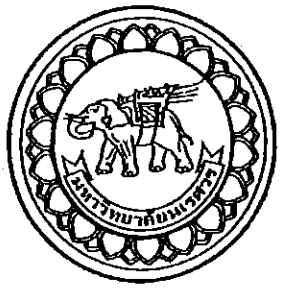


หุ่นยนต์ควบคุมโดยไมโครคอนโทรลเลอร์ผ่านระบบไวเลส

ROBOT CONTROL BY MICROCONTROLLER

USING WIRELESS SYSTEM

นายกิตติพงษ์ สุขเดหา รหัส 49363939

นายชัยสิริ มูลอุด รหัส 49360372

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 ส.ค. 2555
เลขทะเบียน 15747797
เลขเรียกหนังสือ ว.
มหาวิทยาลัยนเรศวร TK174 ก 2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	หุ่นยนต์ควบคุมโดยไมโครคอนโทรลเลอร์ผ่านระบบไวเลส	
ผู้ดำเนินโครงงาน	นายกิตติพงษ์ สุขเคหา	รหัส 49363939
	นายชัยสิริ มูลอุด	รหัส 49360372
ที่ปรึกษาโครงงาน	ดร.อัครพันธ์ วงศ์กังแห	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2552	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

..... ที่ปรึกษาโครงงาน
(ดร.อัครพันธ์ วงศ์กังแห)

..... กรรมการ
(ดร.ชัยรัตน์ พินทอง)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา)

ชื่อหัวข้อโครงงาน	**หุ่นยนต์ควบคุมโดยไมโครคอนโทรลเลอร์ผ่านระบบไวเลส**	
ผู้ดำเนินโครงงาน	**นายกิตติพงษ์ สุขเดหา**	**รหัส 49363939**
	นายชัยสิริ มูลอุด	**รหัส 49360372**
ที่ปรึกษาโครงงาน	**ดร.อัครพันธ์ วงศ์กังแห**	
สาขาวิชา	**วิศวกรรมไฟฟ้า**	
ภาควิชา	**วิศวกรรมไฟฟ้าและคอมพิวเตอร์**	
ปีการศึกษา	2552	

.....

บทคัดย่อ

โครงงานนี้ได้เกิดขึ้นจากแนวคิดที่ต้องการอุปกรณ์ค้นหาสิ่งต่างๆในพื้นที่ที่มนุษย์ไม่อาจจะเข้าถึงได้ อาทิเช่น บริเวณที่มีแก๊สพิษรั่ว บริเวณที่เกิดเพลิงไหม้ หรือเกิดจากภัยพิบัติธรรมชาติที่ทำให้มนุษย์ไม่สามารถเข้าถึงได้ จึงเกิดแนวคิดในการสร้างและออกแบบการควบคุมการทำงานของหุ่นยนต์ระยะไกลผ่านระบบคอมพิวเตอร์ หรืออุปกรณ์ต่างๆที่สามารถเชื่อมต่อได้ ทำให้สามารถรักษาชีวิตเข้าไปช่วยผู้รอดชีวิตที่ยังคงติดอยู่ในภัยพิบัติได้

Project title	Robot Control by Microcontroller using Wireless System
Name	Mr.Kittipong Sukkeha ID. 49363939
	Mr.Chaisiri Moon-Aut ID. 49360372
Project advisor	Mr.Akaraphunt Vongkunghae, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2009

.....

Abstract

This project is a work on a wireless communication system for controlling a mobile rescue robot. The propose of this project is to control the robot wirelessly , gaining the movement abilities of the robot , allowing the robot to be control in case the control wire is not allowed

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้จัดทำโครงงานเรื่อง หุ่นยนต์ควบคุมโดยไมโครคอนโทรลเลอร์ผ่านระบบไวเลส ตั้งแต่ต้นเดือน มิถุนายน พ.ศ. 2552 ถึงเดือน มีนาคม พ.ศ. 2553 ส่งผลให้ข้าพเจ้าได้รับความรู้ และประสบการณ์ต่างๆ ที่มีค่ามากมาย สำหรับโครงงานนี้ สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนหลายฝ่าย ที่กรุณาสละเวลามาเป็นกรรมการการสอบโครงงาน

1. ดร.อัครพันธ์ วงศ์กังแห อาจารย์ที่ปรึกษาโครงงาน
2. ดร.ชัยรัตน์ พินทอง อาจารย์กรรมการสอบโครงงาน
3. ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา อาจารย์กรรมการสอบโครงงาน

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่มีส่วนร่วมในการให้ข้อมูลและเป็นที่ปรึกษาในการทำโครงงานนี้จนเสร็จสมบูรณ์ และให้ความเข้าใจเกี่ยวกับการทำโครงงาน

ผู้จัดทำโครงงาน
นายกิตติพงษ์ สุขเคหา
นายชัยสิริ มูลอุด

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ซ
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบข่ายของโครงงาน 1
1.4 ขั้นตอนการดำเนินการ 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 2
1.6 งบประมาณของโครงงาน 3

บทที่ 2 หลักการและทฤษฎี 4
2.1 Microcontroller 4
2.2 Wireless 4
2.3 H-Bridge Switching 4
2.4 ไอซีรักษาแรงดัน 6
Shunt Regulator(แบบขนาน) 6
Series Regulator (แบบอนุกรม) 6
Map basic circuit of Series Regulator 7
2.5 พอร์ตสื่อสารอนุกรม (RS-232) 8
2.6 โปรแกรม Visual Basic (VB) 9

สารบัญ (ต่อ)

หน้า

บทที่ 3 การดำเนินการและการปฏิบัติการ 10

3.1 การติดต่อสื่อสารผ่านRS-232 11

3.2 อุปกรณ์การรับส่งข้อมูลผ่าน Module Wireless 13

3.3 การติดต่อสื่อสารระหว่างอุปกรณ์ไมโครคอนโทรลเลอร์ 15

3.4 การสั่งอุปกรณ์ต่างๆด้วยไมโครคอนโทรลเลอร์ 16

บทที่ 4 ผลที่ได้จากการทดลอง 19

4.1 การติดต่อสื่อสารผ่าน RS-232 19

4.1.1 ผลการทดลองการรับส่งข้อมูลผ่าน Max 232 19

4.1.2 การเขียนโปรแกรมคอมพิวเตอร์เพื่อควบคุมอุปกรณ์ภายนอก 21

4.2 อุปกรณ์การรับส่งข้อมูลผ่าน Module Wireless 24

4.3 การติดต่อสื่อสารระหว่างอุปกรณ์ไมโครคอนโทรลเลอร์ 26

4.4 การสั่งอุปกรณ์ต่างๆด้วยไมโครคอนโทรลเลอร์ 28

4.5 ทดสอบการทำงานอุปกรณ์ฝั่งรับ 31

4.6 อุปกรณ์เสริม 33

4.6.1 อุปกรณ์แปลงสัญญาณกล้องเป็น AV 33

4.6.2 อุปกรณ์แปลงสัญญาณ AV เข้าคอมพิวเตอร์ 34

4.6.3 โปรแกรมบังคับจอยสติก Joy To Key 34

4.6.4 อุปกรณ์แปลง RS-232 เป็น USB 35

4.7 การทำงานทั้งหมดของระบบ 36

4.7.1 เชื่อมต่ออุปกรณ์เสริมทั้งหมดเข้าสู่คอมพิวเตอร์ 36

4.7.2 เชื่อมสายไฟทั้งหมดเข้าสู่ตัวแผงวงจรของหุ่นยนต์ 36

4.7.3 ทดสอบการควบคุมหุ่นยนต์ผ่านคอมพิวเตอร์ 37

บทที่ 5 สรุปผลโครงงาน 39

5.1 สรุปผลของโครงงาน 39

5.2 ปัญหาที่เกิดขึ้น 39

5.3 การแก้ปัญหาและข้อเสนอแนะ 40

สารบัญ (ต่อ)

หน้า

5.4 แนวทางการพัฒนาเพิ่มเติม 40
ภาคผนวก ก 41
ภาคผนวก ข 43
ประวัติผู้เขียนโครงงาน 66

สารบัญตาราง

ตารางที่ หน้า

4.1 ตารางแสดงผลการทดสอบการเชื่อมต่อโปรแกรม VB เข้ากลับไมโครคอนโทรเลอร์ผ่าน Max เบอร์ต่าง เงื่อนไขการจ่ายไฟเลี้ยง 3 โวลต์..... 24
4.2 ตารางแสดงผลการทดสอบการเชื่อมต่อโปรแกรม VB เข้ากลับไมโครคอนโทรเลอร์ผ่าน Max เบอร์ต่าง เงื่อนไขการจ่ายไฟเลี้ยง 5 โวลต์..... 24
4.3 ตารางแสดงผลการทดสอบการส่งค่าไปที่หุ่นยนต์เพื่อดูผลการทำงาน..... 24
4.4 ตารางแสดงผลการทดสอบของ TRW 2.4 G โดยการส่ง Data = 0x2A..... 26
4.5 ตารางแสดงผลการทดสอบของ TRW 2.4 G โดยการส่ง Data = 0x15..... 26
4.6 ตารางแสดงผลการทดสอบการทำงาน MCU (ไมโครคอนโทรลเลอร์) โดยไม่ผ่าน IC SN74HC245N..... 27
4.7 ตารางแสดงผลการทดสอบการทำงาน MCU (ไมโครคอนโทรลเลอร์) โดยผ่าน IC SN74HC245N..... 28
4.8 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1010..... 30
4.9 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 0101..... 30
4.10 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1001..... 30
4.11 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 0110..... 30
4.12 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1111 และ 0000..... 30
4.13 ตารางแสดงผลการทดสอบการการรับส่งระหว่างตัวส่ง (Transceiver) กับหุ่นยนต์ (Receiver) 33

สารบัญรูป

รูปที่ หน้า

2.1 การทำงานวงจร H-Bridge Switching 4
2.2 การทำงานแบบ Forward 5
2.3 การทำงานแบบ Reverse 5
2.4 แสดงแผนผังการทำงานของเร็กกูเลเตอร์แบบขนาน 6
2.5 แสดงแผนผังการทำงานของเร็กกูเลเตอร์แบบอนุกรม 6
2.6 แสดงแผนผังวงจรพื้นฐานของเร็กกูเลเตอร์แบบอนุกรม 7
2.7 แสดงการทำงานแบบ Half-Duplex กับ Full-Duplex 8
3.1 ภาพรวมของระบบทั้งหมดในโครงงาน 10
3.2 ภาพรวมของการเชื่อมต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์ 11
3.3 โปรแกรมการสั่งการข้อมูลจากคอมพิวเตอร์ 12
3.4 วงจรการทำงาน Computer to transceiver module 12
3.5 ภาพรวมของระบบรับส่งข้อมูลผ่านทาง Wireless 13
3.6 วงจรการรับสัญญาณเพื่อแสดงผลของอุปกรณ์ฝั่งรับข้อมูล 14
3.7 วงจรการส่งข้อมูลทาง wireless ผ่านการควบคุมโดยคอมพิวเตอร์ 14
3.8 ภาพรวมของระบบการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากัน 2 ตัว 15
3.9 วงจรการเชื่อมต่อไมโครคอนโทรลเลอร์ 2 ตัวลงในฝั่งอุปกรณ์รับ 15
3.10 ภาพรวมของระบบไมโครคอนโทรลเลอร์สั่งการอุปกรณ์ 16
3.11 วงจรการเชื่อมต่ออุปกรณ์ต่างๆของอุปกรณ์ฝั่งรับ 17
4.1 การต่อ Max 232N เพื่อเชื่อมต่อเข้ากับคอมพิวเตอร์ 19
4.2 วงจร Max 232N ต่อเข้าไมโครคอนโทรลเลอร์ 20
4.3 โปรแกรมส่งค่าและอ่านค่า RS-232 20
4.4 ส่งค่าตัวหนังสือออกไปจากคอมพิวเตอร์ 21
4.5 อ่านค่าตัวหนังสือที่เคยส่งออกไปจากคอมพิวเตอร์ 21
4.6 ต่อไมโครคอนโทรเลอร์ และ Max232N เข้ากับคอมพิวเตอร์ 22
4.7 พัฒนารูปแบบโปรแกรมควบคุมการทำงาน Robot 22
4.8 เขียนโปรแกรมควบคุมการทำงานอุปกรณ์ภายนอก 23
4.9 ผลการทำงานของโปรแกรม 23

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.10 Module Wireless TRW 2.4G ฝั่งตัวส่ง (Transceiver) 25
4.11 Module Wireless TRW 2.4G ฝั่งตัวรับ (Receiver) 25
4.12 วงจรการต่อไมโครคอนโทรลเลอร์ 2 ตัวเข้าหากันผ่าน SN74HC245N 27
4.13 วงจร IC L293D ต่อเข้ากับไมโครคอนโทรลเลอร์ 28
4.14 วัดแรงดันที่ขา Input ของ L293D 29
4.15 วัดแรงดันที่ขา Output ของ L293D 29
4.16 ประกอบวงจรทั้งหมด ลงไปในอุปกรณ์รับข้อมูล (Receiver Module) 31
4.17 ส่งสัญญาณจากทางด้านอุปกรณ์ส่งข้อมูล (Transceiver Module) 32
4.18 หุ่นยนต์เดินตามการทำงานของอุปกรณ์ส่งข้อมูล (Transceiver Module) 32
4.19 แสดงผลภาพจากกล้องที่หุ่นยนต์ผ่านโทรทัศน์ 33
4.20 การเชื่อมต่ออุปกรณ์แปลงสัญญาณ AV เข้ากับอุปกรณ์แปลงสัญญาณเข้าคอมพิวเตอร์ 34
4.21 การแสดงผลภาพของอุปกรณ์แปลงสัญญาณ AV เข้าคอมพิวเตอร์ 34
4.22 จอยสติกที่ใช้ในการติดต่อกับคอมพิวเตอร์ 35
4.23 ผลการทดสอบจอยสติกติดต่อสื่อสารกับโปรแกรม VB2005 35
4.24 ผลการทดสอบการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่าน RS-232 35
4.25 ต่ออุปกรณ์ทั้งหมดเข้าสู่คอมพิวเตอร์ 36
4.26 ต่อสายไฟต่างๆเข้าสู่วงจร 36
4.27 วัดแรงดันไฟฟ้าจากจุดต่างๆ เพื่อเช็คความพร้อม 37
4.28 สั่งการเปิดไฟที่หน้าตัวหุ่นยนต์ 37
4.29 สั่งการให้หุ่นยนต์เดินไปข้างหน้าและหน้าจอแสดงผลขณะเดิน 38

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันได้มีภัยธรรมชาติมากมายที่เกิดขึ้นโดยไม่คาดคิดเช่น แผ่นดินไหว น้ำท่วม หรือสึนามิ และเหตุการณ์ต่างๆที่ก่อให้เกิดความเสียหายมากมาย ทำให้เกิดตึกถล่ม บ้านเรือนเสียหายพังทลาย ทำให้ต้องมีการกู้ภัยต่อชีวิตและทรัพย์สินอยู่บ่อยครั้ง แต่ในการกู้ภัยในแต่ละครั้งอาจจะช่วยชีวิตได้ไม่สำเร็จเสมอไป เนื่องจากความลำบากในสิ่งกีดขวาง สภาพแวดล้อม และอุปสรรคต่างๆมากมาย โดยในบางสภาพแวดล้อมนั้นอาจจะก่อให้เกิดอันตรายต่อผู้กู้ภัยเองได้ ซึ่งในพื้นที่ที่ยากที่มนุษย์จะค้นหาหรือสำรวจผู้รอดชีวิตก่อนการกู้ภัยนั้น จึงต้องพึ่งอุปกรณ์บางอย่างเพื่อช่วยค้นหาซึ่งโดยทางผู้ดำเนินโครงงานนี้จึงได้คิดค้นหุ่นยนต์สำรวจกู้ภัยในที่ที่ยากต่อการสำรวจของมนุษย์เข้าถึงเพื่อเป็นประโยชน์ต่อการช่วยค้นหาสิ่งต่างๆต่อไป

หุ่นยนต์สำรวจนี้ทางผู้ดำเนินโครงงานได้สร้างขึ้น เพื่อศึกษาการทำงานของหุ่นยนต์ที่ทำงานด้วยไมโครคอนโทรลเลอร์ (Microcontroller) โดยมีการส่งสัญญาณความถี่ในรูปแบบของ Wireless โดยมีการควบคุมการทำงานผ่านอุปกรณ์คอมพิวเตอร์ เพื่อสำรวจค้นหาสิ่งต่างๆหรือสิ่งมีชีวิตที่ต้องการช่วยเหลือ หุ่นยนต์นี้ได้มีการทำงานร่วมกับคอมพิวเตอร์เพื่อประมวลผลทั้งภาพ เสียง และอุปกรณ์ควบคุมการทำงานต่างๆของหุ่นยนต์ โดยจะเป็นการเขียนโปรแกรมควบคุมการทำงานขึ้นมาเพื่อให้ง่ายต่อการใช้งาน และยังสามารถต่ออุปกรณ์เสริมต่างๆได้อาทิเช่น แท่นควบคุมทิศทางในรูปแบบต่างๆ ไมโครโฟน และระบบควบคุมอื่นๆได้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการส่งสัญญาณวิทยุความถี่ 2.4G (Wireless)
2. เพื่อศึกษาทำงานของไมโครคอนโทรลเลอร์ในตระกูลต่างๆ
3. เพื่อศึกษาการเขียนโปรแกรมตอบสนองกับการทำงานของไมโครคอนโทรลเลอร์
4. เพื่อสร้างหุ่นยนต์ต้นแบบ ที่สามารถนำไปประยุกต์ใช้งานในชีวิตจริงได้

1.3 ขอบข่ายของโครงงาน

1. ศึกษาการเชื่อมต่ออุปกรณ์ไมโครคอนโทรลเลอร์กับคอมพิวเตอร์
2. ศึกษาการส่งสัญญาณข้อมูลในความถี่วิทยุ 2.4G (Wireless)
3. ศึกษาการเขียนโปรแกรมควบคุมการทำงานไมโครคอนโทรลเลอร์
4. ออกแบบหุ่นยนต์ให้สามารถเคลื่อนที่ได้ในเกือบทุกสภาพพื้นผิว

1.4 ขั้นตอนการดำเนินการ

กิจกรรม	ปี2552							ปี2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ค้นคว้าหาข้อมูลการทำงานของอุปกรณ์ต่างๆ	←	→								
2.ศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์และโปรแกรมคอมพิวเตอร์		←	→							
3.ออกแบบและประกอบหุ่นยนต์			←	→						
4.ทดลองการส่งสัญญาณระหว่าง Wireless Tx,Rx ผ่านไมโครคอนโทรลเลอร์				←	→					
5.ทดสอบการสั่งงานไมโครคอนโทรลเลอร์ผ่านคอมพิวเตอร์				←	→					
6.ติดตั้งอุปกรณ์ทั้งหมดในตัวหุ่นยนต์						←	→			
7.ทดสอบการทำงานแล้วแก้ไขข้อผิดพลาด							← →			
8.บันทึกข้อมูลการทำงาน							← →			
9.สรุปผลการทดลองและจัดทำรูปเล่ม									← →	

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1. เข้าใจถึงหลักการทำงานของ MCU (ไมโครคอนโทรลเลอร์)
2. เข้าใจหลักการการส่งสัญญาณวิทยุในย่านความถี่ 2.4G (Wireless)
3. สามารถนำความรู้ในการเขียนโปรแกรมไปประยุกต์ใช้ในงานอื่นๆได้
4. สามารถนำข้อมูลที่ค้นคว้านี้ ไปพัฒนาหรือค้นคว้าต่อไปได้

1.6 งบประมาณของโครงงาน

1.	ค่าวัสดุอุปกรณ์ในการประกอบหุ่นยนต์	1,000	บาท
2.	หนังสือ Visual Basic 2008	500	บาท
3.	จัดทำรูปเล่มโครงงาน	300	บาท
4.	ค่าใช้จ่ายอื่นๆ	200	บาท
	รวมเป็นเงิน	2,000 บาท (สองพันบาทถ้วน)	

บทที่ 2
หลักการและทฤษฎี

2.1 Microcontroller

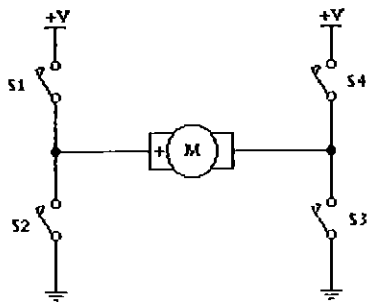
ไมโครคอนโทรลเลอร์เกิดจากคำ 2 คำ คือไมโคร(Micro) ซึ่งหมายถึงขนาดเล็ก และคำว่าคอนโทรลเลอร์ (controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม
ไมโครคอนโทรลเลอร์ จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์คือภายใน ไมโครคอนโทรลเลอร์ ได้รวมเอาซีพียู , หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวเดียวกัน

2.2 Wireless

Wireless คือระบบที่ใช้การเชื่อมต่อ Network แบบไร้สาย ด้วยเทคโนโลยีการสื่อสารภายใต้มาตรฐาน IEEE 802.11 ซึ่งสามารถสื่อสารเข้ากันได้กับอุปกรณ์ที่ต่างรุ่นต่างยี่ห้อ และหากอุปกรณ์นั้นผ่านเกณฑ์มาตรฐานเราจะเห็นได้ว่า มีการประทับตรา Wi-Fi Certified ซึ่งหมายความว่าอุปกรณ์ตัวนี้สามารถทำการเชื่อมต่อแบบไร้สายได้กับ อุปกรณ์อื่นๆ ที่มีตรา Wi-Fi Certified ได้เช่นกัน กลายมาเป็นคำศัพท์ของอุปกรณ์ LAN ไร้สายนั่นเอง

2.3 H-Bridge Switching

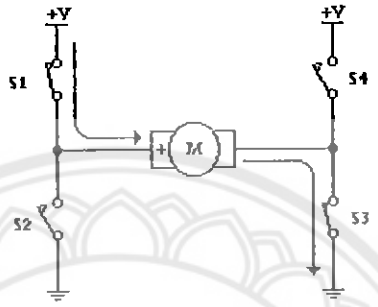
เป็นไอซีที่ใช้ควบคุมทิศทางการหมุนของ DC Motor หลักการของวงจรนั้น จะประกอบไปด้วย สวิทช์ 4 ตัว นั่นก็คือ S1 ,S2 ,S3 และ S4 ซึ่งในรูปตัวอย่าง 2.1 จะใช้ DC-Motor เป็น Load ของวงจร



รูปที่ 2.1 การทำงานวงจร H-Bridge Switching

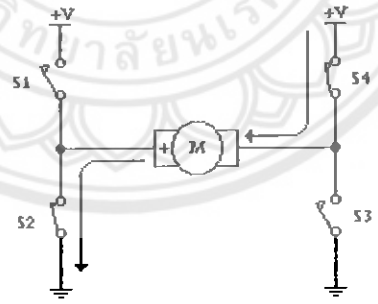
ในสภาวะเริ่มต้น สวิทช์ ทุกตัวจะ Off อยู่ จะทำให้ไม่มีการทำงานของมอเตอร์

เมื่อมีการ On สวิทช์ S1 และ S3 พร้อมกันจะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้า ไหลผ่านมอเตอร์ จากขั้วบวกของมอเตอร์ ไปยังขั้วลบของมอเตอร์ จึงทำให้มอเตอร์สามารหมุนได้ ในทิศทาง Forward (จะหมุนแบบตามเข็มนาฬิกา หรือทวนเข็มนาฬิกานั้น ขึ้นอยู่กับลักษณะของ การพันขดลวดภายในมอเตอร์) ดังรูปที่ 2.2



รูปที่ 2.2 การทำงานแบบ Forward

ในทางกลับกัน ถ้าทำการ On สวิทช์ S2 และ S4 พร้อมกัน ดังรูปที่ 2.3ก็จะเป็นการเชื่อมวงจร และทำให้เกิดกระแสไฟฟ้า ไหลผ่านมอเตอร์ จากขั้วลบของมอเตอร์ ไปยังขั้วบวกของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ และเป็นการหมุนในทิศทาง Reverse



รูปที่ 2.3 การทำงานแบบ Reverse

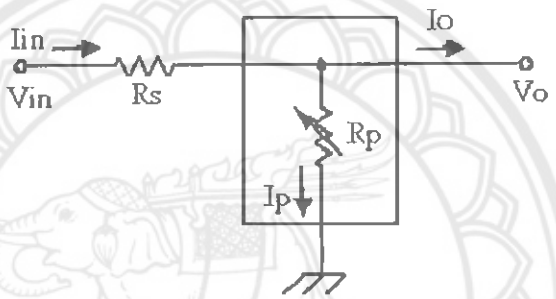
สรุป วงจรนี้จะอาศัยสวิทช์ 4 ตัว เพื่อบังคับทิศทางการไหล ของกระแสไฟฟ้า ที่ไหลผ่านมอเตอร์ เพื่อควบคุมให้มอเตอร์หมุนตามทิศทางที่ต้องการ โดยการผลัดกัน On และ Off สวิทช์พร้อมกัน 2 ตัว

2.4 ไอซีรักษาแรงดัน

วงจรรักษาแรงดันหรือที่เรียกว่าวงจรเร็กกูเลเตอร์(Regulator)เป็นวงจรที่นำมาใช้ลดระดับของแรงดันหรือรักษาแรงดันให้คงที่ จะแบ่งเป็น 2 ประเภท คือ

Shunt Regulator(แบบขนาน)

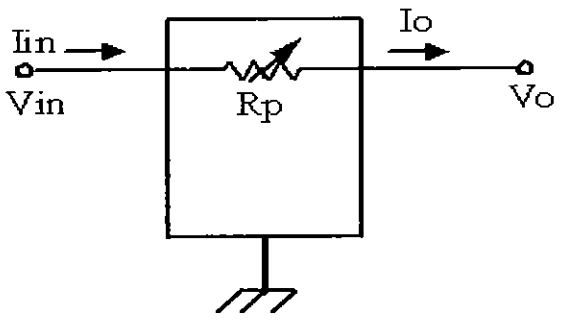
การทำงานของวงจรเร็กกูเลเตอร์แบบขนาน โดยมีแรงดันอินพุท V_{IN} จ่ายให้กับวงจร มีตัวต้านทาน R_S ทำหน้าที่ในการจำกัดกระแสที่จะไหลผ่านวงจรทั้งหมด ตัวต้านทานที่ปรับค่าได้ R_P จะทำการปรับค่าเองโดยอัตโนมัติเพื่อให้แรงดันที่เอาท์พุทคงที่ตลอด สมการของแรงดันเอาท์พุท $V_O = V_{IN} - R_S (I_O + I_P)$ โดยมีวิธีการทำงานของวงจรดังรูปที่ 2.4



รูปที่ 2.4 แสดงแผนผังการทำงานของเร็กกูเลเตอร์แบบขนาน

Series Regulator (แบบอนุกรม)

หลักการทำงานของเร็กกูเลเตอร์แบบอนุกรมนี้ มีการจ่ายแรงดันที่ยังไม่ได้มีการเร็กกูเลทไปยัง R_P โดย R_P จะปรับค่าความต้านทานของตัวเองได้อัตโนมัติ ทำให้เกิดแรงดันตกคร่อมที่ R_P ค่าหนึ่ง จะได้แรงดันเอาท์พุทเท่ากับ แรงดันอินพุทลบด้วยแรงดันตกคร่อมในตัวเร็กกูเลเตอร์ ซึ่งผลของการปรับค่า R_P ที่ถูกต้อง ก็จะทำให้ได้แรงดันเอาท์พุทตามที่ต้องการ และจากหลักการทำงานของเร็กกูเลเตอร์ชนิดนี้เองที่ได้นำมาประยุกต์ทำเป็นไอซี เร็กกูเลเตอร์เบอร์ต่างๆ ทั้งเบอร์ 78XX เบอร์ 79XX และอื่นๆ โดยมีวิธีการทำงานของวงจรดังรูปที่ 2.5



รูปที่ 2.5 แสดงแผนผังการทำงานของเร็กกูเลเตอร์แบบอนุกรม

Map basic circuit of Series Regulator

- Voltage Referent (วงจรแรงดันอ้างอิง) ซึ่งเป็นส่วนที่เป็นอิสระต่อทั้งอุณหภูมิและแรงดันที่จ่ายให้กับเร็กกูเลเตอร์ ดังรูปวงจรที่ 2.6 (แรงดันอ้างอิง)

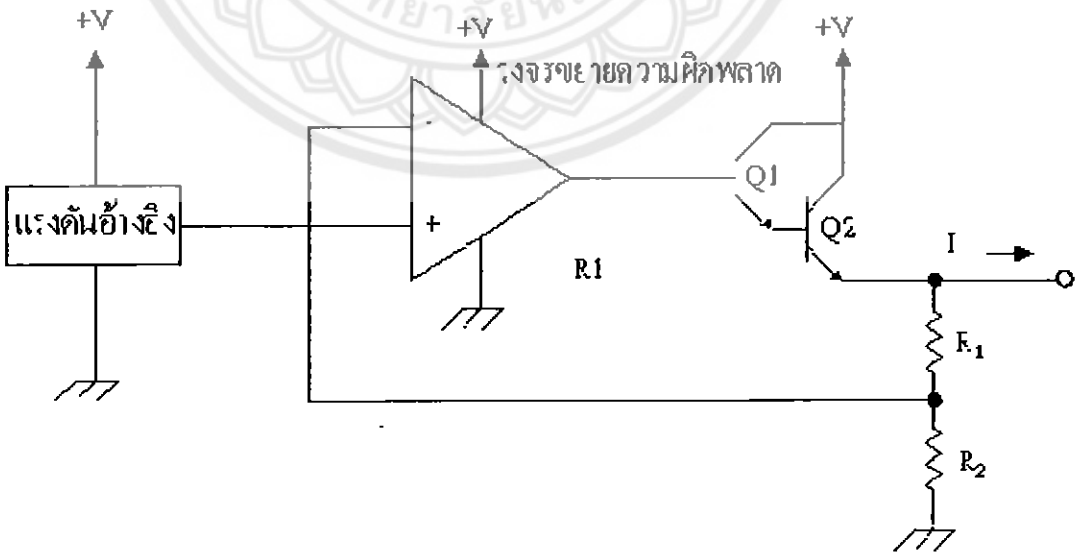
- Error Amplifier (วงจรขยายความผิดพลาด) ทำหน้าที่คอยเปรียบเทียบแรงดัน ระหว่างแรงดันอ้างอิงและสัดส่วนของแรงดันเอาท์พุท ที่ป้อนกลับมาที่ขาอินเวอร์ติ้งของออปแอมป์ ดังรูปวงจรที่ 2.6 (วงจรขยายความผิดพลาด)

- Series Transistor (ซีรีส์พาสทรานซิสเตอร์) ทำหน้าที่จ่ายกระแสเอาท์พุทให้เพียงพอกับความต้องการของโหลด ดังรูปวงจรที่ 2.6

เมื่อป้อนแรงดันอินพุทให้กับไอซีเร็กกูเลเตอร์ แรงดันเอาท์พุทจะถูกป้อนมายังอินพุทโดย R_1 และ R_2 ทำหน้าที่เป็นวงจรแบ่งแรงดัน ซึ่งแรงดันที่ตกคร่อม R_2 จะเป็นสัดส่วนกับแรงดันที่เอาท์พุท วงจรขยายความผิดพลาดจะทำหน้าที่รักษาสัดส่วนของแรงดันอ้างอิงกับแรงดันที่ตกคร่อม R_2 ให้เท่ากัน

ถ้าแรงดัน V_{R2} มากกว่า V_{REF} วงจรขยายความผิดพลาดจะลดระดับการขยายสัญญาณเอาท์พุท ทำให้ทรานซิสเตอร์จ่ายกระแสลดลงเป็นผลให้แรงดันเอาท์พุทที่จ่ายให้โหลดลดลง ด้วย

ถ้าแรงดัน V_{R2} น้อยกว่า V_{REF} วงจรขยายความผิดพลาดจะเพิ่มระดับการขยายสัญญาณเอาท์พุท ทำให้ทรานซิสเตอร์จ่ายกระแสเพิ่มขึ้น เป็นผลให้แรงดันเอาท์พุทที่จ่ายให้โหลดเพิ่มขึ้นด้วย

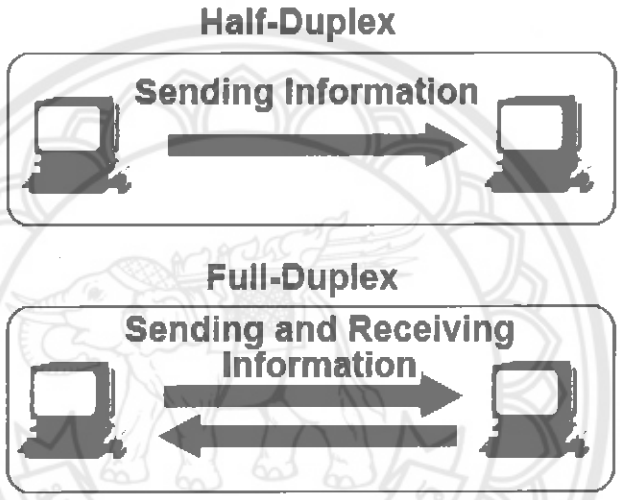


รูปที่ 2.6 แสดงแผนผังวงจรพื้นฐานของเร็กกูเลเตอร์แบบอนุกรม

2.5 พอร์ตสื่อสารอนุกรม (RS-232)

Recommended Standard-232 (RS-232) เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม (Serial Port) มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น

การส่งสัญญาณโดยใช้พอร์ตสื่อสารอนุกรมจะแบ่งออกเป็น 2 ลักษณะคือ Half-Duplex กับ Full-Duplex โดยจะมีลักษณะการรับส่งดังรูปที่ 2.7



รูปที่ 2.7 แสดงการทำงานแบบ Half-Duplex กับ Full-Duplex

Half-Duplex (การส่งสัญญาณกึ่งทางคู่)

การส่งสัญญาณแบบนี้เมื่อ ผู้ส่งได้ทำการส่งสัญญาณไปแล้ว ผู้รับก็จะรับสัญญาณนั้นหลังจากนั้นผู้รับก็สามารถปรับมาเป็นผู้ส่งสัญญาณ แทน ส่วนผู้ส่งเดิมก็ปรับมาเป็นผู้รับแทนสลับกันได้ แต่ไม่สามารถส่งสัญญาณพร้อมกันในเวลาเดียวกันได้ จึงเรียกการส่งสัญญาณแบบนี้ว่า ฮาร์ฟดูเพล็กซ์ (Half Duplex หรือ HD) ดังรูปที่ 2.7 (บน)

Full-Duplex (การส่งสัญญาณทางคู่)

การส่งสัญญาณแบบนี้ สามารถส่งข้อมูลได้พร้อมกันทั้งสองทางในเวลาเดียวกัน ดังรูปที่ 2.7 (ล่าง)

2.6 โปรแกรม Visual Basic (VB)

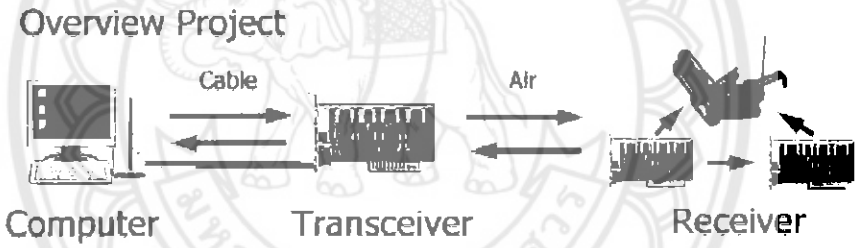
เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่ นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล (Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

บทที่ 3
การดำเนินการและการปฏิบัติการ

บทนี้จะอธิบายถึงการดำเนินงานในการดำเนินงานในโครงงานทั้งหมดโดยการนำหลักการและทฤษฎีจากบทที่ 2 มาปฏิบัติการสร้างและปรับปรุงให้เกิดผลงาน ที่ออกมาอย่างที่ตั้งเป้าเอาไว้

ก่อนอื่นจะเริ่มจากการอธิบายภาพรวมของระบบทั้งหมดก่อน โดยในโครงงานนี้จะเป็นการสร้างระบบติดต่อสื่อสารข้อมูลระหว่างอุปกรณ์ต่างๆกับคอมพิวเตอร์โดยผ่านระบบ Network ไร้สาย หรือที่เรียกกันว่า Wireless นั้นเองโดยจะแยกภาคส่วนการวิเคราะห์ข้อมูลออกเป็น 4 ภาค คือ

1. การติดต่อสื่อสารผ่าน RS-232
2. อุปกรณ์การรับส่งข้อมูลผ่าน Module Wireless
3. การติดต่อสื่อสารระหว่างอุปกรณ์ไมโครคอนโทรลเลอร์
4. การสั่งอุปกรณ์ต่างๆ ด้วยไมโครคอนโทรลเลอร์



รูปที่ 3.1 ภาพรวมของระบบทั้งหมดในโครงงาน

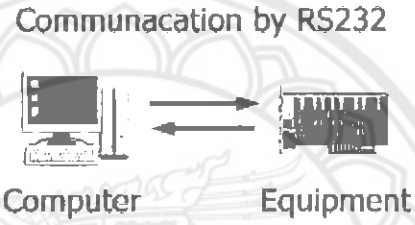
หลักการทำงานของระบบดังรูปที่ 3.1 นั้น เริ่มจากมีการสร้างข้อมูลในการสั่งการจากคอมพิวเตอร์ผ่านโปรแกรม Visual Basic 2005 โดยทำการสร้างข้อมูลมาเป็นอักขระตัวอักษร แล้วนำส่งข้อมูลผ่านพอร์ต Serial port (RS-232) โดยคอมพิวเตอร์จะแปลงข้อมูลให้อยู่ในระดับบิต เพื่อติดต่อสื่อสารข้อมูลกับอุปกรณ์ภายนอกโดยในที่นี้แล้วแต่การตั้งค่าในการส่งข้อมูล หลังจากที่คอมพิวเตอร์ได้ส่งข้อมูลออกมาทาง Serial port แล้ว ทางด้านฝั่งรับข้อมูลก็จะเชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ตระกูล AVR เพื่อเป็นการรับข้อมูลจากคอมพิวเตอร์มาส่งข้อมูลต่อไปยังอุปกรณ์ภายนอกอื่นและแสดงสถานะภาพทำงานไปในตัว

หลังจากที่คอมพิวเตอร์ได้ส่งข้อมูลไปยังอุปกรณ์ Transceiver (Tx) แล้ว ทางอุปกรณ์ (Tx) รับข้อมูลมาแปลงข้อมูลอักขระที่ได้มาเป็นข้อมูลเลขฐาน 16 เพื่อใช้ในการแสดงผลสถานะภาพการทำงานและส่งข้อมูลทั้งหมดไปยัง Receiver (Rx) เพื่อให้ทางอุปกรณ์ฝั่งรับ (Rx) นำข้อมูลทั้งหมดไปสั่งการอุปกรณ์ต่างๆที่ใช้ในการควบคุมอุปกรณ์

ในระบบนี้ได้มีการส่งข้อมูลเป็นแบบ Full-Duplex คือสามารถส่งข้อมูลไปยังตัวรับแล้วตัวรับยังสามารถส่งข้อมูลกลับมาที่ตัวส่งได้ สังเกตได้จากหน้าตาแสดงผลที่หน้าจอที่มีการแจ้งสถานะภาพเชื่อมต่อของอุปกรณ์ต่างๆได้

3.1 การติดต่อสื่อสารผ่านRS-232

ในส่วนของการดำเนินการในส่วนนี้จะเป็นการดำเนินการสร้างอุปกรณ์ทดลองการส่งข้อมูลจากคอมพิวเตอร์ไปยังไมโครคอนโทรลเลอร์เพื่อให้อุปกรณ์ LED ทำงานตามข้อมูลที่ส่งไปให้ดังรูปที่ 3.2



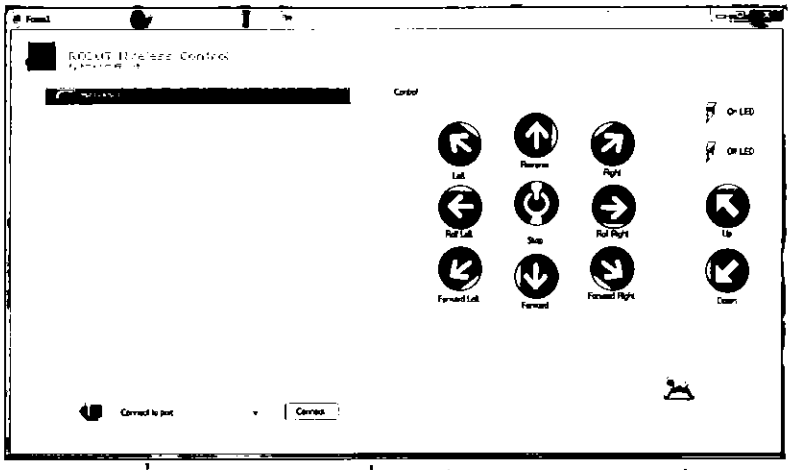
รูปที่ 3.2 ภาพรวมของการเชื่อมต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์

เริ่มจากการเขียนโปรแกรมควบคุมการส่งค่าต่างๆของข้อมูลผ่านโปรแกรม Visual basic 2005 โดยมีการเขียนเงื่อนไขดังนี้

สามารถเลือก Com port ในการเชื่อมต่อได้ เนื่องจากเมื่ออุปกรณ์เกิดการเชื่อมต่อ อาจจะเป็นการใช้อุปกรณ์แปลงสัญญาณจาก USB เป็น Serial port จึงทำให้มีการเชื่อมต่อผ่านช่องสัญญาณ USB ที่ไม่ซ้ำช่องกันทำให้ค่า Com port เปลี่ยนแปลง จึงจำเป็นต้องทำการเขียนโปรแกรมเพื่อรองรับปัญหาทางด้านนี้ ดังรูปที่ 3.3 ช่อง Connect to port

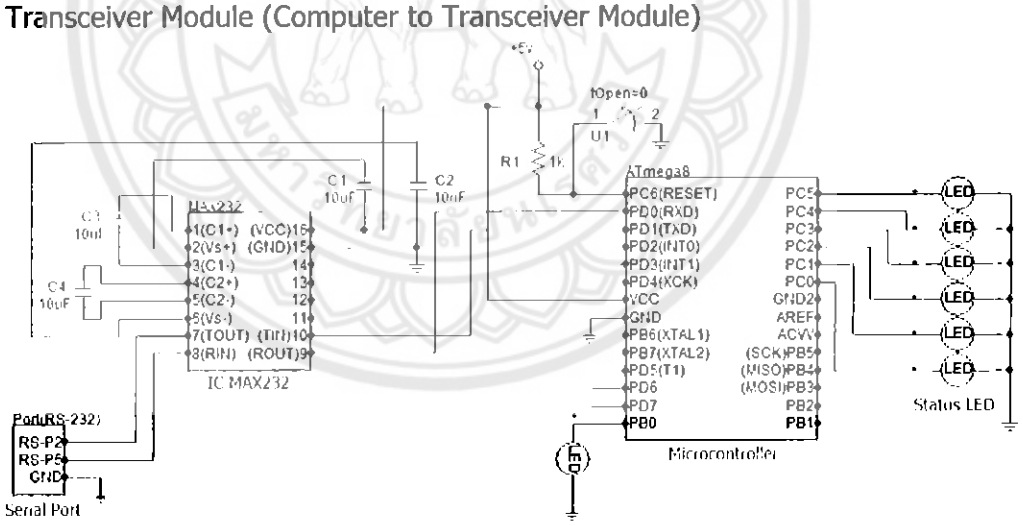
ปุ่มควบคุมทิศทางและควบคุมอุปกรณ์การทำงานของหุ่นยนต์ เพื่อที่จะส่งข้อมูลให้หุ่นยนต์สามารถเคลื่อนที่ไปตามทิศทางและเปิด/ปิดการใช้งานอุปกรณ์ของหุ่นยนต์ที่ต้องการได้ ดังรูปที่ 3.3 ช่อง Control

ช่องแสดงสถานะข้อมูลเป็นการแจ้งการทำงานของอุปกรณ์ที่ทางอุปกรณ์ที่เชื่อมต่อกับคอมพิวเตอร์ส่งมาให้ เพื่อเป็นการยืนยันข้อมูลการทำงาน และแจ้งสถานะภาพทำงาน ดังรูปที่ 3.3 ช่อง messages



รูปที่ 3.3 โปรแกรมการสั่งการข้อมูลจากคอมพิวเตอร์

ส่วนในส่วนของอุปกรณ์รับข้อมูลจากคอมพิวเตอร์หรือในโครงงานนี้เรียกว่า Transceiver Module (ชื่อนี้เรียกจากภาพรวมทั้งหมดของระบบ) จะมีการทำงานรับข้อมูลจากคอมพิวเตอร์เป็นข้อมูลอักขระตัวอักษรเพื่อมาแปลงเป็นเลขฐาน 16 เพื่อใช้ในการสั่งการแสดงผลและส่งข้อมูลต่างๆ ของอุปกรณ์ตามรูปภาพที่ 3.4



รูปที่ 3.4 วงจรการทำงาน Computer to transceiver module

จากรูปภาพที่ 3.4 จะแสดงถึงการเชื่อมต่อของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์เพื่อรับข้อมูลออกมาดำเนินการต่อไป โดยจะมีหลักการทำงานดังนี้

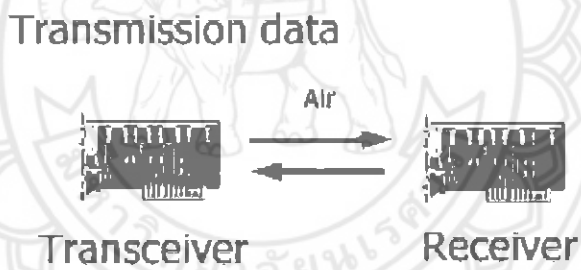
Computer ได้ส่งข้อมูลเข้ามายัง Serial port เป็นเลขฐานต่างๆตามที่ตั้งค่า แล้วนำข้อมูลที่ได้จากคอมพิวเตอร์เข้าไปประมวลผลใน IC เบอร์ Max232 (อุปกรณ์ชนิดนี้เป็นอุปกรณ์แปลงข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่างๆ) และนำค่าที่ได้ส่งข้อเข้าไปยังไมโครคอนโทรลเลอร์

(Microcontroller) เพื่อให้ไมโครคอนโทรลเลอร์ประมวลผลข้อมูล และนำไปแสดงผลที่หลอด LED เพื่อบอกสถานะภาพการส่งข้อมูลออกมาดังรูปที่ 3.4

ภายในอุปกรณ์ไมโครคอนโทรลเลอร์นั้นได้เขียนโปรแกรมเพื่อรับข้อมูลอักขระผ่านคำสั่ง UART หรือคำสั่งที่ใช้ในการเชื่อมต่อ RS-232 แล้วนำข้อมูลมาเช็คค่า เมื่อมีการเชื่อมต่อได้จะส่งข้อมูลกลับไปยังคอมพิวเตอร์เพื่อให้คอมพิวเตอร์รับรู้ว่าสามารถเชื่อมต่อกับอุปกรณ์นี้ได้แล้ว และนำค่าที่คอมพิวเตอร์ส่งมายังไมโครคอนโทรลเลอร์นั้นไปเช็คค่าแล้วเปลี่ยนแปลงค่าอักขระเป็นเลขฐาน 16 เพื่อนำข้อมูลนี้ไปแสดงผลและส่งข้อมูลต่อไป

3.2 อุปกรณ์การรับส่งข้อมูลผ่าน Module Wireless

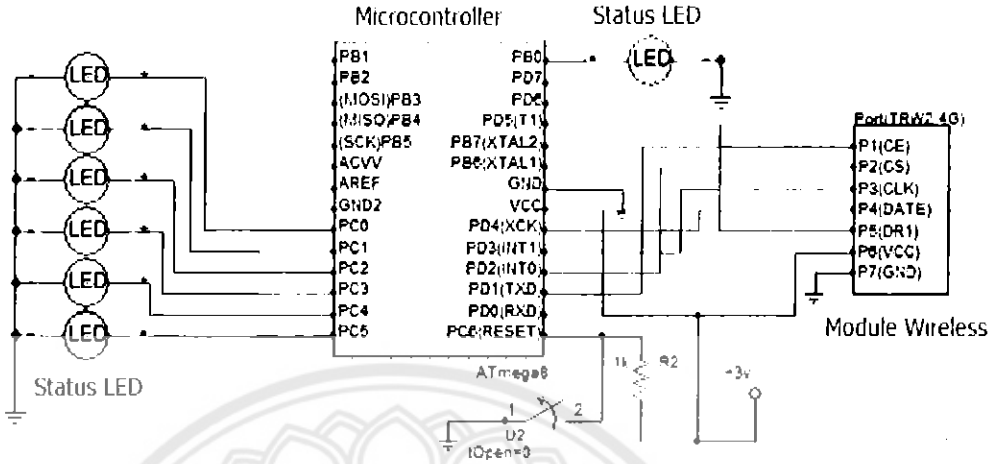
หัวใจหลักของการดำเนินการอยู่ที่ขั้นตอนนี้ ขั้นตอนนี้คือขั้นตอนการส่งข้อมูลที่ทางอุปกรณ์ส่ง (Transceiver Module) ที่ได้แปลงข้อมูลคำสั่งจากคอมพิวเตอร์มาเป็นข้อมูลเลขฐาน16 เพื่อติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ทางอุปกรณ์ด้านรับข้อมูล (Receiver Module) เพื่อให้อุปกรณ์ฝั่งรับไปประมวลผลสั่งการอุปกรณ์ต่อไป ดังรูปที่ 3.5



รูปที่ 3.5 ภาพรวมของระบบรับส่งข้อมูลผ่านทาง Wireless

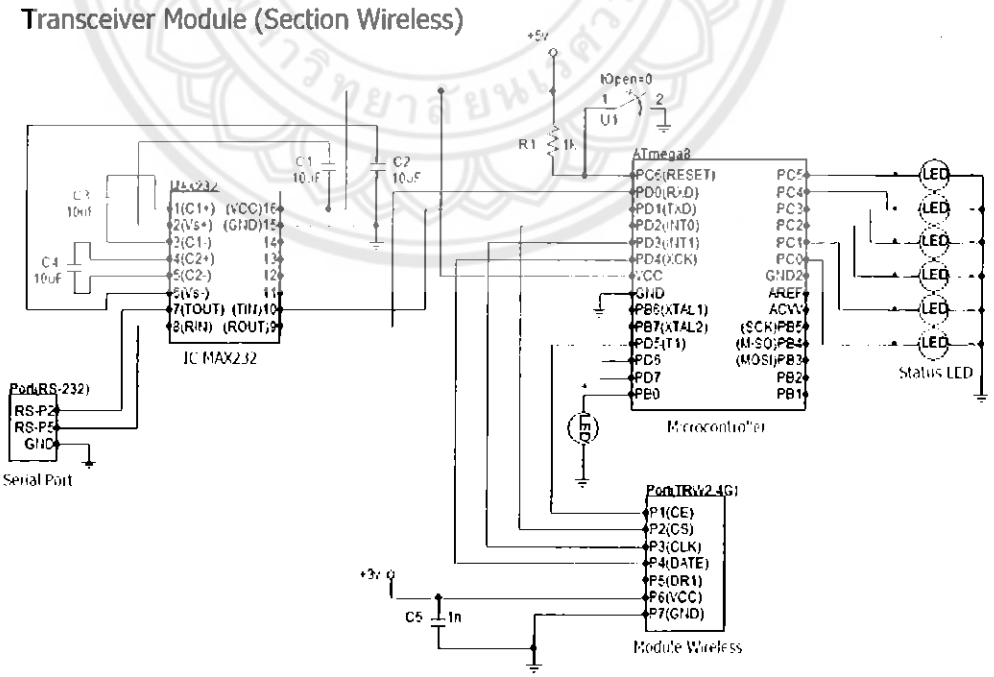
ในขั้นตอนการดำเนินการนี้เริ่มจากการทดสอบการทำงานของอุปกรณ์ Module Wireless TRW2.4G โดยการนำเอา Module Wireless ต่อเข้ากับไมโครคอนโทรลเลอร์ แล้วนำ LED ต่อเข้ากับขาแสดงผล(Output) ของไมโครคอนโทรลเลอร์ดังรูปที่ 3.6 เพื่อจะดูสถานะภาพการแสดงผลของอุปกรณ์ฝั่งรับ (Receiver Module) ว่าเมื่อมีการส่งข้อมูลมาจากทางฝั่งอุปกรณ์ส่ง (Transceiver Module) แล้วแสดงค่าเหมือนกันแสดงว่าอุปกรณ์ระบบนี้ใช้งานได้อย่างสมบูรณ์

Receiver Module (Section Wireless)



รูปที่ 3.6 วงจรการรับสัญญาณเพื่อแสดงผลของอุปกรณ์ฝั่งรับข้อมูล

เมื่อทำการต่ออุปกรณ์ฝั่งรับ (Receiver Module) เสร็จแล้ว ต้องนำอุปกรณ์ฝั่งส่ง (Transceiver Module) มาทำการเพิ่มอุปกรณ์ Module Wireless TRW2.4G ด้วย โดยมีการต่อเพิ่มดังรูปที่ 3.7

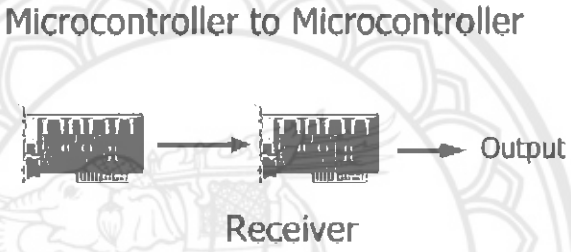


รูปที่ 3.7 วงจรการส่งข้อมูลทาง wireless ผ่านการควบคุมโดยคอมพิวเตอร์

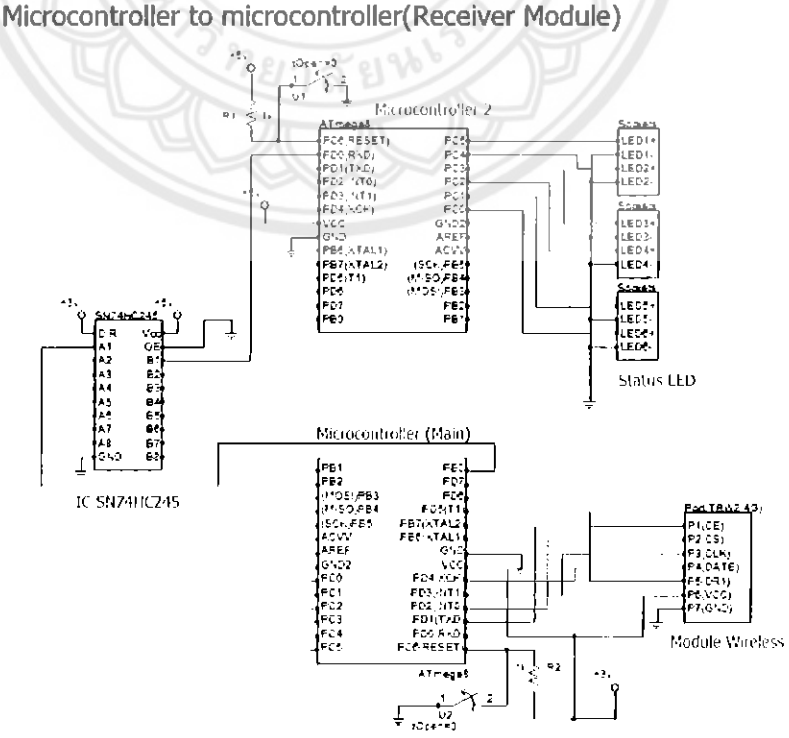
หลังจากที่มีการต่ออุปกรณ์ครบแล้วก็ได้เริ่มทำการทดสอบโดยมีการจ่ายกระแสไฟเข้าไปยังอุปกรณ์ทั้ง 2 และทดลองกดปุ่มบนโปรแกรมที่เขียนขึ้นมา เพื่อทดสอบการทำงานว่ามีการรับส่งข้อมูลอย่างถูกต้องหรือไม่ ถ้าถูกต้องก็จะนำอุปกรณ์ฝั่งรับข้อมูลไปดัดแปลงเพื่อใช้งานกับหุ่นยนต์ต่อไป

3.3 การติดต่อสื่อสารระหว่างอุปกรณ์ไมโครคอนโทรลเลอร์

ขั้นตอนนี้จะเป็นการทำให้อุปกรณ์ไมโครคอนโทรลเลอร์ 2 ตัว สามารถสื่อสารกันได้ โดยใช้หลักการ On/Off พอร์ตเพื่อเป็นการแจ้งสถานะว่าต้องการให้ไมโครคอนโทรลเลอร์นี้ทำงานในชุดคำสั่งที่กำหนดไว้ โดยมีหลักการทำงานภาพรวมดังรูปที่ 3.8 และมีการต่อวงจรดังรูปที่ 3.9



รูปที่ 3.8 ภาพรวมของระบบการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากัน 2 ตัว



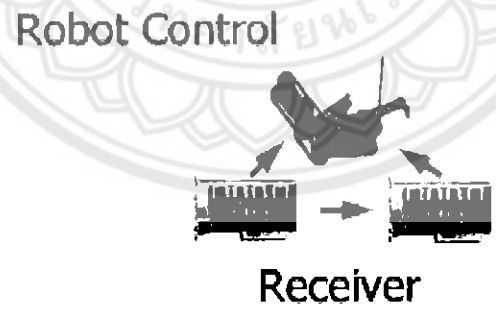
รูปที่ 3.9 วงจรการเชื่อมต่อไมโครคอนโทรลเลอร์ 2 ตัวลงในฝั่งอุปกรณ์รับ

จากวงจรรูปที่ 3.9 จะเห็นได้ว่าในการเชื่อมต่อไมโครคอนโทรลเลอร์นั้นมี IC SN74HC245 มาขั้นเพราะว่าการทำงานของไมโครคอนโทรลเลอร์แต่ละตัวทำงานในระดับแรงดันไฟฟ้าที่ต่างกัน คือไมโครคอนโทรลเลอร์ตัวหลัก (Microcontroller main) จะทำงานในระดับแรงดันไฟฟ้าที่ 3 โวลต์ และไมโครคอนโทรเลอร์ตัวสำรอง (Microcontroller 2) ทำงานในระดับแรงดันไฟฟ้าที่ 5 โวลต์ ทำให้ต้องพึ่ง IC SN74HC245 นี้ในการยกระดับแรงดันไฟฟ้าให้สูงขึ้นเพื่อจะทำให้ไมโครคอนโทรลเลอร์สามารถติดต่อสื่อสารกันได้อย่างสมบูรณ์

หลักการทำงานของวงจรรูปที่ 3.9 คือไมโครคอนโทรเลอร์ตัวหลัก (Microcontroller main) จะรับข้อมูลมาจาก Module wireless เพื่อรับข้อมูลมาเช็คเงื่อนไขเมื่อเงื่อนไขถูกต้อง พอร์ต PB0 จะทำการ On ทำให้มีการส่งสัญญาณไปที่ IC SN74HC245 เพื่อยกระดับแรงดันไฟฟ้า เพื่อนำสัญญาณที่ยกระดับแรงดันไฟฟ้าแล้วไปสื่อสารกับไมโครคอนโทรลเลอร์ตัวสำรอง (Microcontroller 2) เพื่อให้ประมวลผลตามเงื่อนไขที่ตั้งค่าไว้

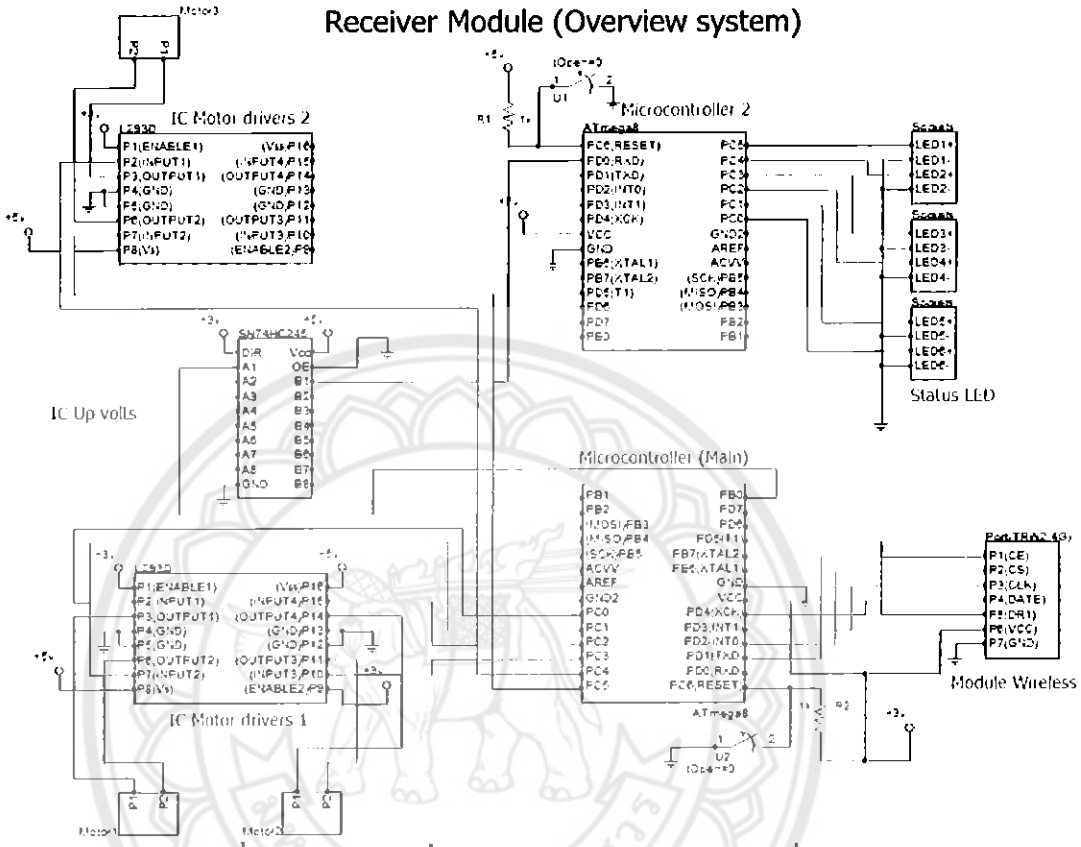
3.4 การสั่งอุปกรณ์ต่างๆด้วยไมโครคอนโทรลเลอร์

หลังจากที่มีการดำเนินการทำอุปกรณ์การรับส่งสัญญาณในระบบเรียบร้อยหมดแล้ว ต่อไปจะเป็นขั้นตอนการดำเนินการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์เพื่อเข้าไปสั่งการอุปกรณ์ มอเตอร์และLED เพื่อให้ทำการขับเคลื่อนอุปกรณ์ไปยังทิศทางต่างๆได้ โดยมีภาพรวมของอุปกรณ์ทั้งหมดดังรูปที่ 3.10



รูปที่ 3.10 ภาพรวมของระบบไมโครคอนโทรลเลอร์สั่งการอุปกรณ์

ในโครงงานนี้จะมีอุปกรณ์ที่ควบคุมโดยไมโครคอนโทรลเลอร์ในอุปกรณ์ฝั่งรับทั้งหมด 2 ชนิด คือ มอเตอร์ DC จำนวน 3 ตัว และ LED 6 ดวง ส่วนกล้องจะเป็นการส่งสัญญาณออกไปยังตัวรับโดยตรง โดยไม่ผ่านอุปกรณ์ไมโครคอนโทรลเลอร์ โดยจะมีการต่ออุปกรณ์ต่างๆ เข้ากับวงจรดังรูปที่ 3.11



รูปที่ 3.11 วงจรการเชื่อมต่ออุปกรณ์ต่างๆของอุปกรณ์ฝั่งรับ

จากรูปที่ 3.11 จะแสดงถึงการต่ออุปกรณ์มอเตอร์และหลอด LED เข้ากับไมโครคอนโทรลเลอร์ทั้ง 2 ตัว โดยการเชื่อมต่อของอุปกรณ์มอเตอร์กับไมโครคอนโทรลเลอร์จะไม่สามารถต่อเข้าโดยตรงได้ ต้องผ่าน IC Motor drivers (IC เบอร์ L293D) ในการช่วยขับมอเตอร์ เนื่องจากมอเตอร์ต้องการกำลังไฟฟ้าทำให้ต้องมีการเพิ่มแหล่งจ่ายไฟฟ้าเข้ามา โดยมีการต่อผ่าน IC Motor Drivers เพื่อเพิ่มกระแสและแรงดันไฟฟ้าทำให้มอเตอร์ทำงานได้อย่างปกติ และยังมีคุณสมบัติที่สำคัญอีกอย่างคือสามารถป้องกันกระแสย้อนกลับไปยังตัวไมโครคอนโทรลเลอร์ได้ด้วย ทำให้เวลามีการกลับขั้วของมอเตอร์อย่างกะทันหันจะไม่ส่งผลกระทบไปยังไมโครคอนโทรลเลอร์

จากการดำเนินการข้อ 3.3 ที่มีการสื่อสารกันระหว่างไมโครคอนโทรลเลอร์หลักกับไมโครคอนโทรลเลอร์ตัวที่ 2 ถ้ามีการส่งข้อมูลทำให้เงื่อนไขของไมโครคอนโทรลเลอร์ที่ 2 ก็จะส่งผลให้อุปกรณ์หลอด LED ติด เพื่อเพิ่มการส่องสว่างให้กับตัวกล้อง

การเขียนโปรแกรมเพื่อสั่งการหลังจากที่ไมโครคอนโทรลเลอร์ตัวหลักได้เขียนโปรแกรมรับข้อมูลที่ส่งมาผ่าน Module wireless แล้ว ทางอุปกรณ์ฝั่งรับ (Receiver Module) ก็ให้เขียนโปรแกรมส่งค่า Output ออกทาง Port C เพื่อเป็นการควบคุมมอเตอร์และสร้างเงื่อนไขการตรวจเช็คการเพิ่มรับค่าไปสื่อสารกับไมโครคอนโทรเลอร์ตัวที่ 2 ต่อไป

สรุปการดำเนินการและการปฏิบัติในการออกแบบสร้างอุปกรณ์และเขียนโปรแกรมเพื่อควบคุมการทำงานของหุ่นยนต์นั้นจำเป็นต้องวิเคราะห์ออกเป็นภาคส่วนเพื่อง่ายต่อการทดลองและการสร้างชิ้นงานของโครงงานขึ้นมา โดยมีการลำดับความคิดก่อนที่จะเริ่มประดิษฐ์โครงงานเพื่อง่ายต่อการจัดทำและง่ายต่อผู้พัฒนาต่อไป

บทที่ 4
ผลที่ได้จากการทดลอง

ในบทนี้จะเป็นการนำการดำเนินการและการปฏิบัติการจากบทที่ 3 ที่ได้สร้างขึ้นมาทดลอง หาผลการทดลองว่าทำงานได้จริงและตรงกับที่คาดหวังไว้หรือไม่

จากบทที่ 3 การดำเนินการและการปฏิบัติการจะแบ่งการทดลองเป็นภาคส่วนเหมือนบทที่ 3 เพื่อง่ายและสะดวกต่อการดำเนินการทดลอง

4.1 การติดต่อสื่อสารผ่าน RS-232

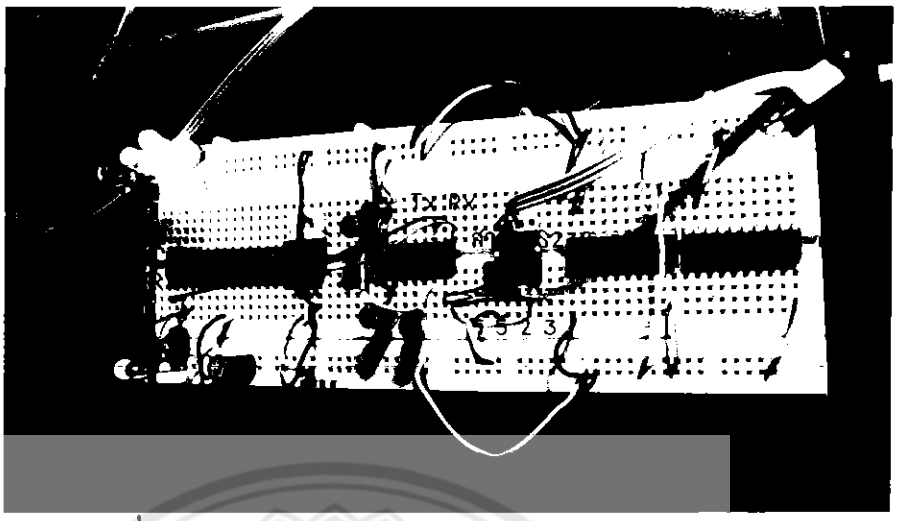
ในภาคส่วนนี้จะมีการดำเนินการหลักๆอยู่ 2 อย่าง คือ การรับส่งข้อมูลผ่าน Max232 การเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ภายนอก

4.1.1 ผลการทดลองการรับส่งข้อมูลผ่าน Max232

MAX232N มีคุณสมบัติทำการติดต่อสื่อสารผ่านพอร์ตอนุกรม (RS232) ของคอมพิวเตอร์ติดต่อสื่อสารกับอุปกรณ์ภายนอก โดยภายใต้การทำงานของแรงดันไฟฟ้า 5 โวลต์

การทดสอบการทำงานของการส่งข้อมูลของพอร์ตอนุกรม ผ่านโปรแกรม Visual Basic 2005 โดยทำการต่อวงจรทดสอบการรับส่งข้อมูลจากโปรแกรม Visual Basic ไปยังไมโครคอนโทรลเลอร์ โดยการนำสายสัญญาณ RS-232 พอร์ต 2 เชื่อมต่อเข้าพอร์ต 7 (Max 232) พอร์ต 5 เชื่อมต่อเข้าพอร์ต 8 (Max 232) และพอร์ต 9(Max 232) เชื่อมเข้าไมโครคอนโทรลเลอร์พอร์ต TxD พอร์ต 10 (Max 232) เชื่อมเข้ากับไมโครคอนโทรลเลอร์พอร์ต RxD ตามรูปที่ 4.1 - 4.2

รูปที่ 4.1 การต่อ Max 232N เพื่อเชื่อมต่อเข้ากับคอมพิวเตอร์



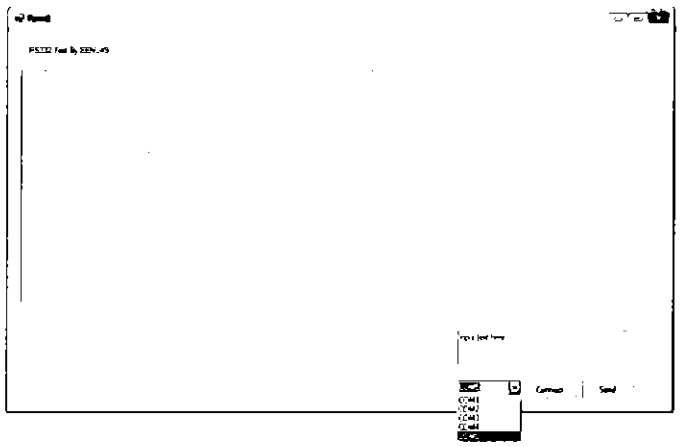
รูปที่ 4.2 วงจร Max 232N ต่อเข้าไมโครคอนโทรลเลอร์

เขียนโปรแกรม Visual basic เพื่อทดสอบการทำงานโดยมีเงื่อนไขดังนี้

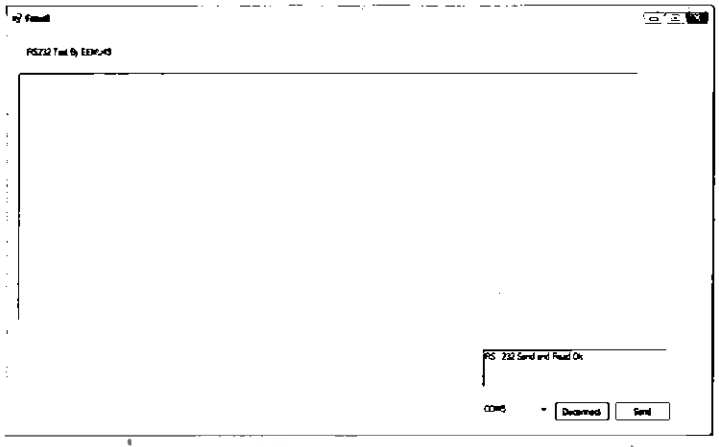
- ปุ่มเชื่อมต่อ Com port เพื่อกำหนดช่องสัญญาณการเชื่อมต่อกับอุปกรณ์ภายนอก
- ช่องการส่งข้อมูล เพื่อจะส่งข้อมูลเข้าไปยังอุปกรณ์ภายนอก
- ช่องแสดงผลข้อมูล เพื่อที่จะดูผลตอบสนองของไมโครคอนโทรลเลอร์ว่าตอบกลับมาถูกต้องหรือไม่

เขียนโปรแกรมในไมโครคอนโทรลเลอร์ เพื่อทดสอบการเชื่อมต่อและการส่งข้อมูลกับไปยังคอมพิวเตอร์ โดยมีเงื่อนไขดังนี้

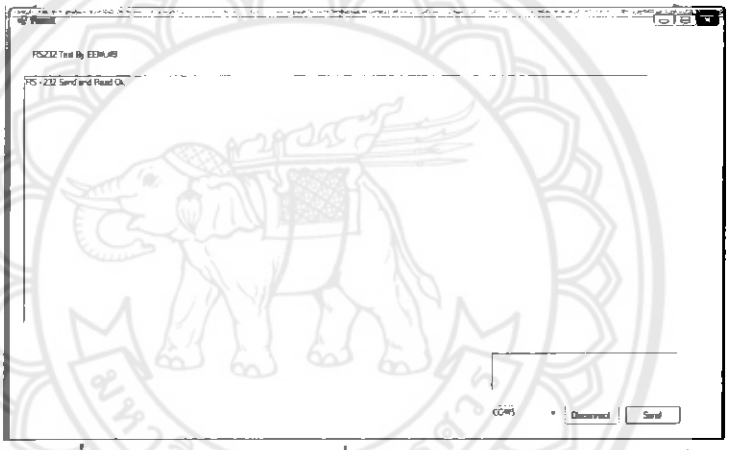
- เขียนโปรแกรมเปิดคำสั่งการทำงานเชื่อมต่อ Serial port ผ่านคำสั่ง UART ของ AVR
- เขียนโปรแกรมส่งค่าแสดงสถานะเมื่อมีการเชื่อมต่อ แต่ถ้ายังไม่เชื่อมต่อให้ Null time เอาไว้จะได้ผลการทดลองดังรูปที่ 4.3 - 4.5



รูปที่ 4.3 โปรแกรมส่งค่าและอ่านค่า RS-232



รูปที่ 4.4 ส่งค่าตัวหนังสือออกไปจากคอมพิวเตอร์



รูปที่ 4.5 อ่านค่าตัวหนังสือที่เคยส่งออกไปจากคอมพิวเตอร์

จากผลการทดลองจะทำให้เห็นว่าคอมพิวเตอร์กับอุปกรณ์ภายนอกได้มีการเชื่อมต่อกันแล้ว และมีการส่งสัญญาณได้อย่างถูกต้องไม่มีปัญหาใดๆ

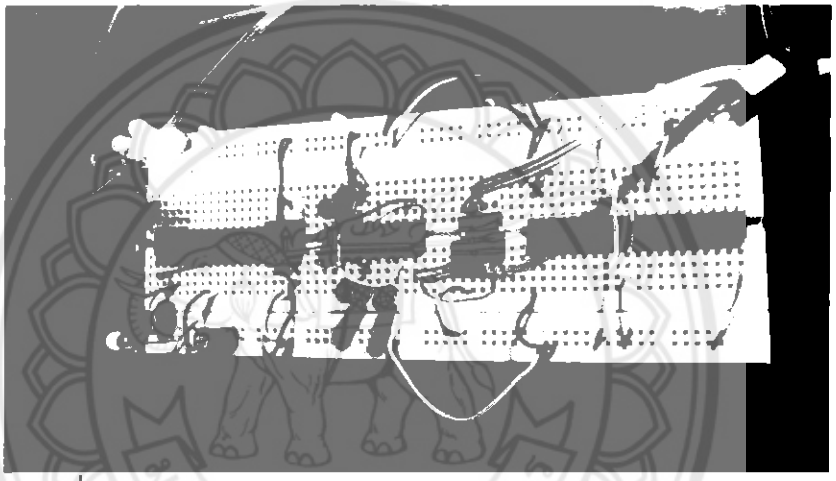
ข้อเสนอแนะถ้าเกิดตัวหนังสือขึ้นเป็น ???? แสดงว่าการต่อ Max 232 กับไมโครคอนโทรลเลอร์นั้น มีแรงดันไฟฟ้าที่ไม่เท่ากันอาทิเช่น Max232 จ่ายแรงดันไฟฟ้าที่ 5 โวลต์ ส่วนไมโครคอนโทรลเลอร์จ่ายแรงดันไฟฟ้าที่ 3 โวลต์ ทำให้ไม่สามารถอ่านค่าได้อย่างถูกต้อง

4.1.2 การเขียนโปรแกรมคอมพิวเตอร์เพื่อควบคุมอุปกรณ์ภายนอก

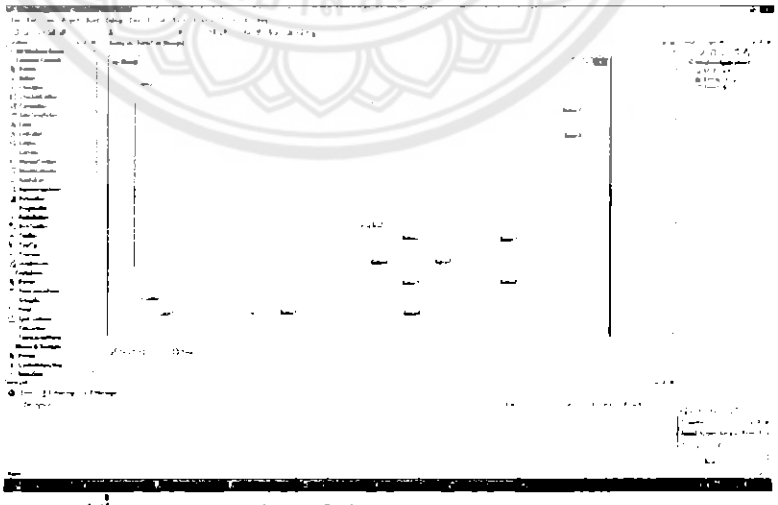
หลังจากได้ทดลองการส่งข้อมูลเพื่อดูการตอบสนองของไมโครคอนโทรลเลอร์แล้ว เราก็จะมาเริ่มสร้างอุปกรณ์ควบคุมการทำงานของตัวอุปกรณ์ส่งข้อมูล (Transceiver Module) โดยการเขียนโปรแกรม Visual basic เพื่อควบคุมการทำงานของอุปกรณ์ส่งข้อมูลนี้

โปรแกรม Visual Basic 2005 มีคุณสมบัติที่สามารถส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 ได้จึงเหมาะและสะดวกต่อการพัฒนาโปรแกรมควบคุมอุปกรณ์ภายนอกด้วยคอมพิวเตอร์อย่างมาก

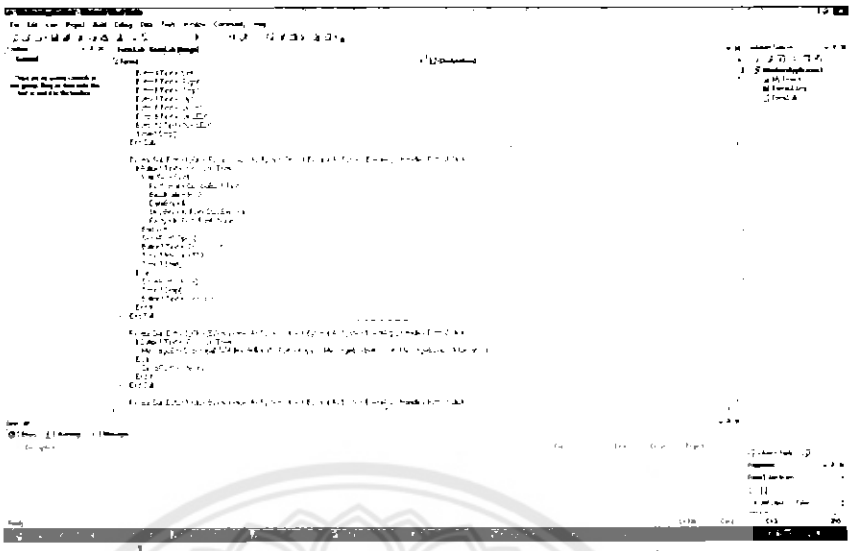
ในการทดลองนี้ ต้องการทดสอบการสั่งการไมโครคอนโทรลเลอร์ผ่านทางโปรแกรม Visual Basic 2005 (VB) โดยผ่าน IC MAX232N หรือ IC MAX3232 เพื่อต้องการให้มีการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ และอ่านค่าที่ส่งกลับมาเพื่อแสดงผลผ่านหน้าจอ และยังสามารถส่งข้อมูลออกไปที่ไมโครคอนโทรลเลอร์เป็น bit เพื่อที่จะให้ไมโครคอนโทรลเลอร์สั่งการ TRW2.4G เพื่อสั่งการการทำงานกับหุ่นยนต์ค้นหาต่อไป เพื่อตรวจสอบดูการทำงานว่าจะใช้งานได้หรือไม่ โดยมีการต่อวงจรดังรูปที่ 4.6 และเขียนโปรแกรมควบคุมตามเงื่อนไขที่ตั้งไว้ตามรูป 4.7 - 4.9



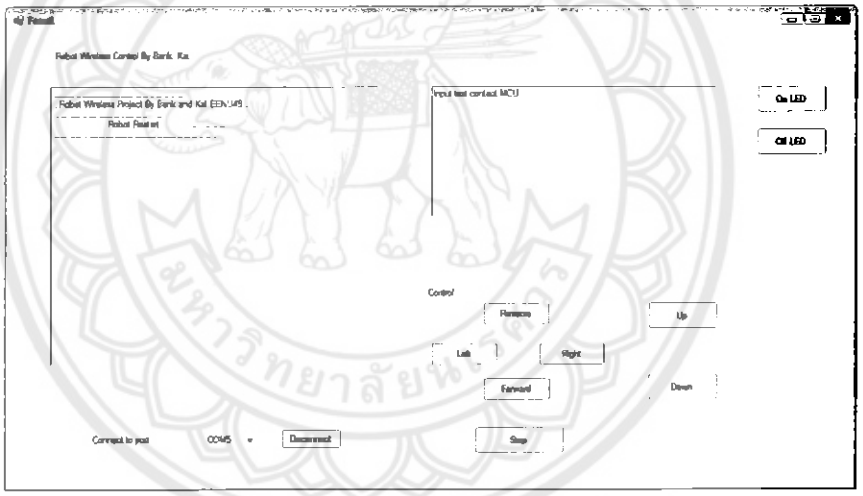
รูปที่ 4.6 ต่อไมโครคอนโทรเลอร์ และ Max232N เข้ากับคอมพิวเตอร์



รูปที่ 4.7 พัฒนารูปแบบโปรแกรมควบคุมการทำงาน Robot



รูปที่ 4.8 เขียนโปรแกรมควบคุมการทำงานอุปกรณ์ภายนอก



รูปที่ 4.9 ผลการทำงานของโปรแกรม

ผลการทดลอง

จากการทดลองโปรแกรมสามารถติดต่อกับไมโครคอนโทรลเลอร์ได้ แต่มีการติดปัญหาทางด้านอุปกรณ์ในการจ่ายไฟ 3 โวลต์และ 5 โวลต์ ต้องมีการสั่งเปลี่ยนจาก Max232N เป็น Max3232 เพื่อทำให้สามารถคุยกับไมโครคอนโทรลเลอร์เป็นภาษาปกติได้ โดยผลการทดลองจะเป็นตามตารางที่ 4.1 - 4.2

ตารางที่ 4.1 ตารางแสดงผลการทดสอบการเชื่อมต่อโปรแกรม VB เข้ากลับไมโครคอนโทรเลอร์ผ่าน Max เบอร์ต่าง เงื่อนไขการจ่ายไฟเลี้ยง 3 โวลต์

ส่งข้อมูล Rs[3]	ffbbllllrrud	ddddddddd	ioddbbll
Max232N	??????????	??????????	????????
Max3232	ffbbllllrrud	ddddddddd	ioddbbll

ตารางที่ 4.2 ตารางแสดงผลการทดสอบการเชื่อมต่อโปรแกรม VB เข้ากลับไมโครคอนโทรเลอร์ผ่าน Max เบอร์ต่าง เงื่อนไขการจ่ายไฟเลี้ยง 5 โวลต์

ส่งข้อมูล Rs[3]	ffbbllllrrud	ddddddddd	ioddbbll
Max232N	ffbbllllrrud	ddddddddd	ioddbbll
Max3232	??????????	??????????	??????????

และในการส่งข้อมูลจากคอมพิวเตอร์ลงไปยังไมโครคอนโทรลเลอร์นั้นจะส่งเป็นข้อมูลแบบ Char array เพื่อทำให้การส่งข้อมูลไหลต่อเนื่อง ในการทดลองได้ตั้งค่าเป็น Char Rs[3]; ให้รับ Bit ได้ 4 ตัว ได้ผลการทดลองดังตาราง 4.3

ตารางที่ 4.3 ตารางแสดงผลการทดสอบการส่งค่าไปที่หุ่นยนต์เพื่อดูผลการทำงาน

สถานะการทำงาน	Rs[3] = 'f'	Rs[3] = 'r'	Rs[3] = 'u'	Rs[3] = 'i'
Motor ตัวที่ 1	On	Off	Off	Off
Motor ตัวที่ 2	On	On	Off	Off
Motor ตัวที่ 3	Off	Off	On	Off
LED	Off	Off	Off	On

สรุปการทดลองนี้แสดงถึงการเขียนโปรแกรมควบคุมอุปกรณ์ภายนอกใช้งานได้ดี แต่มีปัญหาทางด้านการจ่ายไฟในการติดต่อกันระหว่าง Max กับตัวไมโครคอนโทรลเลอร์ อยู่บ้างทำให้อาจจะต้องมีการแก้ไขวงจรหรือสั่งอุปกรณ์ใหม่เพื่อจะได้ใช้งานได้อย่างถูกต้อง และรวดเร็ว

4.2 อุปกรณ์การรับส่งข้อมูลผ่าน Module Wireless

การทดลองนี้เป็นหัวใจสำคัญของโครงงาน เนื่องจากการทดลองนี้จะเป็นการทดลองรับส่งข้อมูลจากอุปกรณ์ส่งข้อมูล (Transceiver Module) ไปยังอุปกรณ์รับข้อมูล (Receiver Module) เพื่อให้อุปกรณ์ฝั่งรับไปประมวลผลต่อไป

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

TRW2.4G (Module Wireless) มีคุณสมบัติในการติดต่อระหว่างตัวส่งสัญญาณ และตัวรับสัญญาณในรูปแบบของ Wireless โดยมีการเข้ารหัสในการติดต่อกันระหว่างตัวรับและตัวส่งเพื่อจะได้รู้ถึงตัวที่ต้องการจะติดต่อในย่านความถี่วิทยุ 2.4 GHz

ในการทดลองนี้ จะทดสอบด้วยการส่งสัญญาณให้หลอด LED ทั้ง 2 ด้านทำงานเหมือนกัน โดยมีการตั้งค่าให้กระพริบสลับกันไปมา โดยมีการหน่วงเวลาครั้งละ 1 วินาที ถ้าโมดูลทั้ง 2 อันใช้งานได้จริงหลอดไฟ LED จะต้องทำงานเหมือนกันทั้ง 2 ฝั่ง (Transceiver และReceiver) และกำหนดให้ LED ที่ PortD0 เป็นการแสดงว่าโมดูลมีทำงาน โดยมีการต่อวงจรดังรูปที่ 4.10 – 4.11

รูปที่ 4.10 Module Wireless TRW 2.4G ฝั่งตัวส่ง (Transceiver)

รูปที่ 4.11 Module Wireless TRW 2.4G ฝั่งตัวรับ (Receiver)

1674729X
ปร.
ก674ส
2552

ผลการทดลอง

หลอด LED ทั้ง 2 ฝั่งอุปกรณ์ ติดและกระพริบเหมือนกัน ตามที่ต้องการ และมีการเข้ารหัสให้ติดต่อกันระหว่างโมดูลได้อย่างถูกต้อง แต่มีการรบกวนของสัญญาณรบกวนอยู่มากจึงทำให้ไม่สามารถส่งได้ไกลเท่าที่ควร เนื่องจากการเชื่อมต่อสายสัญญาณกับโมดูลไม่ได้ทำการเชื่อมผ่านตัว Converter ทำให้เกิดสัญญาณรบกวนจากการส่งสัญญาณเป็นจำนวนมากแต่ยังสามารถทำงานได้ไม่ผิดพลาดดังตารางที่ 4.4 – 4.5

ตารางที่ 4.4 ตารางแสดงผลการทดสอบของ TRW 2.4 G โดยการส่ง Data = 0x2A

TRW2.4G	LED PC0	LED PC1	LED PC2	LED PC3	LED PC4	LED PC5
Tx	On	Off	On	Off	On	Off
Rx	On	Off	On	Off	On	Off

ตารางที่ 4.5 ตารางแสดงผลการทดสอบของ TRW 2.4 G โดยการส่ง Data = 0x15

TRW2.4G	LED PC0	LED PC1	LED PC2	LED PC3	LED PC4	LED PC5
Tx	Off	On	Off	On	Off	On
Rx	Off	On	Off	On	Off	On

สรุปผลทดลองครั้งนี้ มีการส่งสัญญาณได้อยากถูกต้อง แต่ยังเกิดสัญญาณรบกวนในการส่งเป็นจำนวนมาก แต่ถึงกระนั้นก็ยังสามารถทำงานได้ โดยไม่ผิดพลาด แต่อาจจะมีหน่วงเวลาบ้างในการส่งบางครั้ง เพราะมีสัญญาณรบกวนเป็นจำนวนมาก

ข้อเสนอแนะในการทำงานเกี่ยวกับระบบสัญญาณต้องมีการเชื่อมต่อระหว่างขาดีๆ ให้สนิทแล้วห้ามเปลี่ยสาย เพราะจะทำให้เกิดสัญญาณรบกวนได้ และการทำงานพวกนี้ควรมีการชิลอุปกรณ์เพื่อลดปัญหาสัญญาณรบกวนลงกราวด์ จะทำให้สัญญาณออกมาดีที่สุด

4.3 การติดต่อสื่อสารระหว่างอุปกรณ์ไมโครคอนโทรลเลอร์

การทดลองนี้เป็นการทดสอบการสื่อสารระหว่างไมโครคอนโทรลเลอร์ 2 ตัวเข้าด้วยกัน ทำให้เกิดช่องทางการสั่งการมากขึ้น โดยการทำงานของไมโครคอนโทรลเลอร์นี้มีแรงดันไฟฟ้าที่จ่ายเข้าตัวอุปกรณ์ต่างกันคือ 3 โวลต์ กับ 5 โวลต์ จึงจำเป็นต้องมี IC ยกระดับแรงดันเข้ามาช่วย

IC เบอร์ SN74HC245N มีคุณสมบัติในการยกแรงดันจาก 3 โวลต์ เป็น 5 โวลต์ เนื่องจากในการทดลองนี้จำเป็นต้องให้ไมโครคอนโทรลเลอร์ 2 ตัว ติดต่อการ เพราะว่า ATmega8 มี Port ใน

การให้ใช้งานจำนวนน้อยแล้ว และต้องการที่จะให้หลอด LED ส่องสว่างมากจึงต้องใช้ไมโครคอนโทรลเลอร์ 5 V อีกตัวมาใช้งาน

ในการทดลองนี้ต้องการให้ ไมโครคอนโทรลเลอร์ตัวที่ 1 (ทำงานที่ไฟ 3 โวลต์) ส่งสัญญาณที่ พอร์ต B0 เมื่อ PB0 ทำงานให้ไมโครคอนโทรลเลอร์ตัวที่ 2 (ทำงานที่ไฟ 5 โวลต์) สั่งการให้ LED ที่ พอร์ต C ทั้งหมดทำงาน ถ้าส่งสัญญาณ 3โวลต์ เข้าในไมโครคอนโทรลเลอร์ 5 โวลต์ โดยตรง ไมโครคอนโทรลเลอร์ จะไม่สามารถรับค่า Input ได้ เนื่องจากไมโครคอนโทรลเลอร์ตัวที่ 2 รับแรงดันไฟฟ้าต่ำสุดได้เพียง 4.7 โวลต์ ทำให้ไม่สามารถเข้าใจในการสั่งการได้ ถ้าไมโครคอนโทรลเลอร์ตัวที่ 1 ให้ พอร์ต B0 ทำงาน และส่งสัญญาณผ่าน SN74HC245N ไมโครคอนโทรลเลอร์ตัวที่ 2จะทำงานได้หรือไม่ตามผลการทดลองรูปที่ 4.12

รูปที่ 4.12 วงจรการต่อไมโครคอนโทรลเลอร์ 2 ตัวเข้าหากันผ่าน SN74HC245N

ผลการทดลอง

จากการส่งสัญญาณผ่าน SN74HC245N เพื่อเป็นการยกระดับของแรงดันจาก 3 โวลต์ เป็น 5 โวลต์นั้น ทำให้ไมโครคอนโทรลเลอร์ทั้ง 2 ตัว ติดต่อสื่อสารกันได้ โดยที่ ไมโครคอนโทรลเลอร์ตัวที่ 1 สั่งให้ พอร์ต B0 ทำงาน ทำให้ส่งผลให้ไมโครคอนโทรลเลอร์ตัวที่ 2 สั่งการให้ พอร์ต C 0-5 ทำงาน ได้ตามที่ต้องการในระดับแรงดัน 5V และได้ค่าต่างๆในการทดสอบดังตารางที่ 4.6 - 4.7

ตารางที่ 4.6 ตารางแสดงผลการทดสอบการทำงาน MCU (ไมโครคอนโทรลเลอร์) โดยไม่ผ่าน IC SN74HC245N

MCU	PortB0	PortC0	PortC1	PortC2	PortC3	PortC4	PortC5
ตัวที่ 1	On	Off	Off	Off	Off	Off	Off
ตัวที่ 1	Off	Off	Off	Off	Off	Off	Off

ตารางที่ 4.7 ตารางแสดงผลการทดสอบการทำงาน MCU (ไมโครคอนโทรลเลอร์) โดยผ่าน IC SN74HC245N

MCU	PortB0	PortC0	PortC1	PortC2	PortC3	PortC4	PortC5
ตัวที่ 1	On	Off	Off	Off	Off	Off	Off
ตัวที่ 1	Off	On	On	On	On	On	On

สรุปผลการทดลองครั้งนี้ ทำให้ไมโครคอนโทรลเลอร์ 2 ตัว ที่มีระดับแรงดันต่างกันติดต่อกันได้ เป็นอย่างที่ต้องการ

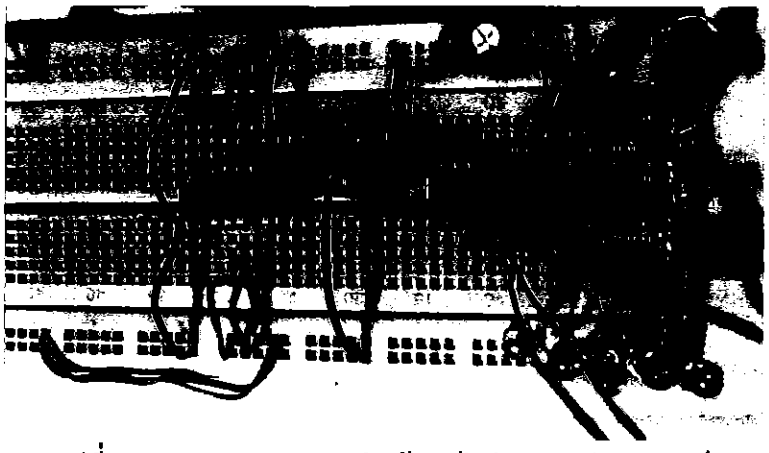
4.4 การสั่งอุปกรณ์ต่างๆด้วยไมโครคอนโทรลเลอร์

หลังจากที่ได้ทดลองขั้นตอนการส่งสัญญาณและการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์แล้ว ก็จะเป็นการทดลองอุปกรณ์ฝั่งรับข้อมูล (Receiver Module) เพื่อนำข้อมูลที่ส่งมาประมวลผลแล้วสั่งการการทำงานอีกที

ในส่วนนี้ก็จะเป็นการทดลองการทำงานของ IC Motor driver เพื่อทดสอบการทำงานของ IC ร่วมกับไมโครคอนโทรลเลอร์และอุปกรณ์ต่างๆว่าสามารถใช้งานได้หรือไม่

IC เบอร์ L293D มีคุณสมบัติในการขับเคลื่อนมอเตอร์ DC จากแหล่งจ่ายต่างกัน โดยมีพารามิเตอร์ต่างๆคือ Enable คือไฟเลี้ยง IC หลัก (แหล่งจ่ายของ Input), Vs กับ Vss คือไฟเลี้ยง IC เสริม (แหล่งจ่ายที่มาช่วยขับ Output), Input คือ สัญญาณที่ส่งมาจากไมโครคอนโทรลเลอร์, Output คือ สัญญาณที่ส่งออกไปควบคุมมอเตอร์

ในการทดลองได้ทดสอบการทำงานการขับเคลื่อนของมอเตอร์ ไปข้างหน้า 3 วินาที และถอยหลัง 3 วินาที โดยมีการทำ LED แสดงการทำงานของ พอร์ต C ที่ใช้ในการส่งข้อมูลไปยัง L293D และเปรียบเทียบกับการทำงานของมอเตอร์ว่าตรงกับที่ LED แสดงหรือไม่ ดังรูปที่ 4.41

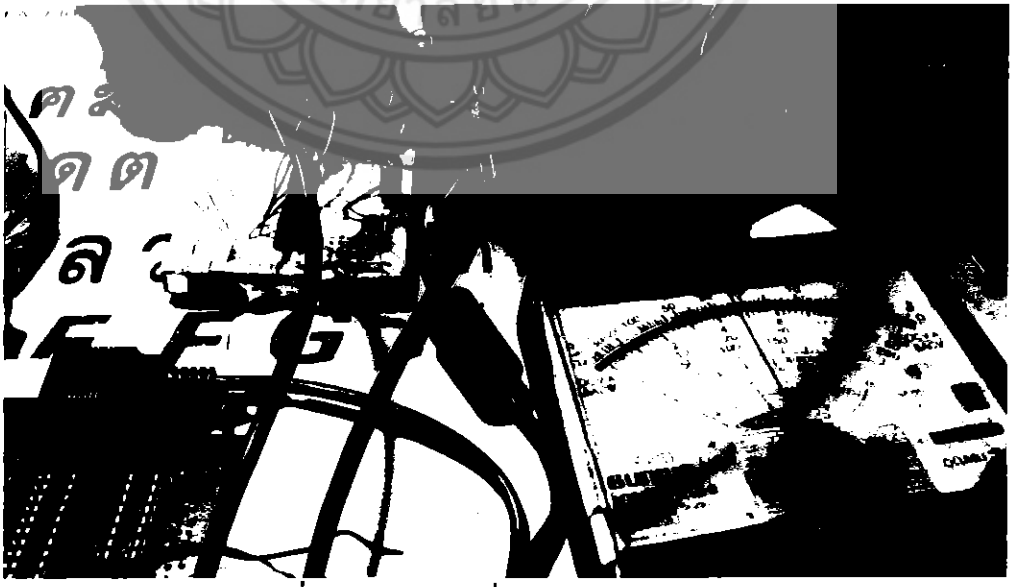


รูปที่ 4.13 วงจร IC L293D ต่อเข้ากับไมโครคอนโทรลเลอร์

ผลการทดลอง

มอเตอร์กับ LED ทำงานตรงกับมอเตอร์ โดยที่แรงดันที่ขา LED (Input) มีแรงดันไฟฟ้า 3 โวลต์ และที่ขามอเตอร์ (Output) มีแรงดันไฟฟ้า 5 โวลต์ ตามที่ต้องการ ได้ผลดังรูปที่ 4.14 - 4.15 และตารางที่ 4.8 - 4.12

รูปที่ 4.14 วัดแรงดันที่ขา Input ของ L293D



รูปที่ 4.15 วัดแรงดันที่ขา Output ของ L293D

ตารางที่ 4.8 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1010

Port MCU	PortC0	PortC1	PortC2	PortC3	สถานะ
LED	3V	0V	3V	0V	1010
Motor ตัวที่ 1	เดินหน้า 5 V				
Motor ตัวที่ 2	เดินหน้า 5 V				

ตารางที่ 4.9 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 0101

Port MCU	PortC0	PortC1	PortC2	PortC3	สถานะ
LED	0V	3V	0V	3V	0101
Motor ตัวที่ 1	ถอยหลัง 5 V				
Motor ตัวที่ 2	ถอยหลัง 5 V				

ตารางที่ 4.10 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1001

Port MCU	PortC0	PortC1	PortC2	PortC3	สถานะ
LED	3V	0V	0V	3V	1001
Motor ตัวที่ 1	เดินหน้า 5 V				
Motor ตัวที่ 2	ถอยหลัง 5 V				

ตารางที่ 4.11 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 0110

Port MCU	PortC0	PortC1	PortC2	PortC3	สถานะ
LED	0V	3V	3V	0V	0110
Motor ตัวที่ 1	ถอยหลัง 5 V				
Motor ตัวที่ 2	เดินหน้า 5 V				

ตารางที่ 4.12 ตารางแสดงผลการทดสอบการทำงานของ IC L293D พอร์ต C = 1111 และ 0000

Port MCU	PortC0	PortC1	PortC2	PortC3	สถานะ
LED	3V หรือ 0V	3V หรือ 0V	3V หรือ 0V	3V หรือ 0V	1111 หรือ0000
Motor ตัวที่ 1	หยุดนิ่ง 5 V				
Motor ตัวที่ 2	หยุดนิ่ง 5 V				

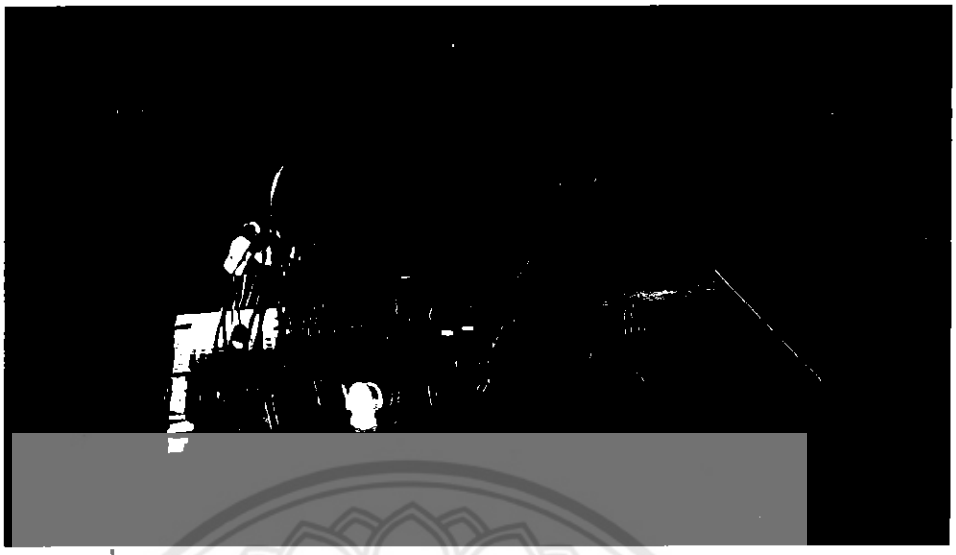
สรุปผลการทดลองครั้งนี้ สามารถใช้ IC L293D ในการขับมอเตอร์ได้ตั้งแต่ 4.5 โวลต์ - 36 โวลต์ สามารถทำงานได้ตามที่ต้องการได้

4.5 ทดสอบการทำงานอุปกรณ์ฝั่งรับ

การทดลองนี้เป็นการประกอบอุปกรณ์ทุกอย่างในภาครับเข้ากับตัวหุ่นยนต์ค้นหา เพื่อทำการทดลองการทำงานทั้งหมดว่าสามารถทำงานได้ตามเงื่อนไขหรือไม่

โดยมีการต่อ Wireless เข้าสู่ตัวไมโครคอนโทรลเลอร์ตัวที่ 1(แรงดันไฟฟ้า 3 โวลต์ เนื่องจาก TRW2.4G ต้องการแรงดันไฟฟ้า 3 โวลต์เท่านั้น) และต่อไมโครคอนโทรลเลอร์ตัวที่ 1 เข้าไปใน IC L293D ทั้ง 2 ตัวเพื่อขับมอเตอร์ ทั้ง 3 ตัว ในการเคลื่อนที่ไปในทิศทางต่างๆ และยังนำไมโครคอนโทรลเลอร์ตัวที่ 1 ไปต่อเข้ากับ SN74HC245N เพื่อยกระดับแรงดันไปติดต่อกับไมโครคอนโทรลเลอร์ตัวที่ 2 แล้วนำ LED ทั้งหมดต่อเข้ากับ พอร์ต C ของไมโครคอนโทรลเลอร์ตัวที่ 2 จากนั้นทำการติดตั้งอุปกรณ์ต่างๆที่เหลือ แล้วทำการทดสอบ ปล่อยสัญญาณจากตัวส่ง ไปยังที่ตัวรับเพื่อตรวจสอบการทำงานของระบบนี้ จะได้ผลดังรูปที่ 4.16-4.18

รูปที่ 4.16 ประกอบวงจรทั้งหมด ลงไปในอุปกรณ์รับข้อมูล (Receiver Module)



รูปที่ 4.17 ส่งสัญญาณจากทางด้านอุปกรณ์ส่งข้อมูล (Transceiver Module)

รูปที่ 4.18 หุ่นยนต์เดินตามการทำงานของอุปกรณ์ส่งข้อมูล (Transceiver Module)

ผลการทดลอง

หุ่นยนต์สามารถทำงานได้ตามการสั่งการของอุปกรณ์ส่งข้อมูล (Transceiver Module) ได้อย่างถูกต้อง แต่มีปัญหาเรื่องเมื่อหุ่นยนต์ห่างจากอุปกรณ์ส่งข้อมูล (Transceiver Module) ไกลเท่าไรสัญญาณอาจมีการรับค่าผิดพลาดบ้าง ทำให้ต้องสั่ง Reset ไมโครคอนโทรลเลอร์ เพื่อเริ่มการทำงานใหม่ ผลของการทำงานได้ผลตามตารางที่ 4.13

ตารางที่ 4.13 ตารางแสดงผลการทดสอบการการรับส่งระหว่างตัวส่ง (Transceiver) กับหุ่นยนต์ (Receiver)

Dataout[3]	101000	000010	010001	111111
Motor ตัวที่ 1	เดินหน้า	หยุดนิ่ง	ถอยหลัง	หยุดนิ่ง
Motor ตัวที่ 2	เดินหน้า	หยุดนิ่ง	หยุดนิ่ง	หยุดนิ่ง
Motor ตัวที่ 3 (กลไกยก)	หยุดนิ่ง	ยกขึ้น	ยกลง	หยุดนิ่ง
LED	ไม่ทำงาน	ไม่ทำงาน	ไม่ทำงาน	ทำงาน

สรุปผลการทดลองจะเห็นได้ว่ามีการทำงานตามที่ได้ตั้งค่าไว้ได้อย่างถูกต้อง โดยที่มีการกำหนดเงื่อนไขให้ถ้ามีการส่งค่า 111111 มาให้ PortB0 ทำงานจึงทำให้ LED ติด จากการทดลองสามารถควบคุมอุปกรณ์ได้อย่างดี แต่อาจจะติดปัญหาเรื่องการรับส่งสัญญาณระยะไกล ซึ่งโดนสัญญาณรบกวน รบกวนเป็นจำนวนมากเนื่องจากกำลังส่งอาจจะไม่พอ

4.6 อุปกรณ์เสริม

ในโครงงานนี้ต้องพึ่งอุปกรณ์เสริมเพื่อความสะดวกในการใช้งาน จึงทำให้ต้องมีการทดสอบการทำงานของอุปกรณ์เสริมร่วมกับการทำงานของหุ่นยนต์ค้นหา โดยจะแบ่งตามหัวข้อต่างๆดังนี้

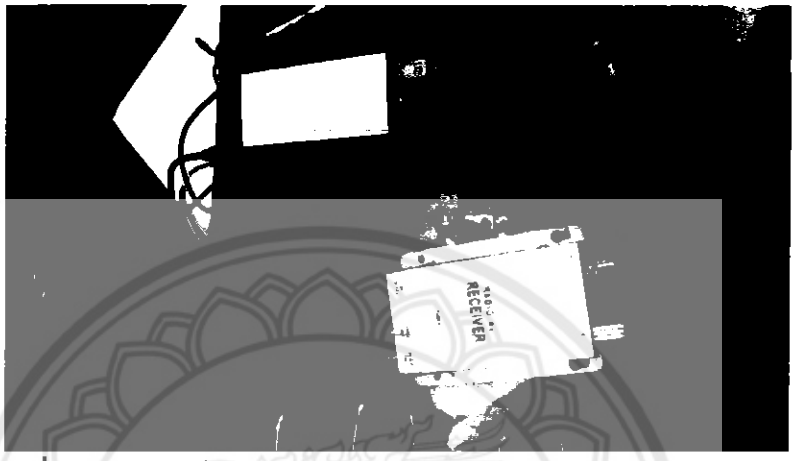
4.6.1 อุปกรณ์แปลงสัญญาณกล้องเป็น AV

ทดสอบโดยการต่อกล้องเข้ากับตัวแปลงสัญญาณ และต่อตัวแปลงสัญญาณเข้ากับโทรทัศน์ ผลการทดลองได้ดังรูปที่ 4.19

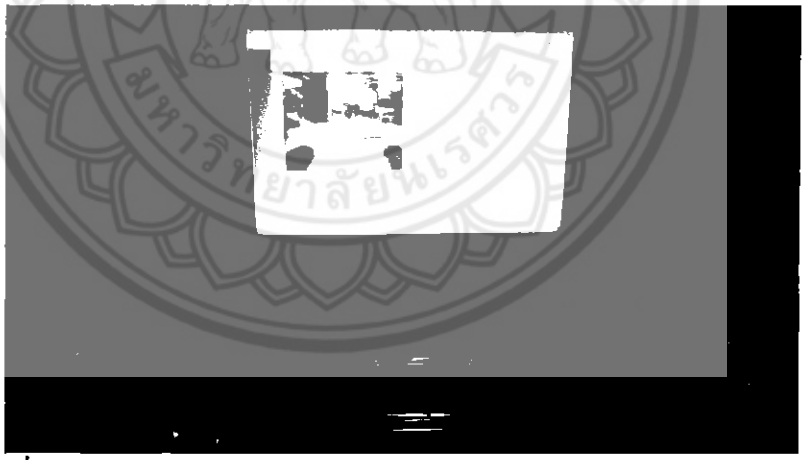
รูปที่ 4.19 แสดงผลภาพจากกล้องที่หุ่นยนต์ผ่านโทรทัศน์

4.6.2 อุปกรณ์แปลงสัญญาณ AV เข้าคอมพิวเตอร์

เป็นการแปลงสัญญาณจากสาย AV เข้าไปสู่ในรูป USB เพื่อติดต่อสื่อสารกับคอมพิวเตอร์ผ่านพอร์ตต่างๆ โดยใช้โปรแกรม EnCoder เป็นตัวช่วยในการแสดงผล ผลการทดลองดังรูปที่ 4.20 - 4.21



รูปที่ 4.20 การเชื่อมต่ออุปกรณ์แปลงสัญญาณ AV เข้ากับอุปกรณ์แปลงสัญญาณเข้าคอมพิวเตอร์

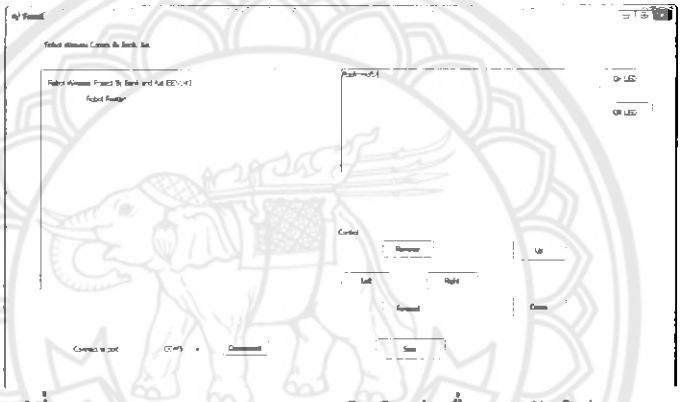


รูปที่ 4.21 การแสดงผลภาพของอุปกรณ์แปลงสัญญาณ AV เข้าคอมพิวเตอร์

4.6.3 โปรแกรมบังคับจอยสติก Joy To Key

เป็นโปรแกรมที่ใช้ในการบังคับจอยสติกให้ออกมาเป็นตัวหนังสือ ในการทดลองนี้ใช้ในการควบคุมการทำงานของโปรแกรม Visual Basic 2005 เพื่อที่จะง่ายต่อการควบคุมหุ่นยนต์ โดยค่าที่ได้จะรับมาเป็นตัวหนังสือ จากผลการทดลองในการกดดังรูปที่ 4.22

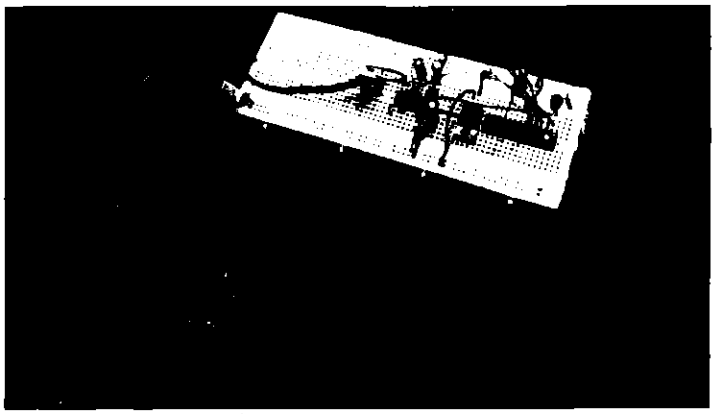
รูปที่ 4.22 จอยสติกที่ใช้ในการติดต่อกับคอมพิวเตอร์



รูปที่ 4.23 ผลการทดสอบจอยสติกติดต่อสื่อสารกับโปรแกรม VB2005

4.6.4 อุปกรณ์แปลง RS-232 เป็น USB

เนื่องจากในโครงงานนี้จำเป็นต้องใช้ช่องสัญญาณ Rs-232 ซึ่งในคอมพิวเตอร์หลายรุ่นจะไม่มีช่องประเภทนี้แล้วจึงจำเป็นต้องทำการแปลงเป็น USB เพื่อจะสามารถใช้งานพอร์ตอนุกรม Rs-232 ได้ การทดลองสามารถเชื่อมต่อได้ และใช้งานได้ดี วิธีการเชื่อมต่อดังรูปที่ 4.24



รูปที่ 4.24 ผลการทดสอบการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่าน RS-232

4.7 การทำงานทั้งหมดของระบบ

ขั้นตอนสุดท้ายในการทดลองนี้จะเป็นการนำอุปกรณ์ทั้งหมดต่อเข้าหากันแล้วทดลองการทำงานจริงกับหุ่นยนต์ค้นหานี้ เพื่อหาจุดบกพร่อง เพื่อแก้ไขและพัฒนาต่อไป โดยจะมีขั้นตอนต่างๆดังนี้

4.7.1 เชื่อมต่ออุปกรณ์เสริมทั้งหมดเข้าสู่คอมพิวเตอร์

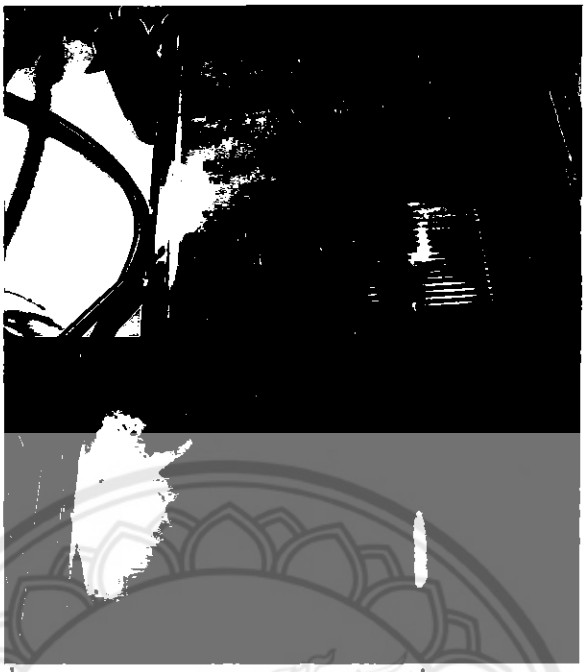
เป็นการนำอุปกรณ์ทั้งหมดในหัวข้อ 4.6 มาต่อรวมกันแล้วทดสอบโปรแกรมดังรูปที่ 4.25

รูปที่ 4.25 ต่ออุปกรณ์ทั้งหมดเข้าสู่คอมพิวเตอร์

4.7.2 เชื่อมสายไฟทั้งหมดเข้าสู่ตัวแผงวงจรของหุ่นยนต์

นำสาย LED สายมอเตอร์ทั้ง 3 ตัว สายกล้อง สายแหล่งจ่าย 9 โวลต์ ต่อเข้าสู่วงจร และทำการทดสอบวัดแรงดันไฟฟ้าที่จุดต่างๆเพื่อเช็คอุปกรณ์อีกครั้งก่อนทำงาน ดังรูปที่ 4.26-4.27

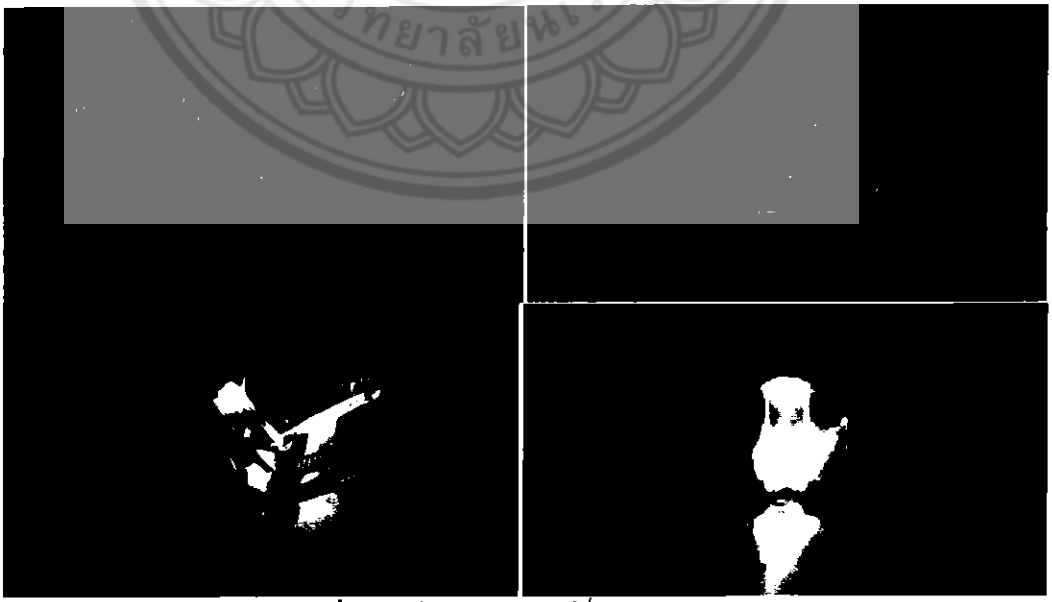
รูปที่ 4.26 ต่อสายไฟต่างๆเข้าสู่วงจร



รูปที่ 4.27 วัดแรงดันไฟฟ้าจากจุดต่างๆ เพื่อเช็คความพร้อม

4.7.3 ทดสอบการควบคุมหุ่นยนต์ผ่านคอมพิวเตอร์

ในการทดสอบครั้งนี้ เป็นการนำทุกสิ่งทุกอย่างมาทดสอบพร้อมกัน ให้ทำงานได้จริง เพื่อทดสอบการทำงานว่าทำงานได้จริงหรือไม่ ผลการทดสอบออกมาดังรูปที่ 4.28 - 4.29



รูปที่ 4.28 สั่งการเปิดไฟที่หน้าตัวหุ่นยนต์

รูปที่ 4.29 สั่งการให้หุ่นยนต์เดินไปข้างหน้าและหน้าจอแสดงผลขณะเดิน

สรุปผลการทดลอง

การทดลองประสบความสำเร็จโดยมีการสั่งการอุปกรณ์ได้ตามที่ต้องการ อาจจะมีปัญหาเรื่องการรับส่งค่าช้าบ้างในบางครั้ง หรือการรับค่าที่ผิดพลาด เกิดจากสาเหตุอุปกรณ์ เมื่อรับค่าเร็วเกินไปอาจจะทำให้อ่านค่าผิดพลาด อาจมีการแก้ไขโดยการเพิ่มความเร็วในการติดต่อสื่อสารกันระหว่างคอมพิวเตอร์กับอุปกรณ์ส่งข้อมูลได้

ข้อเสนอแนะ

ในโครงงานนี้เป็นโครงงานที่มีการทำงานผ่านสัญญาณไวเลสทำให้มีการรบกวนของสัญญาณภายนอกเป็นจำนวนมาก ถ้าต้องการให้สัญญาณออกมามากที่สุดควรมีการป้องกันสัญญาณรบกวนจากอุปกรณ์อย่างดีโดยการชิลอุปกรณ์ต่างๆลงกราวด์ให้หมด เพื่อลดสัญญาณรบกวนจากอุปกรณ์บนวงจรเดียวกัน และถ้าสัญญาณอ่อนให้ทำการเพิ่มกำลังส่งที่ตัวส่งสัญญาณ ข้อควรระวังห้ามเพิ่มแรงดันไฟฟ้าที่ขาไฟเลี้ยงของอุปกรณ์ Module wireless อาจจะทำให้อุปกรณ์พังได้

บทที่ 5
สรุปผลโครงงาน

ในบทที่ 5 จะสรุปผลของโครงงาน รวมถึงปัญหาและข้อเสนอแนะเพิ่มเติม เพื่อเป็นแนวทางในการพัฒนาโครงงานต่อไป

5.1 สรุปผลของโครงงาน

จากการทดลองหุ่นยนต์ค้นหาต้นฉบับที่ทำงานด้วยการส่งข้อมูลผ่านระบบสัญญาณ Wireless สามารถทำงานได้จริงอย่างมีประสิทธิภาพ โดยมีการส่งสัญญาณอยู่ในย่านความถี่ 2.4 GHz เพื่อไปควบคุมอุปกรณ์ต่างๆได้ดี แต่อาจจะพบปัญหาการทำงานบ้างเนื่องจากการส่งสัญญาณอาจจะถูกสัญญาณภายนอกรบกวนทำให้เกิดความผิดพลาดในการสั่งการในบางครั้ง และการกดปุ่มสั่งการเร็วๆอาจจะทำให้การส่งข้อมูลระหว่างคอมพิวเตอร์กับตัวไมโครคอนโทรลเลอร์เกิดข้อผิดพลาดได้ วิธีแก้ไขอาจจะต้องเพิ่มความเร็วในการส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์แต่ก็ยังมีข้อจำกัดในส่วนนี้ ถ้าต้องการให้หุ่นยนต์รับสัญญาณได้ไกลขึ้น ต้องเพิ่มกำลังไฟฟ้าที่อุปกรณ์ส่งข้อมูลเพื่อให้สัญญาณมีกำลังมากขึ้น และอาจจะพบปัญหาข้อผิดพลาดในการส่งข้อมูลระหว่างไวเลส ทำให้ควบคุมผิดพลาดในบางครั้งอาจจะต้องลดความเร็วในการส่งสัญญาณลงจะทำให้รับข้อมูลและเช็คบิตผิดพลาดได้มากขึ้น

ในโครงงานนี้หุ่นยนต์ค้นหาก็ยังสามารถทำงานได้ตามเป้าหมายที่วางไว้ ในโครงงาน แต่ยังต้องการพัฒนาคุณภาพอีกหลายอย่าง เพื่อเป็นหุ่นยนต์ต้นแบบการค้นหาผู้รอดชีวิตจากสถานที่ที่มนุษย์ไม่สามารถเข้าไปถึงได้ เพื่อพัฒนาการกู้ภัยชีวิตของมนุษย์จากเหตุการณ์ต่างๆได้ดียิ่งขึ้น

5.2 ปัญหาที่เกิดขึ้น

1.อุปกรณ์เชื่อมต่อกับ Module wireless ไม่มีขายทำให้ต้องทำการเชื่อมอุปกรณ์ต่อเข้ากับแผงวงจร ทำให้เวลาเกิดปัญหาไม่สามารถแก้ไขได้ และในการเชื่อมต่ออุปกรณ์ Module wireless เข้ากับแผงวงจรทำให้เกิดสัญญาณรบกวนเป็นจำนวนมากถ้าทำการต่อด้วยสาย

2.อุปกรณ์ Module wireless เปราะบางมาก และไม่ทนต่อกระแสไฟฟ้าที่สูงกว่า 3.3 โวลต์ได้

3.ปัญหาด้านคุณสมบัติของอุปกรณ์ เนื่องจากไมโครคอนโทรลเลอร์สามารถปรับค่าสัญญาณนาฬิกาได้จำกัดทำให้ไม่สามารถตั้งค่าความเร็วในการส่งข้อมูลสูงๆได้ ทำให้การส่งข้อมูลติดต่อกันไวๆอาจจะเกิดปัญหาได้ในบ้างครั้ง

4.ปัญหาด้านการเชื่อมต่อสายเป็นลายวงจร อาจจะลัดวงจรได้ เนื่องจากไม่ได้ทำการกดลายวงจร และเมื่อเจอการกระแทกมากอาจจะทำให้ลัดวงจรได้เช่นกัน เพราะไม่มีกล่องเก็บแผงวงจร

5.ปัญหาด้านกลศาสตร์ในตัวหุ่นยนต์มีน้ำหนักถ่วงที่ไม่เท่ากัน เนื่องจากปัญหาของขนาดหุ่นยนต์กับตัวอุปกรณ์ที่แตกต่างกัน และการออกแบบมาเพื่อใช้ทดลองมากกว่าการใช้จริง

5.3 การแก้ปัญหาและข้อเสนอแนะ

1.ปัญหาอุปกรณ์เชื่อมต่อกับ Module wireless ไม่มีขายให้ทำการแผงวงจรแล้วมาเจาะรูให้เล็กเท่ากับ pin ของอุปกรณ์ Module wireless แล้วกัดลายวงจรเพื่อทำตัวเชื่อมต่อออกจากแผงวงจรเพื่อเป็น Output ต่อไป

2.ก่อนจะทำการต่ออุปกรณ์ Module wireless ทุกครั้งตรวจเช็คแหล่งจ่ายไฟฟ้า หรือตรวจสอบเร็กกูเรทเตอร์ว่าพร้อมใช้งานหรือไม่ ทำการวัดกระแสไฟฟ้าก่อนทุกครั้ง เพื่อลดความเสี่ยงต่อการจ่ายกระแสไฟฟ้าผิดทำให้ Module wireless พังได้

3.เปลี่ยนอุปกรณ์ไมโครคอนโทรลเลอร์ที่มีคุณสมบัติเพิ่มมากขึ้น เพื่อสามารถตั้งค่าความเร็วในการส่งสัญญาณได้ แต่ราคากับขนาดก็จะเพิ่มขึ้นมาตามคุณสมบัติ

4.วิธีการออกแบบลายวงจรแล้วทำการกัดลายวงจรเป็นวิธีที่ดีที่สุด แต่ถ้าต้องการต่อเพื่อทดสอบการทำงานของระบบว่าทำงานได้จริง ก็ควรใช้สายขนาดเล็กเชื่อมต่อเข้ากับ Node ต่างๆ และการต่อ Node ต่างๆ ไม่ควรจะมีการพวงสายมากเกินไปเนื่องจากจะทำให้ Node นั้นเกิดความร้อนสะสมแล้วทำให้บอร์ดมีปัญหาได้ในภายหลัง

5.ปัญหาเรื่องทางกลศาสตร์ อาจจะต้องหาอุปกรณ์ถ่วงน้ำหนักเพื่อให้มีน้ำหนักที่เท่ากัน แต่ถ้าต้องการความมั่นคงอาจจะขยายขนาดของหุ่นยนต์ เปลี่ยนวัสดุในการทำ เพื่อจะได้ออกแบบที่วางอุปกรณ์ต่างๆได้อย่างพอดี แต่ก็ต้องคำนึงถึงจุดประสงค์ในการทำงานของอุปกรณ์นั้นๆด้วย

5.4 แนวทางการพัฒนาเพิ่มเติม

1. พัฒนาค่าความผิดพลาดในการส่งสัญญาณให้ได้น้อยที่สุด เมื่อทำการสั่งการทางหุ่นยนต์ต้องกระทำสิ่งนั้นทันทีไม่มีหน่วงเวลา และไม่ผิดพลาด

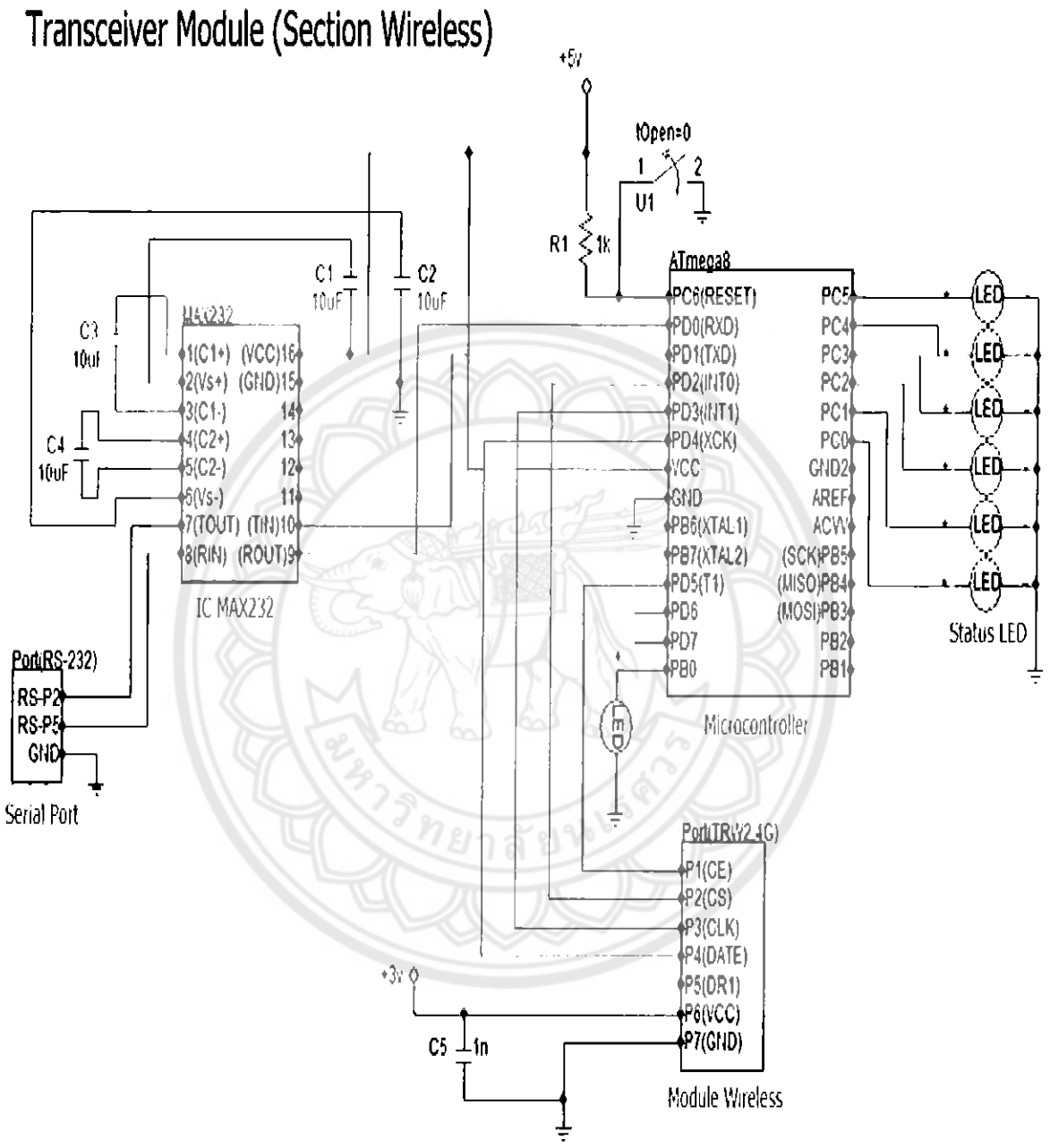
2. เพิ่มความสามารถหุ่นยนต์เช่น การสั่งการหุ่นยนต์ผ่านเสียง หรือใส่อุปกรณ์หยิบจับสิ่งของเข้าไป

3. พัฒนาอุปกรณ์การส่งสัญญาณให้สามารถส่งได้ไกลมากขึ้น

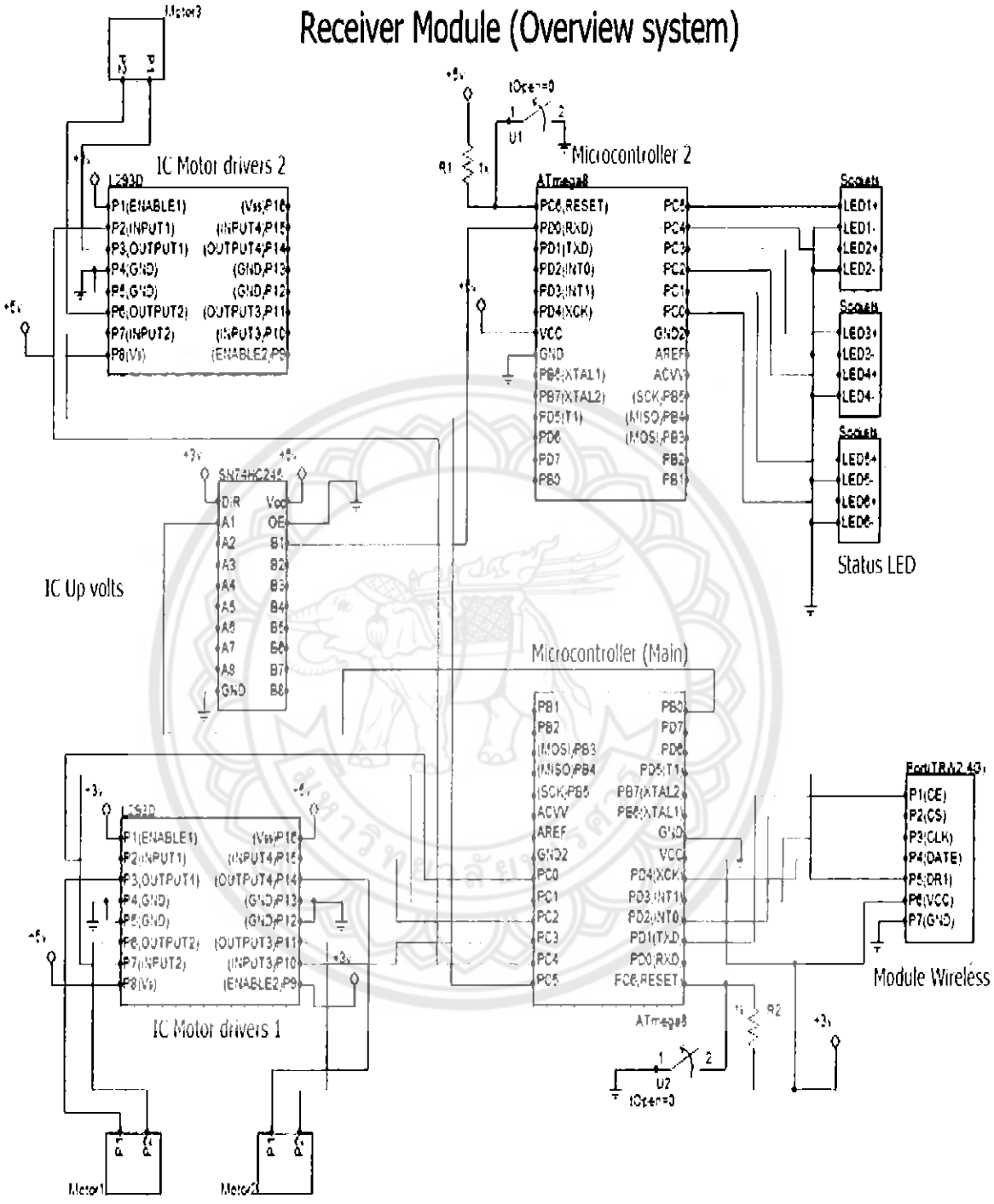
4. พัฒนาหุ่นยนต์ต้นแบบให้สมบูรณ์แบบจนสามารถสร้างแม่แบบในการสร้างหุ่นยนต์ที่นำไปใช้ในชีวิตจริงได้

ภาคผนวก ก

ไดอะแกรมการทำงานของวงจรที่อุปกรณ์ส่งสัญญาณ



ไดอะแกรมการทำงานของวงจรที่อุปกรณ์รับสัญญาณ



ภาคผนวก ข

โปรแกรมควบคุมหุ่นยนต์ผ่าน Rs-232 (Visual Basic 2005)

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        ComboBox1.Items.Add("COM1")
        ComboBox1.Items.Add("COM2")
        ComboBox1.Items.Add("COM3")
        ComboBox1.Items.Add("COM4")
        ComboBox1.Items.Add("COM5")
        GroupBox1.Text = ""
        GroupBox2.Text = "Control"
        Label1.Text = "Connect to port"
        Button1.Text = "Connect"
        Button2.Text = "Reverse"
        Button3.Text = "Forward"
        Button4.Text = "Left"
        Button5.Text = "Right"
        Button6.Text = "Stop"
        Button7.Text = "Up"
        Button8.Text = "Down"
        Button9.Text = "Off LED"
        Button10.Text = "On LED"
        Button11.Text = "Forward Left"
        Button12.Text = "Forward Right"
        Button13.Text = "Roll Left"
        Button14.Text = "Roll Right"
        Button15.Text = "Control By Joy"
        Timer1.Stop()
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If Button1.Text = "Connect" Then
            With SerialPort1
```

```
            .PortName = ComboBox1.Text
            .BaudRate = 9600
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
            .Parity = IO.Ports.Parity.None
        End With
        SerialPort1.Open()
        Button1.Text = "Disconnect"
        Timer1.Interval = 500
        Timer1.Start()
    Else
        SerialPort1.Close()
        Timer1.Stop()
        Button1.Text = "Connect"
    End If
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    If Button1.Text = "Connect" Then
        MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Else
        SerialPort1.Write("f" & vbCrLf)
    End If
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    If Button1.Text = "Connect" Then
        MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Else
        SerialPort1.Write("b" & vbCrLf)
    End If
End Sub
```

```
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("l" & vbCrLf)
        End If
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("r" & vbCrLf)
        End If
    End Sub
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("s" & vbCrLf)
        End If
    End Sub
    Private Sub RichTextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End Sub
    Private Sub RichTextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles RichTextBox1.TextChanged
    End Sub
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        RichTextBox1.AppendText(SerialPort1.ReadExisting)
```

```
    End Sub
    Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End Sub
    Private Sub ProgressBar1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End Sub
    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs)
    End Sub
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("u" & vbCrLf)
        End If
    End Sub
    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("d" & vbCrLf)
        End If
    End Sub
    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("o" & vbCrLf)
        End If
    End Sub
```

```
    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("i" & vbCrLf)
        End If
    End Sub
    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button11.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("z" & vbCrLf)
        End If
    End Sub
    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button12.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("x" & vbCrLf)
        End If
    End Sub
    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button13.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("q" & vbCrLf)
        End If
```

```
    End Sub
    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button14.Click
        If Button1.Text = "Connect" Then
            MessageBox.Show("คุณยังไม่ได้เปิดพอร์ตสื่อสาร", "Port not open", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            SerialPort1.Write("w" & vbCrLf)
        End If
    End Sub
    Private Sub GroupBox2_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
GroupBox2.Enter
    End Sub
    Private Sub Form1_KeyUp(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyUp
        Select Case e.KeyCode
            Case Keys.F
                SerialPort1.Write("f" & vbCrLf)
                e.Handled = True
            Case Keys.B
                SerialPort1.Write("b" & vbCrLf)
                e.Handled = True
            Case Keys.L
                SerialPort1.Write("l" & vbCrLf)
                e.Handled = True
            Case Keys.R
                SerialPort1.Write("r" & vbCrLf)
                e.Handled = True
            Case Keys.S
                SerialPort1.Write("s" & vbCrLf)
                e.Handled = True
            Case Keys.U
                SerialPort1.Write("u" & vbCrLf)
                e.Handled = True
            Case Keys.D
```

```
            SerialPort1.Write("d" & vbCrLf)
            e.Handled = True
        Case Keys.O
            SerialPort1.Write("o" & vbCrLf)
            e.Handled = True
        Case Keys.I
            SerialPort1.Write("i" & vbCrLf)
            e.Handled = True
        Case Keys.Z
            SerialPort1.Write("z" & vbCrLf)
            e.Handled = True
        Case Keys.X
            SerialPort1.Write("x" & vbCrLf)
            e.Handled = True
        Case Keys.Q
            SerialPort1.Write("q" & vbCrLf)
            e.Handled = True
        Case Keys.W
            SerialPort1.Write("w" & vbCrLf)
            e.Handled = True
    End Select
  End Sub
  Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PictureBox1.Click
  End Sub
  Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PictureBox2.Click
  End Sub
  Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PictureBox3.Click
  End Sub
  Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PictureBox4.Click
  End Sub
  Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
    End Sub
    Private Sub Button15_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button15.Click
        Shell("\JoyToKey\JoyToKey.exe", AppWinStyle.NormalNoFocus, False, -1)
    End Sub
End Class
```

โปรแกรมในไมโครคอนโทรลเลอร์อุปกรณ์ฝั่งส่ง (Transceiver Module)

```
#include<avr/io.h>
#include<avr/interrupt.h>
#include<compat/deprecated.h>
#define F_CPU 8000000UL                    // Micro F = 8 MHz
#include <util/delay.h>
#include <string.h>
#include "lib_UART.c"                      // Use UART
#define TRW_Rx cbi(PORTD,PD6)              // Clear bit Rx use PD0
#define TRW_Xx sbi(PORTD,PD6)              // Set bit Rx use PD0
#define CE_HIGH sbi(PORTD,PD5)                  // CE use PD1
#define CE_LOW  cbi(PORTD,PD5)                  // CE use PD1
#define CS_HIGH sbi(PORTD,PD2)                  // CS use PD2
#define CS_LOW  cbi(PORTD,PD2)                  // CS use PD2
#define CLK1_HIGH sbi(PORTD,PD3)           // CLK use PD3
#define CLK1_LOW  cbi(PORTD,PD3)           // CLK use PD3
#define DATA_HIGH sbi(PORTD,PD4)           // DATA use PD4
#define DATA_LOW  cbi(PORTD,PD4)           // DATA use PD4
//#define DR1_HIGH sbi(PORTD,PD5)          // Not Use DR1 in Tx
//#define DR1_LOW  cbi(PORTD,PD5)          // Not Use DR1 in Tx
void Delay_W1();                                    // Function Delay Wireless
void Delay_W2();                                    // Function Delay Wireless
void Delay();                                       // Function Delay Work
void Shift_Data(unsigned char);                     // Function Shift Data
void Config();                                      // Function Config of Wireless
// Function Delay
void Delay_W1(){
   int i;
   for(i=0;i<4;i++){_delay_us(1);}
 }
 void Delay_W2(){
  int i;
  for(i=0; i<6;i++){_delay_us(1);}
}
```

```
void Delay(unsigned int i){
      for(; i>0;i--)
          _delay_ms(1);
 }
// End Function Delay

// Function Main Robot 3 bit
int main(void){
          char buf[3];                                    //ประกาศตัวแปรรับ-ส่งค่า
          DDRD = 0xFF;                                    // Set port D = Output Only
          DDRC = 0xFF;                                    // Set port C = Output Only
          DDRB = 0x01;                                    // Set port B0 = Output
          Init_Serial(96);                                //เปิดใช้การเชื่อมต่อRs-232 ความเร็ว 9600
          PORTC = 0x00;                                   // Clear port D = 0
  unsigned char DataOut[3] = {0,0,0};                     // ประกาศตัวแปรรับ-ส่งค่า
    Config();                                             // Start Config
 fprintf(&uart_str,"\nRobot Wireless Project By Bank & Kat EENU49");
 fprintf(&uart_str,"\n.....");
 fprintf(&uart_str,"\n.....Robot restart.....");
fprintf(&uart_str,"\n.....");
while(1){
//TRW_Rx;
//Check Data from Vb2005 and Sent DataOut
 if(gets(buf)==NULL){
          PORTC = 0xFF;                    // Data = Null ถ้าไม่มีข้อมูลส่งมาหรือเกิด error Led จะติดหมด
             break;          }
if(buf[0]=='f'){
                              PORTC = 0x00;
                              fprintf(&uart_str,"Robot :: Reverse");
                              sbi(PORTC,PC0);
                              sbi(PORTC,PC2);
                              DataOut[0] =0x05;
                              }
 if(buf[0]=='b'){
                              PORTC = 0x00;    '
```

```
                    fprintf(&uart_str,"Robot :: Forward");
                    sbi(PORTC,PC1);
                    sbi(PORTC,PC3);
                    DataOut[0] =0x0A;
                    }
        if(buf[0]=='l'){
                    PORTC = 0x00;
                    fprintf(&uart_str,"Robot :: Left");
                    sbi(PORTC,PC2);
                    DataOut[0] =0x04;
                    }
         if(buf[0]=='r'){
                    PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Right");
                    sbi(PORTC,PC0);
                    DataOut[0] =0x01;
                    }
         if(buf[0]=='s'){
                    PORTC = 0x00;
                    DataOut[0] =0x00;
        fprintf(&uart_str,"Robot :: Stop !!");
                    }
        if(buf[0]=='u'){
                    PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Up");
                    sbi(PORTC,PC4);
                    DataOut[0] =0x10;
                    }
         if(buf[0]=='d'){
                    PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Down");
                    sbi(PORTC,PC5);
                    DataOut[0] =0x20;
                    }
// Loop Left
```

```
        if(buf[0]=='q'){
                        PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Loop Left");
                        sbi(PORTC,PC2);
                        sbi(PORTC,PC1);
                        DataOut[0] =0x06;
// Loop Right
                        }
        if(buf[0]=='w'){
                        PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Loop Right");
                        sbi(PORTC,PC0);
                        sbi(PORTC,PC3);
                        DataOut[0] =0x09;
                        }

// back Left
        if(buf[0]=='z'){
                        PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Forward Left");
                        sbi(PORTC,PC3);
                        DataOut[0] =0x08;
                        }
// back Right
        if(buf[0]=='x'){
                        PORTC = 0x00;
        fprintf(&uart_str,"Robot :: Forward Right");
                        sbi(PORTC,PC1);
                        DataOut[0] =0x02;
                        }
// LED On
        if(buf[0]=='i'){
        fprintf(&uart_str,"Robot :: LED ON !!!");
                        sbi(PORTB,PB0);
                        DataOut[0] = 0x03;
```

```
                    }
// LED Off
        if(buf[0]=='o'){
        fprintf(&uart_str,"Robot :: LED OFF !!!");
                    cbi(PORTB,PB0);
                    DataOut[0] = 0x0C;
                    }
 //*** Start Tx ***
        CE_HIGH;
                     TRW_Xx;
          Delay_W1();
                    Shift_Data(0xAA);          // Send Address
                    Shift_Data(0x55);          // Send Data
                    Shift_Data(0xAA);          // Send Data
                    Shift_Data(0x55);          // Send Data
                    Shift_Data(0xAA);          // Send Data
        Shift_Data(DataOut[0]);    // Send Data
                    Shift_Data(DataOut[1]);  // Send Data
                    Shift_Data(DataOut[2]);  // Send Data
        Delay_W1();
        CE_LOW;
          Delay_W2();
    //*** End Tx ***
Delay_W1();
  }
 return 0;
}
// End Function Main Robot 3 bit
// Function Shift_Data Of Wireless :: Write DATA
void Shift_Data(unsigned char data) {
        unsigned int count;                          // Notice Variables
        unsigned char buf;                           // Notice Variables
   for (count=0;count<8;count++) {
      CLK1_LOW;
       Delay_W1();
```

```
    buf=data & 0x80;
   if (buf==0){
        DATA_LOW;
         Delay_W1();
      } else {
         DATA_HIGH;
         Delay_W1();
      }
        CLK1_HIGH;
        Delay_W1();
                  data<<=1;
}
}
// End Function Shift_Data Of Wireless :: Write DATA
// Function Config bit Of Wireless
void Config(){
   unsigned char conf;
         // Setbit Config
  CE_LOW;
         Delay_W2();
         CS_HIGH;
         Delay_W1();
         // *** Start Config ***
         // Default Config RF (TEST)
       conf =0x8E;
          Shift_Data(conf);  // Bits 143-136 18 byte
          Delay_W2();
    conf =0x08;
          Shift_Data(conf);  // Bits 135-128 17 byte
          Delay_W2();
    conf =0x1C;
          Shift_Data(conf);  // Bits 127-120 16 byte
          Delay_W2();
         // RX2 Data Length = 24 Bit(3 Byte)
    conf =0x18;
```

```
        Shift_Data(conf);  // Bits 119-112 15 byte
        Delay_W2();
      // RX1 Data Length = 24 Bit(3 Byte)
conf =0x18;
        Shift_Data(conf);  // Bits 111-104 14 byte
        Delay_W2();
      // RX2 Address = C0,AA,55,AA,55
conf =0xC0;
        Shift_Data(conf);  // Bits 103-96  13 byte
        Delay_W2();

conf =0xAA;
        Shift_Data(conf);  // Bits  95-88  12 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf);  // Bits  87-80  11 byte
        Delay_W2();
conf =0xAA;
        Shift_Data(conf);  // Bits  79-72  10 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf);  // Bits  71-64   9 byte
        Delay_W2();
      // RX1 Address = AA,55,AA,55,AA
conf =0xAA;
        Shift_Data(conf);  // Bits  63-56   8 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf);  // Bits  55-48   7 byte
        Delay_W2();
conf =0xAA;
        Shift_Data(conf);  // Bits  47-40   6 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf);  // Bits 39-32    5 byte
```

```
        Delay_W2();
    conf =0xAA;
        Shift_Data(conf);  // Bits 31-24    4 byte
        Delay_W2();
       //ADDR_W = 40Bit Address + 16Bit CRC
    conf =0xA3;
        Shift_Data(conf);  // Bits 23-16    3 byte
        Delay_W2();
       //(4FH) CONFIG0 = 1Ch,Shock Burst,250KBPS,0dB
    conf =0x4F;
        Shift_Data(conf);  // Bits 15-8     2 byte
        Delay_W2();
       //CONFIG1 = TX Operation
    conf =0x04;
        Shift_Data(conf);  // Bits 7-0      1 byte
        Delay_W2();
       // ***End Config ***
  CE_HIGH;
  Delay_W2();
  CS_LOW;
  Delay_W2();
}
```

โปรแกรมในไมโครคอนโทรลเลอร์อุปกรณ์ฝั่งรับ ตัวที่ 1 (Receiver Module)

```
#include<avr/io.h>
#include<compat/deprecated.h>
#include<avr/interrupt.h>
#define F_CPU 8000000UL                    // Micro F = 8 MHz
#include <util/delay.h>
#define TRW_Rx cbi(PORTD,PD0)          // Clear bit Rx use PD0
#define TRW_Xx sbi(PORTD,PD0)          // Set bit Rx use PD0
#define CE_HIGH sbi(PORTD,PD1)             // CE use PD1
#define CE_LOW  cbi(PORTD,PD1)             // CE use PD1
#define CS_HIGH sbi(PORTD,PD2)             // CS use PD2
#define CS_LOW  cbi(PORTD,PD2)             // CS use PD2
#define CLK1_HIGH sbi(PORTD,PD3)       // CLK use PD3
#define CLK1_LOW  cbi(PORTD,PD3)       // CLK use PD3
#define DATA_HIGH sbi(PORTD,PD4)       // DATA use PD4
#define DATA_LOW  cbi(PORTD,PD4)       // DATA use PD4
#define DR1_HIGH sbi(PORTD,PD5)            // DR1 use PD5
#define DR1_LOW  cbi(PORTD,PD5)            // DR1 use PD5
void Delay_W1();                                   // Function Delay Wireless
void Delay_W2();                                   // Function Delay Wireless
void Shift_Data(unsigned char);                    // Function Shift Data
void Config();                                     // Function Config of Wireless
unsigned char Rec_Byte(),DataOut[3];   // ประกาศตัวแปร : Res_Byte=  DataOut = Output 3 bit
unsigned char  data = 0x00;                        // ประกาศตัวแปร

// Function Main Robot 3 bit
int main(void){
        DDRD = 0xFF;            // Set port D = Output Only
        DDRB = 0x01;            // Set port B0= Output
        DDRC = 0xFF;            // Set port C = Output Only
        Config();
        DDRD = 0xCF;            // set port D ให้เป็นทั้ง Output และ Input
        // Wireless Code  *** RX Start ***
while(1){
   CS_LOW;                      // PD2 = Clear bit
```

```
        Delay_W1();             // Play Function Delay B1
        CE_HIGH;                // PD1 = Set bit
        Delay_W2();             // Play Function Delay B2
  DATA_LOW;           // PD4 = Clear bit
TRW_Xx;                                     // PD0 = Set bit
        while(!(PIND & 0x20));          // ! PortD = 100000 loop start
  TRW_Rx;                                       // PD0 = Clear bit
  DataOut[0]=Rec_Byte();                    // Rec_Byte() save in DataOut[0]
        DataOut[1]=Rec_Byte();              // Rec_Byte() save in DataOut[1]
        DataOut[2]=Rec_Byte();              // Rec_Byte() save in DataOut[2]
        PORTC = DataOut[0];                 // Output PortC is DataOut[0];
//Check LED On/Off
        if(DataOut[0]==0x03){
                        sbi(PORTB,PB0);
                        DataOut[0] = 0x00;
                        }
        if(DataOut[0]==0x0C){
                        cbi(PORTB,PB0);
                        DataOut[0] = 0x00;
                        }
}
        // Wireless Code *** Rx End ***
 }
// End Function Main Robot 3 bit
// Function Delay
void Delay_W1(){
   int i;
   for(i=0;i<1;i++){_delay_us(10);} //10
 }
 void Delay_W2(){
  int i;
  for(i=0; i<1;i++){_delay_us(50);} //50
}
// End Function Delay
// Function Shift_Data Of Wireless :: Write DATA
```

```
void Shift_Data(unsigned char data) {
  unsigned int count;                                   // Notice Variables
  unsigned char buf;                                    // Notice Variables
   for (count=0;count<8;count++) {
      CLK1_LOW;
      Delay_W1();
      buf=data & 0x80;
    if (buf==0){
        DATA_LOW;
         Delay_W1();
      } else {
        DATA_HIGH;
         Delay_W1();
      }
        CLK1_HIGH;
        Delay_W1();
                    data<<=1;
            }
}
// End Function Shift_Data Of Wireless :: Write DATA
// Function Rec_Byte Of Wireless :: Read DATA
unsigned char Rec_Byte() {
   unsigned int count;
   //unsigned char  data = 0x00;
      if((PIND & 0x10) ==0x10)                  // Input D == 10000 If Play
                                  data|= 0x01;
                else
                            data|= 0x00;
         for (count=0;count<7;count++) {
       data<<=1;
            CLK1_LOW;
        Delay_W1();
       if((PIND & 0x10) ==0x10)                  // Input D == 10000 If Play
                                  data|= 0x01;
                else
```

```
                    data|= 0x00;
    CLK1_HIGH;
                Delay_W1();
    }
                CLK1_LOW;
      Delay_W1();
    if((PIND & 0x10) ==0x10)                    // Input D == 10000 If Play
                                data|= 0x01;
              else
                    data|= 0x00;
    CLK1_HIGH;
                Delay_W1();
return data;                                                    // Sent Data
}
 // End Function Rec_Byte Of Wireless :: Read DATA
// Function Config bit Of Wireless
void Config(){
    unsigned char conf;
        // Setbit Config
  CE_LOW;
        Delay_W2();
        CS_HIGH;
        Delay_W1();
        // *** Start Config ***
            // Default Config RF (TEST)
      conf =0x8E;
        Shift_Data(conf);  // Bits 143-136 18 byte
        Delay_W2();
    conf =0x08;
        Shift_Data(conf);  // Bits 135-128 17 byte
        Delay_W2();
   conf =0x1C;
        Shift_Data(conf);  // Bits 127-120 16 byte
        Delay_W2();
// RX2 Data Length = 8 Bit
```

```
conf =0x18;
        Shift_Data(conf); // Bits 119-112 15 byte
        Delay_W2();
      //RX1 Data Length = 8 Bit(1 Byte)
conf =0x18;
        Shift_Data(conf); // Bits 111-104 14 byte
        Delay_W2();
              //RX2 Address = C0,AA,55,AA,55
conf =0xC0;
        Shift_Data(conf); // Bits 103-96  13 byte
        Delay_W2();
 conf =0xAA;
        Shift_Data(conf); // Bits  95-88  12 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf); // Bits  87-80  11 byte
        Delay_W2();
conf =0xAA;
        Shift_Data(conf); // Bits  79-72  10 byte
        Delay_W2();
 conf =0x55;
        Shift_Data(conf); // Bits  71-64   9 byte
        Delay_W2();
              // RX1 Address = AA,55,AA,55,AA
conf =0xAA;
        Shift_Data(conf); // Bits  63-56   8 byte
        Delay_W2();
  conf =0x55;
        Shift_Data(conf); // Bits  55-48   7 byte
        Delay_W2();
conf =0xAA;
        Shift_Data(conf); // Bits  47-40   6 byte
        Delay_W2();
conf =0x55;
        Shift_Data(conf); // Bits 39-32    5 byte
```

```
        Delay_W2();
    conf =0xAA;
        Shift_Data(conf); // Bits 31-24   4 byte
        Delay_W2();
            //ADDR_W = 40Bit Address + 16Bit CRC
    conf =0xA3;
        Shift_Data(conf); // Bits 23-16   3 byte
        Delay_W2();
            //4FH) CONFIG0 = 1Ch,Shock Burst,250KBPS,0dB
    conf =0x4F;
        Shift_Data(conf); // Bits 15-8    2 byte
        Delay_W2();
            //CONFIG1 = RX Operation
    conf =0x05;
        Shift_Data(conf); // Bits 7-0     1 byte
        Delay_W2();
  // *** End Config ***
   CE_HIGH;
   Delay_W2();
   CS_LOW;
   Delay_W2();
}
// End Function Config bit Of Wireless
```

โปรแกรมในไมโครคอนโทรลเลอร์อุปกรณ์ฝั่งรับตัวที่ 2 (Receiver Module)

```
#include<avr/io.h>
#include<compat/deprecated.h>
#include<avr/interrupt.h>
#define F_CPU 8000000UL    // CPU clock frequency (in Hertz)
#include <util/delay.h>
int main(void)
{
        DDRD = 0x00;                                    // Set port D = Input Only
        DDRC = 0xFF;                                    // Set port C = Output Only
        PORTC = 0x00;
        PORTD = 0x00;
  while(1) {
  if((PIND & 0x01) ==0x01)
                        PORTC = 0xFF;
                else
                        PORTC = 0x00;
  }
  return 0;
}
```

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายกิตติพงษ์ สุขเดหา

ภูมิลำเนา 746/51 ถนนบรมไตรโลกนารถ 2 ซอย 19 ตำบลในเมือง อำเภอเมือง จังหวัดพิษณุโลก รหัสไปรษณีย์ 65000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: bank_volume3_@hotmail.com

ชื่อ นายชัยสิริ มูลอุด

ภูมิลำเนา 85/1 หมู่ 9 ตำบลตากออก อำเภอบ้านตาก จังหวัดตาก รหัสไปรษณีย์ 63120

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนบ้านตาก "ประชาวิทยาคาร"
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail redshadow_kat14@hotmail.com