุณหภูมิ....59

4.5 การแสดงผลกล้องผ่าน Router Wireless....60

4.6 การแสดงผลโปรแกรมควบคุมหุ่นยนต์กู้ภัย....60

4.7 การเคลื่อนที่ในสภาพผิวที่ขรุขระ....62

4.8 การใช้แขนล้อช่วยในทางขรุขระ....62

4.9 แสดงการขึ้นลงบันได....63

4.10 การเคลื่อนที่ในทางลาด....63

4.11 การวัดอุณหภูมิกับแก้วน้ำแข็ง....64

4.12 การวัดอุณหภูมิบนมือคน....65

4.13 การวัดอุณหภูมิกับหัวแร้งบัดกรี....65

4.14 การทดสอบระยะในการควบคุมหุ่นยนต์กู้ภัย....67

4.15 แสดงผลการทดสอบระยะในการควบคุมหุ่นยนต์....67

4.16 ปัญหาการเคลื่อนที่ในทางขรุขระ....68

4.17 การหงายท้องของหุ่นยนต์ตอนขึ้นบันได....69

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

จากภัยธรรมชาติที่เกิดขึ้น ไม่ว่าจะเป็น อัคคีภัย อุทกภัย วาตภัย และยังรวมไปถึงสิ่งที่เกิดจากการกระทำของมนุษย์ อย่างเช่นการก่อวินาศกรรม ซึ่งเป็นเหตุให้มีผู้ได้รับบาดเจ็บและเสียชีวิต ที่เจ้าหน้าที่เข้าไปช่วยเหลือผู้ประสบภัย โดยสะดวกหรือสภาวะแวดล้อมผู้ประสบภัยอาจก่อให้เกิดอันตรายต่อผู้เข้าช่วยเหลือ อย่างเช่นวัตถุมีพิษ, รังสี, วัตถุระเบิด

ดังนั้นโครงงานนี้จึงจัดทำ หุ่นยนต์กู้ภัยเพื่อช่วยค้นหาผู้ประสบภัยในพื้นที่ที่ยากต่อการเข้าถึง และช่วยบอกให้เจ้าหน้าที่กู้ภัยได้ทราบถึงสภาพแวดล้อมของผู้ประสบภัย ซึ่งเทคโนโลยีที่ทันสมัย ทั้งในด้านระบบสื่อสารไร้สาย ระบบอิเล็กทรอนิกส์ ระบบคอมพิวเตอร์ที่มีประสิทธิภาพสามารถนำมาใช้ในการพัฒนาหุ่นยนต์กู้ภัยได้

1.2 จุดประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาระบบโครงสร้างการทำงานของหุ่นยนต์

1.2.2 เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ในการควบคุมมอเตอร์กระแสตรง

1.2.3 เพื่อศึกษาการรับส่งข้อมูลแบบไร้สาย

1.2.4 เพื่อศึกษาและออกแบบโปรแกรมติดต่อระหว่างผู้ควบคุมกับหุ่นยนต์

1.2.5 ใช้ความรู้เชิงวิศวกรรมทางด้านอิเล็กทรอนิกส์

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถแสดงผลภาพจากกล้องที่ส่งเข้ามาแบบไร้สายในระยะทางไม่เกิน 100 เมตร ในที่พื้นโล่งได้

1.3.2 สามารถควบคุมมอเตอร์ด้วยคอมพิวเตอร์ผ่านชุดไมโครคอนโทรลเลอร์ MCS51

1.3.3 หุ่นยนต์สามารถตรวจจับ สิ่งกีดขวาง อุณหภูมิของเป้าหมาย และอุณหภูมิบริเวณรอบข้างได้

1.3.4 สามารถตรวจสอบความเร็วของล้อทั้งสองข้าง

1.3.5 สามารถบังคับหุ่นยนต์ด้วยระบบคอมพิวเตอร์ผ่านระบบการสื่อสารไร้สายในระยะทางไม่เกิน 100 เมตรในที่โล่ง

1.4 ผลคาดว่าที่จะได้รับ

1.4.1 สามารถแสดงผลภาพจากกล้องที่ส่งเข้ามาแบบไร้สายได้

1.4.2 สามารถควบคุมมอเตอร์ด้วยคอมพิวเตอร์ผ่านชุดไมโครคอนโทรลเลอร์ MCS51ได้

1.4.3 สามารถบังคับหุ่นยนต์ด้วยระบบคอมพิวเตอร์ผ่านระบบการสื่อสารไร้สายได้

1.4.4 สามารถควบคุมการเคลื่อนที่ของล้อทั้งสองข้าง ให้ได้ระยะทางเท่ากันเมื่อเดินหน้าหรือถอยหลังได้

1.4.5 ค้นหาเป้าหมาย และตรวจวัดอุณหภูมิโดยรอบเป้าหมายได้

1.4.6 หุ่นยนต์ตรวจจับพื้นผิวที่เป็นอุปสรรคต่อการเคลื่อนที่ได้

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 ตารางแผนดำเนินงาน

รายการ	พ.ศ. 2552							พ.ศ. 2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.5.1 ออกแบบโครงสร้างหุ่นยนต์	←									
1.5.2 จัดหาวัสดุอุปกรณ์ สำหรับโครงสร้างของหุ่นยนต์	←									
1.5.3 หาเครื่องมือและสถานที่การทำโครงงาน	←									
1.5.4 จัดหางบประมาณ			←							
1.5.5 ลงมือสร้างหุ่นยนต์				←						
1.5.6 ทำการทดสอบและแก้ไขโครงสร้างหุ่นยนต์					←					
1.5.7 ทำการประกอบชุดวงจรควบคุมหุ่นยนต์							←			
1.5.8 เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์								←		
1.5.9 ทำการทดสอบและแก้ไขการทำงานของโปรแกรม								←		
1.5.10 ทดสอบและปรับปรุงการทำงานของหุ่นยนต์									←	

1.6 งบประมาณที่ใช้

วัสดุโครงสร้างหุ่นยนต์	10,000 บาท
วงจรควบคุมการทำงาน	6,500 บาท
มอเตอร์	5,500 บาท
ชุดรับส่งข้อมูลไร้สาย	1,500 บาท
เซนเซอร์	1,000 บาท
กล้องไร้สาย	4,900 บาท

ในบทต่อไปจะเป็นทฤษฎีและหลักการทำงาน เพื่อที่จะนำไปประยุกต์ใช้ในการออกแบบและสร้างหุ่นยนต์

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงถูกใช้งานอย่างกว้างขวางในงานอุตสาหกรรมและ เนื่องจากความสัมพันธ์ระหว่างความเร็วกับแรงบิด สามารถที่จะเปลี่ยนแปลงไปตามการใช้งานได้เกือบทุกรูปแบบ สำหรับการใช้งานของทั้งมอเตอร์และการสร้างใหม่ (Regeneration) ในทิศทางและการหมุน การทำงานอย่างต่อเนื่องของ DC Motors โดยทั่วไปจะอยู่ในช่วงความเร็ว 8 ต่อ 1 รวมทั้ง การลดภาระหรือการลดความเร็วในระยะเวลาสั้น ๆ จะอยู่ในช่วงไร้ขอบเขต (ควบคุมการลดความเร็วลงถึงศูนย์รอบ ต่อนาทีได้อย่างราบเรียบนุ่มนวล) มักจะใช้มอเตอร์ไฟฟ้ากระแสตรง เมื่อมันต้องจ่ายแรงบิดที่จะทำให้มอเตอร์หมุนมากกว่าแรงบิดขณะใช้งานปกติ 3 เท่าหรือ มากกว่า และในสถานการณ์ฉุกเฉิน มอเตอร์ไฟฟ้ากระแสตรงสามารถที่จะจ่ายแรงบิดได้มากกว่า 5 เท่าของแรงบิดใช้งานปกติ โดย ปราศจากการหยุดกลางคัน (Stalling) (ต้นกำลังสามารถจ่ายกำลังให้ได้)

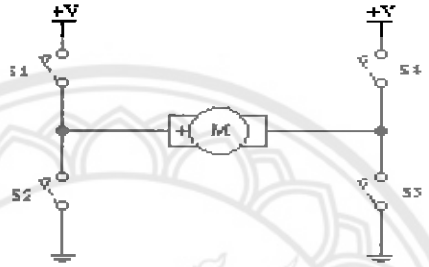
2.1.1 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง สามารถที่จะควบคุมความเร็วจนถึงศูนย์รอบต่อนาทีได้อย่างไม่มีอุปสรรคโดยการเร่งในทิศทางตรงกันข้ามอย่างทันทีทันใด โดยไม่ต้องสับเปลี่ยนวงจรกำลังและมอเตอร์ไฟฟ้ากระแสตรง จะตอบสนองการเปลี่ยนแปลงของสัญญาณควบคุมได้อย่างรวดเร็ว เนื่องจากมีอัตราแรงบิดต่อความเฉื่อยสูงขดลวดสนามแม่เหล็กมอเตอร์ไฟฟ้ากระแสตรง โดยทั่วไปจะแบ่งโดยแยกประเภทของสนามแม่เหล็กของมอเตอร์ ได้แก่ ลวดขนาน (Shunt-wound) ขดลวดอนุกรม (Series-wound) ขดลวดแบบผสม (Compound-wound) นอกจากนี้ ยังมีแบบแม่เหล็กถาวรและแบบไม่มีแปรงถ่าน (Brushless) ใช้งานอยู่บ้างเหมือนกัน ปกติจะเป็นมอเตอร์ที่มีกำลังม้าต่ำ ๆ มอเตอร์อาจจะแบ่งประเภทเป็นแบบใช้งานต่อเนื่องหรือใช้งานเป็นช่วงๆ มอเตอร์ที่ใช้งานต่อเนื่อง สามารถที่จะทำงานโดยไม่ต้องมีเวลาหยุดพักเลย ได้การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไป เช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่า โดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์แต่การควบคุม ในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็น

ค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) [1]

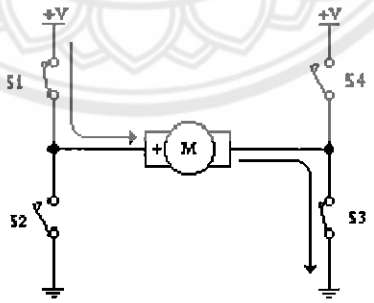
2.1.2 การควบคุมทิศทางการหมุนของมอเตอร์

หลักการของวงจร H-bridge switching จะประกอบไปด้วยสวิทช์ 4 ตัว คือ S1, S2, S3, S4 ซึ่งในรูป จะใช้ DC Motor เป็นโหลดของวงจร ในสภาวะเริ่มต้น สวิทช์ทุกตัว OFF อยู่ก็จะไม่มีอะไรเกิดขึ้น เนื่องจากไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์



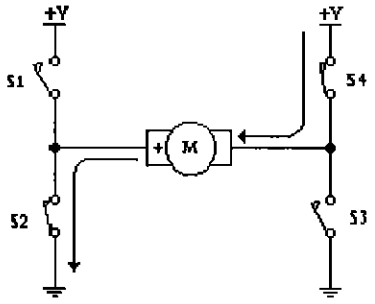
รูปที่ 2.1 ภาพแสดง H-Bridge Switching

เมื่อทำการ ON สวิทช์ S1 และ S3 พร้อมกัน จะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้าไหลผ่านมอเตอร์จากขั้วบวกของมอเตอร์ไปยังขั้วลบของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ในทิศทาง Forward (จะหมุนแบบตามเข็มนาฬิกาหรือทวนเข็มนาฬิกานั้นขึ้นอยู่กับลักษณะของการพันขดลวดภายในมอเตอร์



รูปที่ 2.2 ภาพแสดง H-Bridge Switching เมื่อ S1 และ S3 ON พร้อมกัน

ในทางกลับกัน หากทำการ ON สวิทช์ S2 และ S4 พร้อมกัน จะเป็นการเชื่อมวงจรและทำให้เกิดกระแสไฟฟ้าไหลผ่านมอเตอร์ จากขั้วลบของมอเตอร์ไปยังขั้วบวกของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้และเป็นการหมุนในทิศทาง Backward (กลับทิศทางกับกรณีแรก)

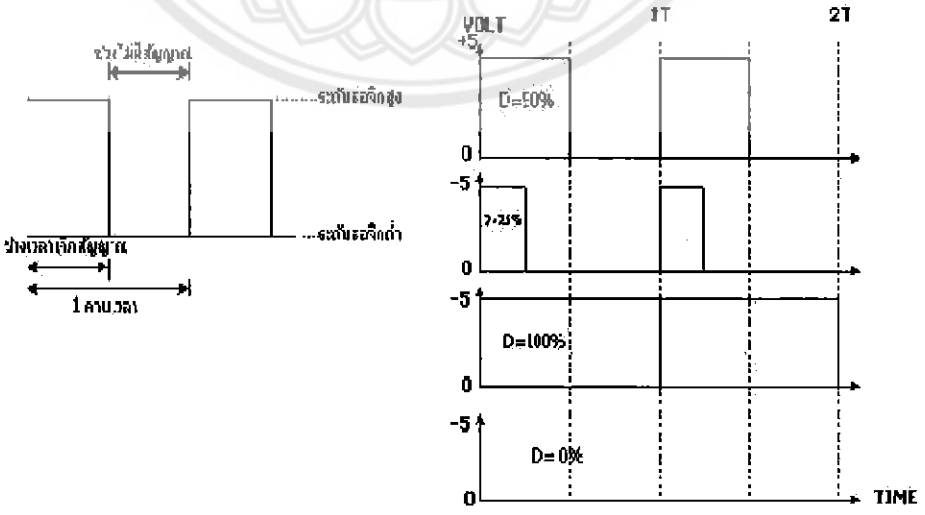


รูปที่ 2.3 ภาพแสดงH-Bridge Switching เมื่อ S2 และ S4 On พร้อมกัน

2.2 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM)

การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนที่สัดส่วน และความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดิวตี้ไซเคิล (Duty Cycle) นั้นเอง ซึ่งค่าของดิวตี้ไซเคิลคือช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดิวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึงใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามากหมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าจะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดิวตี้ไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ดังนี้

ค่าดิวตี้ไซเคิล = (ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ) X100% [2]



รูปที่ 2.4 ภาพแสดงความกว้างของพัลส์ขนาดต่างๆ
และค่าดิวตี้ไซเคิลของช่วงพัลส์ที่มีความถี่คงที่

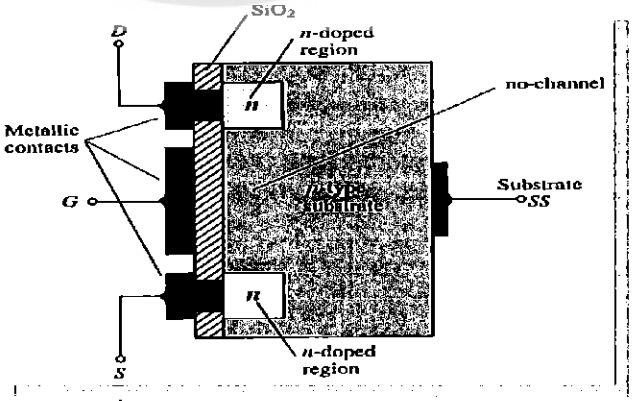
2.3 วงจรไดรเวอร์มอเตอร์

2.3.1 วงจรไดรเวอร์มอเตอร์แบบมอสเฟท

ทรานซิสเตอร์แบบมอสเฟท (Metal–Oxide–Semiconductor field-effect transistor Mosfet) เป็นทรานซิสเตอร์ ที่ใช้อิทธิพลสนามไฟฟ้าในการควบคุมสัญญาณไฟฟ้า โดยใช้ออกไซด์ของโลหะในการทำส่วน GATE นิยมใช้ในวงจรดิจิตอล โดยนำไปสร้างลอจิกเกตต่าง ๆ เพราะมีขนาดเล็ก เป็นเฟสที่ประกอบด้วยสารกึ่งตัวนำซึ่งได้รับ การเคลือบผิวบางส่วนด้วยโลหะออกไซด์ข้อเด่นของเฟสชนิดนี้คือ มีค่าความต้านทานอินพุต (หมายถึงค่าความต่านทานที่เกต) สูงมาก มอสเฟทยังแบ่งเป็น 2 แบบ คือ แบบดีพลีชั่น (Depletion) และแบบเอนฮานซ์เมนต์ (Enhancement) แต่ละประเภทยังแบ่งออกเป็น 2 แบบคือ แบบแชนแนล N และ แบบแชนแนล P มอสเฟทประเภท ดีพลีชั่นหรือดีมอสเฟท (D-MOSFET) ทั้ง 2 แบบจะทำงานได้ 2 โหมด คือ โหมดดีพลีชั่น (Depletion Mode) และ โหมดเอนฮานซ์เมนต์ (Enhancement Mode) กล่าวคือ ถ้าจ่ายแรงดันลบให้กับดีมอสเฟทแชนแนล N จะทำงานในโหมดดีพลีชั่น แต่ถ้าจ่ายแรงดันบวกจะทำงานในโหมดเอนฮานซ์เมนต์ ส่วนดีมอสเฟทแชนแนล P ก็จะทำงานคล้ายกันเมื่อ ได้รับแรงดันที่มีขั้วตรงข้ามกับแบบแชนแนล N มอสเฟทประเภทเอนฮานซ์เมนต์หรืออีมอสเฟท (E-MOSFET) มีโครงสร้างบางอย่างคล้ายกับมอสเฟทแบบดีพลีชั่นแต่จะทำงานได้เฉพาะโหมดเอนฮานซ์เมนต์เท่านั้น

ประเภทของMOSFET

MOS (Negative MOSFET) เป็นทรานซิสเตอร์ประเภท NPN เมื่อมีความต่างศักย์เป็นบวก (สนามไฟฟ้าแรง) สัญญาณไฟฟ้าจึงจะไหลจาก sorce ไป drain ได้ pMOS (Positive MOSFET) เป็นทรานซิสเตอร์ประเภท PNP เมื่อมีความต่างศักย์ต่ำหรือเป็นลบ (สนามไฟฟ้าอ่อน) สัญญาณไฟฟ้าจึงจะไหลจาก sorce ไป drain ได้



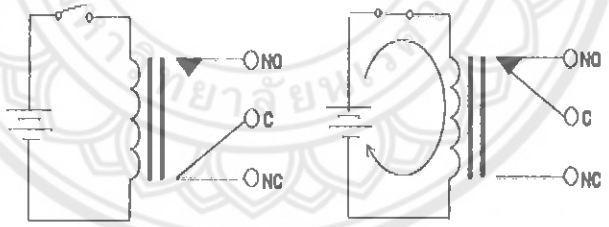
รูปที่ 2.5 โครงสร้างของ MOSFET

nMOS เมื่อปล่อยความต่างศักย์สูง จะเกิดสนามไฟฟ้าในทิศลงอย่างแรง โฮลใน p-type จะถูกผลักลงมาอยู่ด้านล่าง (ตามรูปที่ประกอบข้างบน) ประกอบกับมีอิเล็กตรอนอิสระบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีอิเล็กตรอนอิสระมากจนเป็น n-type ได้เรียกว่า channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง Channel นี้ซึ่งเป็น n-type เหมือนกับ drain และ sorce ได้โดยใช้อิเล็กตรอนอิสระเป็นพาหะ

pMOS จะทำงานกลับกับ nMOS โดยเมื่อปล่อยความต่างศักย์ต่ำ(โดยมากมักจะติดลบ) จะเกิดสนามไฟฟ้าในทิศขึ้นอย่างแรง อิเล็กตรอนอิสระใน n-type จะถูกผลักลงมาอยู่ด้านล่าง ประกอบกับมีโฮลบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีโฮลมากจนเป็น p-type ได้เรียกว่า channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง channel นี้ซึ่งเป็น p-type เหมือนกับ drain และ sorce ได้โดยใช้โฮลเป็นพาหะ

2.3.2 วงจรไดรเวอร์มอเตอร์โดยใช้รีเลย์

ไดรเวอร์มอเตอร์ เป็นอุปกรณ์ส่งพลังงานไฟฟ้าไปให้มอเตอร์เพื่อเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลและเป็นตัวควบคุมการขับเคลื่อนให้กับหุ่นยนต์ เพื่อควบคุมการทำงานของหุ่นให้เดินหน้า ถอยหลัง เลี้ยวซ้ายและเลี้ยวขวา วงจรไดรเวอร์ที่เลือกใช้ จะเป็นวงจรไดรเวอร์ด้วยรีเลย์ ซึ่งจะอธิบายในรายละเอียดดังต่อไปนี้



รูปที่ 2.6 ภาพแสดงโครงสร้างภายในของรีเลย์

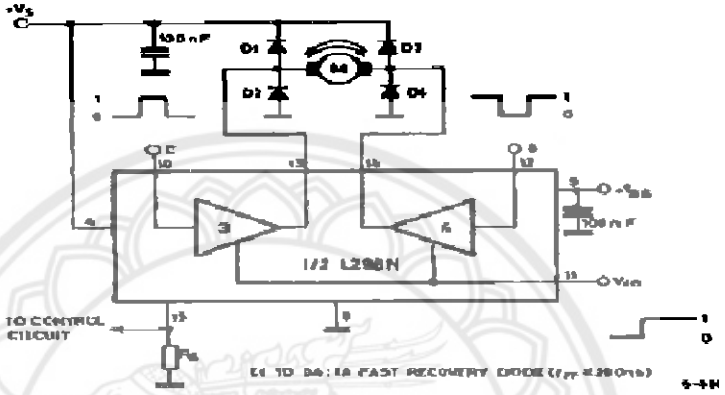
ภายในโครงสร้างของ รีเลย์ จะประกอบไปด้วยขดลวด (Coil) 1 ชุด และ หน้าสัมผัส (Contactor) ซึ่งในหน้าสัมผัส 1 ชุด จะประกอบไปด้วย 2 หน้าสัมผัสดังนี้

1. หน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC.) ซึ่งในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (Common)

2. หน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO.) ขานี้จะต่อเข้ากับขาร่วม (Common) เมื่อขดลวดมีแรงดันตกคร่อม หรือกระแสไหลผ่าน (ในปริมาณที่เพียงพอ)

ใน รีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด เช่น 2 ชุด, 4 ชุด เป็นต้น เมื่อขดลวดได้รับแรงดันตกคร่อม (ขา A และ B) จะทำให้มีกระแสไหลผ่านขดลวด ซึ่งจะทำให้เกิดอำนาจสนามแม่เหล็ก ดึงดูดให้หน้าสัมผัส NO และ C ติดกัน

วงจรไดรเวอร์ขับมอเตอร์ 2 ตัวและใช้รีเลย์ที่ใช้ไฟขับขดลวด 12 โวลต์ และเลือกใช้ IC เบอร์ L298 เป็นไอซีที่สามารถใช้ขับมอเตอร์ได้ 2 ตัวในเวลาเดียวกัน โดยลักษณะภายในจะเป็น Full-Bridge Driver โดยไอซีขับมอเตอร์ (L298) นี้ใช้หลักการ H-Bridge ในการขับมอเตอร์



รูปที่ 2.7 หลักการ H-Bridge ของไอซีเบอร์ L298

จากรูปที่ 2.7 จะเห็นว่ามีการจ่ายแรงดันให้แก่ไอซีขับมอเตอร์ (L298) อยู่ใน 2 ลักษณะ

1. จ่ายแรงดันให้กับไอซี เพื่อให้ไอซีนั้นทำงานได้แก่ Vs เป็นแรงดันจ่ายเข้ามาเพื่อใช้ในการขับมอเตอร์โดยจะค่าตั้งแต่ 2.5 ถึง 46 V และ Vss เป็นแรงดันจ่ายเข้ามาเพื่อเลี้ยงไอซีให้ทำงานโดยมีค่าตั้งแต่ 4.5 ถึง 7 V

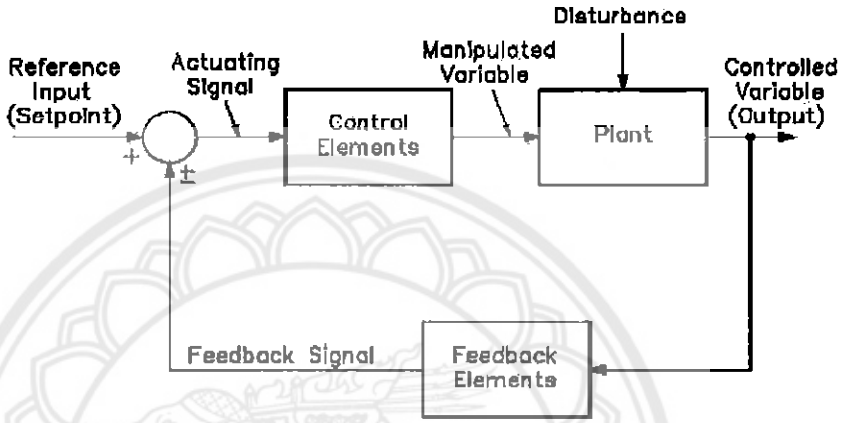
2. จ่ายแรงดันให้กับไอซีเพื่อควบคุมการทำงานของไอซี ซึ่งจะมีอยู่ 3 จุด คือจุดที่ C D และ Ven เป็นส่วนที่รับสัญญาณลอจิก "0" (Low) กับ "1" (High) เพื่อนำไปควบคุมการทำงานของมอเตอร์กระแสตรง ซึ่งจะได้ผลดังตารางนี้ (X= Don't care)

ตารางที่ 2.1 ลักษณะการหมุนของมอเตอร์เมื่อเปลี่ยนอินพุท

Inputs			**Function**
En =H	IN1 = L	IN1 = L	Fast Motor Stop
	IN1 = L	IN1 = H	Reverse
	IN1 = H	IN1 = L	Forward
	IN1 = H	IN1 = H	Fast Motor Stop
En = L	IN1 = X	IN2 = X	Free Running Motor Stop

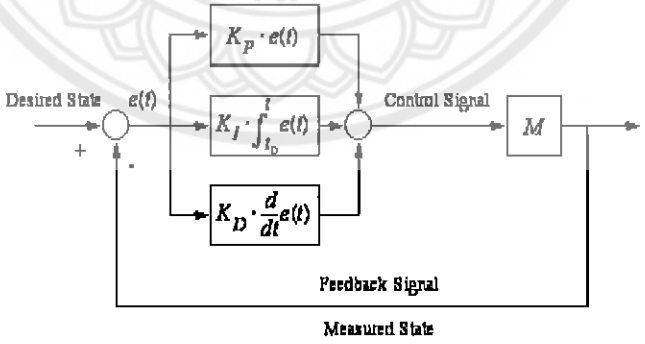
2.4 หลักการควบคุมป้อนกลับ

ระบบการควบคุมแบบป้อนกลับ เป็นระบบการควบคุมที่นิยมใช้ในงานอุตสาหกรรมต่างๆ เป็นอย่างมาก เนื่องจากคุณสมบัติที่ว่า ระบบการควบคุมแบบนี้สามารถปรับค่าการควบคุมได้ เมื่ออุปกรณ์ที่ต้องการควบคุมเปลี่ยนไปซึ่งประสิทธิภาพ ในการควบคุมก็ให้ผลดีเป็นที่น่าพอใจ อีกทั้งการควบคุมก็ทำได้ง่ายและไม่ซับซ้อนจนเกินไป



รูปที่ 2.8 หลักการทำงานแบบป้อยกลับ

โดยปกติแล้วการควบคุมแบบป้อนกลับสามารถพิจารณาโดยแบ่งการควบคุมออกเป็น 3 ส่วนดังรูป



รูปที่ 2.9 ระบบควบคุมแบบ PID

จากรูปจะเห็นว่าเราสามารถแบ่งการควบคุมแบบป้อนกลับออกเป็นสามส่วนดังสมการ

$$m(t) = K_p e(t) + K_I \int e(t)dt + \mathrm{K}_D \frac{de}{dt} \qquad (2.1)$$

ในการที่จะสร้างสมการการควบคุมแบบดิจิตอลจากระบบการควบคุมแบบอนาล็อก ทำได้โดยการแปลงสมการแบบอนาล็อกเป็นสมการแบบดิจิตอลซึ่ง ในที่นี้ได้พิจารณาขีดความสามารถของไมโครคอนโทรลเลอร์ซึ่งมีข้อจำกัดต่างๆค่อนข้างมาก ซึ่งไม่สามารถใช้การคำนวณที่ซับซ้อนได้ การควบคุมแบบ PID ในแบบของสัญญาณอนาล็อก

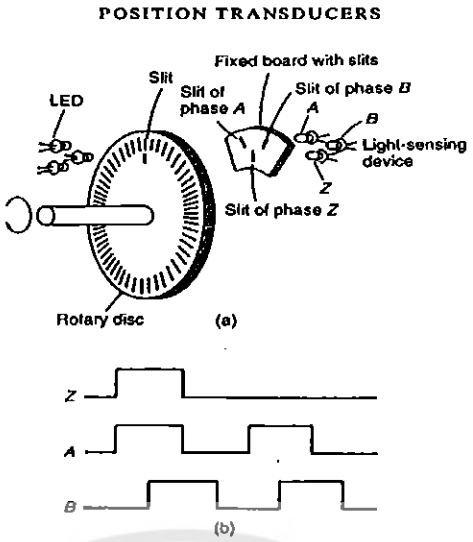
2.5 เซ็นเซอร์ Sensor

2.5.1 เอ็นโค้ดเดอร์ (Encoder)

ในโครงการนี้ได้นำเอาหลักการของ Increment Encoder เข้ามาใช้และอ่านค่าออกมาเพื่อตรวจสอบ ตำแหน่งของการเคลื่อนที่และควบคุมความเร็วของมอเตอร์เพื่อนำมาทำการเปรียบเทียบตำแหน่งการเคลื่อนที่กับค่า Input โดยหลักการทำงานของ Encoder มีดังนี้

Increment Encoder มีลักษณะเป็นแผ่นกลมมีแกนอยู่ตรงกลาง และ ที่แผ่นกลม จะมีช่องเล็ก ที่แสงสามารถส่องผ่านได้เป็นจำนวนมากซึ่งเรียกช่องนี้ว่าช่อง slit ซึ่งที่ด้านหนึ่งของแผ่นกลมนี้จะมีหลอด LED ซึ่งเป็นตัวส่งแสง infrared ไปยังตัวรับสัญญาณแสง infrared ซึ่งจะอยู่ในด้านตรงกันข้าม

โครงสร้างจะประกอบด้วยตัวกำเนิดแสง, ตัวจับแสงซึ่งถูกคั่นกลางด้วยแผ่นจานกลมๆที่มีการทำรูเจาะไว้รอบๆแผ่น (จำนวนรูจะขึ้นอยู่กับความละเอียดของ incremental encoder) และหน้ากากแยกช่องของสัญญาณพัลส์ A ,B และ Z สัญญาณพัลส์ที่ได้จากเอนโค้ดเดอร์ชนิดนี้จะประกอบด้วย 3 แทรค (tracks) คือ A,BและZ พัลส์ที่เกิดจาก แทรค A และ B จะเกิดการเหลื่อมกันมีความต่างเฟสกัน 90 องศา เพื่อทำหน้าที่รายงานผลของความเร็วและทิศทางการหมุนของมอเตอร์ให้คอนโทรลเลอร์ดังนี้กรณีพัลส์ A เกิดขึ้นก่อน B คอนโทรลเลอร์จะรับรู้ว่ามอเตอร์กำลังหมุนด้วยทิศทางตามเข็มนาฬิกา ส่วนแทรค Z หรือพัลส์อ้างอิง จะเกิดขึ้น1พัลส์ในการหมุน 1 รอบ ทำหน้าใช้อ้างอิงตำแหน่งของโรเตอร์

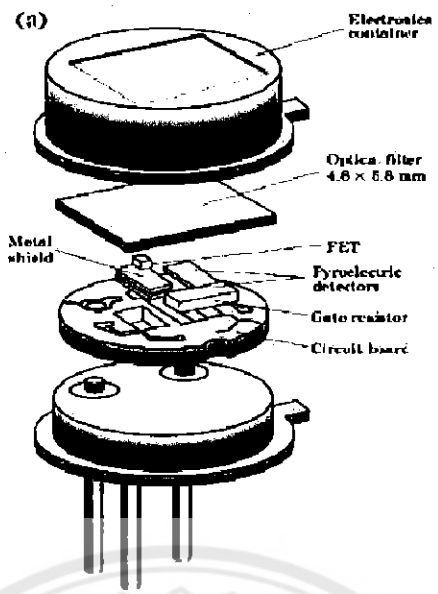


รูปที่ 2.11 ภาพแสดง Incremental Encoder

Incremental Encoder โดยทั่วไปจะไม่นิยมใช้กับระบบเซอร์โวมอเตอร์ที่มีการควบคุมตำแหน่ง เนื่องจากไม่สามารถจำตำแหน่งเดิมได้กรณีที่มีการปิดเครื่องหรือไฟดับ ซึ่งจะต้องทำการหาจุดอ้างอิงใหม่ทุกครั้ง

2.5.2 Passive Infrared Detector (PIR)

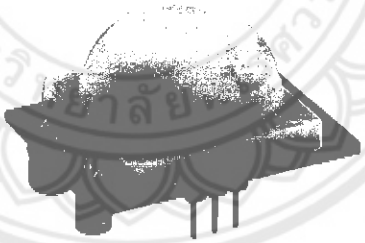
เนื่องจากวัตถุประสงค์ของโครงการนี้ นอกเหนือจากการที่หุ่นยนต์สามารถเคลื่อนที่ไปในบริเวณที่มีสภาพแวดล้อมที่เป็นอันตรายแล้ว การค้นหาผู้รอดชีวิตก็เป็นอีกวัตถุประสงค์หนึ่ง ซึ่งผู้จัดทำโครงการทำการศึกษาและเลือกใช้ PIR sensor เนื่องจากเซ็นเซอร์ดังกล่าว สามารถตรวจจับสิ่งมีชีวิตได้ โดยมีรายละเอียดดังต่อไปนี้ PIR ย่อมาจาก PIR (Passive Infrared Detector) คือตัวตรวจจับรังสีอินฟราเรดแบบหนึ่ง โดยตัวมันจะทำงานเมื่อตรวจจับพบความเปลี่ยนแปลงของรังสีอินฟราเรดที่แผ่ออกมาจากตัวคนหรือตัวสัตว์ ในขณะที่มีการเคลื่อนไหวในตัวคนหรือสัตว์จะมีรังสีความร้อนแผ่ออกมารอบๆตัวในปริมาณที่แน่นอนอยู่จำนวนหนึ่ง เมื่อเกิดการเคลื่อนไหวหรือเคลื่อนที่ก็จะทำให้อุณหภูมิในบริเวณนั้นเกิดการเปลี่ยนแปลงพลังงานความร้อนทำให้เกิดผลการเปลี่ยนแปลง ส่งผลให้คลื่นรังสีความร้อนที่ว่านี้แผ่กระจายออกมามีความยาวคลื่นประมาณ 0.74-300 ไมโครเมตร อันเป็นแถบความถี่ในย่านอินฟราเรดพอดีประสิทธิภาพในการตรวจจับแบบกลมสามารถตรวจจับได้ในระยะ 5 เมตร วงจรนี้ใช้ไฟเลี้ยงวงจร 5-15 โวลต์ดีซี กินกระแสสูงสุดประมาณ 35 มิลลิแอมแปร์



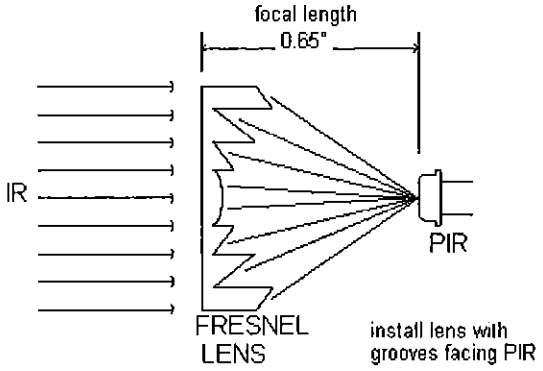
รูปที่ 2.12 ภาพแสดงตัวตรวจจับรังสีอินฟราเรด

2.5.2.1 ส่วนประกอบหลักของ Passive Infrared Detector

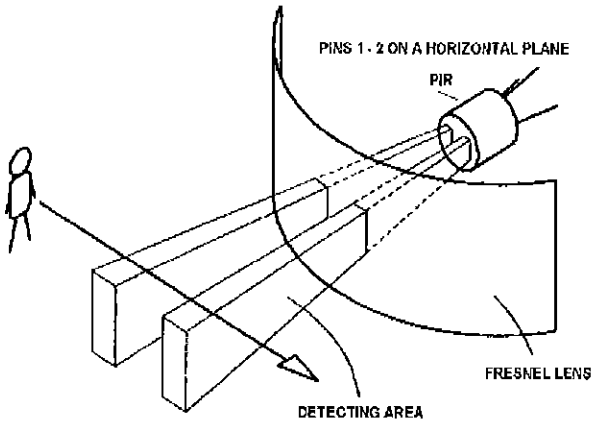
ส่วนประกอบหลักที่ทำหน้าที่ของ PIR detector คล้ายกับตาซึ่งมี Fresnel lens ทำหน้าที่ focus ความร้อนให้ไปตกที่ Pyrosensor, Pyrosensor เปรียบเสมือน จอประสาทตา, Circuit Board เปรียบเสมือนสมองที่ประมวลผลเพื่อส่งสัญญาณไป Microcontroller ของระบบ



รูปที่ 2.13 ภาพแสดง Fresnel lens



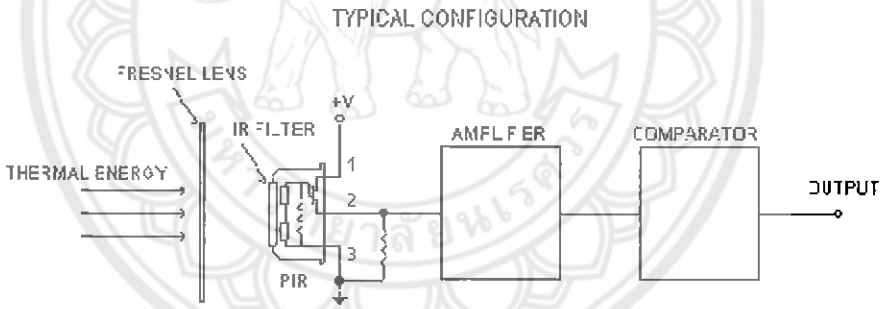
รูปที่ 2.14 ภาพแสดงการทำงานของ PIR



รูปที่ 2.15 ภาพแสดงการทำงานของ PIR เมื่อมีสิ่งมีชีวิตเดินผ่าน

2.5.2.2 การทำงานของ PIR

เมื่อมีคนหรือสัตว์เดินผ่านหน้า PIR จะทำให้ที่ขา S ของ PIR มีพัลส์ลูกเล็ก ๆ เกิดขึ้น เนื่องมาจากตัว PIR จะทำการตรวจจับการเปลี่ยนแปลงความร้อนจากการเปลี่ยนแปลงของรังสีอินฟราเรดที่แผ่ออกมาจากตัวของคนหรือสัตว์ในขณะที่มีการเคลื่อนไหว



รูปที่ 2.16 ภาพแสดง TYPICAL CONFIGURATION OF PYROSENSOR

2.5.3 Ultrasonic Sensor

การตรวจจับความเคลื่อนไหวด้วยอัลตราโซนิคนี้จะแบ่งออกได้เป็น 2 ส่วน คือ ส่วนของตัวส่ง (TX) และส่วนของตัวรับ (RX) ในระหว่างการทำงานเซ็นเซอร์จะทำการส่งสัญญาณเสียงซึ่งเรียกว่า "ซาวด์พาร์เซลส์" (Sound parcels) ให้ขบวนการทางอิเล็กทรอนิกส์ของเวลาทำงานไปเรื่อยๆจนกระทั่งมีการรับการสะท้อนครั้งแรกเกิดขึ้น โดยในส่วนของเครื่องส่งจะทำงานเป็นวงจรออสซิลเลเตอร์กำเนิดความถี่สูงในย่านอัลตราโซนิคเกิดขึ้นมา ซึ่งมีความถี่ขนาด 40 กิโลเฮิรตซ์เพื่อกระจายเสียงออกไป ในส่วนของตัวรับนั้นจะทำหน้าที่รับคลื่นที่ได้รับจากการสะท้อนคลื่น

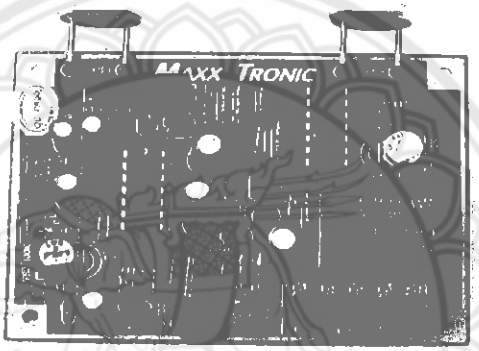
คุณสมบัติของ Ultrasonic sensor

- สามารถตรวจจับวัตถุทุกชนิด ด้วยข้อจำกัดซึ่งน้อยมาก วัตถุทุกชนิดสามารถตรวจจับได้ดีเท่ากันตราบใดที่ขนาดและมุมของการตกกระทบเป็นไปตามที่กำหนด

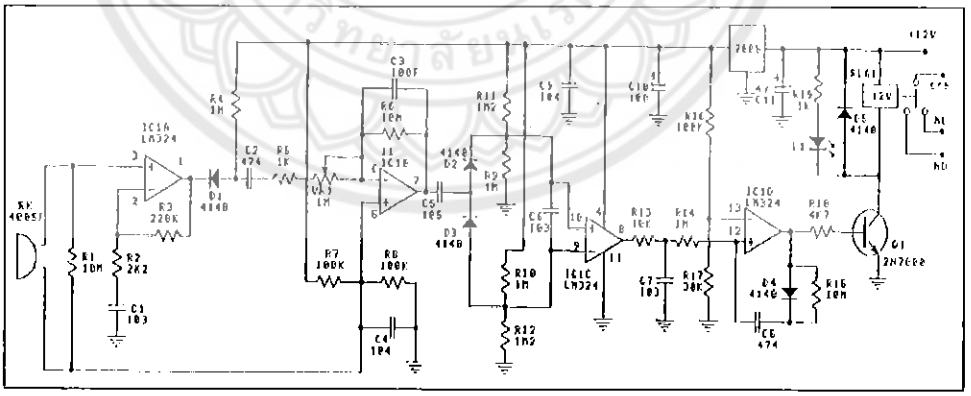
- ไม่มีค่าความผิดพลาดที่ต้องแก้ไขสามารถใช้ได้กับทุกสี อย่างไม่มีปัญหาพื้นที่ผิวของวัตถุที่ตรวจจับไม่มีอิทธิพลต่อการวัด

- ทำงานได้ดีแม้ในสภาวะ หมอก ฝุ่น ควัน หรือในที่ซึ่งมีแสงน้อย ๆ

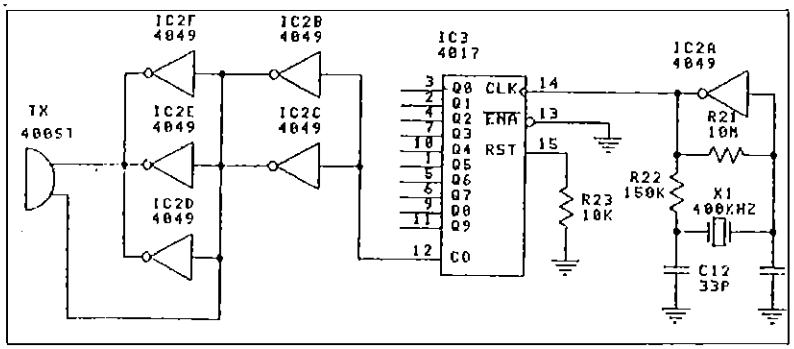
- อัลตราโซนิคดีมากในการตรวจจับวัตถุโปร่งแสงและเป็นมันวาว เช่นกระจก ขวด แผ่นฟอยล์ต่าง ๆ เป็นต้น



รูปที่ 2.17 ภาพแสดง Ultrasonic Sensor



รูปที่ 2.18 ภาพแสดงวงจรชุดรับของ Ultrasonic Sensor



รูปที่ 2.19 ภาพแสดงวงจรชุดส่งของ Ultrasonic

2.5.4 อุณหภูมิแบบอินฟราเรด (แบบไม่สัมผัส) INFRARED THERMOMETER (NON-CONTACT THERMOMETER)

โดยใช้หลักการของการยิงแสงอินฟราเรดระยะไกล ใช้เพื่อวัดค่าอุณหภูมิของวัตถุ และค่าที่อ่านได้ก็เป็นค่าอุณหภูมิที่แท้จริงของวัตถุ โดยไม่มีผลกระทบจากสภาพแวดล้อมรอบข้าง เพียงแค่เล็งไปที่วัตถุก็สามารถอ่านค่าอุณหภูมิบนขอ LCD ได้ในเวลาไม่กี่วินาที ใช้งานกับ วัตถุที่มีอุณหภูมิสูงหรือวัตถุที่ไม่สามารถเข้าไปใกล้ๆได้ ซึ่งเมื่อเปรียบเทียบกับ การวัดอุณหภูมิแบบสัมผัสแล้ว จะเห็นว่าเวลาที่ใช้ในการวัดค่า แต่ละจุดนั้นใช้หลายนาที ทีเดียว

หลักการทำงาน

เนื่องจากการวัดอุณหภูมิเป็นแบบไม่สัมผัส การที่เราจะทราบวัตถุนั้นมีอุณหภูมิเท่าไร สามารถทำได้โดยอาศัยหลักการทั้ง 3 คือ การนำ การพา การแผ่รังสี ซึ่งเครื่องมือวัดอุณหภูมิแบบอินฟราเรดนั้นจะใช้หลักการการแผ่รังสี โดยเครื่องมือจะทำหน้าที่รับรังสีความร้อนที่แผ่ออกมาจากวัตถุ ซึ่งมาจากการที่วัตถุที่มีอุณหภูมิในตัวมันจะสามารถเปล่งรังสีอินฟราเรดออกมา ถ้าวัตถุนั้นเปล่งรังสีอินฟราเรดที่มีความเข้มมาก แสดงว่าวัตถุนั้นมีอุณหภูมิสูง และถ้าวัตถุนั้น เปล่งรังสีอินฟราเรดออกมามีความเข้มน้อยก็แสดงว่า วัตถุนั้นมีอุณหภูมิต่ำนั่นเอง

องค์ประกอบที่มีผลต่อค่าอุณหภูมิ

Emissivity : วัตถุทุกชนิดมีการสะท้อนแสงมาก - น้อยต่างกัน เราจึงต้องปรับค่า ๆ หนึ่งที่เครื่องมือวัด เพื่อชดเชยการสะท้อนของวัตถุนั้น แต่บางรุ่นจะตั้งค่าคงที่ไว้ที่ 0.95 ซึ่งเป็นค่าที่ใช้กับวัตถุส่วนใหญ่ แต่เครื่องมือวัดแบบปรับค่าได้ จะมีความถูกต้องแม่นยำกว่า

Distance to spot size : เป็นการกำหนดระยะทางในการติดตั้งเพื่อให้ได้พื้นที่ที่ต้องการเฉลี่ยค่าอุณหภูมิมาแสดงผล โดยกำหนดจากสูตร d = D/F โดย d แทนขนาดพื้นที่ ที่ต้องการเฉลี่ยค่าอุณหภูมิ D แทนระยะจากหน้าเลนส์ถึงวัตถุ และ F แทนระยะโฟกัสของเครื่องมือวัดนั้น ๆ เช่น

เราใช้เครื่องมือวัดที่ระยะโฟกัสเท่ากับ 100 และติดตั้งเครื่องมือวัดให้ห่างจากวัตถุ 200 cm. พื้นที่เฉลี่ยออกมาเป็นค่าอุณหภูมิมีพื้นที่เท่ากับ 200/100 เท่ากับ 2 ตร.ซม. นั่นเอง

Field of View : การวัดต้องแน่ใจว่าตำแหน่งที่วัดไม่ไกลเกินไป จนทำให้วงที่วัดใหญ่กว่าวัตถุ การวัดที่ดี พื้นที่ที่คำนวณออกมาต้องเล็กกว่าพื้นที่ที่เราต้องการวัด หากวัตถุเล็กควรวัดให้ใกล้หรือเลือกเครื่องมือที่มีระยะโฟกัสสูง ๆ

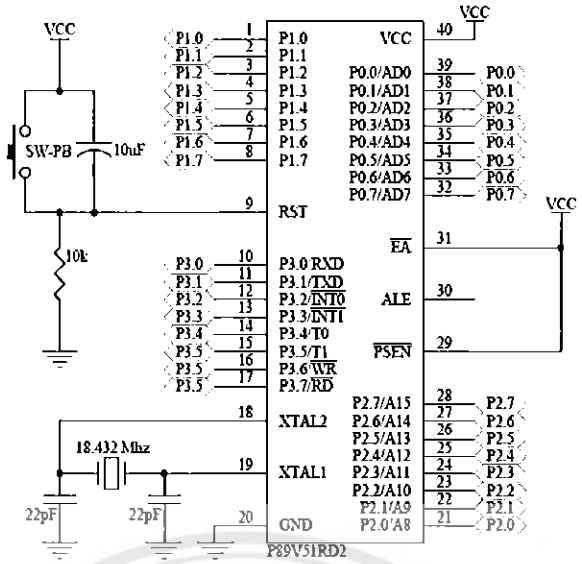
ตัวอย่างรูปร่างหน้าตาของ Sensor อุณหภูมิแบบอินฟราเรด (แบบไม่สัมผัส)

รูปที่ 2.20 ตัววัดอุณหภูมิแบบอินฟราเรด

2.6 ไมโครคอนโทรลเลอร์ MCS-51 (เบอร์ P89V51RD2)

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เริ่มแรกได้ถูกพัฒนาขึ้นจากบริษัท อินเทล (Intel Corporation) และได้มีการนำไปใช้งานกันอย่างแพร่หลายตั้งแต่ปี 1980 ในช่วงเวลาที่ผ่านมาได้มีบริษัทผู้ผลิตหลายบริษัท เช่น Dallas, Philips, Atmel ได้รับลิขสิทธิ์ในการผลิต และจำหน่าย จากบริษัท อินเทล และบริษัทต่าง ๆ ก็ได้พัฒนาความสามารถของไมโครคอนโทรลเลอร์ MCS-51 รุ่นใหม่ ๆ ให้มีความสามารถ และมีความเร็วเพิ่มขึ้น แต่ยังคงโครงสร้างพื้นฐานที่สำคัญของไมโครคอนโทรลเลอร์ตระกูล 8051 ซึ่งมีรายละเอียดดังนี้ [3]

1. เป็นไมโครคอนโทรลเลอร์ที่มีหน่วยประมวลผลกลางแบบ 8 บิต
2. มีคำสั่งคำนวณทางคณิตศาสตร์ และตรรกศาสตร์ (Boolean processor)
3. มีแอดเดรสบัสขนาด 16 บิตทำให้สามารถอ้างตำแหน่งหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลได้ 64 กิโลไบต์
4. มีหน่วยความจำ (RAM) ภายในขนาด 128 ไบต์ (8051/8031) หรือ 256 ไบต์ (8052/8032)
5. มีพอร์ตอนุกรมทำงานแบบดูเพล็กซ์เต็ม (Full Duplex) 1 พอร์ต
6. มีพอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 บิต
7. มีไทเมอร์ 2 ตัว (8051/8031) หรือ 3 ตัว (8052/8032)
8. มีวงจรควบคุมการเกิดอินเตอร์รัปต์ 5 ประเภท (8051/8031) หรือ 6 ประเภท (8052/8032)



รูปที่ 2.21 วงจรไมโครคอนโทรลเลอร์ P89V51RD2

ไมโครคอนโทรลเลอร์ MCS-51 มีวงจรออสซิลเลเตอร์อยู่ภายใน ดังนั้นในการใช้งานจึงสามารถต่อคริสตอล และตัวเก็บประจุเข้ากับคริสตอลได้โดยตรง โดยความถี่ของคริสตอลที่ต่อเข้ากับไมโครคอนโทรลเลอร์จะเป็นตัวระบุความเร็วในการทำงานโดยตรง ในไมโครคอนโทรลเลอร์ MCS-51 ปกติ 1 แมชชีนไซเคิล (Machine Cycle) จะใช้สัญญาณนาฬิกาจำนวน 12 ลูก และในการทำงานแต่ละคำสั่งไมโครคอนโทรลเลอร์จะใช้เวลาในการทำงาน 1 - 4 แมชชีนไซเคิล ขึ้นอยู่กับความซับซ้อนของคำสั่งนั้น

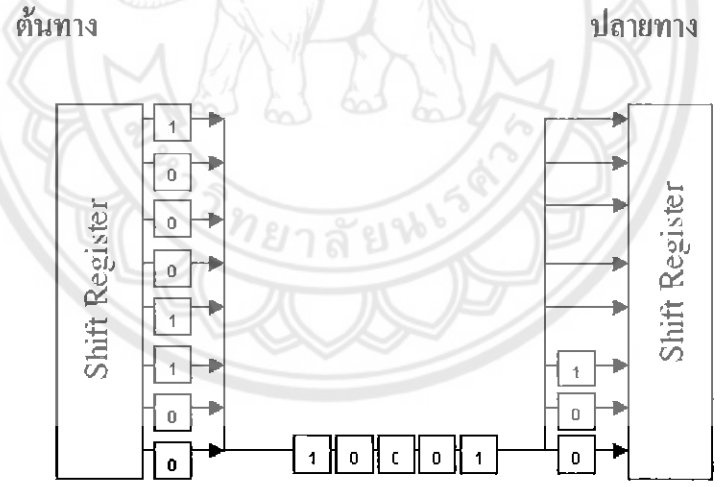
ในปัจจุบันผู้ผลิตได้พัฒนาให้ไมโครคอนโทรลเลอร์ สามารถทำงานได้เร็วขึ้นโดยเพิ่มความสามารถในการรองรับคริสตอลความถี่ที่สูงขึ้น รวมไปถึงการปรับปรุงการทำงานภายในให้ไมโครคอนโทรลเลอร์ใช้จำนวนสัญญาณนาฬิกาในการสร้างแมชชีนไซเคิลน้อยลง โดยในบางรุ่น 1 แมชชีนไซเคิลใช้สัญญาณนาฬิกาเพียงแค่ 1 ลูกเท่านั้น

สำหรับไมโครคอนโทรลเลอร์ MCS-51 ที่เราจะมาลองเล่นกันนั้นเป็นรุ่น P89V51RD2 ของบริษัท Philips ที่เลือกรุ่นนี้เนื่องจากเป็นรุ่นที่สามารถรองรับการดาวน์โหลดโปรแกรมแบบ ISP (In System Programming) ผ่านพอร์ตอนุกรมได้โดยตรง ไม่ต้องอาศัยอุปกรณ์ หรือวงจรเพิ่มเติมในการดาวน์โหลดโปรแกรม จึงทำให้สามารถใช้งานได้อย่างสะดวก รวมถึงราคาของ P89V51RD2 ที่ไม่แพง เมื่อเทียบกับความสามารถ และประสิทธิภาพของมัน P89V51RD2 สามารถทำงานในโหมด X2 ซึ่ง จะทำให้สามารถทำงานได้เร็วกว่า MCS-51 พื้นฐาน 2 เท่า (1 แมชชีนไซเคิล ใช้สัญญาณนาฬิกา 6 ลูก) เมื่อใช้คริสตอลความถี่ที่เท่ากัน ในการทำงานในโหมด X2 นี้ P89V51RD2 สามารถใช้คริสตอลความถี่สูงสุด 20 MHz ส่วนในการทำงานในโหมด X1 สามารถใช้คริสตอลความถี่สูงสุด 40 MHz ภายใน P89V51RD2 มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 64 กิโลไบต์ นอกจากนั้นยังมี หน่วยความจำข้อมูลภายนอกเพิ่มเติมขนาด 1 กิโลไบต์ อยู่ภายในตัวชิพด้วย

2.7 การสื่อสารแบบอนุกรม

การส่งข้อมูลแบบขนานนั้น สายส่งข้อมูลแบบขนานที่ใช้สำหรับเชื่อมต่อระหว่างพอร์ต I/O กับอุปกรณ์ภายนอกนอกจากจะมีความยาวได้เพียง 1 หรือ 2 เมตรเท่านั้น ทั้งนี้เนื่องจากค่าคาพาซิแดนซ์ในสายจะกำจัดระยะทางในการส่งข้อมูล แต่ถ้าเราต้องการให้สามารถส่งข้อมูลได้ในระยะทางไกลขึ้น เราก็ต้องนำวงจรขับพิเศษมาใช้ การส่งข้อมูลแบบขนานจะต้องส่งสัญญาณจำนวน 1 เส้นสำหรับข้อมูลในแต่ละบิต ซึ่งทำให้การโอนย้ายข้อมูลแบบขนาน 1 ไบต์มีราคาสูงกว่าการโอนย้ายข้อมูลแบบอนุกรมถึง 8 เท่า เช่นเดียวกันด้วยเหตุผลด้านราคาและความไม่สะดวกที่พบในการโอนย้ายข้อมูลแบบขนาน จึงได้มีอุปกรณ์หลายชนิดที่ใช้ในการสื่อสารแบบอนุกรม ทั้งๆที่ไมโครโปรเซสเซอร์ ในเครื่องไมโครคอมพิวเตอร์จะยังคงใช้การโอนย้ายข้อมูลแบบขนาน

เมื่อเราศึกษาการโอนย้ายข้อมูลแบบอนุกรม เราก็จะต้องเรียนรู้โปรโตคอลที่เกี่ยวกับการส่งข้อมูลแบบอนุกรม ซึ่งแบ่งออกได้เป็น 3 อย่าง อย่างแรกคือ วิธีการแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม และการแปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนาน อย่างที่สองคือ ชนิดของวงจรและรูปแบบสัญญาณที่ใช้ในการส่งข้อมูลในระยะไกล อย่างที่สามคือรูปแบบของข้อมูลที่ส่งไปและการควบคุมการโอนย้ายข้อมูล [4]

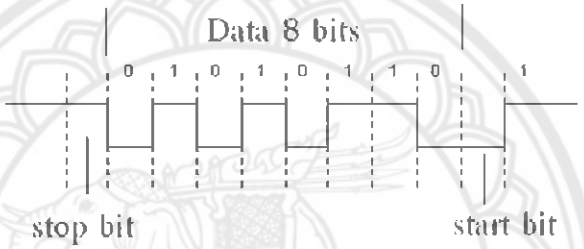


รูปที่ 2.22 รูปแบบการส่งข้อมูลแบบอนุกรม

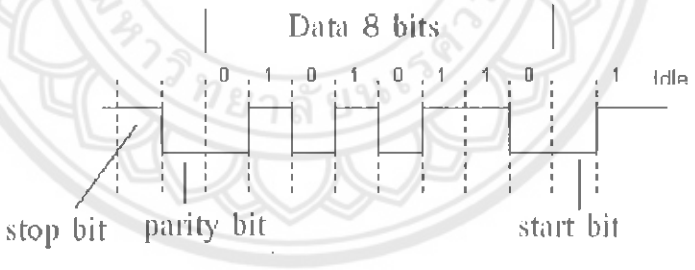
2.7.1 การเชื่อมต่อแบบอนุกรม UART

การแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยเริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ในรีจีสเตอร์ที่เลื่อนค่าได้ (Shift register) จากนั้นเราจะใช้สัญญาณนาฬิกาในการเลื่อนค่าในรีจีสเตอร์ออกมาทีละบิต (โดยจะเลื่อนค่าไปทางขวามือ) โดยบิตแรกที่ถูกเลื่อนออกมาคือ บิต LSB ของข้อมูลและบิตที่สองที่ถูกเลื่อนออกมาก็คือ บิตที่อยู่ถัดไปจากบิต LSB และบิตต่อๆไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาก็คือ บิต MSB ของข้อมูล เมื่อเรานำบิตที่ 8 ของข้อมูลมาใช้ใน

การตรวจสอบความผิดผลาดในการสื่อสารข้อมูลซึ่งเราเรียกบิตนี้ว่า บิตพาริตี้ (Parity bit) UART ส่วนใหญ่สามารถสร้างและทำการตรวจสอบข้อมูลนั้นว่าเป็นพาริตี้คู่หรือเป็นพาริตี้คี่ได้ ในการสร้างพาริตี้คู่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตี้เพื่อให้ข้อมูลทั้ง 8 บิตนั้นมีเลข 1 จำนวนคู่ตัว และ ในการสร้างพาริตี้คี่ UART จะทำการเซตหรือเคลียร์ค่าในพาริตี้เพื่อให้ข้อมูลทั้ง 8 บิตนั้น มีเลข 1 จำนวนคี่ตัว การส่งข้อมูลของ UART จะเป็นแบบอะซิงโครนัส ซึ่งก็หมายความว่าเวลาระหว่างเวิร์ดความว่าอัตราการส่งข้อมูลของ UART จะไม่ขึ้นกับจังหวะการทำงานของไมโครโปรเซสเซอร์โดยไมโครโปรเซสเซอร์และ UART จะมีวงจรสร้างสัญญาณนาฬิกาของมันเอง แต่ถ้าเราพบว่าไมโครโปรเซสเซอร์และ UART ทำงานร่วมกันอย่างเข้าจังหวะแต่การทำเช่นก็เพื่อเป็นการลดส่วนของวงจรฮาร์ดแวร์ที่ใช้สร้างสัญญาณนาฬิกา



รูปที่ 2.23 การสื่อสารแบบอะซิงโคนัสที่ไม่มีพาริตี้บิต



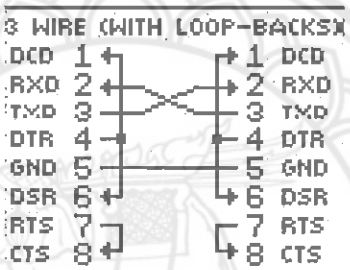
รูปที่ 2.24 การสื่อสารแบบอะซิงโคนัสที่มีพาริตี้บิต

การเชื่อมต่อระหว่างพอร์ตอนุกรม โดยทั่วไปเราจะพบเส้นส่งสัญญาณอนุกรมแบบมาตรฐาน EIA RS-232 มากที่สุดซึ่งเราจะเรียกว่า RS-232 สายส่งสัญญาณ RS-232 นี้ได้ถูกนำไปใช้ในหน่วยแสดงผล เครื่องพิมพ์โมเด็ม และอุปกรณ์อื่น ๆ ซึ่งจะมีความยาวของสายไม่เกิน 50 ฟุต

มาตรฐาน RS-232 ได้กำหนดให้ค่าสัญญาณไฟฟ้าที่มีระดับแรงดันไฟฟ้าเท่ากับ 3 โวลต์หรือสูงกว่า ที่มีค่าทางตรรกะเป็น 1 และกำหนดค่าสัญญาณไฟฟ้าที่มีระดับแรงดันเท่ากับ -3 โวลต์หรือต่ำกว่า มีค่าทางตรรกะเป็น 0 วงจรไอซีที่สร้างสัญญาณเหล่านี้ต้องการแหล่งจ่ายไฟขนาด +12 โวลต์ RS-232 จะใช้สาย 1 เส้นสำหรับส่งข้อมูลและใช้สายอีก 1 เส้นสำหรับข้อมูล โดยสัญญาณใน

แต่ละสายนี้จะถูกอ้างอิงกับกราวนด์ (ขาเบอร์ 7) มาตรฐาน RS-232 นี้ยังได้กำหนดสัญญาณตอบรับเพื่อใช้ในการควบคุมการรับ/ส่งข้อมูลด้วย

รูปที่ 2.25 Serial Port (Com Port) ใช้ในการเชื่อมต่อการส่งสัญญาณ



รูปที่ 2.26 การต่อสายสัญญาณตามมาตรฐาน RS-232

2.8 เครือข่ายไร้สาย Wireless LAN

การใช้งานเครือข่ายไร้สายมีอัตราการเติบโตเพิ่มสูงขึ้นอย่างรวดเร็ว นับตั้งแต่มาตรฐาน IEEE 802.11 เกิดขึ้น เครือข่ายไร้สายก็ได้รับการปรับปรุงและพัฒนามาอย่างต่อเนื่อง จนกระทั่งปัจจุบันเครือข่ายไร้สายสามารถใช้งานได้ด้วยความสะดวก และมีความปลอดภัยสูงขึ้นมาก นอกจากนั้นก็ยังให้อัตราความเร็วของการสื่อสารที่เพิ่มสูงขึ้น จนสามารถตอบรับกับการใช้งานในด้านต่างๆ ได้อย่างดี ไม่ว่าจะเป็นการใช้งานอินเทอร์เน็ตความเร็วสูง การใช้งานวิดีโอสตรีมมิงมัลติเมียและการใช้งานด้านความบันเทิงต่างๆ สำหรับการประยุกต์ใช้งานเครือข่ายไร้สายนับว่ามีอย่างหลากหลาย ซึ่งพอจะยกตัวอย่างได้ต่อไปนี้ ผู้ใช้งานตามบ้านเรือนที่พัก สามารถนำระบบเครือข่ายไร้สายมาใช้งานทั้งการแชร์การใช้งาน Internet ร่วมกับสมาชิกภายในครอบครัว และยังสามารถนำไปเล่นได้จากจุดใดก็ได้ที่อยู่ในบ้านได้สะดวกและรวดเร็วโดยไม่ต้องใช้สาย LAN ในการเดินสายนำสัญญาณ เป็นต้น

2.8.1 มาตรฐานเครือข่ายไร้สาย IEEE 802.11

เครือข่ายไร้สายมาตรฐาน IEEE 802.11 ได้รับการตีพิมพ์เผยแพร่ครั้งแรกเมื่อปี พ.ศ. 2540 โดยสถาบัน IEEE (The Institute of Electronics and Electrical Engineers) ซึ่งมีข้อกำหนดระบุไว้ว่าผลิตภัณฑ์เครือข่ายไร้สายในส่วนของ PHY Layer นั้นมีความสามารถในการรับส่งข้อมูลที่ความเร็ว 1, 2, 5.5, 11 และ 54 เมกะบิตต่อวินาที โดยมีสื่อนำสัญญาณ 3 ประเภทให้เลือกใช้งานอันได้แก่ คลื่นวิทยุย่านความถี่ 2.4 กิกะเฮิรตซ์, 2.5 กิกะเฮิรตซ์และคลื่นอินฟราเรด ส่วน.ในระดับชั้น MAC Layer นั้นได้กำหนดกลไกของการทำงานแบบ CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ซึ่งมีความคล้ายคลึงกับ CSMA/CD (Collision Detection) ของมาตรฐาน IEEE 802.3 Ethernet ซึ่งนิยมใช้งานบนระบบเครือข่ายแลนใช้สาย โดยมีกลไกในการเข้ารหัสข้อมูลก่อนแพร่กระจายสัญญาณไปบนอากาศ พร้อมกับมีการตรวจสอบผู้ใช้งานอีกด้วย

มาตรฐาน IEEE 802.11 ในยุคเริ่มแรกนั้นให้ประสิทธิภาพการทำงานที่ค่อนข้างต่ำ ทั้งไม่มีการรับรองคุณภาพของการให้บริการที่เรียกว่า QoS (Quality of Service) ซึ่งมีความสำคัญในสภาพแวดล้อมที่มีแอพพลิเคชันหลากหลายประเภทให้ใช้งาน นอกจากนั้นกลไกในเรื่องการรักษาความปลอดภัยที่นำมาใช้ก็ยังมีช่องโหว่จำนวนมาก IEEE จึงได้จัดตั้งคณะทำงานขึ้นมาหลายชุดด้วยกัน เพื่อทำการพัฒนาและปรับปรุงมาตรฐานให้มีศักยภาพเพิ่มสูงขึ้น

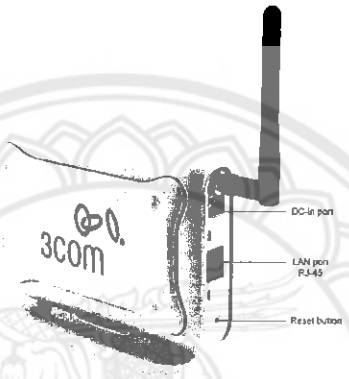
ตารางที่ 2.2 Wireless Networking Standards

802.11 Protocol	Release	Freq. (GHz)	Thru. (Mbit/s)	Data (Mbit/s)	Mod.	Range (Indoor). (m)	Range (Outdoor). (m)
–	1997	2.4	00.9	002		~20	~100
a	1999	5	23	054	OFDM	~35	~120
b	1999	2.4	04.3	011	DSSS	~38	~140
g	2003	2.4	19	054	OFDM	~38	~140
n	2009	2.4, 5	74	248		~70	~250
y	2008	3.7	23	054		~50	~5000

2.8.2 มาตรฐานที่ใช้ในการติดต่อสื่อสารที่ใช้ IEEE 802.11g

เป็นมาตรฐานที่นิยมใช้งานกันมากในปัจจุบัน และได้เข้ามาทดแทนผลิตภัณฑ์ที่รองรับมาตรฐาน IEEE 802.11b เนื่องจากสนับสนุนอัตราความเร็วของการรับส่งข้อมูลในระดับ 54 เมกะบิตต่อวินาที โดยใช้เทคโนโลยี OFDM บนคลื่นสัญญาณวิทยุย่านความถี่ 2.4 กิกะเฮิรตซ์ และให้

รัศมีการทำงานที่มากกว่า IEEE 802.11a พร้อมความสามารถในการใช้งานร่วมกันกับมาตรฐาน IEEE 802.11b ได้ (Backward-Compatible) ดังนั้น ในระบบอุปกรณ์ไร้สาย และ กล้องวงจรปิดไร้สาย ต่างๆ ที่ใช้มาตรฐาน IEEE 802.11b เดิมอยู่ ก็สามารถมาใช้งานร่วมกับอุปกรณ์ที่ใช้ มาตรฐาน IEEE 802.11g ได้ โดย เพียงแต่อัพเกรดเฟิร์มแวร์เท่านั้น จึงทำให้ไม่จำเป็นที่จะต้องเปลี่ยนแปลงโครงสร้างพื้นฐานของระบบ หรือจะใช้งานกับ Access Point เลยก็ทำได้เช่นเดียวกัน โดย IEEE 802.11g ใช้เทคนิคการเข้ารหัสข้อมูลได้ทั้งแบบ CKK ตามมาตรฐาน IEEE 802.11b หรือจะใช้ OFDM ตามมาตรฐาน IEEE 802.11a ก็ได้



รูปที่ 2.27 อุปกรณ์ Wireless Router Access Point

เป็นอุปกรณ์ที่ใช้เป็นตัวกลาง ในการรับส่งข้อมูลแบบกระจายทางคลื่นความถี่ กับ Wireless Card ซึ่งติดตั้งบนเครื่องของผู้ใช้แต่ละคนให้สามารถติดต่อสื่อสารกันได้ ลักษณะการทำงานจะเป็นเช่นเดียวกับ Hub ที่ใช้กับระบบเครือข่ายใช้สาย โดย Access Point จะมีพอร์ต RJ-45 สำหรับใช้เพื่อเชื่อมโยงเข้ากับเครือข่ายใช้สายที่ใช้งานกันอยู่

ในบทต่อไปจะเป็นการออกแบบและชุดควบคุมการทำงานของหุ่นยนต์ ซึ่งจะนำทฤษฎีในบทนี้ไปประยุกต์ใช้ในการออกแบบและสร้างหุ่นยนต์

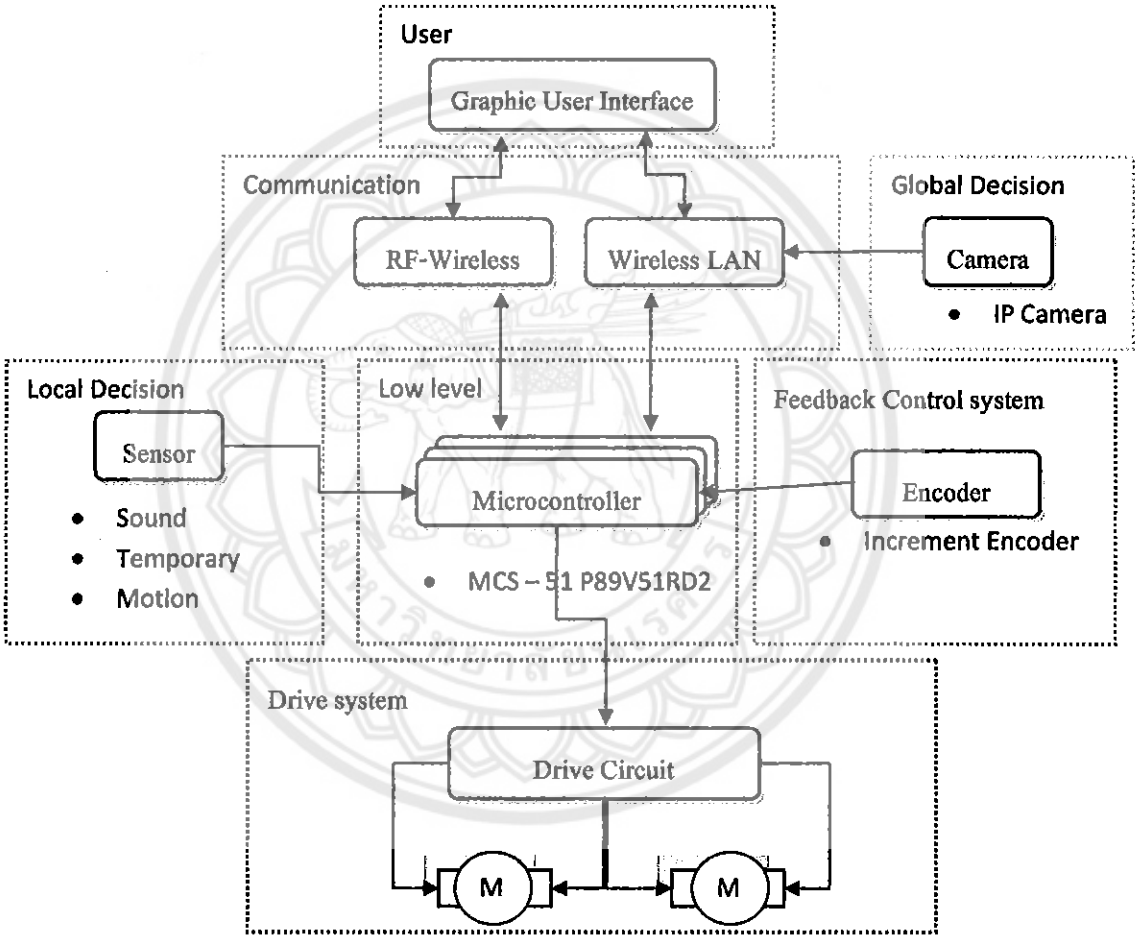
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

บทที่ 3

การออกแบบและชุดควบคุมการทำงาน

3.1 หลักการทำงาน

ในการควบคุมการทำงานของหุ่นยนต์กู้ภัยต้องอาศัยการทำงานหลาย ๆ อย่างเพื่อควบคุมหุ่นยนต์โดยจะแบ่งออกเป็น 2 ส่วนคือ ชุดไมโครคอนโทรลเลอร์ที่ควบคุมการทำงานของเซนเซอร์ต่างๆ และมอเตอร์ ชุดสั่งงานควบคุมการทำงานทั้งหมด



รูปที่ 3.1 หลักการทำงานของชุดควบคุมหุ่นยนต์

จากรูปที่ 3.1 มีการทำงานหลาย ๆ ส่วนประกอบกันโดยจะมีไมโครคอนโทรลเลอร์เป็นตัวจัดการการทำงานของเซนเซอร์ และมอเตอร์ที่ใช้ในการเคลื่อนที่ โดยไมโครคอนโทรลเลอร์จะถูกสั่งงานจากผู้ควบคุมที่ในโปรแกรมในการควบคุมการทำงาน โดยผ่านการติดต่อสื่อสารกันแบบไร้

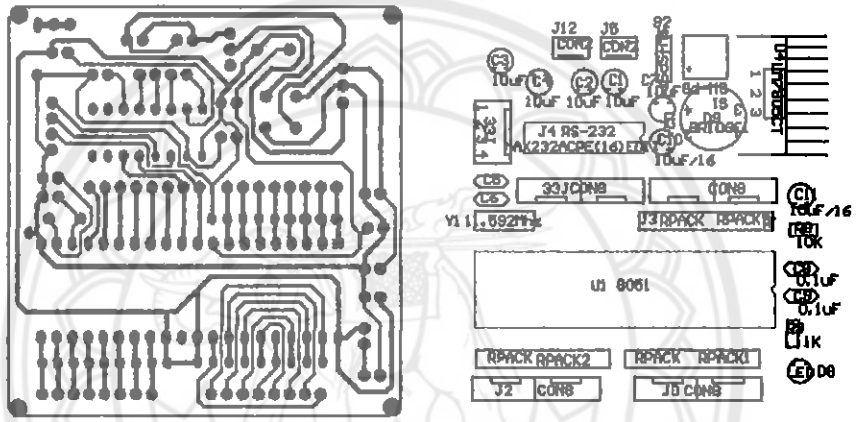
15756856
มร.
26154
2552

สายที่เรียกว่า Wireless LAN ซึ่งเป็นมาตรฐานเครือข่ายไร้สาย และมีการรับภาพจากกล้องส่งผ่านเครือข่ายไปที่ผู้ควบคุม

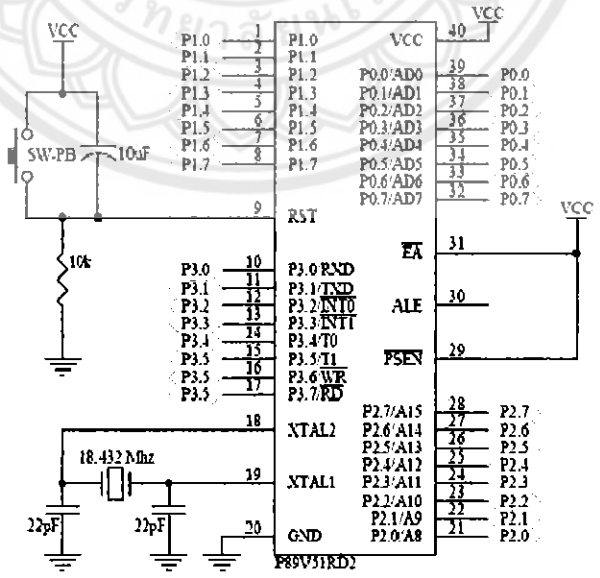
3.2 การออกแบบชุดควบคุมหุ่นยนต์

3.2.1 การออกแบบชุดควบคุมการทำงานโดยไมโครคอนโทรลเลอร์

บอร์ดควบคุมโดยไมโครคอนโทรลเลอร์จะเป็นการ ทำหน้าที่ในการรับคำสั่งจากการจากควบคุมจากเครื่องคอมพิวเตอร์ เพื่อไปสั่งงานควบคุมตัวขับมอเตอร์ โดยจะใช้มอเตอร์ทั้งหมด 6 ตัว ซึ่งชุดควบคุมมอเตอร์สามารถที่จะควบคุมมอเตอร์ได้ 2 ตัวต่อหนึ่งชุดควบคุม โดยชุดควบคุมมอเตอร์จะถูกสั่งงานจากพอร์ทไมโครคอนโทรลเลอร์ 1 พอร์ทต่อ 1 ชุดควบคุม

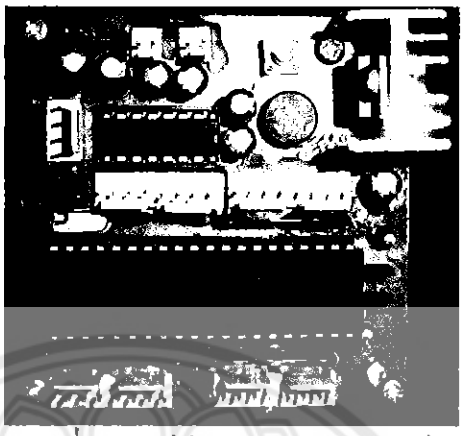


รูปที่ 3.2 การออกแบบวงจรไมโครคอนโทรลเลอร์ (Microcontroller)



รูปที่ 3.3 วงจรไมโครคอนโทรลเลอร์แสดงพอร์ทต่างๆ

จากรูปไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 จะมีพอร์ทอยู่ 4 พอร์ท ซึ่งพอร์ทที่เราใช้ในการควบคุมมอเตอร์คือพอร์ท 0 และพอร์ท 2 ซึ่งเป็นพอร์ทเอาต์พุตของไมโครคอนโทรลเลอร์โดยจะต่อกับชุดควบคุมมอเตอร์

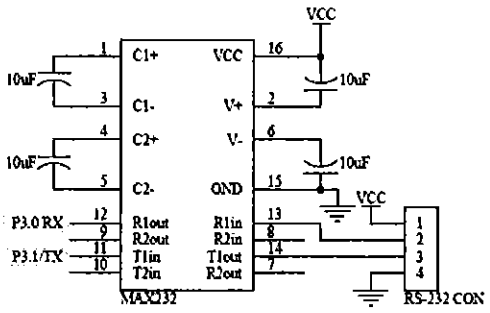


รูปที่ 3.4 บอร์ดไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ จะเป็นตัวควบคุมการทำงานทั้งหมดของหุ่นยนต์ โดยที่ไมโครคอนโทรลเลอร์จะทำงานก็ต่อเมื่อ ได้รับคำสั่งและนำมาประมวลผลเพื่อที่จะสั่งงานให้ชุดขับเคลื่อนหรือเซ็นเซอร์ทำงาน

3.2.2 การติดต่อพอร์ทอนุกรม

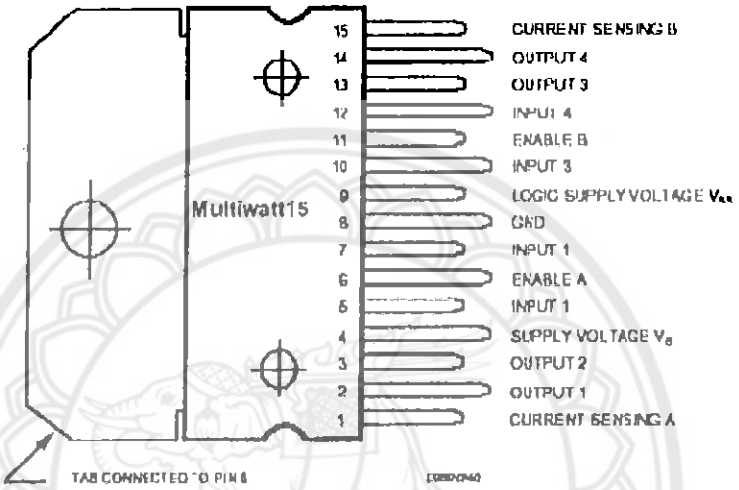
ดังที่ได้กล่าวไปแล้วว่าระบบ RS-232 (Computer Port) จะใช้ระดับแรงดันไฟฟ้าต่างกับระบบ TTL ดังนั้นในการนำข้อมูลจากคอมพิวเตอร์ไปประมวลผลในไมโครคอนโทรลเลอร์จะต้องมีการแปลงระดับสัญญาณซึ่งในการแปลงระดับสัญญาณได้ใช้ไอซี MAX 232 ซึ่งไอซีจะรับสัญญาณ RS-232 และแปลงเป็น TTL และในขณะเดียวกันก็สามารถแปลงระบบ TTL เป็น RS-232 ได้เช่นกัน การจัดการขาและการใช้งานไอซีแสดงดังรูป



รูปที่ 3.5 วงจรติดต่อผ่านพอร์ทสื่อสารอนุกรม RS-232

3.2.3 ชุดควบคุมมอเตอร์แบบ H-Bridge

การเปลี่ยนทิศทางการหมุนของดีซีมอเตอร์แบบแม่เหล็กถาวร ทำได้โดยการสลับขั้วแหล่งจ่ายไฟฟ้าที่ต่อเข้ากับตัวมอเตอร์และในทางปฏิบัติจะใช้เป็นวงจรอิเล็กทรอนิกส์ที่ เรียกว่า H-Bridge เป็นตัวจัดการการทำงาน ซึ่งในปัจจุบันนี้ได้มีการผลิต IC ขับมอเตอร์แบบ H-Bride ขึ้นมามากมาย ในส่วนของโครงงานนี้ได้ศึกษา DUAL FULL-BRIDGE DRIVER L298 ซึ่งมีตัวถังและวงจรภายในดังรูป [5]

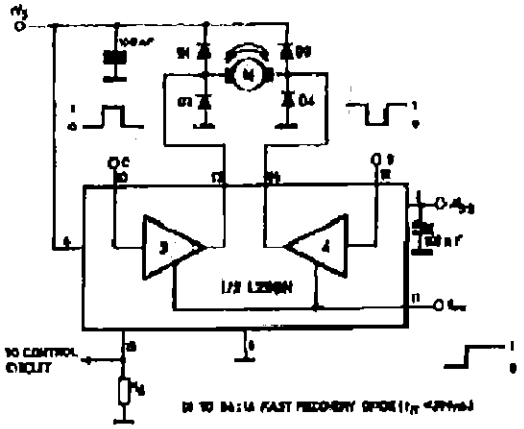


รูปที่ 3.6 การจัดการขาของไอซี L298

จากวงจรภายในจะเห็นว่า L298 สามารถขับโหลดได้ 2 ช่องและสามารถรับสัญญาณควบคุมแบบTTL (Transistor Transistor Logic) เพื่อที่จะควบคุมทิศทางการไหลของกระแส นอกจากนี้ยังมีขา Enable เป็นตัวตัดสินว่าจะให้โหลดที่ต่ออยู่ทำงานหรือไม่โดยไม่สนใจสัญญาณควบคุม

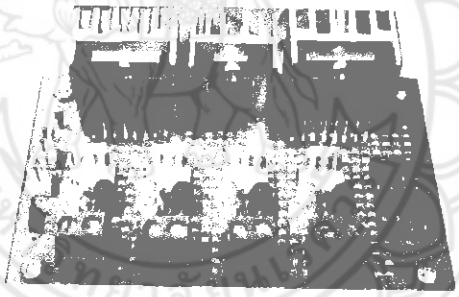
ขา Emitter ของ Transistor ทั้งสองข้างของแต่ละ bridge จะต่อกับตัวต้านทานภายนอกเพื่อที่จะใช้ในการกำหนดค่ากระแสไฟฟ้าที่ไหลได้ โดยหากเกินกว่าที่วงจรและโหลดที่ต่ออยู่สามารถที่จะรับได้ก็อาจจะมีวงจรเพื่อที่ทำการ Disable การทำงานของโหลดได้

จะเห็นว่าวงจรภายในไม่มีการต่อไดโอด เพื่อให้กระแสไฟฟ้าไหลได้เมื่อโหลดที่ใช้เป็นตัวเหนี่ยวนำ (มอเตอร์ รีเลย์) เนื่องจากสำหรับโหลดที่เป็นตัวเหนี่ยวนำค่ากระแสไฟฟ้าจะไม่สามารถเปลี่ยนเป็นศูนย์ได้ในเวลาทันทีทันใด (คล้ายกับกรณีที่ตัวเก็บประจุไม่สามารถเปลี่ยนค่าความต่างศักย์ตกคร่อมได้อย่างทันทีทันใด) จึงต้องมีการต่อไดโอดเพื่อให้กระแสไฟฟ้าไหลในกรณีที่เราเปิดสวิตซ์ แต่มอเตอร์ยังคงหมุนอยู่โดยที่การต่อไดโอดภายนอกแสดงดังรูป

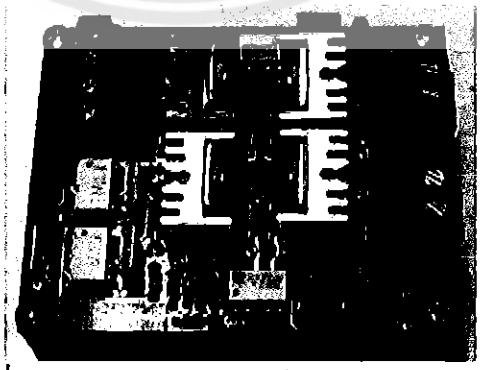


รูปที่ 3.7 การต่อวงจรใช้งานแบบ 1 ช่องของ L298

สำหรับโครงการนี้ได้ทำการดัดแปลงวงจร H-Bridge โดยต้องการให้สามารถควบคุมทิศทางได้ และใช้สัญญาณ Pulse Width Modulation ต่อเข้ากับขา Enable เพื่อให้ได้ระดับความเร็วตามต้องการดังรูป



รูปที่ 3.8 บอร์ดควบคุมมอเตอร์ H-Bridge ขนาด 3A



รูปที่ 3.9 บอร์ดควบคุมมอเตอร์ H-Bridge ขนาด 40A

คุณสมบัติของ Module ควบคุมทิศทางการหมุนของมอเตอร์ขับเคลื่อน แบบ H-Bridge รุ่น SE-HB40-1

Output

- Motor DC Supply 12-24 V 40A (Max.)
- Full-Complementary Power MOSFET Driver
  With ultra-fast reverse recovery protection diodes

Input

- Full Opto-isolated input interface signals
- 5V 8 mA TTL – Level

Drive Mode: independently with

- ON – OFF Control
- Direction Control
- Speed Control (PWM Drives)

PWM Frequency : 400 Hz - 1000 Hz (800 Hz Recommend)

เหตุผลที่เลือกใช้ H-Bridge รุ่น SE-HB40-1

หากมีการขับเกินกำลังจนมอสเฟทพัง ให้ตรวจสอบโดยการใช้มิเตอร์วัดการลัดวงจรของมอสเฟททั้ง 4 ตัว หากตัวใดเสียขั้วมอสเฟทจะลัดวงจร ให้เปลี่ยนมอสเฟทได้เลย วงจรส่วนอื่นปลอดภัย ไม่จำเป็นต้องตรวจสอบ

- การเปลี่ยนมอสเฟท เบอร์ IRF4905 ไม่มีจำหน่าย ให้ใช้เบอร์ IRF9540 แทน
- เบอร์ IRFZ44 มีจำหน่าย หรือให้ใช้เบอร์ HUF75639P3
- IRF3710PBF จะทำให้ทนแรงดันได้ดีกว่า

จากการทดลองไอซีควบคุมมอเตอร์สำเร็จรูป เบอร์ L298D ร่วมกับ Relay 10 A ไม่เหมาะสมกับการสวิตช์ซึ่งจากการบังคับแบบไม่ต่อเนื่อง เช่น เมื่อกดบังคับให้หุ่นยนต์เดินหน้าและหยุด สลับกันอย่างรวดเร็วจะทำให้หน้าสัมผัสของ Relay เกิดอุณหภูมิสูงทำให้ส่วนหน้าสัมผัสโลหะติดกันไม่แยกจากกันทำให้ควบคุมหุ่นยนต์ไม่ได้จึงเลือกใช้โมดูล H-Bridge รุ่น SE-HB40-1 แทน

จากแผ่นริ้นที่หาซื้อตามท้องตลาดเพื่อนำมาออกแบบวงจรเอง จะทำให้เกิดปัญหาเมื่อวงจรเกิดความร้อนและความถี่สูงระหว่าง ลายทองแดงทำให้ประสิทธิภาพของวงจรที่ออกแบบเองเกิดปัญหาความผิดพลาดของการส่งสัญญาณบ่อยครั้งจากการทดลอง และที่สำคัญคือลายทองแดง บางมากจะต้องใช้สายไฟเดินบริเวณที่มีกระแสไฟฟ้าไหลผ่านมาก ๆ สามารถทนกระแสได้ถึง 40 A ซึ่งแตกต่างจากวงจรที่ออกแบบขึ้นเอง ที่ทนกระแสได้น้อยกว่า

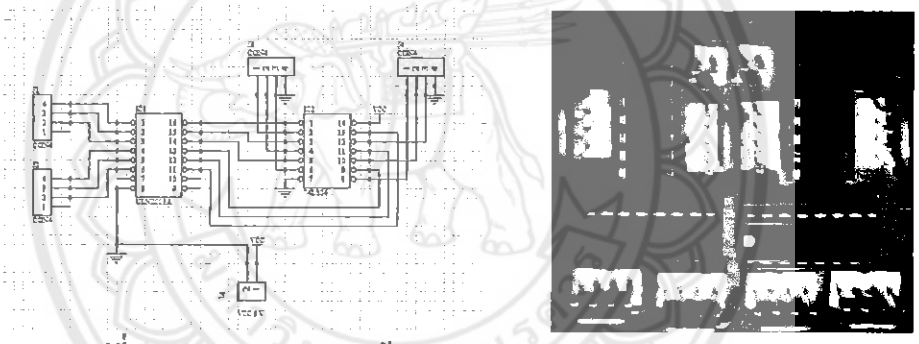
โดยการออกแบบโปรแกรมการควบคุมจะเป็นไปตามตารางที่ 3.1 โดยโปรแกรมจะรับอินพุตจากส่วนควบคุมระดับบน ถ้าส่วนควบคุมระดับบนส่งข้อมูลมาสั่งงานส่วนควบคุมระดับล่าง (MCU) ก็จะส่งแรงดันเอาต์พุต ประมาณ 5 V ตามจำนวนบิตที่วงจรความคุมทิศทางมอเตอร์ H-Bridge รุ่น SE-HB40-1 ดังนี้

ตารางที่ 3.1 การเคลื่อนที่ของหุ่นยนต์

สถานะ	Wheel Left				Wheel Right			
	IN1	IN2	EN	GND	IN1	IN2	EN	GND
Forward Robot	0	1	1	0	0	1	1	0
Backward Robot	1	0	1	0	1	0	1	0
Turn Left Robot	0	0	0	0	0	1	1	0
Turn Right Robot	0	1	1	0	0	0	1	0
Turn Short Left Robot	1	0	1	0	0	1	1	0
Turn Short Right Robot	0	1	1	0	1	0	1	0

จากตารางที่ 3.1 จะทำให้ทราบได้ว่าถ้าต้องการให้หุ่นยนต์เคลื่อนที่ไปในทิศทางใด ก็ต้องเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ให้ส่งแรงดันออกที่พอร์ตของไมโครคอนโทรลเลอร์ ซึ่งในที่นี้การควบคุมจะส่งเอาต์พุตออกที่พอร์ต 2 ของบอร์ดไมโครคอนโทรลเลอร์ตามบิตแต่ละบิตของพอร์ตไมโครคอนโทรลเลอร์ ที่ช่อง สถานะ IN1 และ IN2 คือ ขาควบคุมทิศทางการหมุนของมอเตอร์ และ EN คือ ขา Enable มีหน้าที่ในการเปิดการทำงานของวงจรควบคุมหารหมุนของมอเตอร์ และยังเป็นขาที่ใช้ปรับความเร็วของมอเตอร์ในการใช้วิธีการ Pulse Wide Modulation ตามที่ได้กล่าวแล้วในบทที่ 2 และขา GND คือ ขากราวด์ที่ต้องต่อร่วมกับบอร์ดไมโครคอนโทรเลอร์ด้วย เพื่อให้กระแสไหลได้ครบวงจรหรือมองเป็นกราวด์ร่วมนั้นเอง เมื่อวงจรควบคุมทิศทางการหมุนของมอเตอร์ ได้รับการส่งเอาท์พุตจากไมโครคอนโทรเลอร์เรียบร้อย มอเตอร์จะหมุนโดยมีจุดต่อมอเตอร์ และรับแรงดันจากภายนอกเพื่อให้มอเตอร์ได้รับกระแส โดยแหล่งจ่ายแรงดันจากภาย นอกนี้จะเลือกใช้แบตเตอรี่รถจักรยานยนต์ที่มีแรงดัน 12 VDC 3A/hr ซึ่งมีราคาถูกและสามารถประจุไฟเข้าได้อย่างสะดวก แต่มีน้ำหนักมากถ้าเลือกเป็นแบตเตอรี่แบบใช้น้ำกลั่น และในที่นี้เลือกแบบแบตเตอรี่แห้ง และการเลือกแรงดันที่จ่ายให้มอเตอร์นั้น พิจารณาโดยมอเตอร์เป็นมอเตอร์กระแสตรงที่ทนแรงดันสูงสุดไม่เกิน 12 V

เมื่อออกแบบรหัสควบคุมเพื่อเตรียมการเขียนโปรแกรมแล้วนั้น การออกแบบวงจรเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับชุดควบคุมทิศทางการหมุนของมอเตอร์นั้น กระแสที่ออกจากพอร์ตของไมโครคอนโทรลเลอร์ ไม่เพียงพอต่อการควบคุมชุดควบคุมทิศทางการหมุนของมอเตอร์เนื่องจาก ชุดควบคุมทิศทางการหมุนของมอเตอร์ต้องการกระแสเข้าที่ขา IN1, IN2, EN ที่กระแส สูงกว่า 8 mA แต่ไมโครคอนโทรลเลอร์จ่ายกระแสออกพอร์ตได้เพียง 1 mA. เท่านั้น จึงต้องออกแบบวงจรขยายกระแสให้กับบอร์ดควบคุมทิศทางการหมุนของมอเตอร์ โดยใช้อุปกรณ์ดังต่อไปนี้ คือ IC ULN2003A และ IC Not Gate 7404 โดย IC ULN2003A โดยไอซีตัวนี้จะทำหน้าที่ขยายกระแสให้ ให้กับบอร์ดควบคุมทิศทางการหมุนของมอเตอร์และต่อกับ IC Not Gate 7404 เพื่อกลับบิตและส่งเข้าให้กับบอร์ดควบคุมทิศทางการหมุนของมอเตอร์ การทำงานของวงจรขยายกระแสมีลักษณะเหมือนกับวงจรบัฟเฟอร์ บางวงจรหรือ IC สำเร็จรูปบางตัวเช่น IC 74LS245 ทำหน้าที่ขยายกระแสจากพอร์ตของไมโครคอนโทรลเลอร์ หรืออาจจะใช้วิธีการ ต่อตัวต้านทานอนุกรมกับแหลงจ่ายแรงดัน 5 Volt เพื่อทำให้เกิดกระแสที่แต่ละบิตสูงขึ้นแต่กระแสก็ยังไม่เพียงพอ ตัวอย่างการต่อวงจร โดยใช้ IC ULN2003A และ IC Not Gate 7404 ดังรูปที่ 3.10



รูปที่ 3.10 วงจรขยายกระแสด้วย IC ULN2003A และ IC Not Gate 7404

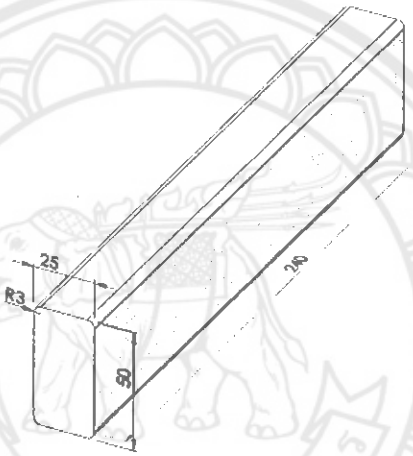
เมื่อประกอบอุปกรณ์ดังแสดงในรูปที่ 3.10 เรียบร้อย ก็จะสามารถนำไปใช้งานในควบคุมการหมุนของมอเตอร์ได้โดยกระแสที่ได้จากวงจร Buffer ที่ออกแบบขึ้นให้กระแสสูงถึง 500 mA จากการแสดงคุณสมบัติจากเอกสารแสดงคุณสมบัติ

3.3 การออกแบบหุ่นยนต์

เนื่องจากโครงการนี้เป็นโครงการหุ่นยนต์กู้ภัย วัสดุที่เลือกใช้จึงต้องการคุณสมบัติที่มีความคงทนและแข็งแรง ส่วนประกอบหลักของหุ่นยนต์คือตัวฐานของหุ่นยนต์, ลือตีนตะขาบและแขนล้อยก โครงสร้างของหุ่นยนต์ส่วนใหญ่จะเป็นเหล็กและอลูมิเนียม เนื่องจากต้องการให้มีความแข็งแรง

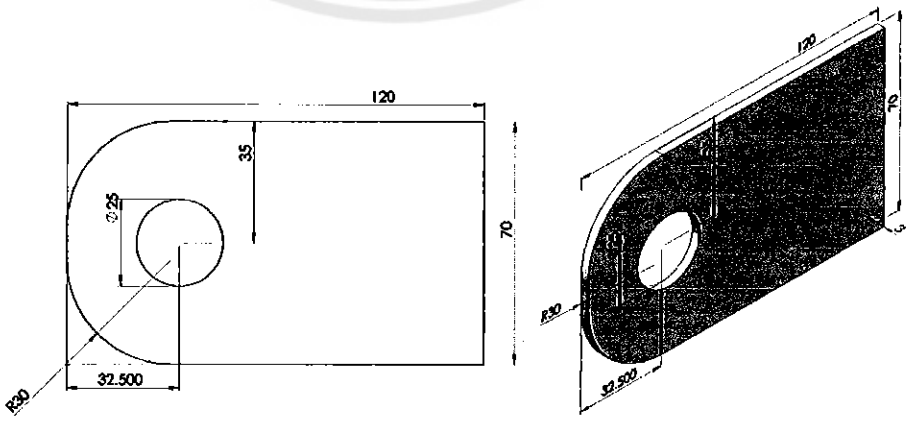
3.3.1 การออกแบบโครงล้อของหุ่นยนต์

โครงสร้างของล้อจะใช้เหล็กกระบอกสี่เหลี่ยมที่มีขนาด 2.5 ซม. × 5 ซม. มีขนาดยาว 25 ซม. ดังรูป

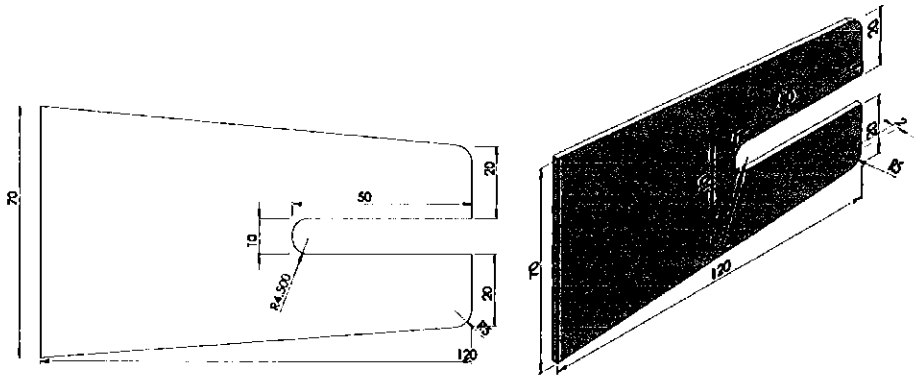


รูปที่ 3.11 เหล็กกระบอกสี่เหลี่ยมสำหรับทำโครงล้อ

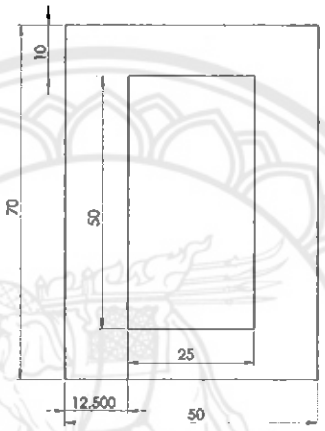
ตัวประล้อนั้นจะใช้แผ่นเหล็ก 2 อันมาประกบกันโดยตัดให้ได้ขนาดตามแบบดังรูป



รูปที่ 3.12 แผ่นเหล็กสำหรับทำที่ประกบล้อหน้า

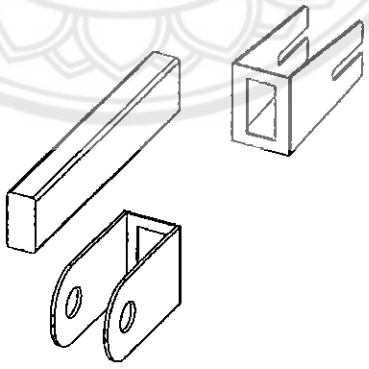


รูปที่ 3.13 แผ่นเหล็กสำหรับทำที่ประกบล้อหลัง

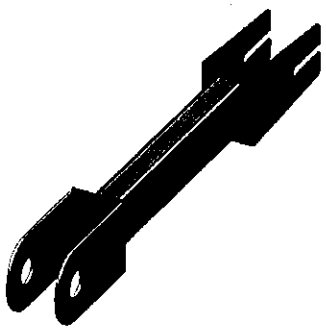


รูปที่ 3.14 แผ่นเหล็กไว้ยึดแผ่นประกบล้อ

เมื่อเราตัดแบบที่เราได้ออกแบบไว้เรียบร้อยแล้วก็จะนำมาประกอบกันโดย**การเชื่อม**

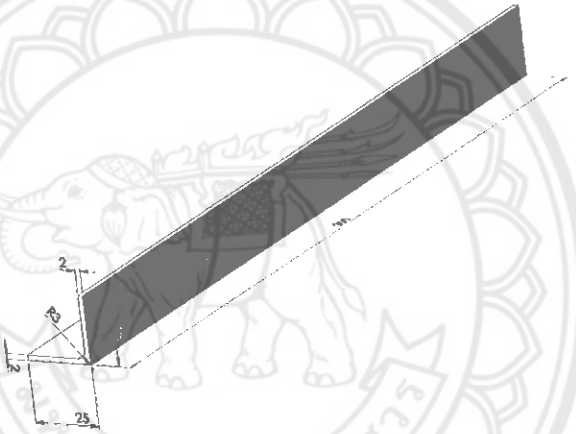


รูปที่ 3.15 การประบอกชิ้นส่วนของฐานล้อ

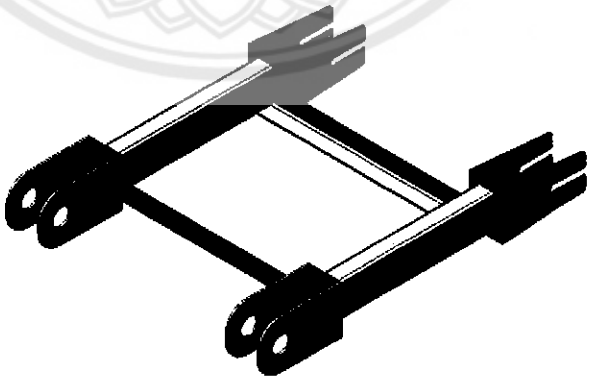


รูปที่ 3.16 รูปฐานล้อเมื่อประกอบแล้ว

เราจะทำชุดของฐานล้อจำนวน 2 ชุด จากนั้นนำเหล็กฉากมาทำเป็นตัวยึดระหว่างโครงล้อทั้ง 2 ข้างเข้าด้วยกัน โดยตัดเหล็กฉาก 1 นิ้ว (25 มม.) ยาว 30 ซม.



รูปที่ 3.17 เหล็กฉากที่จะนำไปยึดเป็นโครง

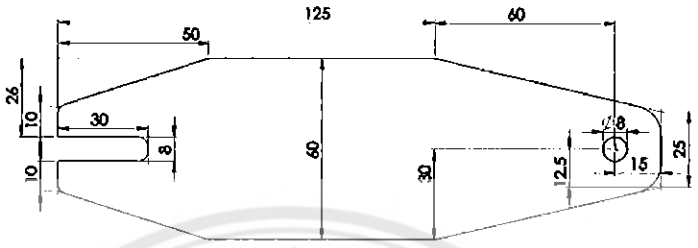


รูปที่ 3.18 โครงสร้างฐานล้อของหุ่นยนต์

3.3.2 การออกแบบแขนล้อของหุ่นยนต์

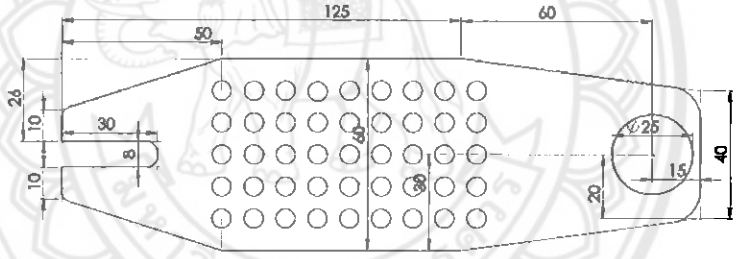
ส่วนประกอบของแขนล้อของหุ่นยนต์จะเป็นแผ่นเหล็ก 2 ชิ้นประกัน แล้วทำการเชื่อมด้วยแผ่นเหล็กที่ขั้นไว้ตรงกลางเพื่อให้มีความแข็งแรง เนื่องจากแขนล้อนั้นต้องทำการยกตัวรถและประคองตัวรถในสภาพพื้นผิวที่ขรุขระได้

ขั้นตอนการออกแบบนั้น ขั้นแรกทำการตัดแผ่นเหล็กให้มีขนาดดังรูปด้านล่าง



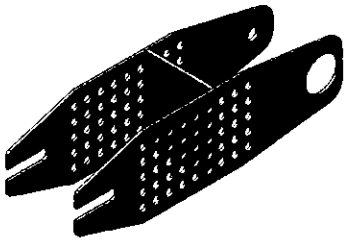
รูปที่ 3.19 ประกบแขนล้อด้านนอก

เมื่อเราตัดแผ่นเหล็กได้ขนาดแล้วจะทำการเจาะรูเพื่อที่จะลดน้ำหนักของตัวรถ



รูปที่ 3.20 ประกบแขนล้อด้านในที่เจาะรูแล้ว

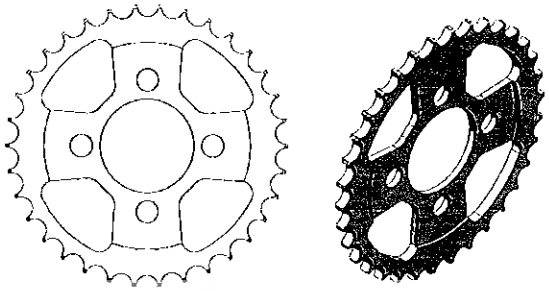
จากนั้นนำแผ่นเหล็กมาทำการประกบกันโดยการเชื่อม จะมีแผ่นเหล็กขั้นไว้ตรงกลางเพื่อให้แผ่นเหล็ก 2 อันนั้นติดกันเป็นชิ้นเดียว ดังรูป



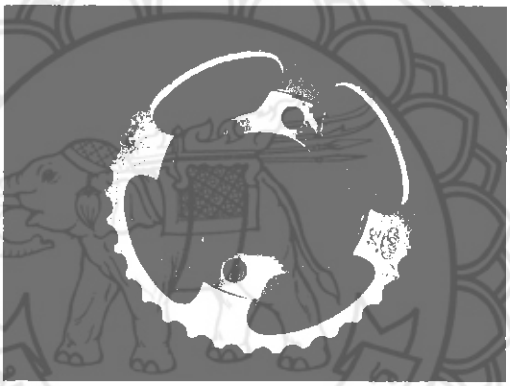
รูปที่ 3.21 แขนล้อของหุ่นยนต์

3.3.3 การออกบแบบล้อของหุ่นยนต์

ล้อและแขนล้อของหุ่นยนต์จะใช้สเตอร์ของรถจักรยานยนต์ ขนาด 34 ฟัน (มีจำนวนฟัน 34 ฟัน) ดังรูป

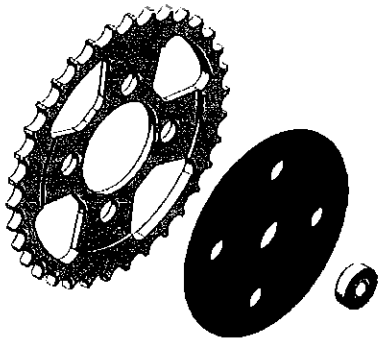


รูปที่ 3.22 สเตอร์รถจักรยานยนต์ขนาด 34 ฟัน



รูปที่ 3.23 รูปจริงของสเตอร์รถจักรยานยนต์ขนาด 34 ฟัน

การประกอบล้อจะใช้ดุมมีประกบกับสเตอร์ ในดุมล้อนั้นจะมีลูกปืนอยู่ข้างในเพื่อที่จะลดแรงเสียดทานของล้อกันเพลา



รูปที่ 3.24 ชุดส่วนประกอบของล้อ

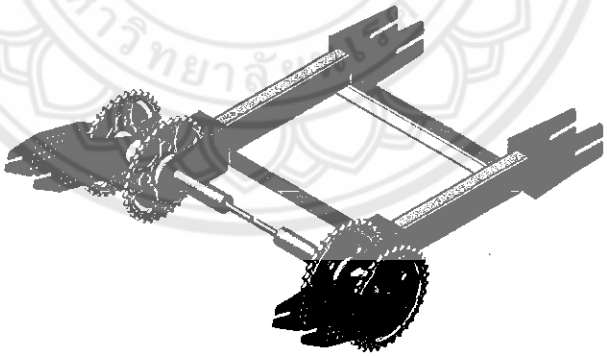
3.3.4 การประกอบฐานของหุ่นยนต์

การประกอบฐานของหุ่นยนต์ โดยการนำชิ้นส่วนแต่ละชิ้นที่เราทำการออกแบบและประกอบกันเป็นชุดแล้วมารวมกันเพื่อให้เป็นโครงของหุ่นยนต์ โดยการประกอบนั้นจะต้องใช้เพลาเพื่อเป็นตัวขับเคลื่อนของหุ่นยนต์ จะประกอบด้วยเพลาขับเคลื่อน และเพลายกแขนล้อ

เพลาขับเคลื่อนจะมีเส้นผ่านศูนย์ขนาด 2.5 เซนติเมตร มีรูขนาด 1.5 เซนติเมตร เราต้องทำการกลึงปลายของเพลาเพื่อที่จะได้ทำการใส่ลูกปืนเข้าไปข้างใน ซึ่งลูกปืนข้างในจะเป็นตัวขับกับเพลาเล็กขนาด 0.8 เซนติเมตรเพื่อยกแขนล้อ

รูปที่ 3.25 เพลาขับเคลื่อนและลูกปืนของหุ่นยนต์

เมื่อได้เพลามา ก็จะทำการประกอบเข้ากับโครงที่เราได้ออกแบบไว้ ซึ่งจะมีเพลา 8 มิลลิเมตร อยู่ข้างในเพื่อเป็นตัวยกแขนล้อของหุ่นยนต์ดังรูป

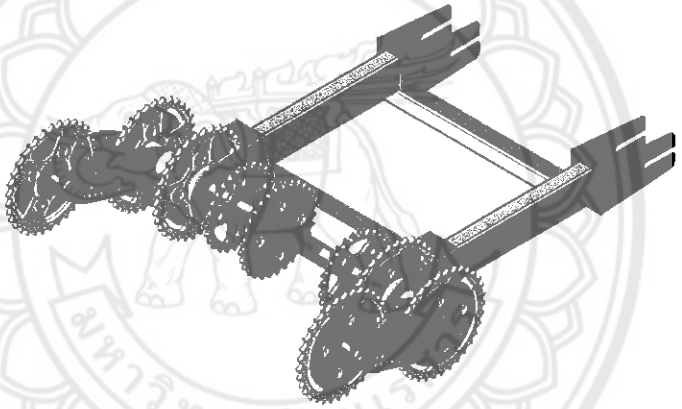


รูปที่ 3.26 โครงหุ่นยนต์ที่ใส่เพลาขับเคลื่อนและเพลายกแขน

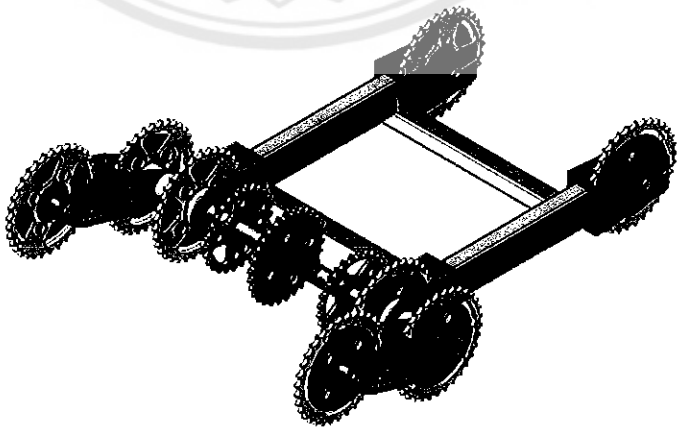
เมื่อได้โครงที่ใส่เพลาแล้ว ต่อไปจะทำการใส่ตัวที่จะทำการขับเคลื่อนยกแขนล้อโดยจะใช้สเตอร์ของรถจักรยานยนต์ขนาด 30(มีจำนวนฟัน 30 ฟัน) มาทำการเชื่อมกับเพลาขนาดเส้นผ่านศูนย์กลาง 2.5 cm. มีรูขนาดเส้นผ่านศูนย์กลาง 0.8 cm. ดังรูป

รูปที่ 3.27 ชุดของตัวขับเคลื่อนตัวแขนล้อ

จากนั้นเราจะทำการประกอบกับโครงเข้ากับเฟืองโซ่ขับเคลื่อน ซึ่งเฟือง**โซ่ขับเคลื่อนจะมี**ขนาด 27 ฟัน จะได้ดังรูป



รูปที่ 3.28 โครงหุ่นยนต์ที่ใส่สเตอร์ขับเคลื่อนแล้ว

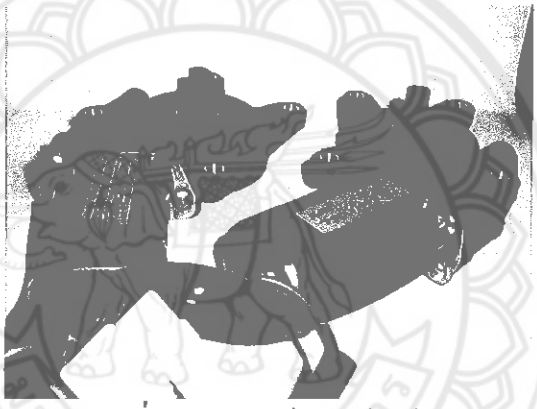


รูปที่ 3.29 โครงฐานของหุ่นยนต์เมื่อใส่ล้อทั้งหมด

เฟืองโซ่ที่ที่ทำหน้าที่เป็นตัวขับเคลื่อนจะติดกับมอเตอร์ ซึ่งมีขนาด 9 ฟัน จะให้อัตราการทดของเฟืองโซ่จะเท่ากับ 1 : 3 (9 ฟัน : 27 ฟัน)

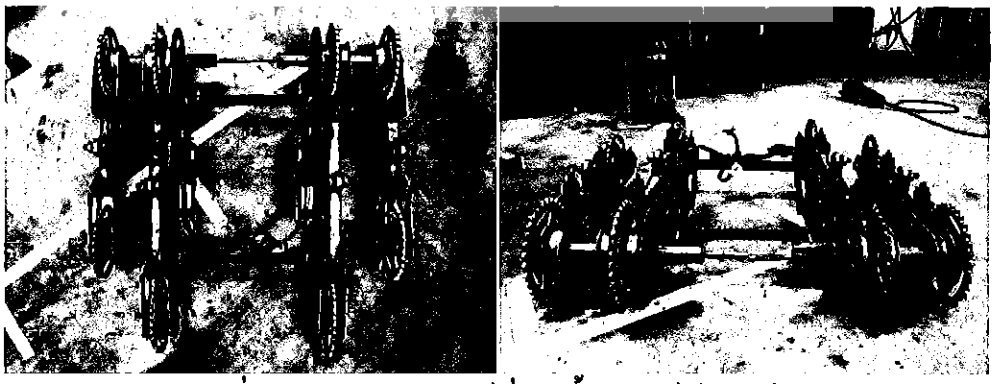
รูปที่ 3.30 เฟืองโซ่ขนาด 9 ฟัน

เมื่อได้โครงมาแล้วก็จะทำการติดตั้งมอเตอร์ โดยมอเตอร์ที่ใช้ขับเคลื่อนและมอเตอร์ของแขนล้อนั้น จะใช้เป็นมอเตอร์กระจกไฟฟ้า



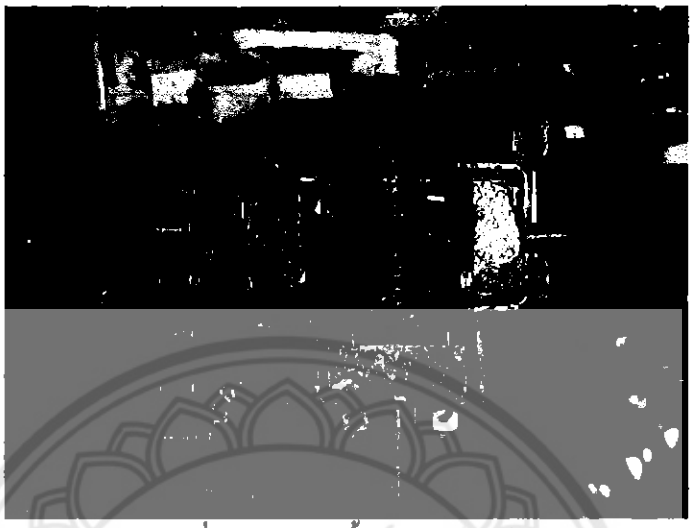
รูปที่ 3.31 มอเตอร์กระจกไฟฟ้า

ทำการติดตั้งมอเตอร์กระจกไฟฟ้าลงไปที่โครงของหุ่นยนต์ โดยจะใช้เหล็กยึดกับมอเตอร์และเชื่อมเหล็กกับโครงของหุ่นยนต์ เมื่อทำการติดตั้งมอเตอร์ขับเคลื่อนแล้วจะได้โครง

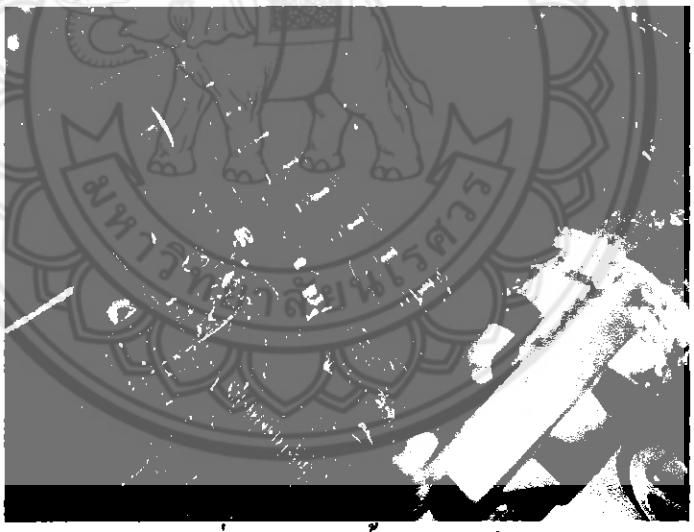


รูปที่ 3.32 โครงของหุ่นยนต์เมื่อติดตั้งมอเตอร์เข้าไปแล้ว

การติดตั้งแขนล้อจะใช้ชุดเฟืองตัวหนอนเข้ามาเป็นตัวขับแขนล้อ เนื่องจากการใช้ชุดเฟืองตัวหนอนนั้นจะช่วยลดกำลังของมอเตอร์ได้ แต่ก็มีขอเสียตรงที่ทำให้ช้า ในลักษณะของงานนี้เราต้องการกำลังของการยกแขนมาก เพราะแขนล้อนั้นต้องยกตัวเองได้

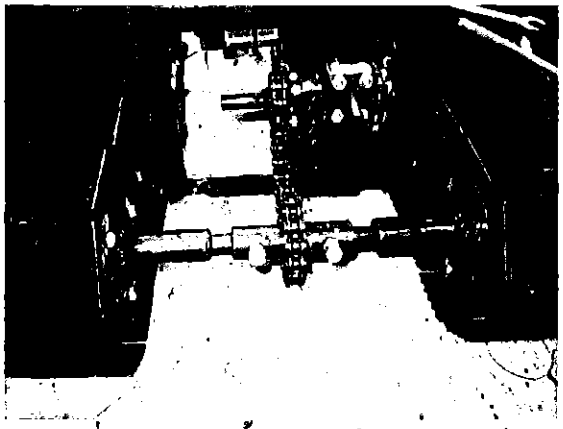


รูปที่ 3.33 การติดตั้งชุดของแขนล้อ



รูปที่ 3.34 การติดตั้งของแขนล้อ

จากนั้นทำการติดตั้งลงไปในโครงของหุ่นยนต์ การติดตั้งจะต้องตั้งโซ่ให้โซ่ตึงที่สุด เพราะจะทำให้แขนล้อของหุ่นยนต์นั้นไม่หลวม



รูปที่ 3.35 การติดตั้งชุดแขนล้อของหุ่นยนต์

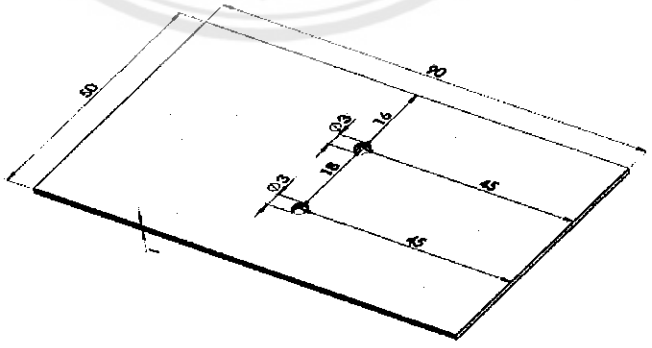
หลังจากติดตั้งเรียบร้อยแล้ว ในส่วนต่อไปจะเป็นการออกแบบและสร้างโซ่ตีนตะขาบเพื่อที่จะนำไปใส่ให้กับหุ่นยนต์

3.3.5 การออกแบบล้อตีนตะขาบ

ล้อตีนตะขาบเป็นชิ้นส่วนในการขับเคลื่อนรถถัง เหมาะสำหรับพื้นผิวที่มีลักษณะตะปุ่มตะป่ำ ขรุขระ มีสิ่งกรีดขวาง หรือกรณีที่ล้อทั่วไปวิ่งได้ลำบาก และสามารถเปลี่ยนทิศทาง ณ จุดที่ต้องการได้โดยทันที แต่มีข้อเสียตรงที่เวลาในการประดิษฐ์ถ้าประกอบไม่ดีอาจทำให้ไม่แข็งแรงและหักได้

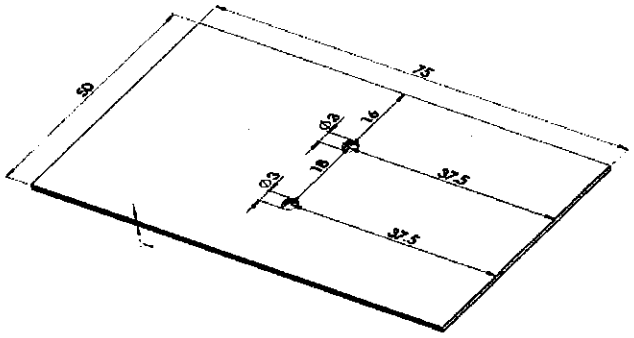
ส่วนประกอบของล้อตีนตะขาบ

ออกแบบสำหรับชิ้นส่วนแผ่นเหล็กและยางที่สำหรับใช้เป็นล้อมีขนาด กว้าง × ยาว เท่ากับ 5 ซม. × 9.5 ซม. ดังรูป



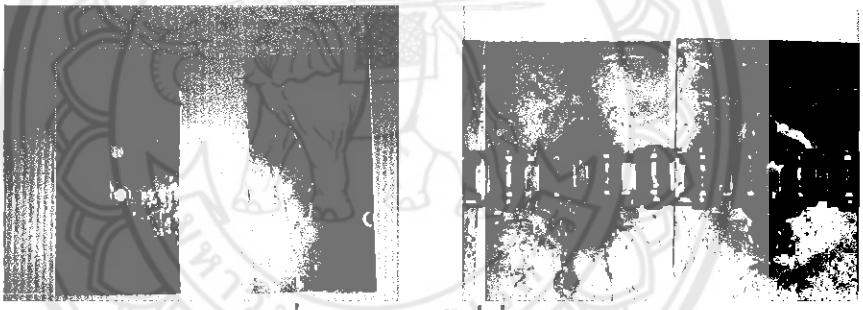
รูปที่ 3.36 ภาพแสดงการออกแบบส่วนประกอบของล้อตีนตะขาบ

ออกแบบสำหรับชิ้นส่วนแผ่นเหล็กและยางที่สำหรับใช้เป็นล้อยกมีขนาด กว้าง × ยาว เท่ากับ 5 ซม. × 9.5 ซม. ดังรูป

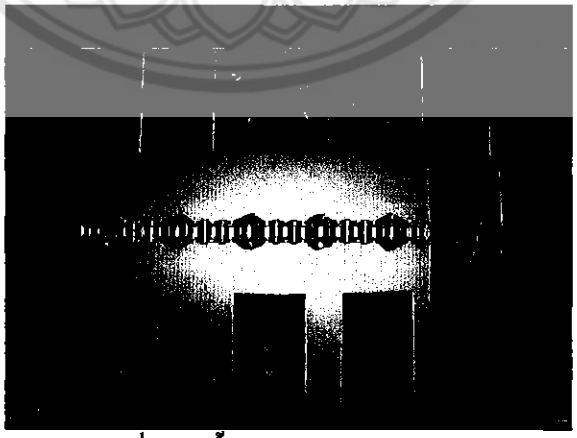


รูปที่ 3.37 ภาพแสดงการออกแบบส่วนประกอบของล้อยกตีนตะขาบ

โซ่ที่จะนำมาทำเป็นตีนตะขาบนั้นจะใช้โซ่เบอร์ 428 โดยจะนำโซ่มาเชื่อมกับแผ่นเหล็กสี่เหลี่ยมที่ได้ออกแบบไว้ข้างต้น



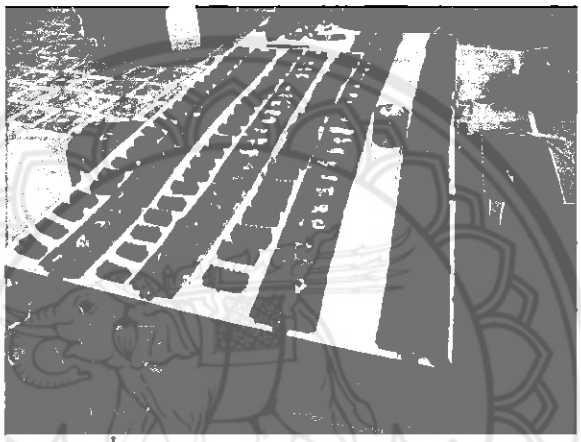
รูปที่ 3.38 แผ่นเหล็กที่เชื่อมกับโซ่



รูปที่ 3.39 ชิ้นส่วนของโซ่ตีนตะขาบ

จากนั้นจะนำแผ่นยางที่ตัดมาจากสายพานของรถจักรยานยนต์ มาติดกับแผ่นเหล็ก เพื่อเพิ่มแรงเสียดทานให้กับล้อของหุ่นยนต์ โซ่ที่ที่เชื่อเสร็จแล้วโดยใช้กาวยางติด

รูปที่ 3.40 สายพานรถจักรยานยนต์ที่ตัดแล้ว



รูปที่ 3.41 ส่วนประกอบของล้อตีนตะขาบ

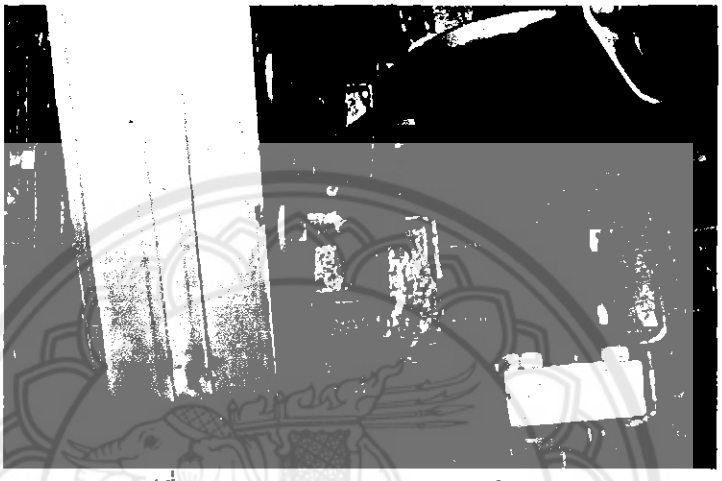
เมื่อประกอบโซ่ตีนตะขาบเรียบร้อยแล้วก็จำนำไปใส่กับล้อที่ได้ออกแบบไว้

รูปที่ 3.42 ล้อเมื่อใส่โซ่ตีนตะขาบแล้ว

3.3.6 การออกแบบแขนของหุ่นยนต์

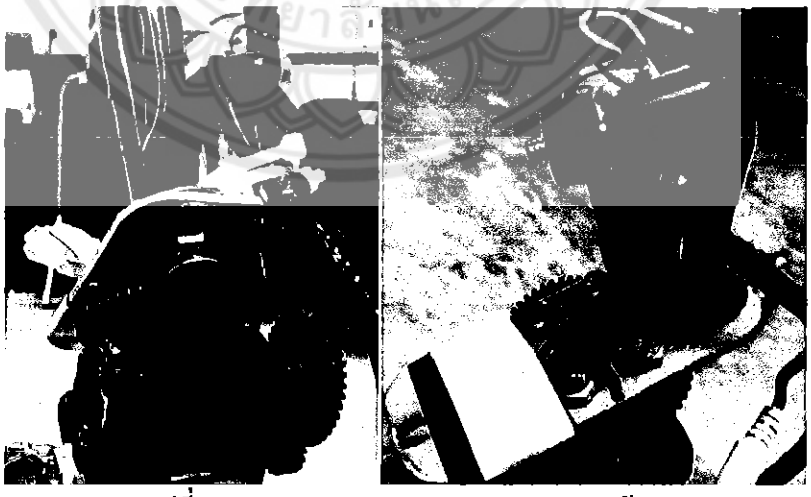
แขนของหุ่นยนต์มีไว้เพื่อไว้ติดเข้ากับกล้อง ซึ่งจะทำให้สามารถหมุนหรือปรับกล้องในมุมต่างๆได้อย่างอิสระโดยไม่ต้องหันตัวรถ และยังสามารถค้นหาเหยื่อได้ในมุมที่รถไม่สามารถไปได้ โดยอุปกรณ์ที่ทำแขนจะต้องมาน้ำหนักเบาและแข็งแรง จึงเลือกใช้อลูมิเนียม

การออกแบบแขน แขนของหุ่นยนต์จะมีความยาวพอประมาณไม่ยาวหรือสั้นเกินไป ซึ่งจะออกแบบให้แขนมีความยาว 50 cm. ซึ่งเมื่อติดตั้งแล้วจะไม่ให้แขนเลยออกจากตัวหุ่น



รูปที่ 3.43 แสดงภาพการออกแบบท่อนแขน

การออกแบบตัวหมุนกล้อง โดยการออกแบบนี้จำให้กล้องหมุนได้ 360 องศา โดยจะต้องออกแบบให้หมุนได้ในทั้งแนวตั้ง (ขึ้น – ลง) และแนวนอน (ซ้าย – ขวา)



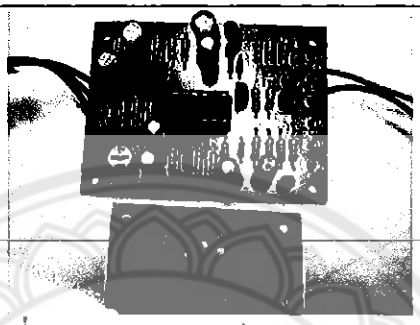
รูปที่ 3.44 แสดงภาพการออกแบบฐานหมุนกล้อง

3.4 ชุดเซ็นเซอร์ของหุ่นยนต์

3.4.1 ชุดเซ็นเซอร์ตรวจจับสิ่งมีชีวิต

เซ็นเซอร์ที่จะทำการติดตั้งเป็นเซ็นเซอร์ที่มีจำหน่ายตามร้านขายอุปกรณ์อิเล็กทรอนิกส์ทั่วไป ซึ่งได้แก่

PIR sensor ส่วนสำคัญที่จะใช้ในการตรวจจับรังสีอินฟราเรดคือ Fresnel lens จึงจำเป็นที่จะต้องติดตั้งไว้ด้านหน้าของหุ่นยนต์ดังรูป

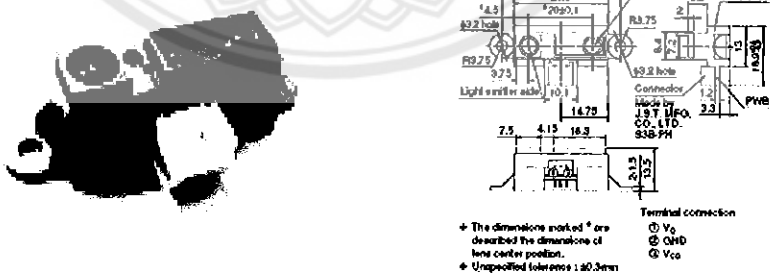


รูปที่ 3.45 ภาพแสดงบอร์ดวงจร PIR sensor

จากราคาของ PIR Sensor มีราคาแพงมากจึงไม่ได้ใส่ลงไปในตัวหุ่นยนต์กู้ภัย

3.4.2 โมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด

- เป็นโมดูลตรวจจับระยะทางแบบอินฟราเรด ที่สามารถวัดระยะทาง ได้ถูกต้อง
- GP2Y0A21YK0F วัดระยะทางในช่วง 10 - 80 เซนติเมตร



ก)　　ข)

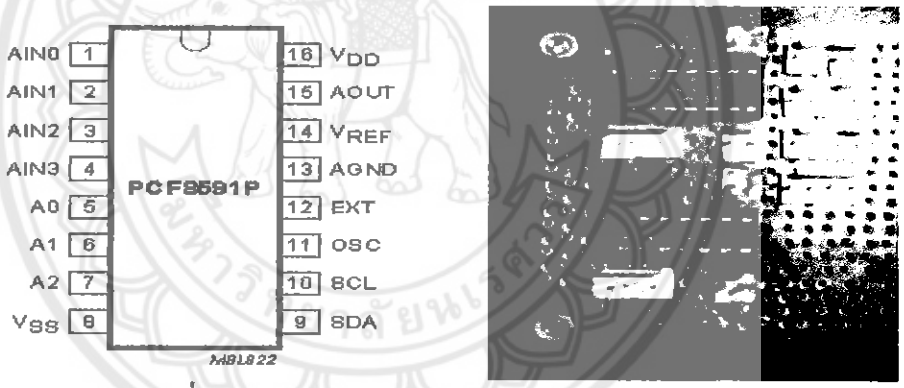
รูปที่ 4.46 โมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด GP2Y0A21YK0F

ก) ลักษณะภายนอกของโมดูลตรวจวัดระยะทาง

ข) ขนาดของโมดูลตรวจวัดระยะทาง

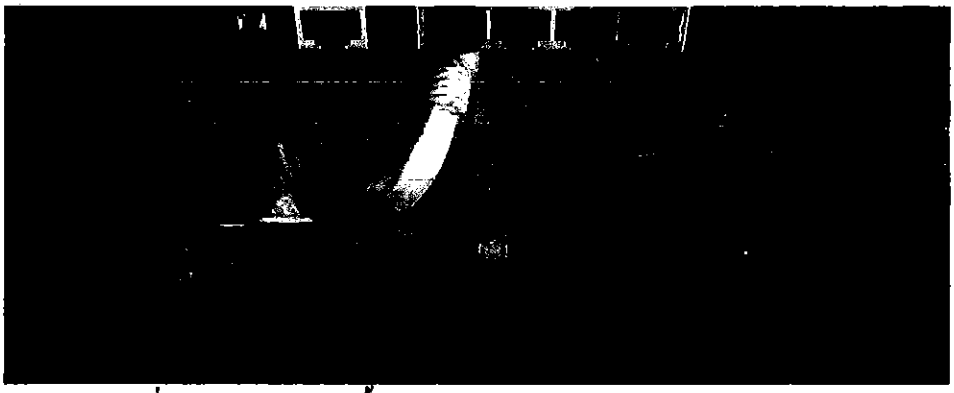
ที่มีวัตถุมาขวางกั้น ทำให้โดยให้ผลการตรวจจับเป็นแรงดันไฟตรง ในย่าน 0.4 - 2.4V ประกอบด้วยตัวส่ง และตัวรับอินฟราเรด ที่ติดตั้ง ภายใต้ ตัวถัง เดียวกัน จะทำงานทันทีที่มีไฟเลี้ยง 5V จ่ายให้โดยตัวส่ง อินฟราเรด จะขับแสงอินฟราเรด จาก ตัวมันตลอดเวลา และเมื่อใดเกิดการสะท้อนของแสงอินฟราเรดกลับไปยังตัวรับภายใน

โดยโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรดนี้ จะต้องใช้คู่กับ โมดูล Analog to Digital เบอร์ PCF8591 8-bit A/D เนื่องจากโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด GP2Y0A21YK0F ให้ผลการตรวจจับเป็นแรงดันไฟตรง ในย่าน 0.4 - 2.4V ดังนั้นจึงต้องมีการทำการ แปลงแปลงแรงดันนี้จากอะนาล็อกเป็นดิจิตอล (Analog to Digital) โดยการแปลงจะมีอุปกรณ์ที่แปลงจากดิจิตอลเป็นอะนาล็อก(Digital to Analog) ซึ่งการแปลงนี้จะเป็นการแปลงแรงดันให้อยู่ในระดับ 0 และ 1 หรือ 0V และ 5V ดังนั้นการเลือกใช้อุปกรณ์ในการทำการ A/D นั้นจะเลือกใช้ เบอร์ PCF8591 8-bit A/D and D/A converter โดยเป็นการสื่อสารในรูปแบบ I^2C BUS การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ ตามที่กล่าวมาข้างต้นว่าจะใช้พอร์ตที่ 1 ในบิตที่ 1 และบิตที่ 2 ในการรับสัญญาณ SDA และ SCL ตามลำดับ และจากนี้จะเป็นการออกแบบวงจรในการสร้างโมดูล Analog to Digital เบอร์ PCF8591 8-bit A/D and D/A converter ดังแสดงในรูป



รูปที่ 3.47 โมดูล Analog to Digital เบอร์ PCF8591 8-bit A/D

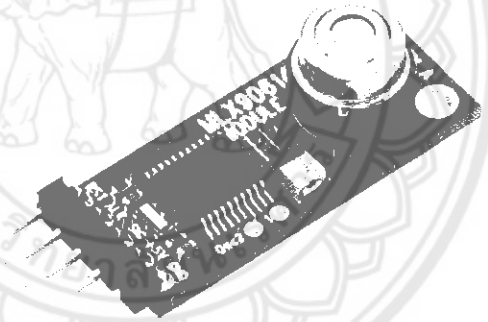
การติดตั้งโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด จะทำการติดตั้งไว้ทางด้านข้างของหุ่นยนต์ เพราะบางที่ในตอนที่เราบังคับมุมกล้องอาจไม่เห็นทางด้านข้าง และอาจทำให้ชนได้ จึงต้องติดตั้ง ไว้ด้านข้างเพื่อเป็นตัวตรวจจับสิ่งกีดขวางทางด้านข้างที่มองไม่เห็น



รูปที่ 3.48 แสดงการติดตั้งโมดูลตรวจวัดระยะทางด้วยแสงอินฟราเรด

3.4.3 ชุดตรวจอุณหภูมิแบบอินฟราเรด

เป็นชุดตรวจจับอุณหภูมิแบบอินฟราเรดระยะไกล วัดค่าอุณหภูมิของวัตถุ และค่าที่อ่านได้ก็เป็นค่าอุณหภูมิที่แท้จริงของวัตถุ โดยไม่มีผลกระทบจากสภาพแวดล้อมรอบข้าง เพียงแค่เล็งไปที่วัตถุก็สามารถอ่านค่าอุณหภูมิส่งค่าไปที่ไมโครคอนโทรลเลอร์ ได้ในเวลาไม่กี่วินาที ใช้งานกับวัตถุที่มีอุณหภูมิสูงหรือวัตถุที่ไม่สามารถเข้าไปใกล้ ๆ



รูปที่ 3.49 ตัวตรวจจับอุณหภูมิแบบอินฟราเรด

การติดตั้งตัวตรวจจับอุณหภูมินั้นจะติดตั้งเข้ากับฐานหมุนกล้อง เพราะจะได้หมุนไปตรวจสอบในที่ต่างๆได้

รูปที่ 3.50 แสดงภาพการติดตั้งตัวตรวจจับอุณหภูมิ

3.4.4 Encoder

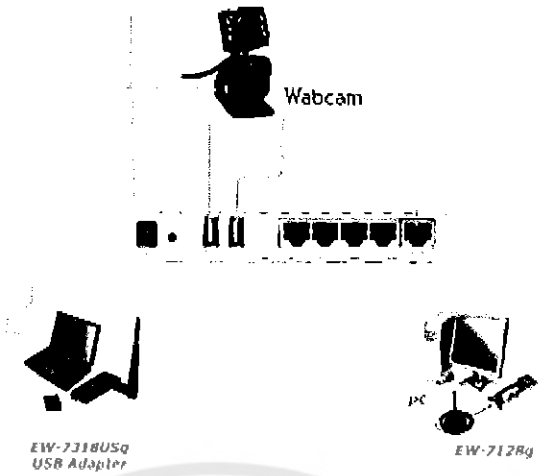
Encoder ที่ใช้จะมี 4 สาย คือสายไฟ กับ GND และอีก 3 สายจะเป็นสายข้อมูล จะประกอบด้วยสาย A กับ B ซึ่งสัญญาณของ A กับ B นั้นจะเหลื่อมกัน 90 องศา ความละเอียดจานของ Encoder จะมีความละเอียด 288

รูปที่ 3.51 การติดตั้ง Encoder

3.5 กล้อง

3.5.1 กล้อง Webcam

การใช้กล้อง Webcam จะใช้ในการตรวจดูสภาพแวดล้อมต่างๆ ซึ่งคุณสมบัติจะใช้เซนเซอร์แบบ CCD ซึ่งนิยมใช้ในกล้องดิจิตอล สามารถให้ความละเอียดที่สูงไวต่อแสง และก็มี Noise สัญญาณรบกวนไม่มาก ซึ่งในกล้องเว็บแคมจะใช้การเชื่อมต่ออินเทอร์เฟซแบบ USB ในการทำโครงงานนี้เป็นการใช้ Wireless LAN ในการติดต่อกับเครื่องคอมพิวเตอร์โดยตัว Wireless LAN สามารถที่จะเชื่อมต่อแบบ USB กับกล้องเว็บแคมได้



รูปที่ 3.52 การเชื่อมต่อกล้องเว็บแคมเข้า Wireless Router

3.5.2 กล้อง IP Camera

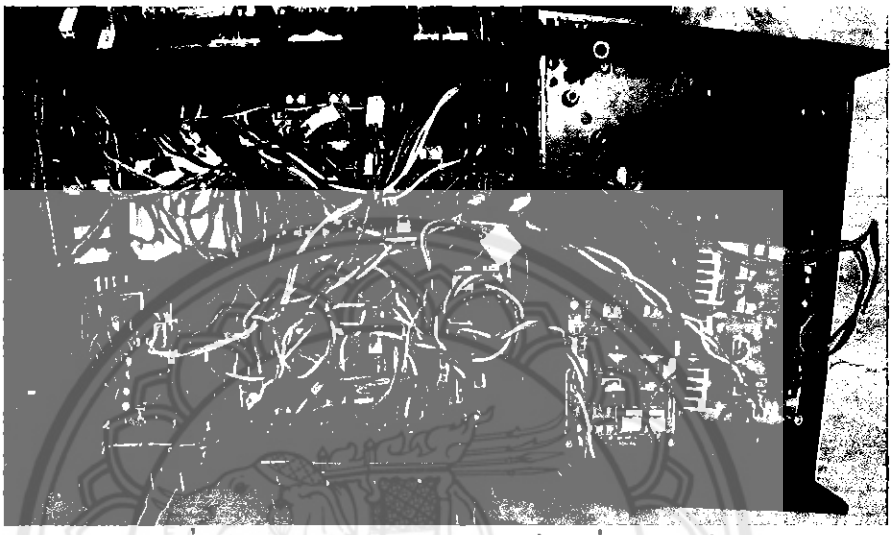
การใช้กล้อง IP เพื่อเป็นการบอกสถานะรอบข้างของหุ่นยนต์เพื่อสะดวกต่อการควบคุมหุ่นยนต์ เพื่อให้ไปตามที่ต่างๆได้ตามความต้องการของผู้ควบคุม โดยการใช้งานจะทำการเชื่อมต่อกับ Wireless LAN

รูปที่ 3.53 ภาพแสดงกล้อง IP

3.6 การประกอบชุดควบคุมและกล้อง

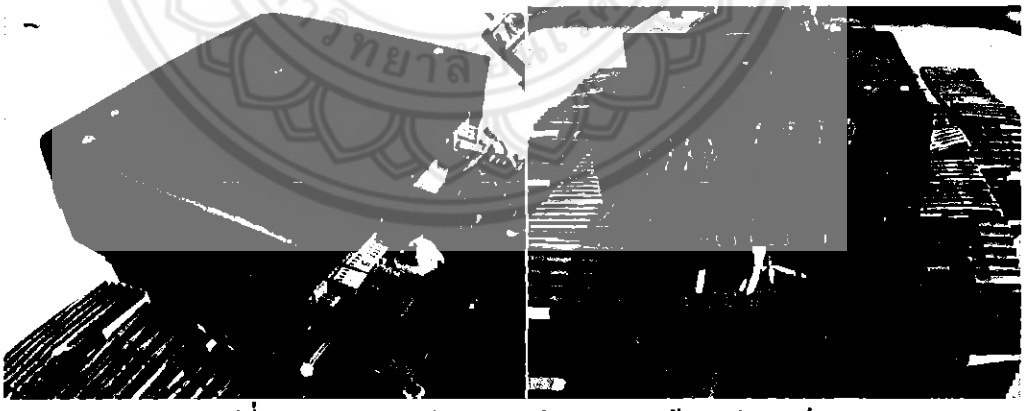
3.6.2 การติดตั้งชุดควบคุม

เมื่อสร้างวงจรและทดสอบเรียบร้อยแล้วจะต้องทำให้การติดตั้งชุดควบคุมเข้ากับกล่องอเนกประสงค์ โดยเลือกขนาดให้พอเหมาะกับขนาดของชุดควบคุม และต้องติดตั้งให้แน่นหนา เนื่องจากป้องกันการกระทบกระเทือนของวงจร



รูปที่ 3.54 แสดงการประกอบอุปกรณ์การเชื่อมต่อสายไฟ

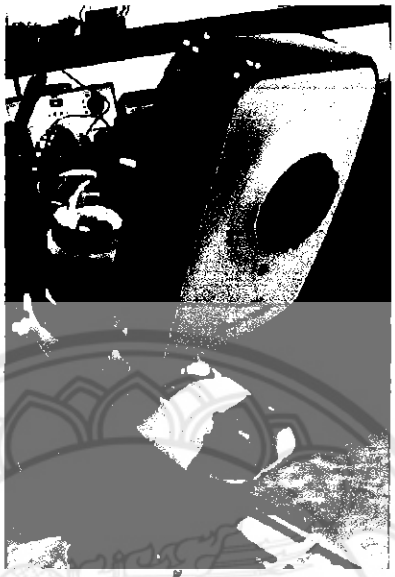
จากนั้นทำการติดตั้งกล่องควบคุมเข้ากับตัวหุ่นยนต์



รูปที่ 3.55 แสดงการประกอบกล่องควบคุมเข้ากับหุ่นยนต์

3.6.3 การติดตั้งกล้อง

การประกอบและการติดตั้งกล้อง IP เข้ากับหุ่นยนต์ โดยกล้องจะติดอยู่กับแขนของหุ่นยนต์ เนื่องจาก จะได้ปรับมุมกล้องได้หลายมุม เพื่อให้ดูง่ายในตอนบังคับหุ่นยนต์



รูปที่ 3.56 แสดงการติดตั้งกล้อง IP เข้ากับตัวหุ่นยนต์

3.7 การติดติดตั้งอุปกรณ์การติดต่อสื่อสาร

3.7.1 การติดต่อโดยใช้ Router Wireless

การใช้ Router Wireless เพื่อส่งสัญญาณกล้องจากตัวหุ่นกลับมาที่เครื่องควบคุม ซึ่ง Router ที่ใช้นี้สามารถส่งไกลได้ถึง 100 เมตรในที่โล่ง และ Router รุ่นนี้ยังสามารถปรับแต่งได้หลากหลาย ซึ่งเหมาะกับงานนี้อย่างมาก การติดตั้งนั้นจะทำการติดตั้งลงไปในตัวหุ่นยนต์ โดยจะต้องสร้างกล่องใส่ให้ใหม่ เนื่องจาก ของคู่ตัวนั้นมีขนานใหญ่เกินไป ไม่สามารถใส่กับหุ่นยนต์ได้

รูปที่ 3.57 การติดตั้ง Router Wireless เข้ากับหุ่นยนต์

3.7.2 การติต่อโดยใช้ RS-232 Wireless

การใช้ RS-232 Wireless นั้นเพื่อต้องการรับส่งข้อมูลแบบไร้สาย โดยการใช้นั้นจะต้องมี 2 ตัว คือ ทางฝั่งของหุ่นยนต์ 1 ตัว และฝั่งผู้ควบคุมอีก 1 ตัว ซึ่งก่อนใช้งานนั้นต้องทำการตั้งค่าให้ทั้ง 2 ตัวทำงานร่วมกันได้ คือต้องตั้งสัญญาณรับส่งของทั้ง 2 ตัว

รูปที่ 3.58 RS-232 Wireless

รูปที่ 3.59 แสดงรูปหุ่นยนต์กู้ภัย

ในบทต่อไปจะเป็นการทดลองในสถานการณ์ต่าง ว่าหุ่นยนต์สามารถทำได้ตามที่ต้องการหรือไม่ และเกิดปัญหาอะไรขึ้นบ้าง แล้วจะมีวิธีแก้ไขได้อย่างไร

3.8 การออกแบบการเขียนโปรแกรมติดต่อสื่อสารผ่านระบบ RS-232 Wireless

จากหลักการออกแบบระบบการทำงานของหุ่นยนต์ เมื่ออุปกรณ์ในการติดต่อสื่อสารติดตั้งเสร็จสมบูรณ์แล้ว โดยเราจะกำหนดการติดต่อสื่อสารให้สามารถติดต่อกับไมโครคอนโทรลเลอร์ได้ โดยจะแบ่งการเขียนโปรแกรมออกเป็น 2 ส่วนคือ

1. การเขียนโปรแกรมที่ตัวของไมโครคอนโทรลเลอร์
2. การเขียนโปรแกรมที่ตัวโปรแกรมควบคุม

3.8.1 การเขียนโปรแกรมที่ตัวของไมโครคอนโทรลเลอร์

โดยตัวของไมโครคอนโทรลเลอร์เราจะกำหนดให้มีการรับค่าจากโปรแกรมควบคุมโดยจะส่งเป็นตัวอักษรเช่น a b c เป็นต้น จากนั้นไมโครคอนโทรลเลอร์จะรับตัวอักษรนั้นมาเพื่อที่จะไปสั่งงานให้มีการเคลื่อนที่ต่าง ๆ ที่ถูกกำหนดไว้ และส่วนการส่งค่าของไมโครคอนโทรลเลอร์จะส่งค่าที่ได้จากการคำนวณของตัว Encoder จะนำมาใช้ในการใช้วัดระยะทางของหุ่นยนต์ส่งกับไปที่โปรแกรมควบคุมหุ่นยนต์และส่วนการส่งอีกอย่างคืออุณหภูมิจะทำการคำนวณค่าที่ได้จากอุปกรณ์เซนเซอร์อุณหภูมิจะส่งค่าไปให้โปรแกรมควบคุม

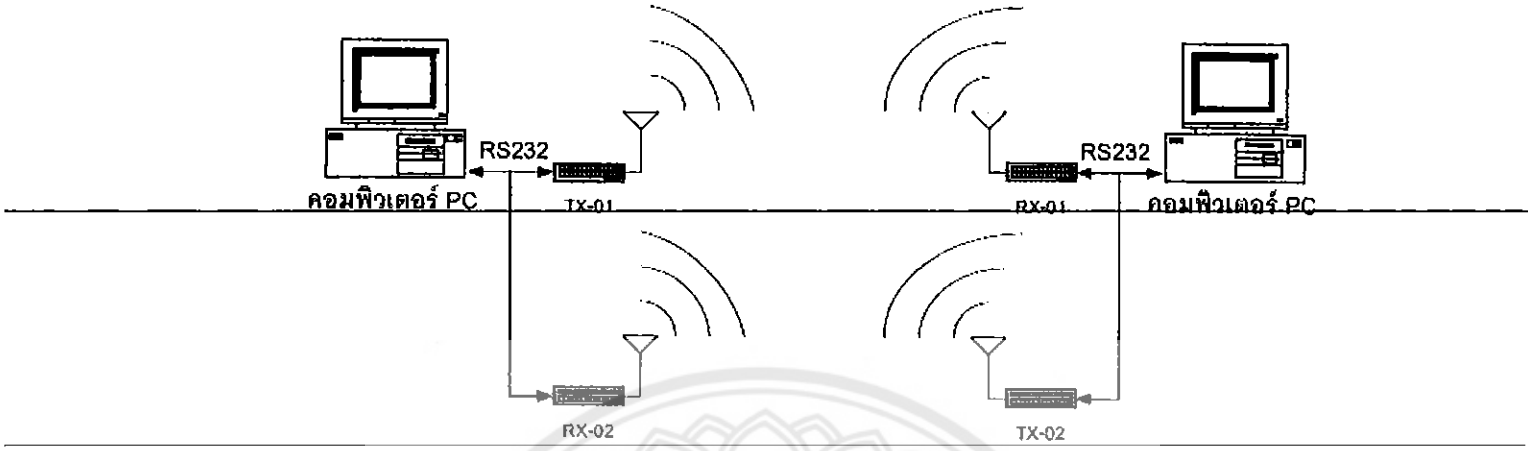
3.8.2 การเขียนโปรแกรมที่ตัวโปรแกรมควบคุม

การเขียนโปรแกรมควบคุมเราจะทำให้โปรแกรมมีการรับค่าจากไมโครคอนโทรลเลอร์ก็คือค่าที่ได้จาก Encoder และอุณหภูมิซึ่งเป็นค่าตัวเลขกับมาแสดงในส่วนที่ได้ออกแบบไว้ ส่วนการส่งค่าออกเราจะส่งค่าเป็นตัวอักษรไปให้กับไมโครคอนโทรลเลอร์โดยค่าตัวอักษรจะได้จากการกำหนดค่าด้วย Joy Stick ตามปุ่มของจอยสติกจะเป็นค่าตัวอักษรประจำปุ่ม

3.9 การติดต่อสื่อสารผ่านระบบ RS-232 Wireless

เป็นการสื่อสารที่ใช้อุปกรณ์ RS232 to RF-wireless ในสัญญาณความถี่ 2.4GHz เป็นชุดแปลงสัญญาณระหว่าง RS232 และ RF-wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ก็จะทำหน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX โดยตัวอุปกรณ์จะต้องมีการกำหนดค่าต่าง ๆ ที่เหมือนกันที่ทำให้สามารถรับส่งข้อมูลกันได้ โดยต้องมีการกำหนดอัตราความเร็วในการรับส่งส่งข้อมูลของตัวเครื่อง กำลังในในการรับส่งข้อมูล และโหมดการทำงานของช่องรับส่งข้อมูลให้

ถูกต้อง โดยการรับส่งข้อมูลจะเป็นแบบ Full Duplex คือสามารถที่จะรับส่งในเวลาเดียวกันได้พร้อม ๆ กัน โดยรูปจะแสดงดังนี้



รูปที่ 3.60 การรับส่งข้อมูลแบบ Full Duplex

โดยจะกำหนดโหมดการทำงานเป็น RF Receive Only และ RF Transmit Only ฝ่ายละ 1 ชุด ดังนี้

ตารางที่ 3.2 การกำหนดค่าของอุปกรณ์ RF แบบ Full Duplex

ค่า Configuration	ET-RF24G V2.0 ฝ่ายต้นทาง	ET-RF24G V2.0 ฝ่ายปลายทาง
	ตัวที่1	ตัวที่2
User RS232 Baud Rate	9200 Bps	9200 Bps
RF Data Rate	250 Kbps	250 Kbps
RF Operation Mode	Auto Direction	Auto Direction
RF Power Gain	+0dBm	+0dBm
RXD ID Code	01	02
TXD ID Code	02	01
RF Frequency Channel	0	0

โดยการกำหนดค่าต่าง ๆ นี้จะทำให้อุปกรณ์สามารถที่จะรับส่งข้อมูลในเวลาเดียวกันได้ เราสามารถที่จะเปลี่ยนแปลงค่าต่าง ๆ ได้โดยสังเกตที่ RXD ID Code และ TXD ID Code จะมีค่าต่อข้ามกันเสมือนแบ่งเป็นช่องในการรับส่งข้อมูลให้ถูกต้อง

3.10 การเขียนโปรแกรมในการรับภาพจากกล้อง IP Camera

การเขียนโปรแกรมควบคุมให้สามารถติดต่อกับกล้องนั้นจะใช้ตัวควบคุม ActiveX เป็นคอมโพเนนต์ที่ Internet Explorer® ใช้ ตัวควบคุมหรือ "Add-on" เหล่านี้จะปรับปรุงประสบการณ์ในการเรียกดูเว็บด้วยการอนุญาตให้มีภาพเคลื่อนไหว ซึ่งสามารถเรียกใช้ได้ในทุก ๆ เครือข่าย โดยต้องทำการติดตั้ง ActiveX ใน Internet Explorer จากนั้นเราจะเขียนโปรแกรมควบคุมให้มีการแสดงผลเหมือนกับ Internet Explorer โดยในภาษา Visual Basic 2008 จะมี Tool หรือเครื่องมือที่ใช้ในการแสดงผลเหมือน Internet Explorer

การเขียนโปรแกรมให้สามารถเรียกใช้อุปกรณ์กล้อง IP Camera นั้นเราจะใช้การเรียกใช้ผ่านโปรโตคอล HTTP โดยจะกำหนดไฟล์ที่จะสามารถแสดงภาพนั้นจะใช้ภาษา PHP สร้างไฟล์ที่กำหนดค่าที่ใช้ในการติดต่อกับกล้องโดยไม่จำเป็นจะต้องมีการจำลอง Web Server โดยเราเขียนโปรแกรมให้สามารถเรียกที่อยู่ของไฟล์ที่สร้างขึ้น โดยตัวไฟล์จะมีที่อยู่หรือ IP กล้องและค่าต่าง ๆ ดังได้แสดงในภาคผนวก

3.10 การเขียนโปรแกรมติดต่อกับจอยสติก Joy Stick

การที่ใช้จอยสติกในการควบคุมก็เพื่อที่จะให้ผู้ควบคุมสามารถที่จะควบคุมได้ง่ายและคนอื่นสามารถเข้าใจง่ายเพราะคล้ายกับเกม โดยการเขียนโปรแกรมเราจะใช้ DirectX เป็นซอฟต์แวร์ที่บริษัทไมโครซอฟต์เป็นผู้เขียนขึ้นมาสำหรับช่วยในการรองรับการติดต่อกับระบบ Hardware ได้โดยตรง โดยไม่ต้องผ่านระบบปฏิบัติการทำให้สามารถเข้าถึงคุณสมบัติต่าง ๆ ของ Hardware ได้ โดยเราจะใช้ DirectX SDK ซึ่งเป็นส่วนของผู้พัฒนาโปรแกรมโดย DirectX SDK เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมโดยไมโครซอฟต์พัฒนา DirectX SDK สำหรับใช้การพัฒนาโปรแกรมในหลายภาษาเช่น C/C++, C# และ Visual Basic .NET โดยผู้พัฒนาโปรแกรมสามารถ Download ได้ฟรีจากเว็บไซต์ของ Microsoft โดยในชุดของ DirectX SDK ซึ่งสิ่งเราใช้คือ Direct Input จะประกอบด้วยฟังก์ชั่นในการรับข้อมูลจากผู้ใช้ผ่านอุปกรณ์ต่าง ๆ เช่น Mouse, Keyboard และ Joystick เป็นต้น

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบการทำงานของโปรแกรมและหุ่นยนต์กู้ภัยที่พัฒนาขึ้นมา ว่าสามารถทำงานได้ตามจุดประสงค์ที่ได้ต้องไว้อย่างไร สารถสั่งการทำงานควบคุมหุ่นยนต์แบบไร้สายได้ จากนั้นหาข้อผิดพลาดของโปรแกรมและหุ่นยนต์กู้ภัยที่พัฒนาขึ้นและนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป

4.1 ขั้นตอนการสอบของหุ่นยนต์

ขั้นตอนการทดลองหุ่นยนต์นั้นจะแบ่งเป็นการทดลองโปรแกรม และการทดลองตัวหุ่นยนต์

4.1.1 การทดลองโปรแกรม

การทดลองโปรแกรมที่พัฒนานั้นจะแบ่งเป็น 2 ส่วนด้วยกัน คือ

4.1.1.1 การเขียนโปรแกรมในไมโครคอนโทลเลอร์

4.1.1.2 การเขียนโปรแกรมควบคุมและติดต่อกับไมโครคอนโทลเลอร์

4.1.2 การทดลองตัวหุ่นยนต์

ในการทดลองตัวหุ่นยนต์ จะมีสนามในการทดสอบในแต่ละพื้นที่ เช่น ทางขรุขระ ทางขึ้นหรือลงบันได ทางลาดในระดับต่างๆ การทดสอบการเคลื่อนที่ เช่น เดินหน้า ถอยหลัง เลี้ยวซ้าย เลียวขวา เป็นต้น และทำการทดสอบเซนเซอร์ต่างๆที่ได้ติดตั้งบนตัวหุ่นยนต์กู้ภัย

รูปที่ 4.1 สนามที่ใช้ในการทดลองหุ่นยนต์กู้ภัย

4.2 การทดสอบโปรแกรมที่พัฒนาขึ้นมา

โปรแกรมการติดต่อและควบคุมหุ่นยนต์ เมื่อเปิดโปรแกรมขึ้นมาจะมีช่องให้ใส่ Address ของกล้อง IP เพื่อจะให้กล้องเข้ามาแสดงผลบนโปรแกรม และมีช่องให้ใส่ค่าของพอร์ตที่จะติดต่อ เพื่อที่จะทำการติดต่อควบคุมหุ่นยนต์

ในส่วนแรกจะเป็นส่วนของการติดต่อ Serial Port ซึ่งจะเป็นการติดต่อกับหุ่นยนต์ โดยจะมีส่วนต่างๆ ดังนี้

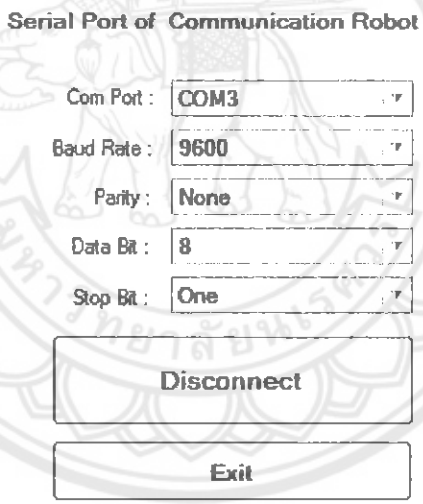
4.2.1 Com Port จะเป็นการกำหนดว่าจะติดต่อทาง RS232 ที่ Port ไหน

4.2.2 Baud Rate จะเป็นการกำหนดว่าจะใช้ความเร็วในการรับส่งข้อมูลเท่าไหร่

4.2.3 Parity จะเป็นการกำหนดความสำคัญในการติดต่อ

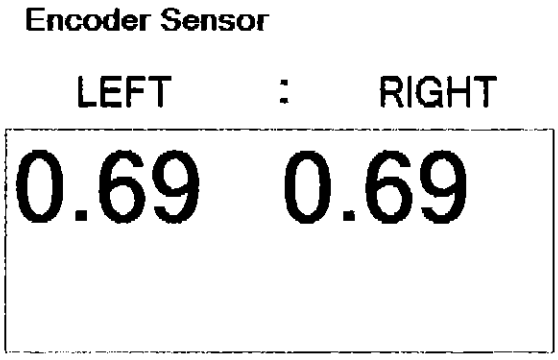
4.2.4 Data Bit จะเป็นการกำหนดอัตราการส่งข้อมูลว่าจะใช้กี่ bit

เมื่อใส่ค่าต่างๆเรียบร้อยแล้ว จะกดปุ่ม Connect เพื่อทำการติดต่อกับหุ่นยนต์ ถ้าต้องการยกเลิกการติดต่อก็ให้กดปุ่ม Disconnect



รูปที่ 4.2 แสดงภาพโปรแกรมติดต่อ Serial Port

ใสส่วนต่อไปจะเป็นส่วนของการวัดระยะทาง ซึ่งจะใช้ Encoder ในการวัดระยะทาง จะแสดงผลออกมาในหน่วยเมตร

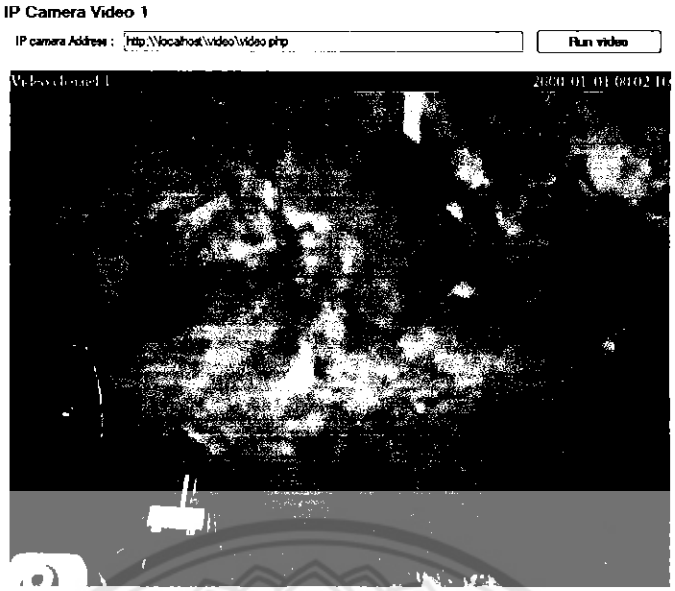


รูปที่ 4.3 การแสดงผลการวัดระยะทาง

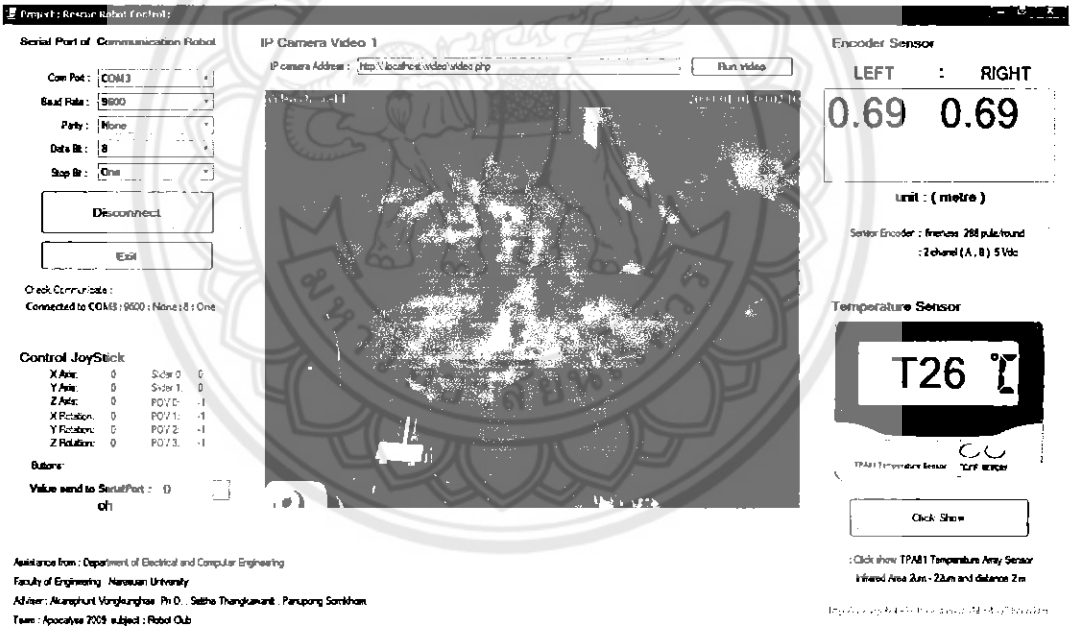
การวัดอุณหภูมิ ก็จะเป็นการแสดงผลของอุณหภูมิซึ่งมีหน่วยเป็น องศา โดยถ้าต้องการการวัดอุณหภูมินั้นสามารถทำได้โดยกดปุ่ม Click Show จากนั้นอุณหภูมิก็จะแสดงขึ้นมา

รูปที่ 4.4 แสดงผลอุณหภูมิ

ในส่วนการแสดงผลของกล้องนั้น จะเป็นการติดต่อผ่านทาง Router Wireless ถ้าต้องการติดต่อกัลป์กล้องนั้นสามารถทำได้โดยต้อง Connect กับ Router Wireless ก่อน จากนั้นทำการกดปุ่ม Run Video กล้องก็จะแสดงผลออกมา



รูปที่ 4.5 การแสดงผลกล้องผ่าน Router Wireless



รูปที่ 4.6 การแสดงผลโปรแกรมควบคุมหุ่นยนต์กู้ภัย

4.3 การทดสอบตัวหุ่นยนต์

4.3.1 เดินหน้า – ถอยหลัง

การทดลองการเดินหน้าและถอยหลังของหุ่นยนต์ทำได้ตามที่เขียนโปรแกรมไว้

4.3.2 การเลี้ยวของหุ่นยนต์

การทดลองการเลี้ยวของหุ่นยนต์นั้นจะมี 2 แบบ คือ เลี้ยวแบบมุมกว้าง และเลี้ยวมุมแคบ

4.3.2.1 การเลี้ยวแบบมุมกว้าง คือการเลียวโดยใช้ล้อข้างเดียวส่วนอีกล้อจะหยุดนิ่ง เช่นจะเลี้ยวขวา ก็ให้ล้อขวาหมุนไปข้างหน้า ส่วนล้อซ้ายนั้นจะหยุดนิ่ง จะเป็นการเลี้ยวแบบมุมกว้าง

4.3.2.2 การเลี้ยวแบบมุมแคบ คือการเลียวโดยใช้ล้อทั้ง 2 ข้างหมุนไปในทางทิศตรงข้ามกัน เช่น เลี้ยวขวา ก็ให้ล้อหมุนไปข้างหน้า ส่วนล้อซ้ายจะหมุนไปข้างหลัง จะเป็นการเลี้ยวแบบมุมแคบ การเลียวแบบมุมแคนจะเลี้ยวได้เร็วกว่ามุมกว้าง

4.3.3 การเคลื่อนที่แขนล้อของหุ่นยนต์

หุ่นยนต์สามารถใช้แขนล้อยกตัวเองได้ และใช้แขนล้อช่วยพยุงตัวในตอนที่หุ่นยนต์เคลื่อนที่ในสภาพผิวที่ขรุขระได้เป็นอย่างดี แต่แขนของหุ่นยนต์นั้นทำงานช้ามาก เป็นเพราะอัตราการทดของเฟืองโซ่ เพราะจะได้แรงบิดที่สูงแต่จะแรกกับความเร็วที่เสียไป

4.3.4 การเคลื่อนที่แขนของหุ่นยนต์

หุ่นยนต์สมมารถยกแขนได้ตามต้องการ แต่จะมีปัญหาตรงเมื่อหุ่นเคลื่อนที่จะเกิดการสั้นของแขน ทำให้แขนไม่นิ่ง จึงต้องทำการแก้ไขโดยการใส่ตัวซับแรงกระแทกเข้าไป ซึ่งดัดแปลงมาจากตัวกันสะบัดของรถจักรยานยนต์แล้วนำมาใส่เข้ากับหุ่น ทำให้แขนของหุนนั่นนิ่งมากขึ้น

4.3.5 การหมุนกล้อง

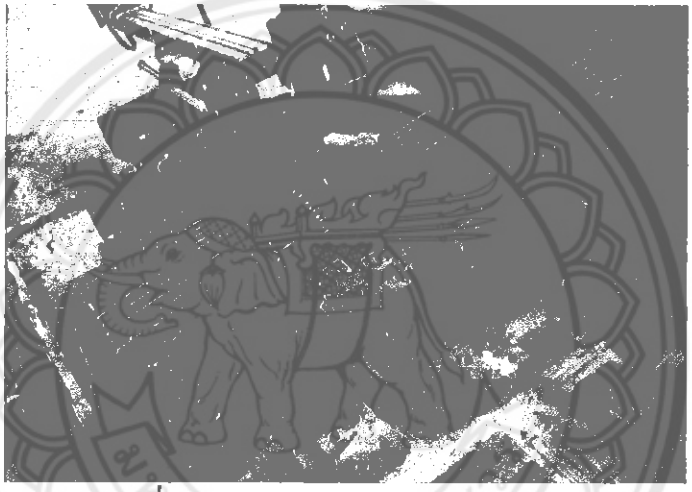
การหมุนกล้องนั้น สามารถหมุนได้ตามความต้องการ แต่จะมีปัญหาของมอเตอร์ตรงที่เพลาของมอเตอร์หลวม ทำให้ตอนที่หุ่นยนต์เคลื่อนที่นั้นทำให้ภาพสั่น ซึ่งจะแก้ไขโดยการทำให้เฟืองที่หมุนนั้นขบกันให้แน่นที่สุดจะทำให้ลดการสั่นสะเทือนลงไป แต่ก็จะทำให้กินแรงของมอเตอร์มากขึ้นด้วย

4.4 การทดลองการเคลื่อนที่ของหุ่นยนต์ในสภาพแวดล้อมต่างๆ

4.4.1 การเคลื่อนที่ในสภาพผิวที่ขรุขระ

หุ่นยนต์สามารถเคลื่อนที่ได้ดี และใช้แขนล้อช่วยเป็นบางพื้นที่ที่สูงจากพื้นไม่เกิน 10 cm. แต่ถ้ามีพื้นที่ที่ขรุขระสูงมาก ซึ่งสูงกว่า 10 cm. นั้น หุ่นยนต์ก็พอที่จะเคลื่อนที่ไปได้ได้ แต่ต้องใช้แขนล้อช่วยพยุงตัวตลอด และอาจมีติดบ้างในบางที่ เนื่องจากล้อตีนตะขาบเข้าไปติดแทรกในอิฐทำให้ล้อไม่สามารถขยับได้ การปัญหาในข้อนี้ต้องเปลี่ยนและออกแบบล้อตีนตะขาบใหม่ทั้งหมด

รูปที่ 4.7 การเคลื่อนที่ในสภาพผิวที่ขรุขระ

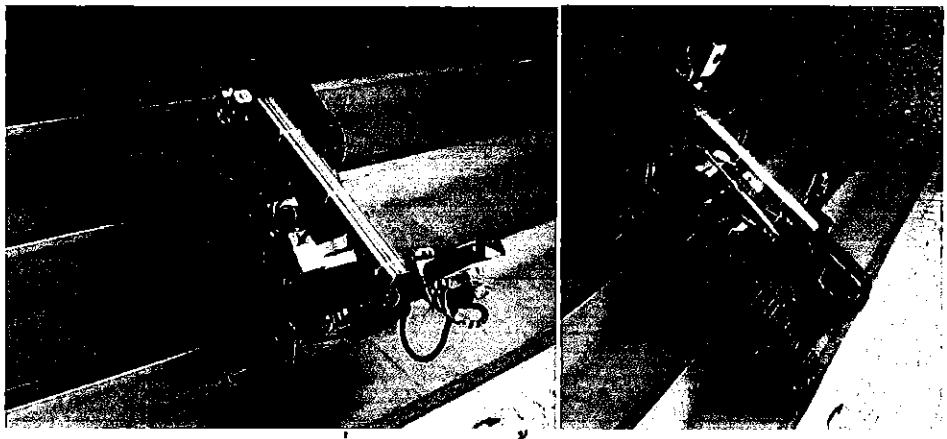


รูปที่ 4.8 การใช้แขนล้อช่วยในทางขรุขระ

จากรูปที่ 4.7 ต้องใช้แขนล้อช่วยพยุงตัวขึ้น เนื่องจากก้อนอิฐที่ยื่นออกมาทำให้ใต้ท้องของหุ่นยนต์ติด ไม่สามารถเคลื่อนที่ไปข้างหน้าได้

4.4.2 การเคลื่อนที่ขึ้น – ลงบันได

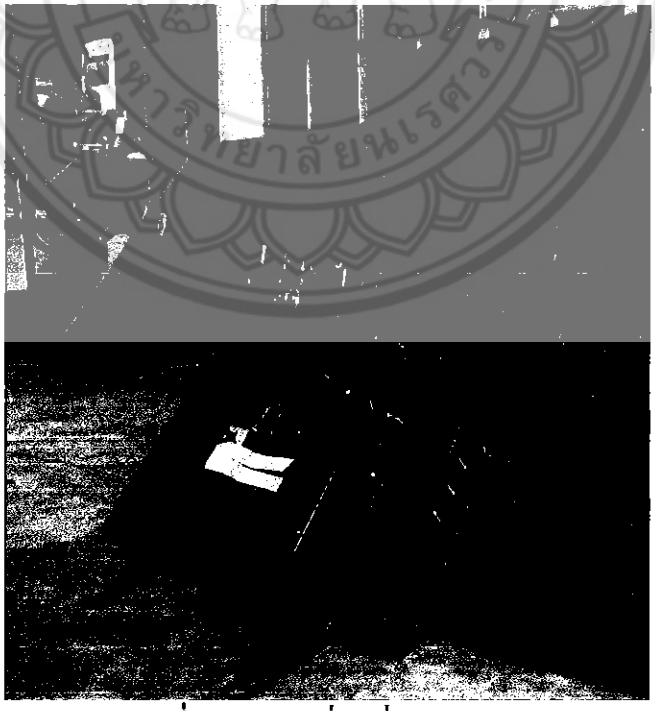
หุ่นยนต์สามารถขึ้นบันไดได้ แต่ถ้าขั้นบันไดสูงเกินกว่า 20 cm. หุ่นจะไม่สามารถขึ้นบันไดได้ เนื่องจาก ตัวหุ่นยนต์นั้นสั้นและน้ำหน้าข้างหน้านั้นมีขนาดเบา เวลาหุ่นขึ้นบันไดที่มีขั้นบันไดสูงมากกว่า 20 cm. จะทำให้หุ่นยนต์หงายท้อง



รูปที่ 4.9 แสดงการขึ้นลงบันได

4.4.3 การเคลื่อนที่ในทางลาด

จากการทดลองในทางลาดขนาดต่างๆ หุ่นยนต์เคลื่อนที่ค่อนข้างจะลำบาก เนื่องจากยางที่ทำเป็นล้อของหุ่นยนต์นั้นมีแรงเสียดทานน้อยมากทำให้เวลาขึ้นหรือลงทางลาด ทำให้หุ่นยนต์ลื่นและเสียหลักในการควบคุม วิธีแก้ไขนั้นทำได้โดยการ ปรับท่าการเดินของหุ่นยนต์โดยการใช้แขนล้อเป็นตัวช่วย ปรับแขนล้อให้เป็นท่าทิ่มลงไปดังรูป จะทำให้หุ่นยนต์เคลื่อนที่ได้ดีขึ้น และการแก้ไขอีกวิธีคือการต้องเปลี่ยนยางล้อของหุ่นยนต์ใหม่



รูปที่ 4.10 การเคลื่อนที่ในทางลาด

4.5 การทดสอบเซนเซอร์

4.5.1 เซนเซอร์อุณหภูมิ

ทำการทดสอบอุณหภูมิในลักษณะต่างๆ ในสภาพแวดล้อมที่ต่างกันไป และระยะทางที่สามารถตรวจจับได้ จะได้ผลตามตารางที่ 4.1

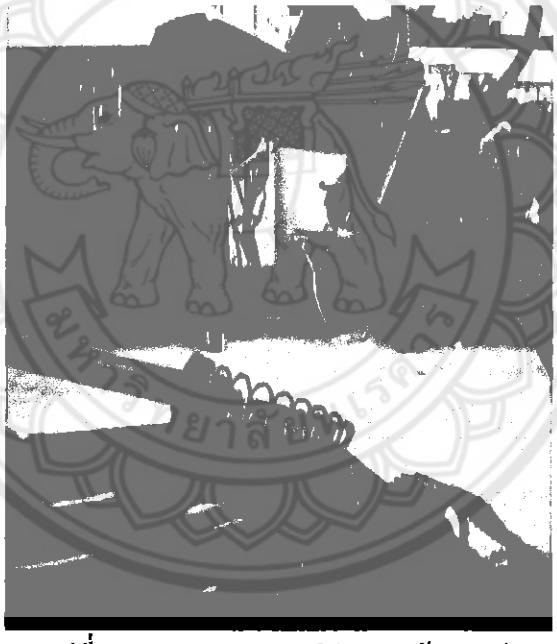
ตารางที่ 4.1 ผลการทดสอบเซนเซอร์อุณหภูมิ

สภาพแวดล้อม	อุณหภูมิ (องศาเซลเซียส)	ระยะ (เซนติเมตร)	หมายเหตุ
แก้วน้ำแข็ง	0 - 2	10	ห้องแอร์
แก้วน้ำแข็ง	8 - 14	20	ห้องแอร์
หัวแร้งบัดกรี	147 - 162	10	ห้องแอร์
หัวแร้งบัดกรี	117 - 134	20	ห้องแอร์
มือคน	31 - 33	10	ห้องแอร์
หน้าคน	32 - 34	10	ห้องแอร์
รอบห้อง	25 - 29	-	ห้องแอร์

จากตารางที่ 4.1 ผลการทดลองนั้น ถ้าระยะไกลออกไปมากๆ ค่าของอุณหภูมิก็จะเปลี่ยนไปด้วย แต่ถ้าระยะไม่เกิน 10 - 15 cm. จะให้ค่าที่ใกล้เคียงกับอุณหภูมิจริงมาก

รูปที่ 4.11 การวัดอุณหภูมิกับแก้วน้ำแข็ง

รูปที่ 4.12 การวัดอุณหภูมิบนมือคน



รูปที่ 4.13 การวัดอุณหภูมิกับหัวแร้งบัดกรี

4.5.2 Encoder

Encoder จะใช้เป็นการตรวจสอบระยะทางและความเร็วของล้อทั้ง 2 ข้างขณะที่หุ่นยนต์เคลื่อนที่ ความละเอียดของ Encoder จะเท่ากับ 288 ในโปรแกรมจะนับทุกๆครั้งเมื่อ Encoder นับได้ 16 ดังนั้น 1 รอบของ Encoder ตัวไมโครคอนโทลเลอร์จะนับได้ $\frac{288}{16} = 18$ ครั้ง

การทดลองจะทำการทดลองในระยะต่างๆ เพื่อตรวจสอบดูว่า Encoder จะบอกระยะทางได้เท่ากันทุกครั้งหรือไม่ ผลการทดลองเป็นไปตามตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองการวัดระยะทางโดยใช้ Encoder

ระยะทางจริง (cm.)	จำนวนพัลส์ของ Encoder	ระยะทาง / จำนวนพัลส์
50	7	7.14
100	15	6.67
150	23	6.52
200	30	6.67
250	37	7.14
300	42	7.00
350	50	6.90
400	58	6.82
450	66	6.85
500	73	6.89
600	87	6.90
700	101	6.93
800	115	6.96
900	132	6.82
1000	147	6.80
เฉลี่ย		6.87

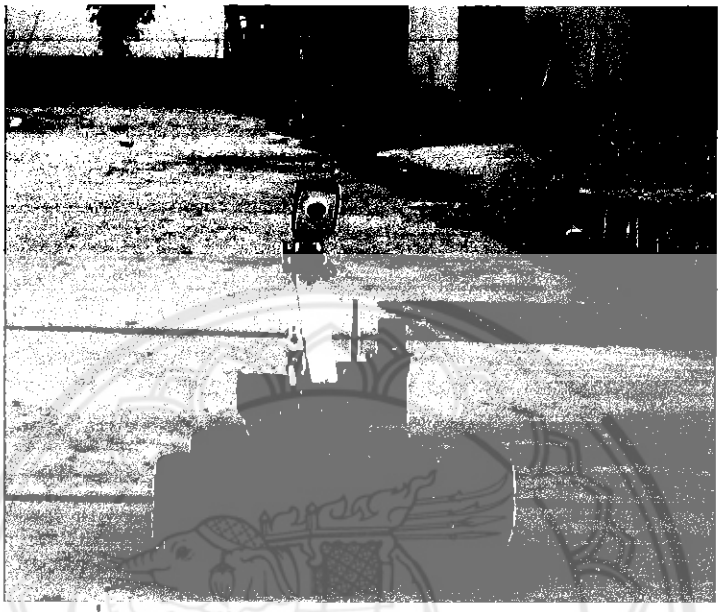
จากตารางที่ 4.2 จากการหาค่าเฉลี่ยของ ระยะทาง / จำนวนพัลส์ ซึ่งจะได้ 6.87 ซึ่งแสดงว่าใน 1 พัลส์ของ Encoder นั้นจะได้ระยะทางเท่ากับ 6.87 เซนติเมตร ซึ่งจะนำตัวเลขนี้ไปใส่ในโปรแกรมเพื่อที่จะสามารถรู้ได้ว่าหุ่นยนต์วิ่งไปได้ระยะทางเท่าไหร่

4.5.3 เซนเซอร์ตรวจวัดระยะทางด้วยแสงอินฟราเรด

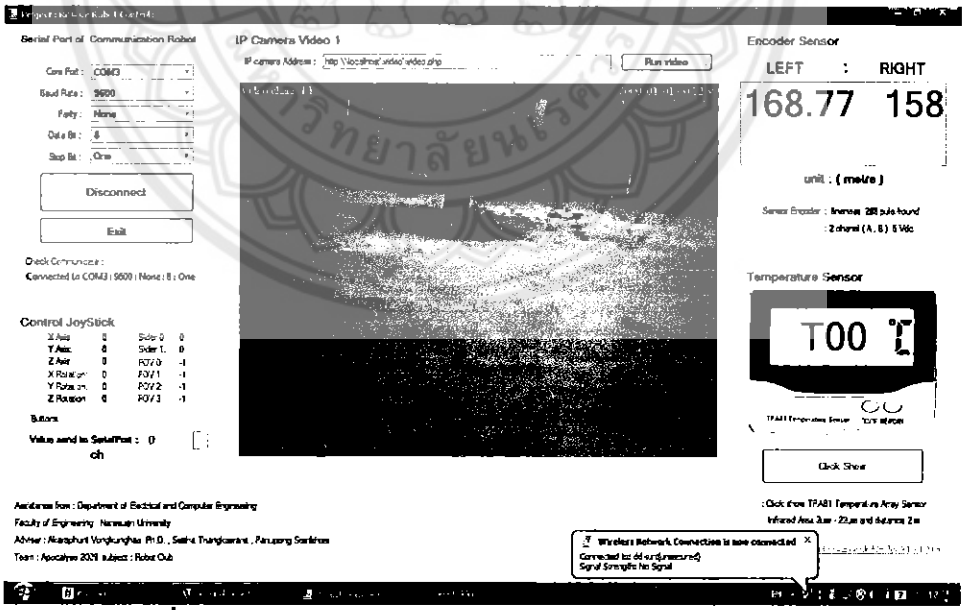
การใช้เซนเซอร์วัดระยะทางนั้น เพื่อตรวจสอบว่าห่างจากสิ่งกีดขวางเป็นระยะทางเท่าไหร่ เพราะบางครั้งการดูจากมุมกล้องอย่างเดียวนั้นอาจไม่เพียงพอ อาจมีบางมุมที่มองไม่เห็น จึงได้นำตัวตรวจวัดระยะทางเข้ามาใช้ช่วยในการควบคุมด้วย ซึ่งจะติดไว้ทางด้านหลังของหุ่นยนต์ซึ่งจะได้บอกระยะห่างได้ โดยไม่ต้องหมุนกล้องมาดู

4.6 การทดสอบระยะที่สามารถควบคุมได้

การทดสอบระยะที่สามารถควบคุมได้นั้นจะทำการทดสอบในพื้นที่โล่ง โดยจะบังหุ่นเดินตรงไปเรื่อยๆ และระหว่างที่บังคับหุ่นยนต์ไปนั้นจะทำการตรวจสอบสัญญาณไปด้วย



รูปที่ 4.14 การทดสอบระยะในการควบคุมหุ่นยนต์กู้ภัย



รูปที่ 4.15 แสดงผลการทดสอบระยะในการควบคุมหุ่นยนต์

จากการทดลองหุ่นยนต์สามารถบังคับได้ไกลถึง 168 เมตรในพื้นที่โล่ง แต่จากการทดลองนั้น ถ้าไปไกลกว่า 100 เมตร สัญญาณภาพที่ส่งมาจะมีการกระตุก แต่ก็พอที่จะสามารถบังคับไปได้

4.7 ปัญหาและการแก้ไขปัญหาที่พบในการทดลอง

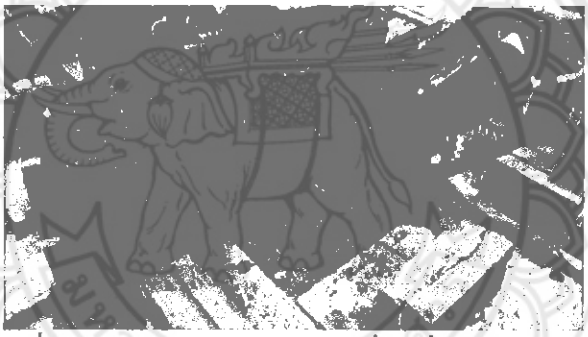
4.7.1 เคลื่อนที่ทางขรุขระ

ปัญหา

จากการทดลองในสนามที่ได้จัดเตรียมไว้ ในทางขรุขระหุ่นยนต์เคลื่อนที่ได้ลำบาก เนื่องจากใต้ท้องของหุ่นยนต์มีความเตี้ยเกินไป และช่วงระยะห่างระหว่างล้อทั้ง 2 ข้างนั้นมากเกินไป ทำให้เวลาเคลื่อนที่หุ่นยนต์จะติดใต้ท้องรถ และในส่วนความกว้างของล้อตีนตะขาบมีขนาดเล็กเกินไป ทำให้ไปติดกับล่องของสนาม ทำให้เคลื่อนที่ไม่ได้

การแก้ไขปัญหา

การแก้ไขปัญหานี้ต้องทำการออกแบบช่วงล่างของหุ่นยนต์ และตีนตะขาบใหม่ โดยให้หน้าล้อตีนตะขาบนั้นมาขนาดกว้างขึ้น จะทำให้ใต้ท้องของหุ่นยนต์มีช่องแคบลง จะช่วยลดปัญหาการติดใต้ท้องได้ และยังช่วยการที่ล้อตีนตะขาบเข้าไปติดได้อีกด้วย



รูปที่ 4.16 แสดงภาพปัญหาการเคลื่อนที่ในทางขรุขระ

4.7.2 การมองผ่านกล้อง

ปัญหา

จากการทดลองเวลาบังคับหุ่นยนต์ผ่านกล้องนั้น มุมมองในการมองจะแคบมาก เพราะมีกล้องเพียงตัวเดียว และชุดฐานหมุนกล้องได้มีความหลวมซึ่งเกิดจากเฟืองและมอเตอร์ ทำให้เวลาเคลื่อนที่นั้นกล้องจะสั่น ยิ่งทำให้การมองนั้นลำบากยิ่งขึ้นอีก

การแก้ไขปัญหา

การแก้ไขปัญหานี้ต้องทำการเปลี่ยนเฟืองเกียร์ของมอเตอร์ใหม่ และใช้ตัวซับแรงกระแทกเข้าไปช่วย ซึ่งจะใช้กันสะบัดของรถจักรยานยนต์ จะทำให้ช่วยซับแรงกระแทกได้มาก และต้องทำการเพิ่มกล้องเพื่อเพิ่มมุมในการมองเห็นตอนบังคับหุ่นยนต์ แต่การเพิ่มกล้องเข้าไปนั้นก็จะทำให้หุ่นยนต์นั้นกินไฟมากขึ้น เพราะกล้อง IP 1 ตัวใช้กระแสไฟถึง 2 แอมแปร์

4.7.3 การเคลื่อนที่ทางลาด

ปัญหา

จากากรทดลองเวลาหุ่นยนต์ขึ้นทางลาด ซึ่งถ้าทางลาดมีความชันมาก หุ่นยนต์จะไม่สามารถขึ้นไปได้ เนื่องจากยางล้อตีนตะขาบของหุ่นยนต์มีความลื่น เวลาขึ้นจะทำให้ล้อฟรีไม่สามารถขึ้นได้ เช่นรูปที่ 4.9 นั้นทางลาดชันมาดวามชันมาก หุ่นยนต์ไม่สามารถขึ้นไปได้

การแก้ไขปัญหา

การแก้ไขปัญหานี้ต้องเปลี่ยนยางของล้อตีนตะขาบใหม่ เพื่อเพิ่มแรงเสียดทานให้กับพื้นและล้อของหุ่นยนต์

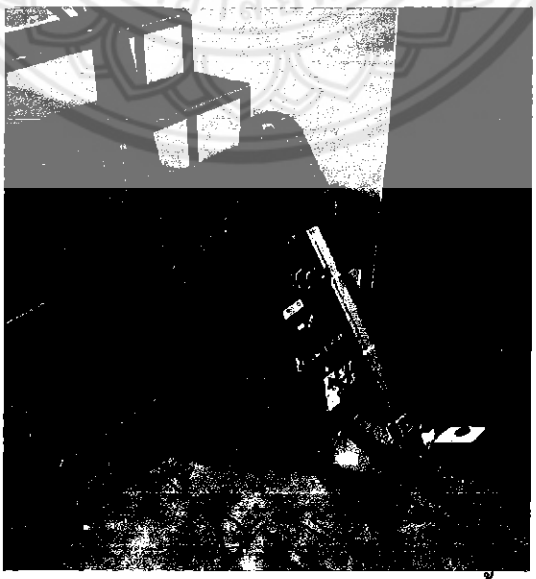
4.7.4 การขึ้นบันได

ปัญหา

จากการทดลองการขึ้นบันไดในสนามที่ได้สร้างขึ้นมานั้น ขั้นของบันไดมีความสูง 20 เซนติเมตร หุ่นยนต์ไม่สามารถขึ้นบันไดได้ ตอนขึ้นหุ่นยนต์จะหงายท้อง แต่ถ้าขั้นบันไดสูงไม่เกิน 18 เซนติเมตรหุ่นยนต์ก็สามารถขึ้นไปได้

การแก้ไขปัญหา

จากปัญหานั้นหุ่นยนต์ในช่วงหน้านั้นมีน้ำหนักเบา ซึ่งทำให้ตอนขึ้นหุ่นยนต์จึงหงายท้อง ถ้าแก้ปัญหาเฉพาะหน้านั้น ก็ทำการถ่วงให้ข้างหน้าหนักขึ้น แต่ถ้าแก้ปัญหาที่ดีควรออกแบบใหม่ โดยให้แขนล้อมีความยาวมากขึ้น ซึ่งจะช่วยเพิ่มน้ำหนักและเพิ่มระยะแขนออกไปเกาะกับบันได



รูปที่ 4.17 แสดงการหงายท้องของหุ่นยนต์ตอนขึ้นบันได

ข้อมูลหุ่นยนต์

น้ำหนัก	32	กิโลกรัม
ความสูง (ยืดแขนสุด)	95	เซนติเมตร
ความสูง (ไม่ยืดแขน)	46	เซนติเมตร
ความกว้าง	55	เซนติเมตร
ความยาว (ยืดแขนล้อ)	75	เซนติเมตร
ความยาว (ไม่ยืดแขนล้อ)	58	เซนติเมตร

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

1. สามารถแสดงผลภาพจากกล้องที่ส่งเข้ามาแบบไร้สายได้ โดยระยะทางระหว่างผู้ควบคุมกับตัวหุ่นยนต์นั้นจะมีระยะไกลได้ไม่เกิน 168 เมตร ถ้าเกิน 168 เมตรสัญญาณจะขาดหายไปและไม่สามารถติดต่อกล้องกับตัวหุ่นยนต์ได้

2. สามารถควบคุมมอเตอร์ด้วยคอมพิวเตอร์ผ่านชุดไมโครคอนโทรลเลอร์ MCS51 แบบไร้สายได้ โดยระยะทางระหว่างผู้ควบคุมกับตัวหุ่นยนต์นั้นจะมีระยะไกลได้ไม่เกิน 180 เมตร ซึ่งในการควบคุมนั้นต้องอาศัยการดูภาพจากกล้อง ซึ่งการติดต่อกล้องนั้นมีระยะไม่เกิน 168 เมตร ถ้าเกินก็จะมองไม่เห็นภาพ แต่ก็ยังสามารถสั่งการควบคุมได้ แต่จะมองไม่เห็น ดังนั้นการควบคุมจึงควบคุมจึงไม่ควรเกินระยะ 168 เมตรในพื้นที่โล่ง

3. หุ่นยนต์สามารถตรวจจับ อุณหภูมิของเป้าหมาย และอุณหภูมิบริเวณรอบข้างได้ โดยการตรวจจับอุณหภูมิแต่ละครั้ง จะต้องมีระยะไม่เกิน 10 เซนติเมตร ถ้าเกินระยะ 10 เซนติเมตรจะทำให้ค่าของอุณหภูมินั้นเปลี่ยนไปไม่ตรงกับอุณหภูมิจริง

4. สามารถตรวจสอบระยะทางของล้อทั้งสองข้างได้มีหน่วยเป็นเมตร ค่าความละเอียด 0.068 เมตร

5.2 ปัญหาที่พบ

1. การหาวัดสุอุปกรณ์ที่เหมาะสมในการสร้างหุ่นยนต์ ซึ่งหายากและไม่เหมาะสมกับหุ่นยนต์

2. การใช้เครื่องมือเครื่องมือช่าง เช่น เชื่อมไฟฟ้า กลึง ยังขาดประสบการณ์และความชำนาญ

3. ข้อจำกัดในเรื่องการแข่งขัน เกี่ยวกับน้ำหนัก ขนาด ของตัวหุ่นยนต์

4. การผลิตชิ้นส่วนมาประกอบกันมักจะไม่พอดี เช่น รูเจาะ ขนาด

5. ยางที่ติดกับตีนตะขาบไม่เกาะกับพื้นทางลาดชัน

6. งบประมาณที่จำกัดในการสร้างหุ่นยนต์ ซึ่งอุปกรณ์แต่ละชิ้นส่วนมีราคาสูงในการทดสอบสร้างจึงมีความเสี่ยงเกิดขึ้นถ้าหุ่นยนต์สามารถปฏิบัติงานได้จริง

7. แบตเตอรี่หมดเร็ว เนื่องจากการกล้องที่ติดกับหุ่นยนต์ใช้กระแสไฟมากและต่อเนื่อง

8. ประสบการณ์ในการออกแบบตัวหุ่นยนต์ เนื่องจากไม่ได้ศึกษาทางด้านระบบเครื่องกลจึงทำให้การออกแบบหุ่นยนต์ ไม่เป็นไปตามทฤษฎีต่าง ๆ ที่เกี่ยวข้อง

5.3 แนวทางการแก้ปัญหาและข้อเสนอแนะ

1. ข้อเสนอแนะสำหรับการหาซื้อวัสดุในการสร้างหุ่นยนต์ จากการปฏิบัติงานที่ผ่านมาพบว่า ยังมีปัญหาและอุปสรรค์พอสมควร เช่น การจัดหาซื้อวัสดุอุปกรณ์ ซึ่งวัสดุที่ใช้ไม่ได้มีขายตามตลาดทั่วไป และไม่ได้แบ่งขาย ทำให้ต้องหาวัสดุมาทดแทนกัน เช่นจากที่จะใช้อลูมิเนียมในบางที่ต้องเปลี่ยนมาใช้เป็นเหล็กแทน

2. ข้อเสนอแนะในการติดตั้งอุปกรณ์ต่างๆ เช่นตัวหนอนและเฟืองตัวหนอนเนื่องจากต้องขบกันแน่นพอดี หากตัวหนอนและเฟืองตัวหนอนขบกันไม่พอดีจำทำให้กลไกลนั้นไม่ทำงาน และอาจทำให้มอเตอร์ทำงานหนักและมีเสียงดัง ซึ่งต้องแก้ปัญหา โดยการติดตั้งชุดเฟืองและมอเตอร์ใหม่ทั้งหมด อาจจะทำให้เสียเลาพอสมควร

3. การเลือกใช้ยางและชนิดของยางที่เหมาะสมเพื่อเพิ่มแรงเสียดทานให้กับล้อของหุ่นยนต์ ซึ่งยางที่สั่งทำจากโรงงานหรือยางที่มีคุณภาพดีนั้นจะมีราคาแพง

4. การติดตั้งอุปกรณ์แต่ละขั้นตอนไม่ควรเร่งรีบ เพราะถ้าไม่ดีแล้วการแก้ไขนั้นจะทำได้ยากและเสียเวลานาน หรืออาจทำให้อุปกรณ์นั้นเสียหายได้

5.4 แนวทางในการพัฒนาเพิ่มเติม

5.4.1 แนวทางการพัฒนาในส่วนของตัวหุ่นยนต์

1. การติดตั้ง เข็มทิศอิเล็กทรอนิกส์ เพื่อแสดงเส้นทางและระบุตำแหน่งผู้ประสบภัยได้อย่างชัดเจน

2. การติดตั้งอุปกรณ์ตรวจจับสภาพแวดล้อมในบริเวณรอบตัวของผู้ประสบภัย เช่น ติดตั้งอุปกรณ์ตรวจวัดก๊าซคาร์บอนไดออกไซด์ อุปกรณ์ตรวจจับความเคลื่อนไหวของผู้ประสบภัย อุปกรณ์ตรวจรับเสียงของผู้ประสบภัย เป็นต้น

3. การตรวจสอบแบตเตอรี่และการสำรองแบตเตอรี่ เพื่อป้องกันไม่ให้หุ่นยนต์ติดอยู่ในภารกิจและสามารถเคลื่อนที่ออกจากภารกิจได้เมื่อแบตเตอรี่ใกล้หมด

4. ติดตั้งกล้องเพิ่ม เพื่อเพิ่มมุมมองให้แก่คนบังคับ ทำให้เห็นสภาพแวดล้อมได้กว้างขึ้น

5. ติดตั้ง Encoder ให้กับมอเตอร์ทุกจุด เพื่อที่จะสามารถบอกลักษณะของหุ่นยนต์ได้อย่างชัดเจน ซึ่งจะทำให้ตอนบังคับรู้ได้ว่าหุ่นยนต์อยู่ในลักษณะใด

5.4.2 แนวทางการพัฒนาในส่วนของการควบคุมและการเชื่อมต่อ

1. การพัฒนาระบบควบคุม Microcontroller เป็นประเภทอื่นหรือตระกูลอื่นที่มีความเร็วมากขึ้น เช่น ARM7, AVR, PIC

2. เปลี่ยนอุปกรณ์การสื่อสารแบบไร้สาย เพื่อให้ควบคุมได้ไกลมากยิ่งขึ้น เช่น การใช้เทคโนโลยี 3G (Third Generation) เข้ามาควบคุมหุ่นยนต์ซึ่งจะบังคับได้ไกลมาก สามารถบังคับที่ไหนก็ได้ที่มีสัญญาณโทรศัพท์

5.4.3 แนวทางในการพัฒนาในส่วนของ Application

1. การพัฒนาจากการใช้คอมพิวเตอร์ควบคุม เป็นการใช้โทรศัพท์มือถือหรืออุปกรณ์อื่นที่มีขนาดพกพาควบคุมหุ่นยนต์แบบไร้สาย

2. การใช้ระบบ AI เข้ามาช่วยวิเคราะห์ เช่นวิเคราะห์ว่าเป็นสิ่งมีชีวิตหรือไม่ และใช้ AI เข้ามาช่วยในการควบคุมหุ่นยนต์เช่น ข้างหลังเป็นทางชันหรือมีสิ่งกีดขวางอยู่ไม่สามารถไปได้ เป็นต้น

3. การแสดงแผนที่ ตำแหน่งและเส้นทางที่หุ่นยนต์เคลื่อนที่พบผู้ประสบภัย

4. การแสดงการจำลองลักษณะของหุ่นยนต์ว่าหุ่นยนต์อยู่ในลักษณะใด

เอกสารอ้างอิง

[1] นิรันด์ อุดมเจริญไมตรี. วสันต์ ดวงประทุม. **"การควบคุมมอเตอร์ไฟฟ้ากระแสตรง"** วิยานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า. มหาวิทยาลัยขอนแก่น 2550.

[2] ทวีทรัพย์ สัญจรดี. วชรพร สุวรรณแสน. **"หุ่นยนต์กู้ภัย"** วิยานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า. มหาวิทยาลัยขอนแก่น 2549.

[3] อุดม จีนประดับ. **ไมโครคอนโทลเลอร์ MCS-51**. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: ศูนย์ตำรา สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ 2541.

[4] **ธีรวัฒน์** ประกอบผล. **การประยุกต์ใช้งานไมโครคอนโทลเลอร์**. พิมพ์ครั้งที่ 8. กรุงเทพมหานคร: สำนักพิมพ์สมาคมส่งเสริมเทคโนโลยี(ไทย - ญี่ปุ่น) 2547.

[5] **จรูญ** แดงวิจิตร. ธนัย อุทัยกัน. "หุ่นยนต์แขนกล" วิยานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า. มหาวิทยาลัยขอนแก่น 2548.

[6] Microsoft Online Safety [Online]. Available: Http://www.microsoft.com/protect/terms/activex.aspx. 2010.

ภาคผนวก ก

โปรแกรมการทำงานของไมโครคอนโทรลเลอร์

```
#include <reg52.h>
#include <stdio.h>
#include <intrins.h>
//-------------------- Function ----------------------
int cmdRS;
void dusec (unsigned int count);
void start232 (void);
void startinterupt(void);
void serial_ISR(void);
void serial_2(void);
void serial_ISR_Job(void);
void showEncoder();
void showTemp();
/*************Termal*************/
int termal;
void dmsec (unsigned int count);
void ipdel (void);
void ipclow (void);
void ipstart (void);
void ipstop (void);
bit irwrbyte (unsigned dat);
unsigned char irrdbyte ();
unsigned char irrdbytex ();

double distanceL,distanceR;
//-----------------------------------------
void start232 (void)
{
      SCON = 0x52;      // set RS232 parameter
    TMOD = 0x55;
      RCAP2H =0xFF;     // Set Baud rate to 9600 bps
      RCAP2L = 0xDC;    // Set Baud rate to 9600 bps and 0xDC =
Crystal = 11.059200 MHz
      T2CON = 0x34;     // Set Timer2 as Baudrate Generate

      //-------------------------
      TH0 = TH1 = 0xff;
      TL0 = TL1 = 0x80;

      TR1 = 1;
      TR0 = 1;
}
void startinterupt(void)            //Founction Interrupt Enable
register
{
      EA = 1;                                    //add Interrupt
Enable
      ES = 1;

      ET0 = 1;
      ET1 = 1;
      //Interrupt priority register
      PT1 = 1;     //Timer 1
      PT0 = 1;     //Timer 0
```

```
	PS = 0;			//serial port
	//*************************

}
/*************************************/
/*************** Encoder ***************/
/*************************************/
int count0 = 0,count1 = 0;
int pls,T_2;
//Count Timer 1 ************************************
void Timer1_Handler(void) interrupt 3 //Right
{
	TF1 = 0;
	if(cmdRS == 'w' || cmdRS == 'a' || cmdRS == 'c')
	{
		count0++;
	}
	else
	{
		count0--;
	}
	distanceR = (count0 * 6.9)/ 100;
	TH1 = 0xff;
	TL1 = 0x80;
	TR1 = 1;
}
//Count1 Timer 0 ************************************
void Timer0_Handler(void) interrupt 1 //Left
{
	TF0 = 0;
	if(cmdRS == 'w' || cmdRS == 'd' || cmdRS == 'z')
	{
		count1++;
	}
	else
	{
		count1--;
	}
	distanceL = (count1 * 6.9)/ 100;
	TH0 = 0xff;
	TL0 = 0x80;
	TR0 = 1;
}
/************** Main Function *****************/
 /*************************************/
 /*************** Control ***************/
 /*************************************/
//--------- Delay Control -----------
void dusec (unsigned int count) {  // mSec Delay
    unsigned int i;                // Keil CA51 (x2)
    while (count)
      {
		serial_2();
		if(cmdRS == '0')
		{
			break;
		}
		//else{
		i = 100;
			while (i>0)
			i--;
```

```
                count--;
            //}
    }
}
void serial_ISR(void) interrupt 4
{
      if(RI)
      {
            cmdRS = SBUF;
            //---------------------------------------
            serial_ISR_Job();
            dusec(100000);
            //---------------------------------------
            RI = 0;
      }
}
//-----------------------------------------------
void serial_2(void)
{
      if(RI)
      {
            cmdRS = SBUF;
            //---------------------------------------
            serial_ISR_Job();
            //---------------------------------------
            RI = 0;
      }
}
void serial_ISR_Job(void)
{
 if(cmdRS == 'w'){          //forword
    P2 = 0xC6;    //
       showEncoder();
 }else if(cmdRS == 'd'){    //forword Left
    P2 = 0xC0;    //
       showEncoder();
 }else if(cmdRS == 'f'){     //backword Left
    P2 = 0x60;    //
      showEncoder();
 }else if(cmdRS == 'a'){     //forword Right
    P2 = 0x06;    //
      showEncoder();
 }else if(cmdRS == 's'){    //backword Right
    P2 = 0x0C;    //
      showEncoder();
 }else if(cmdRS == 'x'){    //backword
    P2 = 0x6C;    //
      showEncoder();
 }else if(cmdRS == 'z'){    //forword left  backword Right
    P2 = 0xCC;    //
      showEncoder();
 }else if(cmdRS == 'c'){    //forword Right backword Left
    P2 = 0x66;    //
      showEncoder();
 }else if(cmdRS == 't'){    //Camara L
    P2 = 0x10;    //
      showEncoder();
 }else if(cmdRS == 'y'){    //Camara R
    P2 = 0x01;    //
      showEncoder();
 }else if(cmdRS == 'h'){      //Camara Up
```

```
    P0 = 0x10;    //
      showEncoder();
  }else if(cmdRS == 'g'){       //Camara Down
    P0 = 0x01;    //
      showEncoder();
  }else if(cmdRS == 'k'){     //Up ARM Wheel Backward
    P0 = 0x06;    //
      showEncoder();
  }else if(cmdRS == ';'){     //Down ARM Wheel Forkward
    P0 = 0x0C;    //
      showEncoder();
  }else if(cmdRS == 'u'){     //HandDown Up
    P0 = 0xC0;    //
      showEncoder();
  }else if(cmdRS == 'i'){     //HandDown Down
    P0 = 0x60;    //
      showEncoder();
  }else if(cmdRS == '1'){     //HandDown Down
    showTemp();
  }
  else{
    P2 = 0x00;
      P1 = 0x00;
    P0 = 0x00;
  }
  }
/***************** Termal ***********************/
/********** I/O PORT **********/

sbit     IPSCL  = P1^4;                  // I2C I/O Bit
sbit     IPSDA  = P1^5;

/********** BASIC FUNCTION **********/

void dmsec (unsigned int count) {       // mSec Delay
    unsigned int i;                     // Keil CA51 (x2)
    while (count) {
        i = 230; while (i>0) i--;
        count--;
    }
}

/********** SPECIFIC FUNCTION **********/
void ipdel (void) {                     // I2C delay
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
}
void ipchigh (void) {                   // I2C clock high
    IPSCL = 1;
    ipdel ();
}
void ipclow (void) {                    // I2C clock low
    IPSCL = 0;
    ipdel ();
}
```

```
void ipstart (void) {                    // start condition
    IPSDA = 1;
    IPSCL = 1;
    IPSDA = 0;
    ipdel ();
    IPSCL = 0;
    IPSDA = 1;
}
void ipstop (void) {                     // stop condition
    IPSDA = 0;
    IPSCL = 1;
    ipdel ();
    IPSDA = 1;
}
bit irwrbyte (unsigned dat) {            // write one byte for ds1307
    unsigned char i;                     // return 0 = ok
    bit outbit;                          // return 1 = error
    for (i=1;i<=8;i++) {
        outbit = dat & 0x80;
        IPSDA = outbit;
        dat = dat << 1;
        ipchigh ();
        ipclow ();
    }
    //IPSDA = 1;
      //IPSDA = 0;
    ipchigh ();
    ipclow ();
      outbit = IPSDA;
    return (outbit);
}
unsigned char irrdbyte () {              // read last byte for ds1307
(not ack)
    unsigned char i,dat;
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++) {
        ipchigh ();
        inbit = IPSDA;
        dat = dat << 1;
        dat = dat | inbit;
        ipclow ();
    }
    //IPSDA = 1;
    ipchigh ();
    ipclow ();
    return (dat);
}
unsigned char irrdbytex () {             // read one byte for DS1307
    unsigned char i,dat;
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++) {
        ipchigh ();
        inbit = IPSDA;
        dat = dat << 1;
        dat = dat | inbit;
        ipclow ();
    }
    //IPSDA = 0;
    ipchigh ();
```

```
    ipclow ();
    //IPSDA = 1;
    return (dat);
}

/*************************************************/
/******************** IR2 ************************/
/*************************************************/

/*********************Show Sensor*********************/
void showEncoder()
{
      /********* Encoder ***********/
      printf("%3.2f   ",distanceL); //Left
      printf("%3.2f\r",distanceR);   //Right

}
void showTemp()
{
       /*********** Tomal ******/
      ipstart();
      irwrbyte(0xd0);
      irwrbyte(0x02);
      ipstop();
      ipstart();
      irwrbyte(0xd1);
      termal =  irrdbytex();
      ipstop();
      printf("T%d\r",termal);

}
/************** Main Function *****************/
void main(void)
{
      start232();
      startinterupt();
      printf ("Microcontroller OK \n\r");
      while(1)
      {
      }
}
```

โปรแกรมควบคุมการทำงานโดยภาษา Visual Basic 2008

```
Option Explicit On
Option Strict On

' Import file aboot JoyStick open
Imports Microsoft.DirectX
Imports Microsoft.DirectX.DirectInput
Imports System.IO.Ports

'*****************************************************************
Public Class ControlRobot
    Inherits System.Windows.Forms.Form
    Private mIniFile As w2IniFile
    Private mPort As New IO.Ports.SerialPort
```

```
    Private mTxt As w2TxtBuffer
    Friend WithEvents ImageList As System.Windows.Forms.ImageList
    Friend WithEvents bgWorker As
System.ComponentModel.BackgroundWorker
    Friend WithEvents StatusStrip1 As
System.Windows.Forms.StatusStrip
    Friend WithEvents Label12 As System.Windows.Forms.Label
    Friend WithEvents Label13 As System.Windows.Forms.Label
    Friend WithEvents SendSerialPort As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents GroupBox5 As System.Windows.Forms.GroupBox
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Link As System.Windows.Forms.TextBox
    Friend WithEvents runvideo As System.Windows.Forms.Button
    Friend WithEvents WebBrowser1 As System.Windows.Forms.WebBrowser
    Friend WithEvents RXencoder As System.Windows.Forms.Label
    Friend WithEvents lstData As System.Windows.Forms.ListBox
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label14 As System.Windows.Forms.Label
    Friend WithEvents Label15 As System.Windows.Forms.Label
    Friend WithEvents LinkLabel1 As System.Windows.Forms.LinkLabel
    Friend WithEvents Label16 As System.Windows.Forms.Label
    Friend WithEvents Temp As System.Windows.Forms.Label
    Friend WithEvents tempdata As System.Windows.Forms.Label
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents Label17 As System.Windows.Forms.Label
    Friend WithEvents Label18 As System.Windows.Forms.Label
    Friend WithEvents Label19 As System.Windows.Forms.Label
    Friend WithEvents Label20 As System.Windows.Forms.Label
    Friend WithEvents Label21 As System.Windows.Forms.Label
    Friend WithEvents lblPort As
System.Windows.Forms.ToolStripStatusLabel


    ' code added to dispose in windows for designer generated code
#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        'timer1.Stop()
        ' Unacquire all DirectInput objects.
```

```
        If Not applicationDevice Is Nothing Then
applicationDevice.Unacquire()
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents buttonExit As System.Windows.Forms.Button
    Friend WithEvents groupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents labelButtons As System.Windows.Forms.Label
    Friend WithEvents label1 As System.Windows.Forms.Label
    Friend WithEvents labelPOV3 As System.Windows.Forms.Label
    Friend WithEvents labelSlider1 As System.Windows.Forms.Label
    Friend WithEvents labelSlider0 As System.Windows.Forms.Label
    Friend WithEvents labelPOV2 As System.Windows.Forms.Label
    Friend WithEvents labelPOV1 As System.Windows.Forms.Label
    Friend WithEvents labelPOV0 As System.Windows.Forms.Label
    Friend WithEvents labelZRotation As System.Windows.Forms.Label
    Friend WithEvents labelYRotation As System.Windows.Forms.Label
    Friend WithEvents labelXRotation As System.Windows.Forms.Label
    Friend WithEvents labelZAxis As System.Windows.Forms.Label
    Friend WithEvents labelYAxis As System.Windows.Forms.Label
    Friend WithEvents labelXAxis As System.Windows.Forms.Label
    Friend WithEvents labelPOV3Text As System.Windows.Forms.Label
    Friend WithEvents labelPOV2Text As System.Windows.Forms.Label
    Friend WithEvents labelPOV1Text As System.Windows.Forms.Label
    Friend WithEvents labelPOV0Text As System.Windows.Forms.Label
    Friend WithEvents labelSlider1Text As System.Windows.Forms.Label
    Friend WithEvents labelSlider0Text As System.Windows.Forms.Label
    Friend WithEvents labelZRotationText As
System.Windows.Forms.Label
    Friend WithEvents labelYRotationText As
System.Windows.Forms.Label
    Friend WithEvents labelXRotationText As
System.Windows.Forms.Label
    Friend WithEvents labelZAxisText As System.Windows.Forms.Label
    Friend WithEvents labelYAxisText As System.Windows.Forms.Label
    Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox4 As System.Windows.Forms.GroupBox
    Friend WithEvents Temperature As System.Windows.Forms.Button
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Panel3 As System.Windows.Forms.Panel
```

```
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Friend WithEvents btnConnect As System.Windows.Forms.Button
    Friend WithEvents cbostop As System.Windows.Forms.ComboBox
    Friend WithEvents cboData As System.Windows.Forms.ComboBox
    Friend WithEvents cboParity As System.Windows.Forms.ComboBox
    Friend WithEvents cboRate As System.Windows.Forms.ComboBox
    Friend WithEvents cboPort As System.Windows.Forms.ComboBox
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents txtSend As System.Windows.Forms.TextBox
    Friend WithEvents Panel1 As System.Windows.Forms.Panel
    Friend WithEvents labelXAxisText As System.Windows.Forms.Label
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Dim resources As
System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(ControlRobot))
        Me.buttonExit = New System.Windows.Forms.Button
        Me.groupBox1 = New System.Windows.Forms.GroupBox
        Me.SendSerialPort = New System.Windows.Forms.Label
        Me.Label13 = New System.Windows.Forms.Label
        Me.labelButtons = New System.Windows.Forms.Label
        Me.Panel3 = New System.Windows.Forms.Panel
        Me.txtSend = New System.Windows.Forms.TextBox
        Me.label1 = New System.Windows.Forms.Label
        Me.labelPOV3 = New System.Windows.Forms.Label
        Me.labelSlider1 = New System.Windows.Forms.Label
        Me.labelSlider0 = New System.Windows.Forms.Label
        Me.labelPOV2 = New System.Windows.Forms.Label
        Me.labelPOV1 = New System.Windows.Forms.Label
        Me.labelPOV0 = New System.Windows.Forms.Label
        Me.labelZRotation = New System.Windows.Forms.Label
        Me.labelYRotation = New System.Windows.Forms.Label
        Me.labelXRotation = New System.Windows.Forms.Label
        Me.labelZAxis = New System.Windows.Forms.Label
        Me.labelYAxis = New System.Windows.Forms.Label
        Me.labelXAxis = New System.Windows.Forms.Label
        Me.labelPOV3Text = New System.Windows.Forms.Label
        Me.labelPOV2Text = New System.Windows.Forms.Label
        Me.labelPOV1Text = New System.Windows.Forms.Label
        Me.labelPOV0Text = New System.Windows.Forms.Label
        Me.labelSlider1Text = New System.Windows.Forms.Label
        Me.labelSlider0Text = New System.Windows.Forms.Label
        Me.labelZRotationText = New System.Windows.Forms.Label
        Me.labelYRotationText = New System.Windows.Forms.Label
        Me.labelXRotationText = New System.Windows.Forms.Label
        Me.labelZAxisText = New System.Windows.Forms.Label
        Me.labelYAxisText = New System.Windows.Forms.Label
        Me.labelXAxisText = New System.Windows.Forms.Label
```

```
        Me.GroupBox2 = New System.Windows.Forms.GroupBox
        Me.Panel1 = New System.Windows.Forms.Panel
        Me.StatusStrip1 = New System.Windows.Forms.StatusStrip
        Me.lblPort = New System.Windows.Forms.ToolStripStatusLabel
        Me.Label6 = New System.Windows.Forms.Label
        Me.Label11 = New System.Windows.Forms.Label
        Me.btnConnect = New System.Windows.Forms.Button
        Me.Label7 = New System.Windows.Forms.Label
        Me.cbostop = New System.Windows.Forms.ComboBox
        Me.Label8 = New System.Windows.Forms.Label
        Me.cboData = New System.Windows.Forms.ComboBox
        Me.Label9 = New System.Windows.Forms.Label
        Me.cboParity = New System.Windows.Forms.ComboBox
        Me.Label10 = New System.Windows.Forms.Label
        Me.cboRate = New System.Windows.Forms.ComboBox
        Me.cboPort = New System.Windows.Forms.ComboBox
        Me.GroupBox3 = New System.Windows.Forms.GroupBox
        Me.Label17 = New System.Windows.Forms.Label
        Me.Label16 = New System.Windows.Forms.Label
        Me.Label15 = New System.Windows.Forms.Label
        Me.Label5 = New System.Windows.Forms.Label
        Me.Label2 = New System.Windows.Forms.Label
        Me.RXencoder = New System.Windows.Forms.Label
        Me.Label12 = New System.Windows.Forms.Label
        Me.lstData = New System.Windows.Forms.ListBox
        Me.GroupBox4 = New System.Windows.Forms.GroupBox
        Me.Temp = New System.Windows.Forms.Label
        Me.tempdata = New System.Windows.Forms.Label
        Me.PictureBox1 = New System.Windows.Forms.PictureBox
        Me.LinkLabel1 = New System.Windows.Forms.LinkLabel
        Me.Label14 = New System.Windows.Forms.Label
        Me.Label4 = New System.Windows.Forms.Label
        Me.Temperature = New System.Windows.Forms.Button
        Me.ImageList = New
System.Windows.Forms.ImageList(Me.components)
        Me.bgWorker = New System.ComponentModel.BackgroundWorker
        Me.GroupBox5 = New System.Windows.Forms.GroupBox
        Me.runvideo = New System.Windows.Forms.Button
        Me.Label3 = New System.Windows.Forms.Label
        Me.Link = New System.Windows.Forms.TextBox
        Me.WebBrowser1 = New System.Windows.Forms.WebBrowser
        Me.Label18 = New System.Windows.Forms.Label
        Me.Label19 = New System.Windows.Forms.Label
        Me.Label20 = New System.Windows.Forms.Label
        Me.Label21 = New System.Windows.Forms.Label
        Me.groupBox1.SuspendLayout()
        Me.Panel3.SuspendLayout()
        Me.GroupBox2.SuspendLayout()
        Me.Panel1.SuspendLayout()
        Me.StatusStrip1.SuspendLayout()
        Me.GroupBox3.SuspendLayout()
        Me.GroupBox4.SuspendLayout()
```

```
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.GroupBox5.SuspendLayout()
        Me.SuspendLayout()
        '
        'buttonExit
        '
        Me.buttonExit.ForeColor = System.Drawing.Color.Black
        Me.buttonExit.Location = New System.Drawing.Point(22, 219)
        Me.buttonExit.Name = "buttonExit"
        Me.buttonExit.Size = New System.Drawing.Size(206, 35)
        Me.buttonExit.TabIndex = 3
        Me.buttonExit.Text = "Exit"
        '
        'groupBox1
        '
        Me.groupBox1.Controls.Add(Me.SendSerialPort)
        Me.groupBox1.Controls.Add(Me.Label13)
        Me.groupBox1.Controls.Add(Me.labelButtons)
        Me.groupBox1.Controls.Add(Me.Panel3)
        Me.groupBox1.Controls.Add(Me.label1)
        Me.groupBox1.Controls.Add(Me.labelPOV3)
        Me.groupBox1.Controls.Add(Me.labelSlider1)
        Me.groupBox1.Controls.Add(Me.labelSlider0)
        Me.groupBox1.Controls.Add(Me.labelPOV2)
        Me.groupBox1.Controls.Add(Me.labelPOV1)
        Me.groupBox1.Controls.Add(Me.labelPOV0)
        Me.groupBox1.Controls.Add(Me.labelZRotation)
        Me.groupBox1.Controls.Add(Me.labelYRotation)
        Me.groupBox1.Controls.Add(Me.labelXRotation)
        Me.groupBox1.Controls.Add(Me.labelZAxis)
        Me.groupBox1.Controls.Add(Me.labelYAxis)
        Me.groupBox1.Controls.Add(Me.labelXAxis)
        Me.groupBox1.Controls.Add(Me.labelPOV3Text)
        Me.groupBox1.Controls.Add(Me.labelPOV2Text)
        Me.groupBox1.Controls.Add(Me.labelPOV1Text)
        Me.groupBox1.Controls.Add(Me.labelPOV0Text)
        Me.groupBox1.Controls.Add(Me.labelSlider1Text)
        Me.groupBox1.Controls.Add(Me.labelSlider0Text)
        Me.groupBox1.Controls.Add(Me.labelZRotationText)
        Me.groupBox1.Controls.Add(Me.labelYRotationText)
        Me.groupBox1.Controls.Add(Me.labelXRotationText)
        Me.groupBox1.Controls.Add(Me.labelZAxisText)
        Me.groupBox1.Controls.Add(Me.labelYAxisText)
        Me.groupBox1.Controls.Add(Me.labelXAxisText)
        Me.groupBox1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 11.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.groupBox1.Location = New System.Drawing.Point(12, 389)
        Me.groupBox1.Name = "groupBox1"
        Me.groupBox1.Size = New System.Drawing.Size(275, 211)
        Me.groupBox1.TabIndex = 2
        Me.groupBox1.TabStop = False
```

```
        Me.groupBox1.Text = "Control JoyStick"
        '
        'SendSerialPort
        '
        Me.SendSerialPort.AutoSize = True
        Me.SendSerialPort.Font = New System.Drawing.Font("Microsoft
Sans Serif", 10.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.SendSerialPort.ForeColor = System.Drawing.Color.Red
        Me.SendSerialPort.Location = New System.Drawing.Point(178,
158)
        Me.SendSerialPort.Name = "SendSerialPort"
        Me.SendSerialPort.Size = New System.Drawing.Size(52, 17)
        Me.SendSerialPort.TabIndex = 27
        Me.SendSerialPort.Text = "Empty"
        '
        'Label13
        '
        Me.Label13.AutoSize = True
        Me.Label13.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label13.Location = New System.Drawing.Point(19, 160)
        Me.Label13.Name = "Label13"
        Me.Label13.Size = New System.Drawing.Size(152, 13)
        Me.Label13.TabIndex = 26
        Me.Label13.Text = "Value send to SerialPort :"
        '
        'labelButtons
        '
        Me.labelButtons.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelButtons.Location = New System.Drawing.Point(74, 132)
        Me.labelButtons.Name = "labelButtons"
        Me.labelButtons.Size = New System.Drawing.Size(184, 13)
        Me.labelButtons.TabIndex = 25
        '
        'Panel3
        '
        Me.Panel3.Controls.Add(Me.txtSend)
        Me.Panel3.Location = New System.Drawing.Point(238, 153)
        Me.Panel3.Name = "Panel3"
        Me.Panel3.Size = New System.Drawing.Size(27, 29)
        Me.Panel3.TabIndex = 8
        '
        'txtSend
        '
        Me.txtSend.ForeColor = System.Drawing.Color.Red
        Me.txtSend.Location = New System.Drawing.Point(3, 3)
        Me.txtSend.Name = "txtSend"
        Me.txtSend.Size = New System.Drawing.Size(20, 24)
        Me.txtSend.TabIndex = 0
```

```
        '
        'label1
        '
        Me.label1.AutoSize = True
        Me.label1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.label1.Location = New System.Drawing.Point(21, 132)
        Me.label1.Name = "label1"
        Me.label1.Size = New System.Drawing.Size(46, 13)
        Me.label1.TabIndex = 24
        Me.label1.Text = "Buttons:"
        '
        'labelPOV3
        '
        Me.labelPOV3.AutoSize = True
        Me.labelPOV3.Enabled = False
        Me.labelPOV3.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV3.Location = New System.Drawing.Point(220, 105)
        Me.labelPOV3.Name = "labelPOV3"
        Me.labelPOV3.Size = New System.Drawing.Size(13, 13)
        Me.labelPOV3.TabIndex = 23
        Me.labelPOV3.Text = "0"
        '
        'labelSlider1
        '
        Me.labelSlider1.AutoSize = True
        Me.labelSlider1.Enabled = False
        Me.labelSlider1.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelSlider1.Location = New System.Drawing.Point(220, 39)
        Me.labelSlider1.Name = "labelSlider1"
        Me.labelSlider1.Size = New System.Drawing.Size(13, 13)
        Me.labelSlider1.TabIndex = 22
        Me.labelSlider1.Text = "0"
        '
        'labelSlider0
        '
        Me.labelSlider0.AutoSize = True
        Me.labelSlider0.Enabled = False
        Me.labelSlider0.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelSlider0.Location = New System.Drawing.Point(220, 23)
        Me.labelSlider0.Name = "labelSlider0"
        Me.labelSlider0.Size = New System.Drawing.Size(13, 13)
        Me.labelSlider0.TabIndex = 21
        Me.labelSlider0.Text = "0"
        '
        'labelPOV2
```

```
        '
        Me.labelPOV2.AutoSize = True
        Me.labelPOV2.Enabled = False
        Me.labelPOV2.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV2.Location = New System.Drawing.Point(220, 89)
        Me.labelPOV2.Name = "labelPOV2"
        Me.labelPOV2.Size = New System.Drawing.Size(13, 13)
        Me.labelPOV2.TabIndex = 20
        Me.labelPOV2.Text = "0"
        '
        'labelPOV1
        '
        Me.labelPOV1.AutoSize = True
        Me.labelPOV1.Enabled = False
        Me.labelPOV1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV1.Location = New System.Drawing.Point(220, 73)
        Me.labelPOV1.Name = "labelPOV1"
        Me.labelPOV1.Size = New System.Drawing.Size(13, 13)
        Me.labelPOV1.TabIndex = 19
        Me.labelPOV1.Text = "0"
        '
        'labelPOV0
        '
        Me.labelPOV0.AutoSize = True
        Me.labelPOV0.Enabled = False
        Me.labelPOV0.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV0.Location = New System.Drawing.Point(220, 57)
        Me.labelPOV0.Name = "labelPOV0"
        Me.labelPOV0.Size = New System.Drawing.Size(13, 13)
        Me.labelPOV0.TabIndex = 18
        Me.labelPOV0.Text = "0"
        '
        'labelZRotation
        '
        Me.labelZRotation.AutoSize = True
        Me.labelZRotation.Enabled = False
        Me.labelZRotation.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelZRotation.Location = New System.Drawing.Point(116,
105)
        Me.labelZRotation.Name = "labelZRotation"
        Me.labelZRotation.Size = New System.Drawing.Size(13, 13)
        Me.labelZRotation.TabIndex = 17
        Me.labelZRotation.Text = "0"
        '
        'labelYRotation
```

```
        '
        Me.labelYRotation.AutoSize = True
        Me.labelYRotation.Enabled = False
        Me.labelYRotation.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelYRotation.Location = New System.Drawing.Point(116,
89)
        Me.labelYRotation.Name = "labelYRotation"
        Me.labelYRotation.Size = New System.Drawing.Size(13, 13)
        Me.labelYRotation.TabIndex = 16
        Me.labelYRotation.Text = "0"
        '
        'labelXRotation
        '
        Me.labelXRotation.AutoSize = True
        Me.labelXRotation.Enabled = False
        Me.labelXRotation.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelXRotation.Location = New System.Drawing.Point(116,
73)
        Me.labelXRotation.Name = "labelXRotation"
        Me.labelXRotation.Size = New System.Drawing.Size(13, 13)
        Me.labelXRotation.TabIndex = 15
        Me.labelXRotation.Text = "0"
        '
        'labelZAxis
        '
        Me.labelZAxis.AutoSize = True
        Me.labelZAxis.Enabled = False
        Me.labelZAxis.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelZAxis.Location = New System.Drawing.Point(116, 55)
        Me.labelZAxis.Name = "labelZAxis"
        Me.labelZAxis.Size = New System.Drawing.Size(13, 13)
        Me.labelZAxis.TabIndex = 14
        Me.labelZAxis.Text = "0"
        '
        'labelYAxis
        '
        Me.labelYAxis.AutoSize = True
        Me.labelYAxis.Enabled = False
        Me.labelYAxis.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelYAxis.Location = New System.Drawing.Point(116, 39)
        Me.labelYAxis.Name = "labelYAxis"
        Me.labelYAxis.Size = New System.Drawing.Size(13, 13)
        Me.labelYAxis.TabIndex = 13
        Me.labelYAxis.Text = "0"
        '
```

```
        'labelXAxis
        '
        Me.labelXAxis.AutoSize = True
        Me.labelXAxis.Enabled = False
        Me.labelXAxis.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelXAxis.Location = New System.Drawing.Point(116, 23)
        Me.labelXAxis.Name = "labelXAxis"
        Me.labelXAxis.Size = New System.Drawing.Size(13, 13)
        Me.labelXAxis.TabIndex = 12
        Me.labelXAxis.Text = "0"
        '
        'labelPOV3Text
        '
        Me.labelPOV3Text.AutoSize = True
        Me.labelPOV3Text.Enabled = False
        Me.labelPOV3Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV3Text.Location = New System.Drawing.Point(164,
105)
        Me.labelPOV3Text.Name = "labelPOV3Text"
        Me.labelPOV3Text.Size = New System.Drawing.Size(41, 13)
        Me.labelPOV3Text.TabIndex = 11
        Me.labelPOV3Text.Text = "POV 3:"
        '
        'labelPOV2Text
        '
        Me.labelPOV2Text.AutoSize = True
        Me.labelPOV2Text.Enabled = False
        Me.labelPOV2Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV2Text.Location = New System.Drawing.Point(164, 89)
        Me.labelPOV2Text.Name = "labelPOV2Text"
        Me.labelPOV2Text.Size = New System.Drawing.Size(41, 13)
        Me.labelPOV2Text.TabIndex = 10
        Me.labelPOV2Text.Text = "POV 2:"
        '
        'labelPOV1Text
        '
        Me.labelPOV1Text.AutoSize = True
        Me.labelPOV1Text.Enabled = False
        Me.labelPOV1Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV1Text.Location = New System.Drawing.Point(164, 73)
        Me.labelPOV1Text.Name = "labelPOV1Text"
        Me.labelPOV1Text.Size = New System.Drawing.Size(41, 13)
        Me.labelPOV1Text.TabIndex = 9
        Me.labelPOV1Text.Text = "POV 1:"
        '
```

```
        'labelPOV0Text
        '
        Me.labelPOV0Text.AutoSize = True
        Me.labelPOV0Text.Enabled = False
        Me.labelPOV0Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelPOV0Text.Location = New System.Drawing.Point(164, 57)
        Me.labelPOV0Text.Name = "labelPOV0Text"
        Me.labelPOV0Text.Size = New System.Drawing.Size(41, 13)
        Me.labelPOV0Text.TabIndex = 8
        Me.labelPOV0Text.Text = "POV 0:"
        '
        'labelSlider1Text
        '
        Me.labelSlider1Text.AutoSize = True
        Me.labelSlider1Text.Enabled = False
        Me.labelSlider1Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelSlider1Text.Location = New System.Drawing.Point(164,
39)
        Me.labelSlider1Text.Name = "labelSlider1Text"
        Me.labelSlider1Text.Size = New System.Drawing.Size(45, 13)
        Me.labelSlider1Text.TabIndex = 7
        Me.labelSlider1Text.Text = "Slider 1:"
        '
        'labelSlider0Text
        '
        Me.labelSlider0Text.AutoSize = True
        Me.labelSlider0Text.Enabled = False
        Me.labelSlider0Text.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelSlider0Text.Location = New System.Drawing.Point(164,
23)
        Me.labelSlider0Text.Name = "labelSlider0Text"
        Me.labelSlider0Text.Size = New System.Drawing.Size(45, 13)
        Me.labelSlider0Text.TabIndex = 6
        Me.labelSlider0Text.Text = "Slider 0:"
        '
        'labelZRotationText
        '
        Me.labelZRotationText.AutoSize = True
        Me.labelZRotationText.Enabled = False
        Me.labelZRotationText.Font = New
System.Drawing.Font("Microsoft Sans Serif", 8.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(222, Byte))
        Me.labelZRotationText.Location = New System.Drawing.Point(44,
105)
        Me.labelZRotationText.Name = "labelZRotationText"
        Me.labelZRotationText.Size = New System.Drawing.Size(60, 13)
```

```
        Me.labelZRotationText.TabIndex = 5
        Me.labelZRotationText.Text = "Z Rotation:"
        '
        'labelYRotationText
        '
        Me.labelYRotationText.AutoSize = True
        Me.labelYRotationText.Enabled = False
        Me.labelYRotationText.Font = New
System.Drawing.Font("Microsoft Sans Serif", 8.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(222, Byte))
        Me.labelYRotationText.Location = New System.Drawing.Point(44,
89)
        Me.labelYRotationText.Name = "labelYRotationText"
        Me.labelYRotationText.Size = New System.Drawing.Size(60, 13)
        Me.labelYRotationText.TabIndex = 4
        Me.labelYRotationText.Text = "Y Rotation:"
        '
        'labelXRotationText
        '
        Me.labelXRotationText.AutoSize = True
        Me.labelXRotationText.Enabled = False
        Me.labelXRotationText.Font = New
System.Drawing.Font("Microsoft Sans Serif", 8.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(222, Byte))
        Me.labelXRotationText.Location = New System.Drawing.Point(44,
73)
        Me.labelXRotationText.Name = "labelXRotationText"
        Me.labelXRotationText.Size = New System.Drawing.Size(60, 13)
        Me.labelXRotationText.TabIndex = 3
        Me.labelXRotationText.Text = "X Rotation:"
        '
        'labelZAxisText
        '
        Me.labelZAxisText.AutoSize = True
        Me.labelZAxisText.Enabled = False
        Me.labelZAxisText.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelZAxisText.Location = New System.Drawing.Point(44, 55)
        Me.labelZAxisText.Name = "labelZAxisText"
        Me.labelZAxisText.Size = New System.Drawing.Size(39, 13)
        Me.labelZAxisText.TabIndex = 2
        Me.labelZAxisText.Text = "Z Axis:"
        '
        'labelYAxisText
        '
        Me.labelYAxisText.AutoSize = True
        Me.labelYAxisText.Enabled = False
        Me.labelYAxisText.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
```

```
        Me.labelYAxisText.Location = New System.Drawing.Point(44, 39)
        Me.labelYAxisText.Name = "labelYAxisText"
        Me.labelYAxisText.Size = New System.Drawing.Size(39, 13)
        Me.labelYAxisText.TabIndex = 1
        Me.labelYAxisText.Text = "Y Axis:"
        '
        'labelXAxisText
        '
        Me.labelXAxisText.AutoSize = True
        Me.labelXAxisText.Enabled = False
        Me.labelXAxisText.Font = New System.Drawing.Font("Microsoft
Sans Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.labelXAxisText.Location = New System.Drawing.Point(44, 23)
        Me.labelXAxisText.Name = "labelXAxisText"
        Me.labelXAxisText.Size = New System.Drawing.Size(39, 13)
        Me.labelXAxisText.TabIndex = 0
        Me.labelXAxisText.Text = "X Axis:"
        '
        'GroupBox2
        '
        Me.GroupBox2.Controls.Add(Me.Panel1)
        Me.GroupBox2.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.GroupBox2.Location = New System.Drawing.Point(12, 12)
        Me.GroupBox2.Name = "GroupBox2"
        Me.GroupBox2.Size = New System.Drawing.Size(275, 360)
        Me.GroupBox2.TabIndex = 4
        Me.GroupBox2.TabStop = False
        Me.GroupBox2.Text = "Serial Port of  Communication Robot"
        '
        'Panel1
        '
        Me.Panel1.Controls.Add(Me.StatusStrip1)
        Me.Panel1.Controls.Add(Me.Label6)
        Me.Panel1.Controls.Add(Me.Label11)
        Me.Panel1.Controls.Add(Me.btnConnect)
        Me.Panel1.Controls.Add(Me.Label7)
        Me.Panel1.Controls.Add(Me.cbostop)
        Me.Panel1.Controls.Add(Me.buttonExit)
        Me.Panel1.Controls.Add(Me.Label8)
        Me.Panel1.Controls.Add(Me.cboData)
        Me.Panel1.Controls.Add(Me.Label9)
        Me.Panel1.Controls.Add(Me.cboParity)
        Me.Panel1.Controls.Add(Me.Label10)
        Me.Panel1.Controls.Add(Me.cboRate)
        Me.Panel1.Controls.Add(Me.cboPort)
        Me.Panel1.Location = New System.Drawing.Point(13, 28)
        Me.Panel1.Name = "Panel1"
        Me.Panel1.Size = New System.Drawing.Size(248, 308)
        Me.Panel1.TabIndex = 12
        '
```

```
        'StatusStrip1
        '
        Me.StatusStrip1.Items.AddRange(New
System.Windows.Forms.ToolStripItem() {Me.lblPort})
        Me.StatusStrip1.Location = New System.Drawing.Point(0, 286)
        Me.StatusStrip1.Name = "StatusStrip1"
        Me.StatusStrip1.Size = New System.Drawing.Size(248, 22)
        Me.StatusStrip1.TabIndex = 12
        Me.StatusStrip1.Text = "StatusStrip1"
        '
        'lblPort
        '
        Me.lblPort.Font = New System.Drawing.Font("Segoe UI", 9.0!)
        Me.lblPort.Name = "lblPort"
        Me.lblPort.Size = New System.Drawing.Size(121, 17)
        Me.lblPort.Text = "ToolStripStatusLabel1"
        '
        'Label6
        '
        Me.Label6.AutoSize = True
        Me.Label6.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label6.Location = New System.Drawing.Point(28, 17)
        Me.Label6.Name = "Label6"
        Me.Label6.Size = New System.Drawing.Size(59, 13)
        Me.Label6.TabIndex = 0
        Me.Label6.Text = "Com Port : "
        '
        'Label11
        '
        Me.Label11.AutoSize = True
        Me.Label11.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label11.ForeColor = System.Drawing.Color.Red
        Me.Label11.Location = New System.Drawing.Point(1, 271)
        Me.Label11.Name = "Label11"
        Me.Label11.Size = New System.Drawing.Size(111, 13)
        Me.Label11.TabIndex = 11
        Me.Label11.Text = "Check Communicate :"
        '
        'btnConnect
        '
        Me.btnConnect.Cursor = System.Windows.Forms.Cursors.Hand
        Me.btnConnect.Font = New System.Drawing.Font("Microsoft Sans
Serif", 10.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.btnConnect.ForeColor = System.Drawing.Color.Red
        Me.btnConnect.Location = New System.Drawing.Point(22, 159)
        Me.btnConnect.Name = "btnConnect"
        Me.btnConnect.Size = New System.Drawing.Size(207, 52)
        Me.btnConnect.TabIndex = 10
```

```
        Me.btnConnect.Text = "Click Connect"
        Me.btnConnect.UseVisualStyleBackColor = True
        '
        'Label7
        '
        Me.Label7.AutoSize = True
        Me.Label7.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label7.Location = New System.Drawing.Point(19, 46)
        Me.Label7.Name = "Label7"
        Me.Label7.Size = New System.Drawing.Size(67, 13)
        Me.Label7.TabIndex = 1
        Me.Label7.Text = "Baud Rate : "
        '
        'cbostop
        '
        Me.cbostop.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.cbostop.FormattingEnabled = True
        Me.cbostop.Location = New System.Drawing.Point(90, 126)
        Me.cbostop.Name = "cbostop"
        Me.cbostop.Size = New System.Drawing.Size(138, 21)
        Me.cbostop.TabIndex = 9
        '
        'Label8
        '
        Me.Label8.AutoSize = True
        Me.Label8.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label8.Location = New System.Drawing.Point(43, 74)
        Me.Label8.Name = "Label8"
        Me.Label8.Size = New System.Drawing.Size(39, 13)
        Me.Label8.TabIndex = 2
        Me.Label8.Text = "Parity :"
        '
        'cboData
        '
        Me.cboData.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.cboData.FormattingEnabled = True
        Me.cboData.Location = New System.Drawing.Point(90, 98)
        Me.cboData.Name = "cboData"
        Me.cboData.Size = New System.Drawing.Size(138, 21)
        Me.cboData.TabIndex = 8
        '
        'Label9
        '
        Me.Label9.AutoSize = True
```

```
        Me.Label9.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label9.Location = New System.Drawing.Point(30, 102)
        Me.Label9.Name = "Label9"
        Me.Label9.Size = New System.Drawing.Size(54, 13)
        Me.Label9.TabIndex = 3
        Me.Label9.Text = "Data Bit : "
        '
        'cboParity
        '
        Me.cboParity.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.cboParity.FormattingEnabled = True
        Me.cboParity.Location = New System.Drawing.Point(90, 70)
        Me.cboParity.Name = "cboParity"
        Me.cboParity.Size = New System.Drawing.Size(138, 21)
        Me.cboParity.TabIndex = 7
        '
        'Label10
        '
        Me.Label10.AutoSize = True
        Me.Label10.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label10.Location = New System.Drawing.Point(31, 131)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(50, 13)
        Me.Label10.TabIndex = 4
        Me.Label10.Text = "Stop Bit :"
        '
        'cboRate
        '
        Me.cboRate.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.cboRate.FormattingEnabled = True
        Me.cboRate.Location = New System.Drawing.Point(90, 42)
        Me.cboRate.Name = "cboRate"
        Me.cboRate.Size = New System.Drawing.Size(138, 21)
        Me.cboRate.TabIndex = 6
        '
        'cboPort
        '
        Me.cboPort.Cursor = System.Windows.Forms.Cursors.Default
        Me.cboPort.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.cboPort.FormattingEnabled = True
        Me.cboPort.Location = New System.Drawing.Point(90, 14)
        Me.cboPort.Name = "cboPort"
        Me.cboPort.Size = New System.Drawing.Size(138, 21)
```

```
        Me.cboPort.TabIndex = 5
        '
        'GroupBox3
        '
        Me.GroupBox3.Controls.Add(Me.Label17)
        Me.GroupBox3.Controls.Add(Me.Label16)
        Me.GroupBox3.Controls.Add(Me.Label15)
        Me.GroupBox3.Controls.Add(Me.Label5)
        Me.GroupBox3.Controls.Add(Me.Label2)
        Me.GroupBox3.Controls.Add(Me.RXencoder)
        Me.GroupBox3.Controls.Add(Me.Label12)
        Me.GroupBox3.Controls.Add(Me.lstData)
        Me.GroupBox3.Font = New System.Drawing.Font("Microsoft Sans
Serif", 11.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.GroupBox3.Location = New System.Drawing.Point(985, 12)
        Me.GroupBox3.Name = "GroupBox3"
        Me.GroupBox3.Size = New System.Drawing.Size(275, 296)
        Me.GroupBox3.TabIndex = 5
        Me.GroupBox3.TabStop = False
        Me.GroupBox3.Text = "Encoder Sensor"
        '
        'Label17
        '
        Me.Label17.AutoSize = True
        Me.Label17.Location = New System.Drawing.Point(82, 184)
        Me.Label17.Name = "Label17"
        Me.Label17.Size = New System.Drawing.Size(115, 18)
        Me.Label17.TabIndex = 15
        Me.Label17.Text = "unit : ( metre )"
        '
        'Label16
        '
        Me.Label16.AutoSize = True
        Me.Label16.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label16.Location = New System.Drawing.Point(110, 253)
        Me.Label16.Name = "Label16"
        Me.Label16.Size = New System.Drawing.Size(129, 13)
        Me.Label16.TabIndex = 14
        Me.Label16.Text = ": 2 chanel ( A , B )  5 Vdc "
        '
        'Label15
        '
        Me.Label15.AutoSize = True
        Me.Label15.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label15.Location = New System.Drawing.Point(28, 230)
        Me.Label15.Name = "Label15"
        Me.Label15.Size = New System.Drawing.Size(216, 13)
        Me.Label15.TabIndex = 13
```

```
        Me.Label15.Text = "Sensor Encoder  :  fineness  288
pule/round"
        '
        'Label5
        '
        Me.Label5.AutoSize = True
        Me.Label5.Font = New System.Drawing.Font("Microsoft Sans
Serif", 16.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label5.Location = New System.Drawing.Point(131, 30)
        Me.Label5.Name = "Label5"
        Me.Label5.Size = New System.Drawing.Size(19, 26)
        Me.Label5.TabIndex = 12
        Me.Label5.Text = ":"
        '
        'Label2
        '
        Me.Label2.AutoSize = True
        Me.Label2.Font = New System.Drawing.Font("Microsoft Sans
Serif", 15.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label2.Location = New System.Drawing.Point(182, 33)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(72, 25)
        Me.Label2.TabIndex = 8
        Me.Label2.Text = "RIGHT"
        '
        'RXencoder
        '
        Me.RXencoder.AutoSize = True
        Me.RXencoder.Font = New System.Drawing.Font("Microsoft Sans
Serif", 20.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.RXencoder.Location = New System.Drawing.Point(43, 81)
        Me.RXencoder.Name = "RXencoder"
        Me.RXencoder.Size = New System.Drawing.Size(0, 31)
        Me.RXencoder.TabIndex = 9
        '
        'Label12
        '
        Me.Label12.AutoSize = True
        Me.Label12.Font = New System.Drawing.Font("Microsoft Sans
Serif", 15.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label12.Location = New System.Drawing.Point(30, 33)
        Me.Label12.Name = "Label12"
        Me.Label12.Size = New System.Drawing.Size(61, 25)
        Me.Label12.TabIndex = 10
        Me.Label12.Text = "LEFT"
        '
        'lstData
        '
```

```
        Me.lstData.Font = New System.Drawing.Font("Microsoft Sans
Serif", 35.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.lstData.ForeColor = System.Drawing.Color.Red
        Me.lstData.FormattingEnabled = True
        Me.lstData.ImeMode = System.Windows.Forms.ImeMode.NoControl
        Me.lstData.ItemHeight = 54
        Me.lstData.Location = New System.Drawing.Point(0, 64)
        Me.lstData.Name = "lstData"
        Me.lstData.RightToLeft = System.Windows.Forms.RightToLeft.No
        Me.lstData.SelectionMode =
System.Windows.Forms.SelectionMode.None
        Me.lstData.Size = New System.Drawing.Size(275, 112)
        Me.lstData.TabIndex = 7
        '
        'GroupBox4
        '
        Me.GroupBox4.Controls.Add(Me.Temp)
        Me.GroupBox4.Controls.Add(Me.tempdata)
        Me.GroupBox4.Controls.Add(Me.PictureBox1)
        Me.GroupBox4.Controls.Add(Me.LinkLabel1)
        Me.GroupBox4.Controls.Add(Me.Label14)
        Me.GroupBox4.Controls.Add(Me.Label4)
        Me.GroupBox4.Controls.Add(Me.Temperature)
        Me.GroupBox4.Font = New System.Drawing.Font("Microsoft Sans
Serif", 11.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.GroupBox4.Location = New System.Drawing.Point(985, 327)
        Me.GroupBox4.Name = "GroupBox4"
        Me.GroupBox4.Size = New System.Drawing.Size(275, 405)
        Me.GroupBox4.TabIndex = 6
        Me.GroupBox4.TabStop = False
        Me.GroupBox4.Text = "Temperature Sensor"
        '
        'Temp
        '
        Me.Temp.AutoSize = True
        Me.Temp.BackColor = System.Drawing.SystemColors.Control
        Me.Temp.Font = New System.Drawing.Font("Microsoft Sans
Serif", 37.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Temp.ForeColor = System.Drawing.Color.Red
        Me.Temp.Location = New System.Drawing.Point(71, 60)
        Me.Temp.Name = "Temp"
        Me.Temp.Size = New System.Drawing.Size(115, 58)
        Me.Temp.TabIndex = 9
        Me.Temp.Text = "T00"
        '
        'tempdata
        '
        Me.tempdata.AutoSize = True
```

```
        Me.tempdata.Font = New System.Drawing.Font("Microsoft Sans
Serif", 32.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.tempdata.ForeColor = System.Drawing.Color.Red
        Me.tempdata.Location = New System.Drawing.Point(110, 61)
        Me.tempdata.Name = "tempdata"
        Me.tempdata.Size = New System.Drawing.Size(0, 51)
        Me.tempdata.TabIndex = 5
        '
        'PictureBox1
        '
        Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(14, 25)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(253, 201)
        Me.PictureBox1.TabIndex = 12
        Me.PictureBox1.TabStop = False
        '
        'LinkLabel1
        '
        Me.LinkLabel1.AutoSize = True
        Me.LinkLabel1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.LinkLabel1.LinkColor = System.Drawing.Color.CornflowerBlue
        Me.LinkLabel1.Location = New System.Drawing.Point(5, 367)
        Me.LinkLabel1.Name = "LinkLabel1"
        Me.LinkLabel1.Size = New System.Drawing.Size(268, 13)
        Me.LinkLabel1.TabIndex = 14
        Me.LinkLabel1.TabStop = True
        Me.LinkLabel1.Text = "http://www.robot-
electronics.co.uk/htm/tpa81tech.htm"
        '
        'Label14
        '
        Me.Label14.AutoSize = True
        Me.Label14.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label14.Location = New System.Drawing.Point(36, 328)
        Me.Label14.Name = "Label14"
        Me.Label14.Size = New System.Drawing.Size(210, 13)
        Me.Label14.TabIndex = 13
        Me.Label14.Text = "Infrared Area 2um - 22um and distance 2 m"
        '
        'Label4
        '
        Me.Label4.AutoSize = True
        Me.Label4.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Label4.Location = New System.Drawing.Point(28, 306)
```

```
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(226, 13)
        Me.Label4.TabIndex = 3
        Me.Label4.Text = ": Click show TPA81 Temperature Array
Sensor"
        '
        'Temperature
        '
        Me.Temperature.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(222, Byte))
        Me.Temperature.ForeColor =
System.Drawing.SystemColors.ActiveCaption
        Me.Temperature.Location = New System.Drawing.Point(31, 239)
        Me.Temperature.Name = "Temperature"
        Me.Temperature.Size = New System.Drawing.Size(215, 47)
        Me.Temperature.TabIndex = 2
        Me.Temperature.Text = "Click Show"
        Me.Temperature.UseVisualStyleBackColor = True
        '
        'ImageList
        '
        Me.ImageList.ColorDepth =
System.Windows.Forms.ColorDepth.Depth8Bit
        Me.ImageList.ImageSize = New System.Drawing.Size(16, 16)
        Me.ImageList.TransparentColor =
System.Drawing.Color.Transparent
        '
        'GroupBox5
        '
        Me.GroupBox5.Anchor =
CType((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom),
System.Windows.Forms.AnchorStyles)
        Me.GroupBox5.Controls.Add(Me.runvideo)
        Me.GroupBox5.Controls.Add(Me.Label3)
        Me.GroupBox5.Controls.Add(Me.Link)
        Me.GroupBox5.Controls.Add(Me.WebBrowser1)
        Me.GroupBox5.Font = New System.Drawing.Font("Microsoft Sans
Serif", 11.0!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.GroupBox5.Location = New System.Drawing.Point(296, 12)
        Me.GroupBox5.Name = "GroupBox5"
        Me.GroupBox5.RightToLeft =
System.Windows.Forms.RightToLeft.No
        Me.GroupBox5.Size = New System.Drawing.Size(671, 588)
        Me.GroupBox5.TabIndex = 8
        Me.GroupBox5.TabStop = False
        Me.GroupBox5.Text = "IP Camera Video 1"
        '
        'runvideo
        '
```

```
        Me.runvideo.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.runvideo.ForeColor = System.Drawing.Color.DarkRed
        Me.runvideo.Location = New System.Drawing.Point(525, 26)
        Me.runvideo.Name = "runvideo"
        Me.runvideo.Size = New System.Drawing.Size(123, 24)
        Me.runvideo.TabIndex = 3
        Me.runvideo.Text = "Run video"
        Me.runvideo.UseVisualStyleBackColor = True
        '
        'Label3
        '
        Me.Label3.AutoSize = True
        Me.Label3.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label3.Location = New System.Drawing.Point(18, 31)
        Me.Label3.Name = "Label3"
        Me.Label3.Size = New System.Drawing.Size(102, 13)
        Me.Label3.TabIndex = 2
        Me.Label3.Text = "IP camera Address :"
        '
        'Link
        '
        Me.Link.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Link.Location = New System.Drawing.Point(126, 28)
        Me.Link.Name = "Link"
        Me.Link.Size = New System.Drawing.Size(386, 20)
        Me.Link.TabIndex = 1
        '
        'WebBrowser1
        '
        Me.WebBrowser1.Anchor =
System.Windows.Forms.AnchorStyles.None
        Me.WebBrowser1.Location = New System.Drawing.Point(15, 65)
        Me.WebBrowser1.MinimumSize = New System.Drawing.Size(20, 20)
        Me.WebBrowser1.Name = "WebBrowser1"
        Me.WebBrowser1.ScrollBarsEnabled = False
        Me.WebBrowser1.Size = New System.Drawing.Size(640, 501)
        Me.WebBrowser1.TabIndex = 0
        Me.WebBrowser1.Url = New System.Uri("",
System.UriKind.Relative)
        '
        'Label18
        '
        Me.Label18.AutoSize = True
        Me.Label18.Location = New System.Drawing.Point(12, 637)
        Me.Label18.Name = "Label18"
        Me.Label18.Size = New System.Drawing.Size(331, 13)
        Me.Label18.TabIndex = 9
```

```
        Me.Label18.Text = "Assistance from : Department of Electrical
and Computer Engineering"
        '
        'Label19
        '
        Me.Label19.AutoSize = True
        Me.Label19.Location = New System.Drawing.Point(12, 660)
        Me.Label19.Name = "Label19"
        Me.Label19.Size = New System.Drawing.Size(216, 13)
        Me.Label19.TabIndex = 10
        Me.Label19.Text = "Faculty of Engineering   Naresuan
University"
        '
        'Label20
        '
        Me.Label20.AutoSize = True
        Me.Label20.Location = New System.Drawing.Point(12, 683)
        Me.Label20.Name = "Label20"
        Me.Label20.Size = New System.Drawing.Size(429, 13)
        Me.Label20.TabIndex = 11
        Me.Label20.Text = "Adviser : Ph.D. Akaraphunt Vongkunghae  ,
Settha Thangkawanit , Panupong Sornkhom" & _
            ""
        '
        'Label21
        '
        Me.Label21.AutoSize = True
        Me.Label21.Location = New System.Drawing.Point(11, 706)
        Me.Label21.Name = "Label21"
        Me.Label21.Size = New System.Drawing.Size(221, 13)
        Me.Label21.TabIndex = 12
        Me.Label21.Text = "Team : Apocalyse 2009  subject : Robot
Club"
        '
        'ControlRobot
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.BackColor = System.Drawing.SystemColors.ControlLight
        Me.ClientSize = New System.Drawing.Size(1272, 744)
        Me.Controls.Add(Me.Label21)
        Me.Controls.Add(Me.Label20)
        Me.Controls.Add(Me.Label19)
        Me.Controls.Add(Me.Label18)
        Me.Controls.Add(Me.GroupBox5)
        Me.Controls.Add(Me.groupBox1)
        Me.Controls.Add(Me.GroupBox3)
        Me.Controls.Add(Me.GroupBox4)
        Me.Controls.Add(Me.GroupBox2)
        Me.Name = "ControlRobot"
        Me.Text = "Rescue Robot Control"
        Me.WindowState =
System.Windows.Forms.FormWindowState.Maximized
        Me.groupBox1.ResumeLayout(False)
```

```
            Me.groupBox1.PerformLayout()
            Me.Panel3.ResumeLayout(False)
            Me.Panel3.PerformLayout()
            Me.GroupBox2.ResumeLayout(False)
            Me.Panel1.ResumeLayout(False)
            Me.Panel1.PerformLayout()
            Me.StatusStrip1.ResumeLayout(False)
            Me.StatusStrip1.PerformLayout()
            Me.GroupBox3.ResumeLayout(False)
            Me.GroupBox3.PerformLayout()
            Me.GroupBox4.ResumeLayout(False)
            Me.GroupBox4.PerformLayout()
            CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
            Me.GroupBox5.ResumeLayout(False)
            Me.GroupBox5.PerformLayout()
            Me.ResumeLayout(False)
            Me.PerformLayout()

        End Sub

#End Region

        ' Translation of the MS DirectX SDK Joystick example to vb
        ' These three fields hold common data that
        ' different threads will have to access
        Public Shared state As New JoystickState
        Private applicationDevice As Device = Nothing
        Public Shared numPOVs As Integer = 0
        Private SliderCount As Integer = 0 ' number of returned slider
controls
        Private WithEvents timer1 As New Timer


        Private Function InitDirectInput() As Boolean
            ' loop through attached game controller devices
            For Each instance As DeviceInstance In
Manager.GetDevices(DeviceClass.GameControl,
EnumDevicesFlags.AttachedOnly)
                ' example just picks the first available
                applicationDevice = New Device(instance.InstanceGuid)
                Exit For
            Next

            ' If we didn't find one then let user know and quit
            If applicationDevice Is Nothing Then
                MessageBox.Show("Unable to create a Joystick device.
Check please.", "[No Communicate] No joystick found")
                Return False
            End If

            ' Set the data format to the c_dfDIJoystick pre-defined
format.
```

```
        applicationDevice.SetDataFormat(DeviceDataFormat.Joystick)
        ' Set the cooperative level for the device.
        ' (The logical OR here has the effect of making it Exclusive
and foreground)
        applicationDevice.SetCooperativeLevel(Me,
CooperativeLevelFlags.Exclusive Or CooperativeLevelFlags.Foreground)
        ' Enumerate all the objects on the device.

        For Each d As DeviceObjectInstance In
applicationDevice.Objects
            ' For axes that are returned, set the DIPROP_RANGE
property for the
            ' enumerated axis in order to scale min/max values.
            If Not (d.ObjectId And
Convert.ToInt32(DeviceObjectTypeFlags.Axis)) = 0 Then
                ' Set the range for the axis.

applicationDevice.Properties.SetRange(ParameterHow.ById, d.ObjectId,
New InputRange(-1000, +1000))
                ' Update the controls to reflect what
                ' objects the device supports.
                UpdateControls(d)
            End If
        Next
        Return True
    End Function
    Private Sub buttonExit_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles buttonExit.Click
        Close()
    End Sub

    Public Sub GetData()
        timer1.Start()
        ' Make sure there is a valid device.
        If applicationDevice Is Nothing Then Return

        Try

            ' Poll the device for info.
            applicationDevice.Poll()

        Catch notAcqEx As NotAcquiredException
            ' Check to see if either the app
            ' needs to acquire the device, or
            ' if the app lost the device to another
            ' process.
            Try
                ' Acquire the device.
                applicationDevice.Acquire()
            Catch iex As InputException
                ' Failed to acquire the device.
                ' This could be because the app
                ' doesn't have focus.
```

```
                Return
            End Try
        Catch inpLostEx As InputLostException
            ' Check to see if either the app
            ' needs to acquire the device, or
            ' if the app lost the device to another
            ' process.
            Try
                ' Acquire the device.
                applicationDevice.Acquire()
            Catch iex As InputException
                ' Failed to acquire the device.
                ' This could be because the app
                ' doesn't have focus.
                Return
            End Try
        End Try

        ' Get the state of the device.
        Try
            state = applicationDevice.CurrentJoystickState
            ' Catch any exceptions. None will be handled here,
            ' any device re-aquisition will be handled above.
        Catch iex As InputException
            Return
        End Try
        UpdateUI()
    End Sub
    '********Joystick Value to serialPort*********
    Private Sub UpdateUI()

        ' This function updated the UI with
        ' Joystick state information.

        Dim strText As String = ""

        labelXAxis.Text = state.X.ToString()
        labelYAxis.Text = state.Y.ToString()
        labelZAxis.Text = state.Z.ToString()

        labelXRotation.Text = state.Rx.ToString()
        labelYRotation.Text = state.Ry.ToString()
        labelZRotation.Text = state.Rz.ToString()

        Dim slider As Integer() = state.GetSlider()

        labelSlider0.Text = slider(0).ToString()
        labelSlider1.Text = slider(1).ToString()

        Dim pov As Integer() = state.GetPointOfView()

        labelPOV0.Text = pov(0).ToString()
        labelPOV1.Text = pov(1).ToString()
```

```
            labelPOV2.Text = pov(2).ToString()
            labelPOV3.Text = pov(3).ToString()

            ' Fill up text with which buttons are pressed
            Dim buttons As Byte() = state.GetButtons()

            Dim _button As Integer = 0
            For Each b As Byte In buttons
                _button += 1
                If Not ((b And &H80) = 0) Then
                    strText = _button.ToString()
                    _button += 1
                End If
            Next
            labelButtons.Text = strText

            '-----------------------------------------------------------------
----------------------

'*****************************************************************************
**************
            'Joystick Value to serialPort

'*****************************************************************************
**************
            ' ^ Forward :
            If labelPOV0.Text = "0" Then
                SendSerialPort.Text = "w"
                '----------------------------------------
                ' v Backward :
            ElseIf labelPOV0.Text = "18000" Then
                SendSerialPort.Text = "x"
                '----------------------------------------
                ' < Left :
            ElseIf labelPOV0.Text = "27000" Then
                SendSerialPort.Text = "a"
                '----------------------------------------
                ' > Right :
            ElseIf labelPOV0.Text = "9000" Then
                SendSerialPort.Text = "d"
                '----------------------------------------
                'Arm Down :
            ElseIf labelButtons.Text = "1" Then
                SendSerialPort.Text = ";"
                '----------------------------------------
                'Arm Up :
            ElseIf labelButtons.Text = "3" Then
                SendSerialPort.Text = "k"
                '----------------------------------------
                ' < Left speed:
            ElseIf labelButtons.Text = "7" Then
                SendSerialPort.Text = "c"
                '----------------------------------------
```

```
            ' > Right speed:
        ElseIf labelButtons.Text = "8" Then
            SendSerialPort.Text = "z"
            '----------------------------------------

            'Arm Camera ----------------------------
            'Arm Camera: Up
        ElseIf labelYAxis.Text = "-1000" Then
            SendSerialPort.Text = "u"
            '----------------------------------------
            'Arm Camera : Down
        ElseIf labelYAxis.Text = "1000" Then
            SendSerialPort.Text = "i"
            '----------------------------------------
            'Camera : Up
        ElseIf labelZRotation.Text = "-1000" Then
            SendSerialPort.Text = "g"
            '----------------------------------------
            'Camera : Down
        ElseIf labelZRotation.Text = "1000" Then
            SendSerialPort.Text = "h"
            '----------------------------------------
            'Camera : Left
        ElseIf labelZAxis.Text = "-1000" Then
            SendSerialPort.Text = "t"
            '----------------------------------------
            'Camera : Right
        ElseIf labelZAxis.Text = "1000" Then
            SendSerialPort.Text = "y"
            '----------------------------------------
        Else
            SendSerialPort.Text = "0"
        End If

'*************************************************************************
*************
        txtSendvalue()
        '-------------------------------------------------------------
----------------------
    End Sub
    Public Sub UpdateControls(ByVal d As DeviceObjectInstance)

        ' Set the UI to reflect what objects the Joystick supports.
        If (ObjectTypeGuid.XAxis.Equals(d.ObjectType)) Then
            labelXAxis.Enabled = True
            labelXAxisText.Enabled = True
        End If

        If (ObjectTypeGuid.YAxis.Equals(d.ObjectType)) Then
            labelYAxis.Enabled = True
            labelYAxisText.Enabled = True
        End If
```

```
        If (ObjectTypeGuid.ZAxis.Equals(d.ObjectType)) Then
            labelZAxis.Enabled = True
            labelZAxisText.Enabled = True
        End If

        If (ObjectTypeGuid.RxAxis.Equals(d.ObjectType)) Then
            labelXRotation.Enabled = True
            labelXRotationText.Enabled = True
        End If

        If (ObjectTypeGuid.RyAxis.Equals(d.ObjectType)) Then
            labelYRotation.Enabled = True
            labelYRotationText.Enabled = True
        End If

        If (ObjectTypeGuid.RzAxis.Equals(d.ObjectType)) Then
            labelZRotation.Enabled = True
            labelZRotationText.Enabled = True
        End If

        If (ObjectTypeGuid.Slider.Equals(d.ObjectType)) Then
            SliderCount += 1
            Select Case SliderCount
                Case 0
                    labelSlider0.Enabled = True
                    labelSlider0Text.Enabled = True
                Case 1
                    labelSlider1.Enabled = True
                    labelSlider1Text.Enabled = True
            End Select

        End If

        If (ObjectTypeGuid.PointOfView.Equals(d.ObjectType)) Then
            numPOVs += 1
            Select Case numPOVs

                Case 0
                    labelPOV0.Enabled = True
                    labelPOV0Text.Enabled = True
                Case 1
                    labelPOV1.Enabled = True
                    labelPOV1Text.Enabled = True
                Case 2
                    labelPOV2.Enabled = True
                    labelPOV2Text.Enabled = True
                Case 3
                    labelPOV3.Enabled = True
                    labelPOV3Text.Enabled = True
            End Select
        End If

    End Sub
```

```
    Private Sub timer1_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles timer1.Tick
        GetData()
    End Sub
    'Group Box Name Serial Port of  Communication the Robot

'**********************************************************************
********************
    '******** : Serial Port of Communication the Robot :
**********************************

'**********************************************************************
********************
    Private Sub ConnectPort()
        'Check port properties ตรวจสอบว่าพอร์ต
        If cboPort.SelectedIndex = -1 Then Return
        If cboRate.SelectedIndex = -1 Then Return
        If cboParity.SelectedIndex = -1 Then Return
        If cboData.SelectedIndex = -1 Then Return
        If cbostop.SelectedIndex = -1 Then Return

        'close port
        If mPort IsNot Nothing Then mPort.Close()

        'set port properties
        mPort.PortName = cboPort.Text
        mPort.BaudRate = Convert.ToInt32(cboRate.Text)
        mPort.DataBits = Convert.ToInt32(cboData.Text)

        mPort.Parity = CType([Enum].Parse(GetType(IO.Ports.Parity),
cboParity.Text), IO.Ports.Parity)
        mPort.StopBits =
CType([Enum].Parse(GetType(IO.Ports.StopBits), cbostop.Text),
IO.Ports.StopBits)

        'try connection
        Try
            mPort.Open()
        Catch ex As Exception
            MessageBox.Show(ex.Message, Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Return
        End Try

        'set
        mPort.NewLine = ControlChars.Cr
        'mPort.ReadTimeout = 1000   'this is not needed as we are
going to use DataReceived event
        AddHandler mPort.DataReceived, AddressOf DataReceived

        'save info
```

```
        mIniFile.WriteParemeter("Port", "Port", cboPort.Text)
        mIniFile.WriteParemeter("Port", "BaudRate", cboRate.Text)
        mIniFile.WriteParemeter("Port", "DataBits", cboData.Text)
        mIniFile.WriteParemeter("Port", "StopBits", cbostop.Text)
        mIniFile.WriteParemeter("Port", "Parity", cboParity.Text)
    End Sub
    ''' <summary>
    ''' Note that this is called from a backrbound thread
    ''' </summary>
    ''' <param name="sender"></param>
    ''' <param name="e"></param>
    ''' <remarks></remarks>
    Private Sub DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs)
        If e.EventType = IO.Ports.SerialData.Chars Then
ShowReceived()
    End Sub
    '-------------------------------------------------------------------
-------------
    ''' <summary>
    ''' The following handles the actual call from DataReceived in a
thread safe way.
    ''' Value Temperature Sensor and Encoder:
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub ShowReceived()
        Static s As String = ""
        Dim idx As Integer = 0
        Dim sData() As String
        If Me.InvokeRequired Then
            'invoke the thread

            Me.Invoke(New MethodInvoker(AddressOf ShowReceived))
        Else
            'attach the left over of the line
            s += mPort.ReadExisting()
            sData = s.Split(ControlChars.Cr)
            For i As Integer = 0 To sData.Length - 1
                s = sData(i)
                'remove line feed, if any
                s = s.Replace(ControlChars.Lf, "")
                'removed the last line if it is part of the line
being added
                If (i = 0) Then
                    lstData.Items.Clear()
                    idx = lstData.Items.Count - 1
                    If idx > 0 Then
                        If s.Contains(lstData.Items(idx).ToString)
Then
                            'select one item above to prevent the
RemoveAt forces the TopIndex to 0
                            lstData.SelectedIndex = idx - 1
                            'remove item
```

```
                                lstData.Items.RemoveAt(idx)
                                'remove fake selection,
                                lstData.SelectedIndex = -1

                            End If
                        End If
                    End If
                    'show line
                    idx = lstData.Items.Add(s)
                Next
                'show the data
                If Me.Visible And idx > lstData.TopIndex Then
lstData.TopIndex = idx


'*****************************************************************************
********
                'Received of Value at Temperature Sensor:
**********************************
                Dim groupTemp() As String = New String() {"T-10", "T-9",
"T-8", "T-7", "T-6", "T-5", "T-4", "T-3", "T-2", "T1", "T0", "T1",
"T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9", "T10", "T11", "T12",
"T13", "T14", "T15", "T16", "T17", "T18", "T19", "T20", "T21", "T22",
"T23", "T24", "T25", "T26", "T27", "T28", "T29", "T30", "T31", "T32",
"T33", "T34", "T35", "T36", "T37", "T38", "T39", "T40", "T41", "T42",
"T43", "T44", "T45", "T46", "T47", "T48", "T49", "T50"}
                'Chack a group Temp array
                If Me.Visible Then
                    For Each i In groupTemp
                        If lstData.Items(idx).ToString = i Then
                            Temp.Text = lstData.Items(idx).ToString
                        Else
                        End If
                    Next i
                End If

'*****************************************************************************
********
                'save the non-complete line
                s = sData(sData.Length - 1)
            End If
        End Sub
        Private Sub ControlRobot_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
            If InitDirectInput() = False Then Close()
            timer1.Start()
            Dim i As Integer
            Dim s As String
            Dim x As Object()

            'ini file
```

```
        mIniFile = New
w2IniFile(IO.Path.Combine(System.AppDomain.CurrentDomain.BaseDirector
y(), "wterm.ini"))

        'set port properties
        cboPort.DropDownStyle = ComboBoxStyle.DropDownList
        For Each s In My.Computer.Ports.SerialPortNames
            cboPort.Items.Add(s)
        Next
        s = mIniFile.ReadParameter("Port", "Port", "")
        cboPort.SelectedIndex = cboPort.Items.IndexOf(s)

        cboRate.DropDownStyle = ComboBoxStyle.DropDownList
        x = New Object() {1200, 1800, 2400, 4800, 7200, 9600, 14400,
19200, 38400, 57600, 115200, 128000}
        cboRate.Items.AddRange(x)
        i = mIniFile.ReadParameter("Port", "BaudRate", 0)
        cboRate.SelectedIndex = cboRate.Items.IndexOf(i)

        cboParity.DropDownStyle = ComboBoxStyle.DropDownList
        For Each s In [Enum].GetNames(GetType(IO.Ports.Parity))
            cboParity.Items.Add(s)
        Next
        s = mIniFile.ReadParameter("Port", "Parity", "")
        cboParity.SelectedIndex = cboParity.Items.IndexOf(s)

        cboData.DropDownStyle = ComboBoxStyle.DropDownList
        x = New Object() {4, 5, 6, 7, 8}
        cboData.Items.AddRange(x)
        i = mIniFile.ReadParameter("Port", "DataBits", 0)
        cboData.SelectedIndex = cboData.Items.IndexOf(i)

        cbostop.DropDownStyle = ComboBoxStyle.DropDownList
        For Each s In [Enum].GetNames(GetType(IO.Ports.StopBits))
            cbostop.Items.Add(s)
        Next
        s = mIniFile.ReadParameter("Port", "StopBits", "")
        cbostop.SelectedIndex = cbostop.Items.IndexOf(s)

        'button
        btnConnect.ImageList = ImageList
        btnConnect.Text = "Connecting"
        btnConnect.ImageKey = btnConnect.Text

        'terminal and enhancement
        lblPort.Text = ""
        'PanelIO.Enabled = False แบ่ง Panal คือแบ่งเป็นโซน

        mTxt = New w2TxtBuffer(txtSend)
        mTxt.ClearLastCommand = True
        '' chkCap.Checked = mIniFile.ReadParameter("Port", "CAPS",
False)
```

```
        lstData.MultiColumn = False
        lstData.SelectionMode = SelectionMode.MultiExtended
        Dim Menu As New w2ContextMenu(lstData, ImageList)

        'me
        Me.Text = "Project : Rescue Robot Control : "
        Me.MinimumSize = Me.Size()
    End Sub
    Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnConnect.Click
        If btnConnect.Text = "Connecting" Then
            ConnectPort()
        Else
            If mPort.IsOpen Then
                mPort.Close()
                RemoveHandler mPort.DataReceived, AddressOf
DataReceived
            End If
        End If

        If mPort.IsOpen Then
            btnConnect.Text = "Disconnect"
            'Panel1.Enabled = True แก้ Panal ล็อคไม่ให้ใช้งาน
            '
            '
            lblPort.Text = "Connected to " & cboPort.Text & " : " &
cboRate.Text & " : " & cboParity.Text & " : " & cboData.Text & " : "
& cbostop.Text
        Else
            btnConnect.Text = "Connecting"
            lblPort.Text = "Disconnected : Click Connecting"
            'Panel1.Enabled = False

        End If
        btnConnect.Image = ImageList.Images(btnConnect.Text)
    End Sub

    Private Sub chkCap_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        ''mTxt.Uppercase = chkCap.Checked
        mIniFile.WriteParemeter("Port", "CAPS",
mTxt.Uppercase.ToString)
    End Sub

'*********************************************************************
    'Send SerialPort RS232
'*********************************************************************
    Private Sub txtSendvalue()
```

```
'*************************************************************************
'Joystick Value to serialPort
'*************************************************************************
        ' ^ Forward :
        If SendSerialPort.Text = "w" Then
            SendSerialPort.Text = "w"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            ' v Backward :
        ElseIf SendSerialPort.Text = "x" Then
            SendSerialPort.Text = "x"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            ' < Left :
        ElseIf SendSerialPort.Text = "a" Then
            SendSerialPort.Text = "a"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            ' > Right :
        ElseIf SendSerialPort.Text = "d" Then
            SendSerialPort.Text = "d"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            'Arm Down :
        ElseIf SendSerialPort.Text = "k" Then
            SendSerialPort.Text = "k"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            'Arm Up :
        ElseIf SendSerialPort.Text = ";" Then
            SendSerialPort.Text = ";"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            ' < Left speed:
        ElseIf SendSerialPort.Text = "c" Then
            SendSerialPort.Text = "c"
            If btnConnect.Text = "Disconnect" Then
                mPort.WriteLine(SendSerialPort.Text)
            End If
            '-----------------------------------------
            ' > Right speed:
```

```
ElseIf SendSerialPort.Text = "z" Then
    SendSerialPort.Text = "z"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------

    'Arm Camera ------------------------------
    'Arm Camera: Up
ElseIf SendSerialPort.Text = "u" Then
    SendSerialPort.Text = "u"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
    'Arm Camera : Down
ElseIf SendSerialPort.Text = "i" Then
    SendSerialPort.Text = "i"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
    'Camera : Up
ElseIf SendSerialPort.Text = "g" Then
    SendSerialPort.Text = "g"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
    'Camera : Down
ElseIf SendSerialPort.Text = "h" Then
    SendSerialPort.Text = "h"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
    'Camera : Left
ElseIf SendSerialPort.Text = "y" Then
    SendSerialPort.Text = "y"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
    'Camera : Right
ElseIf SendSerialPort.Text = "t" Then
    SendSerialPort.Text = "t"
    If btnConnect.Text = "Disconnect" Then
        mPort.WriteLine(SendSerialPort.Text)
    End If
    '----------------------------------------
ElseIf SendSerialPort.Text = "0" Then
    SendSerialPort.Text = "0"
    If btnConnect.Text = "Disconnect" Then
```

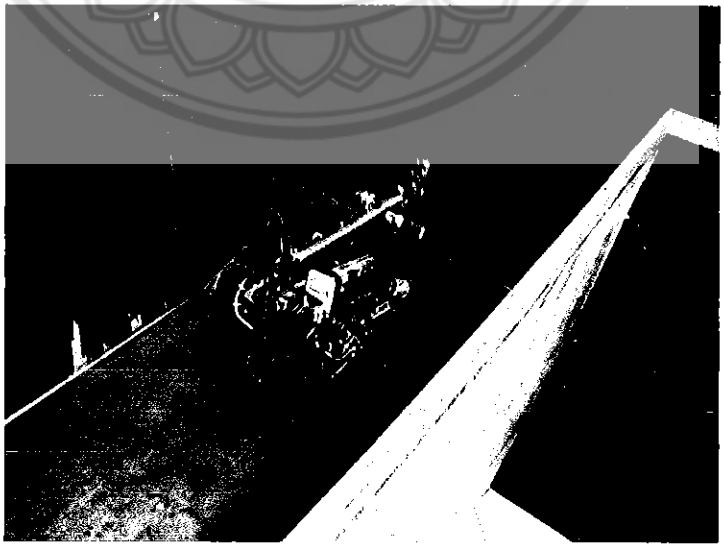
```
                mPort.WriteLine(SendSerialPort.Text)
            End If
        End If
        '-----------------------------------------
    End Sub

    Private Sub runvideo_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles runvideo.Click
        Link.Text = "http:\\localhost\video\video.php"
        If Link.Text = "" Then
            MessageBox.Show("No IP Camera Address . Check please.",
"[No Link] No IP Camera")
        Else
            WebBrowser1.Navigate(Link.Text)
        End If
    End Sub


'*************************************************************************
**********************************
    'Sensor Temperature
    Private Sub Temperature_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Temperature.Click
        mPort.WriteLine("1")
    End Sub
End Class
```

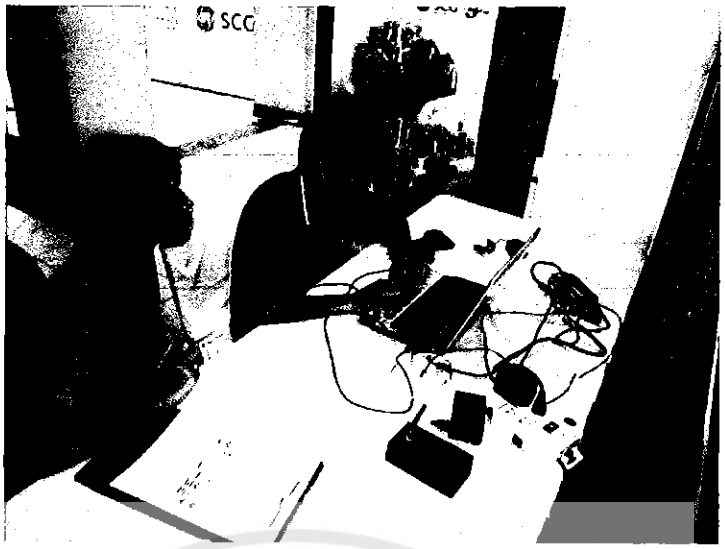
ภาคผนวก ข

การเข้าร่วมการแข่งขัน Thailand Rescue Robot 2009

การแข่งขันหุ่นยนต์กู้ภัย Thailand Rescue Robot Championship เป็นการแข่งขันประดิษฐ์หุ่นยนต์ เพื่อกู้ภัยในสถานการณ์จำลองอุบัติภัยที่เกิดขึ้น อาทิ อัคคีภัย อุทกภัย ภัยจากแผ่นดินไหว หรือภัยอื่นๆ โดยหุ่นยนต์ที่สร้างขึ้นจะต้องมีความสามารถในการค้นหาผู้รอดชีวิต หรือผู้เสียชีวิต ที่ตกค้างอยู่ในซากปรักหักพังต่างๆ ซึ่งเป็นการแข่งขันที่ท้าทาย และมีประโยชน์ต่อสังคม โดยทีมชนะเลิศจะได้เป็นตัวแทนประเทศไทยเข้าร่วม การแข่งขันหุ่นยนต์ ระดับโลก World RoboCup ประเภทหุ่นยนต์กู้ภัย World RoboCup Rescue โดยประเทศต่างๆ จะสลับกันเป็นเจ้าภาพ

World RoboCup เป็นการแข่งขันหุ่นยนต์ระดับโลก จัดขึ้นอย่างต่อเนื่องเป็นปีที่ 13 แบ่งการแข่งขันออกเป็น 4 ประเภท ได้แก่ ฟุตบอลหุ่นยนต์ (RoboCup Soccer) หุ่นยนต์กู้ภัย (RoboCup Rescue) หุ่นยนต์ทำงานบ้าน (RoboCup @Home) และหุ่นยนต์ระดับเยาวชนอายุไม่เกิน 18 ปี (RoboCup Junior) โดยในปี 2010 นี้ จัดขึ้นที่ ประเทศสิงค์โปร มีผู้สมัครเข้าแข่งขันกว่า 300 ทีม จากกว่า 30 ประเทศทั่วโลก ในปีนี้ทางมหาวิทยาลัยมหิดลเป็นเจ้าภาพจัดการแข่งเป็นปีที่ 3 กำหนดการแข่งขันรอบคัดเลือกวันที่ 23-26 ตุลาคม 2552 ณ มหาวิทยาลัยมหิดล ศาลายา จ. นครปฐมรอบชิงชนะเลิศ 7- 10 ธันวาคม 2552 ณ อาคารนิมิตรบุตร สนามกีฬาแห่งชาติ ซึ่งทางเราได้มีการเข้าร่วมการแข่งขันหุ่นยนต์กู้ภัยปี 2552 โดยส่งในนามมหาวิทยาลัยนเรศวร ชื่อทีมว่า Apocalypse ชมรมหุ่นยนต์ มหาวิทยาลัยนเรศวร ซึ่งเป็นการส่งเข้าร่วมแข่งขันในปีแรก โดยแสดงรูปภาพประกอบดังนี้




SCG

SCG
SCG

จากการแข่งขันทีม Apocalypse ได้อันดับที่ 27 จาก 108 ทีม ทั่วประเทศ ถือว่าเป็นการเริ่มต้นที่ดีและทำให้เราสมาชิกมีประสบการณ์ในการแข่งขันและได้เห็นถึงความสามารถของทีมอื่น ๆ ประโยชน์ที่ได้รับจะต้องกับไปพัฒนาหุ่นยนต์ให้มีความสามารถในการเคลื่อนที่ได้อย่างเร็วในการเข้าถึงเป้าหมายและพัฒนาด้านความสามารถต่าง ๆ ให้เท่าเทียบกับนานาประเทศ

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายนนท์ จูเทศ

ภูมิลำเนา 8/776 หมู่ 2 ต.นครสวรรค์ อ.เมือง
จ.นครสวรรค์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียน
  สตรีนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิศวกรรมคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **naj_narak@hotmail.com**

ชื่อ นายมฆวาล ผ่องใส

ภูมิลำเนา 8 หมู่ 8 ต.หนองไม้กอง อ.ไทรงาม
จ.กำแพงเพชร

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียน
  ไทรงามพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิศวกรรมคอมพิวเตอร์
  คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **Mpongsai@gmail.com**