สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบนำร่องรถยนต์ด้วยแสงเลเซอร์

CAR GUIDING SYSTEM

โดย

นายวิฑูร ชุริวัน

นายสาธิต ทวีตา

นายอนุชา อาจหาญ

เลขหมู่.....
เลขทะเบียน.....62367
วัน,เดือน,ปี.....16 ส.ค. 49

b.....11621722
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนำร่องรถยนต์ด้วยแสงเลเซอร์

CAR GUIDING SYSTEM

โดย

นายวิฑูร ชุริวัน 46015280

นายสาธิต ทวีตา 46015286

นายอนุชา อาจหาญ 46015290

อาจารย์ที่ปรึกษา

อาจารย์เทอดศักดิ์ ลิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2548

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบนำร่องรถยนต์ด้วยแสงเลเซอร์

1. นายวิฑูร ชุริวัน รหัส 46015280

2. นายสาธิต ทวีตา รหัส 46015286

3. นายอนุชา อาจหาญ รหัส 46015290

ลงชื่อ.....อาจารย์ที่ปรึกษา

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบนำร่องรถยนต์ด้วยแสงเลเซอร์

นายวิฑูร ชุริวัน รหัส 46015280
นายสาธิต ทวีตา รหัส 46015286
นายอนุชา อาจหาญ รหัส 46015290
อาจารย์ เทอดศักดิ์ ลิ่วหาทอง อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

รายงานโครงงานฉบับนี้เสนอวิธีสร้างระบบนำร่องรถยนต์ ซึ่งเป็นการนำเอาแสงเลเซอร์มาใช้ในการสแกนสิ่งของหรือวัตถุที่อยู่ข้างหน้ารถยนต์ เพื่อที่จะนำมาวิเคราะห์ว่าเส้นทางข้างหน้าว่าสามารถผ่านไปได้หรือไม่ โดยในขณะที่รถยนต์เคลื่อนที่เลเซอร์ก็จะทำการสแกนอยู่ตลอดเวลาซึ่งจะทำงานร่วมกับกล้องที่อยู่บนตัวรถ โดยจะเก็บภาพของเส้นเลเซอร์บนวัตถุมาประมวลผลใน **Note Book** ที่ทำการติดตั้งไว้หลังรถ ในการประมวลผลก็จะมีเงื่อนไขหลักๆคือ ถ้าสิ่งของหรือวัตถุที่อยู่ข้างหน้าไม่สูงหรือใหญ่เกินไปให้รถวิ่งผ่านวัตถุนั้นไปได้ แต่ถ้าวัตถุนั้นสูงและใหญ่เกินไปก็จะสั่งให้รถเลี้ยวซ้ายหรือเลี้ยวขวาแล้วแต่เงื่อนไขของโปรแกรม จุดมุ่งหมายก็คือทำให้รถหลีกเลี่ยงสิ่งของหรือวัตถุที่ไม่สามารถข้ามไปได้นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CARGUIDING SYSTEM

Mr. Witton Churiwan (ID.46015280)
Mr. Satit Taweeta (ID.46015286)
Mr. Anucha Arthan (ID.46015290)
Mr.Thurdsak Leauhatong (Advisor)
Education Year 2/2005

Abstract

This report presents a method for building the system of which the laser is used for scanning the objects that are in front of the car. The front objects are computerized by CPU in order to analyze whether or not the car are able to pass those object. While a car is moving on the road, The laser is scanning and are sent the everything with a camera. This camera will keep the photos of laser line. The camera that is installed at the back of the car CPU in notebook. The main condition, object must be not over height or big, but if object be over height or big a car will turn left or turn light by a defined condition of program. The objective of project is to move a car without going over the object.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินการทำโครงงานนี้ ผู้จัดทำได้รับคำแนะนำและความช่วยเหลือจากอาจารย์ที่ปรึกษา และรุ่นพี่เป็นอย่างดี ขอขอบคุณภาควิชาอิเล็กทรอนิกส์ ซึ่งได้เอื้อเฟื้ออุปกรณ์และเครื่องมือในการทำงาน และขอขอบคุณผู้มีอุปการคุณอีกหลายท่าน ที่มีส่วนเกี่ยวข้อง จนโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

นาย (วิฑูร ชุริวัน)

นาย (สาธิต ทวีตา)

นาย (อนุชา อาจหาญ)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
บทที่ 1 บทนำ	1
บทที่ 2 โครงสร้างของระบบโดยรวม	2
2.1 หลักการของระบบโดยรวม	4
2.2 หลักการของระบบ	5
บทที่ 3 ทฤษฎีการสื่อสารอนุกรม	8
3.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม	8
3.2 รูปแบบของการสื่อสารข้อมูลอนุกรม	8
3.3 การสื่อสารข้อมูลอนุกรมแบบซิงโครนัส	10
3.4 การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส	10
3.5 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	12
3.6 โหมดการการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	12
3.7 รีจิสเตอร์ที่ใช้ในการควบคุม และรับส่งข้อมูลของพอร์ตอนุกรม	14
3.8 มาตรฐานพอร์ตอนุกรมแบบ RS – 232	17
3.9 คอนเน็คเตอร์สำหรับพอร์ต RS – 232 และการเชื่อมต่อ	17
3.10 รายละเอียดหน้าที่การทำงานในแต่ละขอของพอร์ตอนุกรม RS – 232	20
3.11 แอดเดรสของพอร์ตอนุกรม	21
บทที่ 4 ทฤษฎีพื้นฐานและหลักการทำงานของมอเตอร์กระแสตรง และหลักการทำงานของมอเตอร์กระแสตรงชนิดเฟืองทด	22
4.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง	22
4.2 แรงบิด (Torque) ในมอเตอร์กระแสตรง	26
4.3 วิธีการกลับทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรง	27
4.4 มอเตอร์ไฟฟ้ากระแสตรงชนิดทดเฟือง	27
บทที่ 5 ไมโครโปรเซสเซอร์	29
5.1 ความหมายของไมโครคอนโทรลเลอร์	29
5.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	29
5.3 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 หน้าที่การทำงานของรีจีสเตอร์ 40
5.5 อินเตอร์รัพท์ (Interrupt) 43
5.6 รีจิสเตอร์ที่ใช้งาน 45
5.7 ตัวนับและตัวกำหนดเวลา Counter/Timer 49
บทที่ 6 โครงสร้างและหลักการทำงานของชุดทดลองต้นแบบ 59
6.1 โครงสร้างและส่วนประกอบของระบบนำร่องรถยนต์ 59
6.2 อุปกรณ์ที่ติดตั้งบนรถทดสอบ 60
บทที่ 7 วงจรการทำงานและโปรแกรมการทำงานของชุดทดลองต้นแบบ 67
7.1 ไมโครโปรเซสเซอร์ MCS 51 (AT89C51) 67
7.2 วงจรพัลส์วิดมอดดูเลชั่น (Pulse width Modulation) 69
7.3 การทำงานของวงจรสร้างลำแสงเลเซอร์ 70
7.4 การทำงานของวงจรรีเลย์ 72
7.5 วงจรแหล่งจ่ายไฟ 72
7.6 วงจรอินพุทสวิทช์ 72
7.7 โปรแกรมการทำงานของชุดทดลองต้นแบบ 75
บทที่ 8 คณิตศาสตร์ในการสร้างภาพ 79
8.1 ทฤษฎี กราฟิก 2 มิติ และกราฟิก 3 มิติ 79
8.2 กราฟิก 3 มิติ 81
8.3 การคำนวณหาความสัมพันธ์ของพิกเซล (Pixel) ระยะต่างๆ 82
บทที่ 9 โปรแกรมควบคุมการทำงานของคอมพิวเตอร์แบบพกพา 85
9.1 โครงสร้างการทำงานของโปรแกรมระบบนำร่องรถยนต์ ด้วยแสงเลเซอร์ 85
9.2 หลักการทำงานของโปรแกรมระบบนำร่องรถยนต์ด้วยแสงเลเซอร์
บนคอมพิวเตอร์แบบพกพา 87
บทที่ 10 การทดลอง 93
10.1 การเตรียมอุปกรณ์ในการทดลองการสร้างเส้นลำแสงเลเซอร์สแกนและ
เก็บภาพที่ได้จากกล้องเว็บแคม 93
10.2 วิธีการทดลองอุปกรณ์สร้างเส้นสแกนลำแสงเลเซอร์ 93
10.3 วิธีการทดลองการหามุมเลี้ยวของรถทดสอบ 94
10.4 วิธีการทดลองโปรแกรมระบบนำร่องรถยนต์ด้วยแสงเลเซอร์บนคอมพิวเตอร์
แบบพกพา 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

บทที่ 2

รูปที่ 2.1 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม (ในเทอม 1) 2

รูปที่ 2.2 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม (ในเทอม 2) 3

รูปที่ 2.3 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม ในการใช้งานจริง 3

รูปที่ 2.4 แสดงเส้นเลเซอร์ที่ถูกส่งจากแหล่งกำเนิดไปกระทบวัตถุ 4

รูปที่ 2.5 แสดงโครงสร้างของระบบโดยรวม 4

รูปที่ 2.6 แสดงหลักการทำงานของระบบโดยรวม 5

บทที่ 3

รูปที่ 3.1 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม 7

รูปที่ 3.2 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที 8

รูปที่ 3.3 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น,บิตพาริตี้, บิตหยุด ด้วยความเร็ว 9600 บิตต่อวินาที 9

รูปที่ 3.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส 10

รูปที่ 3.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 10

รูปที่ 3.6 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0 12

รูปที่ 3.7แสดงสัญญาณการส่งข้อมูลในโหมด 1 13

รูปที่ 3.8 แสดงสัญญาณการส่งข้อมูลในโหมด 2 13

รูปที่ 3.9 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register) 14

รูปที่ 3.10 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน 15

รูปที่ 3.11 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล 16

รูปที่ 3.12 การจัดขาของคอนเน็คเตอร์พอร์ตออนุกรมตามมาตรฐาน RS – 232 ทั้งแบบ DB – 9 และ DB – 25 18

รูปที่ 3.13 การเชื่อมต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ ในลักษณะต่างๆ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

รูปที่ 4.1 แสดงกฎมือซ้ายของเฟลมมิ่ง 22

รูปที่ 4.2 วงจรเทียบเคียงแรงดันไฟฟ้าด้านกลับ 23

รูปที่ 4.3 แสดงมอเตอร์ไฟฟ้ากระแสตรงอย่างง่าย 24

รูปที่ 4.4 แสดงการเกิดสนามแม่เหล็กที่ขดลวดอาร์เมเจอร์ 25

รูปที่ 4.5 แสดงการคำนวณแรงบิด 26

รูปที่ 4.6 แสดงการกลับทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแส 27

รูปที่ 4.7 ลักษณะของมอเตอร์ไฟฟ้ากระแสตรงและเกียร์เฟืองทด 28

บทที่ 5

รูปที่ 5.1 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89xx 30

รูปที่ 5.2 โครงสร้างพื้นฐานของอนุกรม AT89xx 31

รูปที่ 5.3 โครงสร้างภายในของ MCS-51 31

รูปที่ 5.4 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51 32

รูปที่ 5.5 แสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 35

รูปที่ 5.6 การต่อใช้งานขาพอร์ต P0.0-P0.7 และ P2.0-P2.7 36

รูปที่ 5.7 การเชื่อมต่อหน่วยความจำข้อมูลภายนอก 37

รูปที่ 5.8 รีจีสเตอร์ PSW (Program Status Word) 40

รูปที่ 5.9 แสดงตำแหน่งขาต่างๆ และการร้องขอการอินเตอร์รัพท์ของไมโครคอนโทรลเลอร์ 44

รูปที่ 5.10 แสดงการเลือกใช้สัญญาณแอคทีฟที่ระดับลอจิก "0" หรือที่ขอบขาลงของสัญญาณ 48

รูปที่ 5.11 แสดงบิตต่างๆของรีจิสเตอร์ TMOD 50

รูปที่ 5.12 แสดงวงจรการทำงานของไทเมอร์ (Timer) ในส่วนของขา GATE INT และ T1 51

รูปที่ 5.13 แสดงการทำงานของ Timer/Counter ในโหมดที่ 0 54

รูปที่ 5.14 แสดงการทำงานของ Timer/Counter ในโหมด1 (16 บิต) 55

รูปที่ 5.15 แสดงการทำงานของ Timer/counter ในโหมด 2 (8 bit Auto reload) 56

รูปที่ 5.16 แสดงการทำงานของ Timer0 ในโหมด 3 (8 bit Auto reload) 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.17 แสดงการกำหนดการทำงานของ Timer/Counter 58

บทที่ 6

รูปที่ 6.1 โครงสร้างและส่วนประกอบโดยรวมของระบบนำร่องรถยนต์ 59
รูปที่ 6.2 กล้องเว็บแคมที่ใช้งาน 60
รูปที่ 6.3 การติดตั้งกล้องเว็บแคม 61
รูปที่ 6.4 โครงสร้างและส่วนประกอบการทำงานของรถทดสอบ 63
รูปที่ 6.5 โครงสร้างและส่วนประกอบของระบบสร้างเส้นสแกนสำแสงเลเซอร์ แบบเก่า (ภาคเรียนที่ 1) 64
รูปที่ 6.6 โครงสร้างและส่วนประกอบของระบบสร้างเส้นสแกนสำแสงเลเซอร์ แบบใหม่ (ภาคเรียนที่ 2) 65

บทที่ 7

รูปที่ 7.1 บล็อกการทำงานของไมโคร โปรเซสเซอร์โดยรวมของระบบทั้งหมด 67
รูปที่ 7.2 วงจรการใช้งานของไมโคร โปรเซสเซอร์ MCS 51 (AT89C51) 68
รูปที่ 7.3 แสดงการรับคำสั่งควบคุมของไมโคร โปรเซสเซอร์ 69
รูปที่ 7.4 แสดงการทำงานของวงจรพัลส์วิดมอดดูเลชั่นในการให้สัญญาณควบคุม แบบต่างๆ 70
รูปที่ 7.5 แสดงวงจรการพัลส์วิดมอดดูเลชั่น วงจรรีเลย์ควบคุมการเคลื่อนที่ และวงจรสร้างลำแสงเลเซอร์ 71
รูปที่ 7.6 สัญญาณที่อุปกรณ์ยิงแสงเลเซอร์ที่ใช้ออสซิเลเตอร์เดียวกับ วงจรพัลส์วิดมอดดูเลชั่น 72
รูปที่ 7.7 แสดงวงจรรีเลย์และวงจรแหล่งจ่ายไฟ 73
รูปที่ 7.8 แสดงวงจรอินพุทสวิทช์ 74
รูปที่ 7.9 แผนภาพแสดงการทำงานของโปรแกรมควบคุมแบบรีโมท 75
รูปที่ 7.10 แผนภาพแสดงการทำงานของโปรแกรมควบคุมแบบออโตเมติก 77

บทที่ 8

รูปที่ 8.1 จุดในระบบกราฟิก 2 มิติ 79
รูปที่ 8.2 เส้นตรงในระบบกราฟิก 2 มิติ 80
รูปที่ 8.3 รูปแสดงตำแหน่งของจุดในระบบ 3 มิติ 81
รูปที่ 8.4 รูปแสดงการหาระยะความสูงของวัตถุจากพิกเซล 82
รูปที่ 8.5 แสดงการหาระยะ Δx_1 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8.6 แสดงการอ่านค่าพิกเซลเพื่อหาระยะต่างๆ 84

บทที่ 9

รูปที่ 9.1 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมระบบนำร่องรถยนต์ 85

รูปที่ 9.2 แผนภาพแสดงโครงสร้างโดยรวมของโปรแกรม 89

รูปที่ 9.3 แผนภาพแสดงโปรแกรมตรวจสอบ Pixel 90

รูปที่ 9.4 แผนภาพแสดงโปรแกรมการจำกัดขอบเขตข้อมูล 91

รูปที่ 9.5 แผนภาพแสดงโปรแกรมคำนวณหาค่าความสูงและระยะของวัตถุและ
โปรแกรมเปรียบเทียบ 92

บทที่ 10

รูปที่ 10.1 เส้นสแกนเลเซอร์ของอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 1 93

รูปที่ 10.2 เส้นสแกนเลเซอร์ของอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 2 93

รูปที่ 10.3 แสดงมุมเลี้ยวของล้อรถทดสอบทั้งสองล้อ 94

รูปที่ 10.4 แสดงวงเลี้ยวของรถทดสอบทั้งขวาและซ้าย 94

รูปที่ 10.5 แสดงรูปที่ผ่านโปรแกรมเก็บภาพจากกล้องเว็บแคม 95

รูปที่ 10.6 แสดงรูปที่ผ่านโปรแกรมตรวจสอบ Pixel และ โปรแกรมการ
จำกัดขอบเขตข้อมูล 96

รูปที่ 10.7 แสดงรูปการแสดงผลค่าที่ได้จากการคำนวณ 96

รูปที่ 10.8 แสดงรูปที่ผ่านโปรแกรมเปรียบเทียบแล้วมีค่าน้อยกว่าค่าที่ตั้งไว้ 97

รูปที่ 10.9 แสดงรูปที่ผ่านโปรแกรมเปรียบเทียบแล้วมีค่ามากกว่าค่าที่ตั้งไว้
และแสดง Alram และหยุดการเคลื่อนที่ของรถทดสอบ 98

รูปที่ 10.10 แสดงผลของโปรแกรมเมื่อใช้วัตถุทดสอบที่มีความสูง 3 ซม. 99

รูปที่ 10.11 แสดงรูปค่าคงที่ต่างๆ ที่ใช้ในการทดลอง 100

บทที่ 11

รูปที่ 11.1 แสดงผลการทดลองเมื่อใช้วัตถุทดสอบที่มีความสูง 10 ซม. 101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

บทที่ 3

ตารางที่ 3.1 แสดงตัวอย่างของบิตพาริตี้ในการรับ – ส่งข้อมูลอนุกรม 11

ตารางที่ 3.2 การจัดขาของคอนเน็คเตอร์พอร์ตออนุกรมตามมาตรฐาน RS – 232 ทั้งแบบ DB – 9 และ DB – 25 18

บทที่ 5

ตารางที่ 5.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่ผลิตโดยบริษัท ATMEL 38

ตารางที่ 5.2 ชื่อและตำแหน่งรีจิสเตอร์ในไมโครคอนโทรลเลอร์ตระกูล MCS-51 39

ตารางที่ 5.3 ตำแหน่งรีจีสเตอร์ R0-R7 41

ตารางที่ 5.4 แสดงสัญญาณ และความหมายของการอินเตอร์รัพท์ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ระบบนำร่องรถยนต์ เป็นโครงงานซึ่งสร้างขึ้นจากแนวคิดของการนำเทคโนโลยีด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ รวมเข้ากับระบบเคลื่อนที่ของรถยนต์ ซึ่งเทคโนโลยีนี้ยังแพร่หลายไม่มากนัก ซึ่งโครงการนี้ใช้หลักการการสร้างภาพ 3 มิติ จึงจำเป็นต้องผ่านขั้นตอนที่ยุ่งยากซับซ้อนมาก คือ

- ขั้นตอนแรก คือ ตรวจสอบสภาวะของรถยนต์(สถานะของรถ, ความเร็ว) สภาวะพื้นถนน (ในระยะสแกนเป็นพื้นที่เรียบหรือไม่) และสภาวะสิ่งแวดล้อม (มีสิ่งกีดขวางที่บริเวณต่างๆ ที่รถต้องเคลื่อนที่ผ่านหรือไม่) เก็บข้อมูลต่างๆ แล้วเมื่อได้รับคำสั่งให้เคลื่อนที่รถยนต์จะเคลื่อนที่ตามคำสั่งที่ได้รับ
- ขั้นตอนที่สอง คือ การใช้ลำแสงเลเซอร์สแกนเป็นเส้นตรงบนพื้นถนนเพื่อตรวจสอบเส้นทาง โดยการสแกนเป็นช่วงๆ บนพื้นถนนตามทิศทางการเคลื่อนที่ของรถยนต์ และนำภาพที่กล้องเว็บแคม (Web Cam) ที่เป็นลำเส้นสแกนมาถอดเป็นรูปโครงร่าง โดยใช้โปรแกรมที่สร้างขึ้น ซึ่งโปรแกรมที่สร้างขึ้นสามารถรู้ว่าเป็นพื้นที่เรียบหรือมีวัตถุที่มีมุมขนาดก็องศา และนำข้อมูลที่ได้มาประมวลผลเป็นภาพโครงสร้าง 3 มิติออกมา
- ขั้นตอนที่สาม คือ นำโครงร่างที่ได้มาพิจารณา เช่น ไม่มีวัตถุอยู่ตรงหน้า หรือมีวัตถุที่มีรูปทรงเป็นเช่นไร มีความสูงเท่าไร และสามารถผ่านไปได้หรือไม่

จากขั้นตอนที่กล่าวข้างต้น เป็นการใช้ประโยชน์โดยการนำภาพเส้นสแกนของลำแสงเลเซอร์มาประมวลผลเป็นข้อมูล 3 มิติ แต่ประโยชน์ของเทคโนโลยีนี้ยังสามารถนำไปใช้ประโยชน์ต่างๆ ได้อีก เช่น การตรวจสอบวัตถุของหุ่นยนต์ ภารตรวจสอบความเร็วของการเคลื่อนที่ของวัตถุ การหาตำแหน่งของวัตถุ

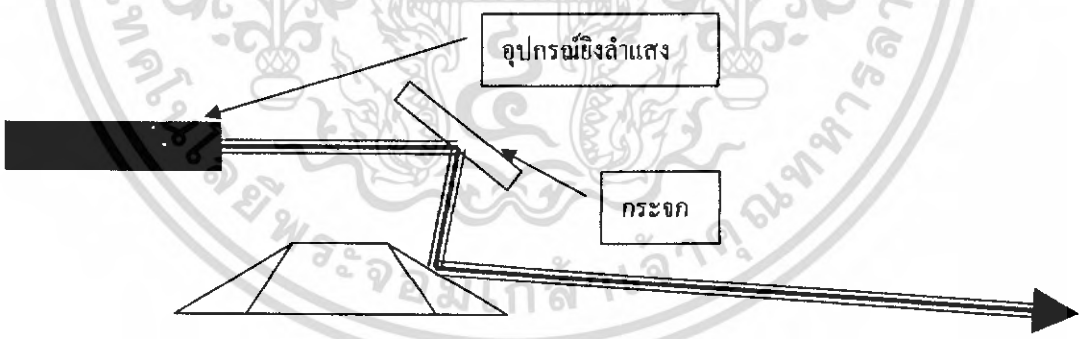
ที่กล่าวมาเป็นเพียงประโยชน์ส่วนหนึ่งเท่านั้น เทคโนโลยีนี้สามารถนำหลักการไปประยุกต์ใช้งานต่างๆ ได้อีกมากมาย โดยโครงงานนี้ก็ทำเพื่อเป็นจุดเริ่มต้นในการใช้งานเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
โครงสร้างของระบบโดยรวม

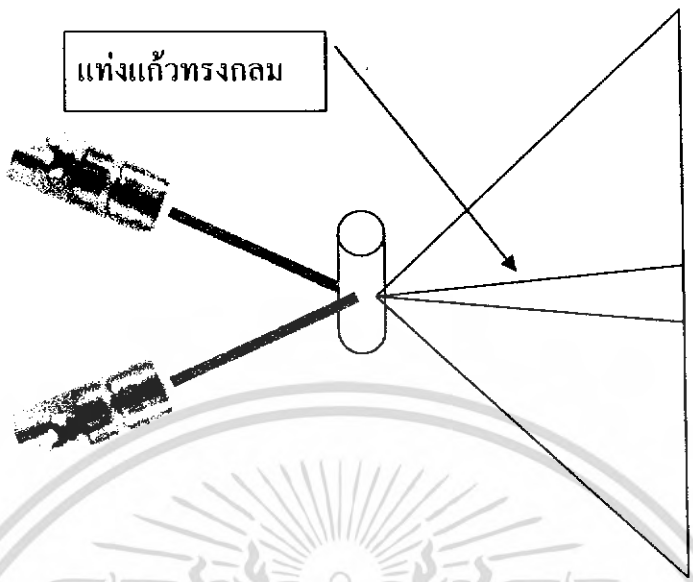
ระบบนำร่องรถยนต์ด้วยแสงเลเซอร์ (Car Guiding System) เป็นอุปกรณ์ที่ใช้ลำแสงเลเซอร์สแกนลักษณะของพื้นผิวถนน หรือวัตถุที่อยู่บนพื้นผิวถนน ออกมาเป็นรูปโครงสร้างสามมิติ แล้วจึงแสดงออกมาหน้าจอมอนิเตอร์ของเครื่องคอมพิวเตอร์แบบพกพา (Computer Note Book) ลักษณะการทำงานของอุปกรณ์นี้คือ การถ่ายภาพผ่านกล้องเว็บแคม แล้วนำมาถอดข้อมูลที่ได้มาเป็นเส้นสแกนหลายๆ ภาพ แล้วนำมาต่อกัน จะมีช่วงการสแกนตามความเร็วรถยนต์ และนำมาประมวลผลผลว่าวัตถุมีรูปร่างลักษณะใด แล้วจึงสั่งให้รถยนต์ทำงานหรือเคลื่อนที่ต่อไป โดยจะใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ควบคุมการทำงานต่างๆ ของรถยนต์และส่งค่าข้อมูลที่จำเป็นในการประมวลผลให้กับคอมพิวเตอร์แบบพกพา โดยใช้การสื่อสารแบบอนุกรม (Serial Port)

ส่วนประกอบของระบบนำร่องรถยนต์ ที่สำคัญได้แก่ รถทดสอบ, กล้องเว็บแคม, การสร้างสร้างเส้นสแกนลำแสงเลเซอร์ และคอมพิวเตอร์แบบพกพา ซึ่งรายละเอียดในการออกแบบในแต่ละส่วนจะแสดงไว้ในบทต่อๆ ไป

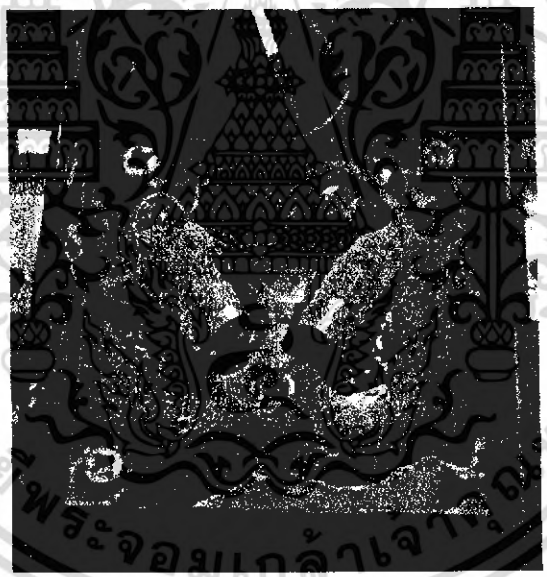


รูปที่ 2.1 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม (ในเทอม 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

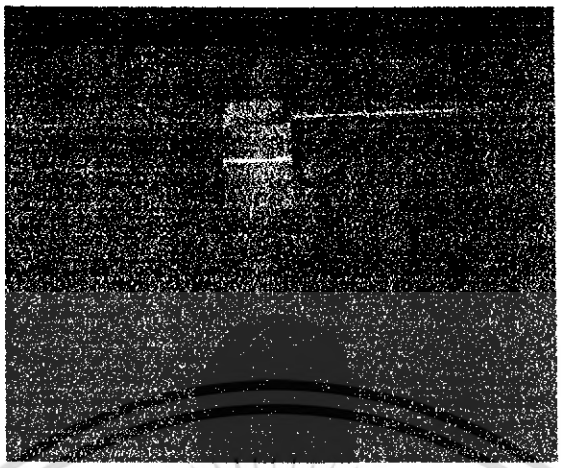


รูปที่ 2.2 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม (ในเทอม 2)

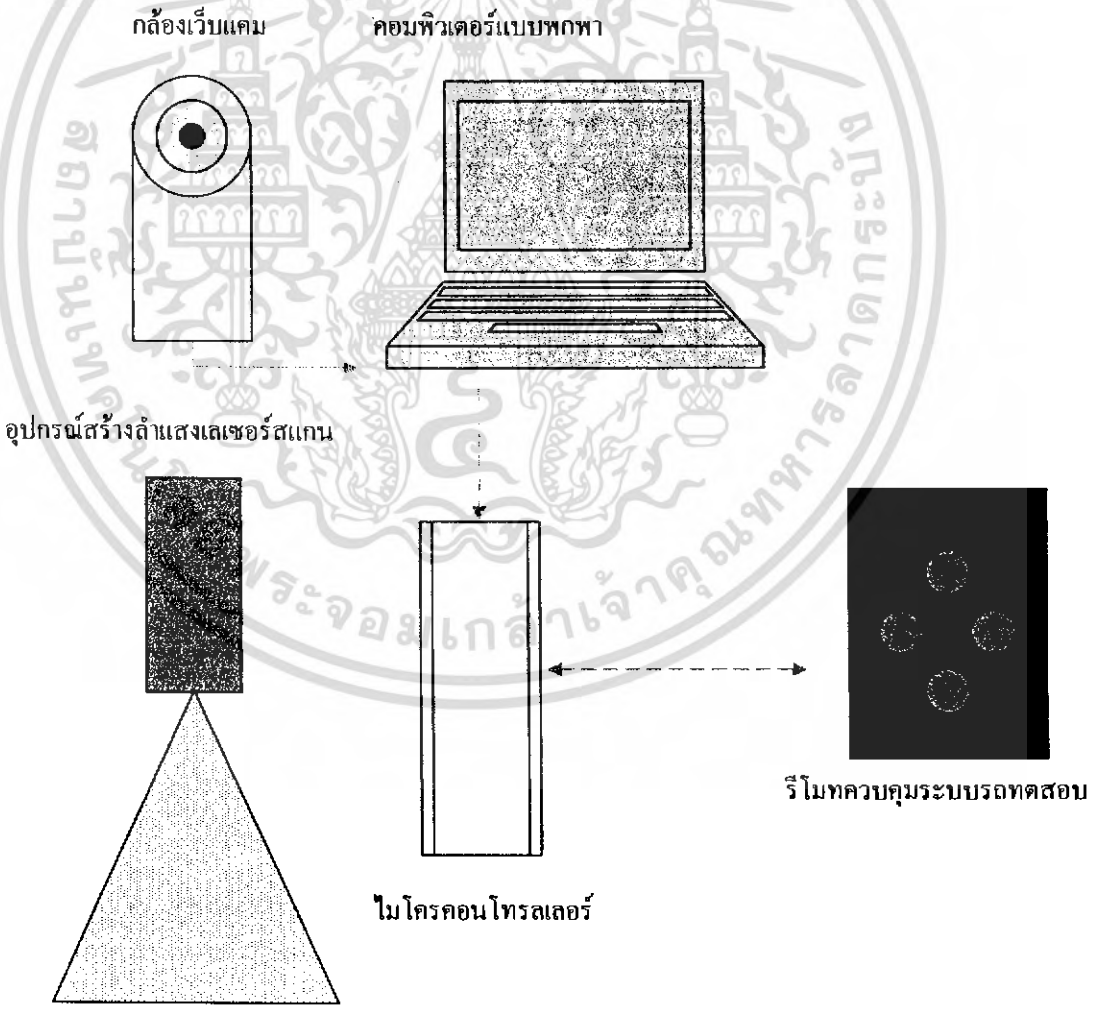


รูปที่ 2.3 แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคมในการใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



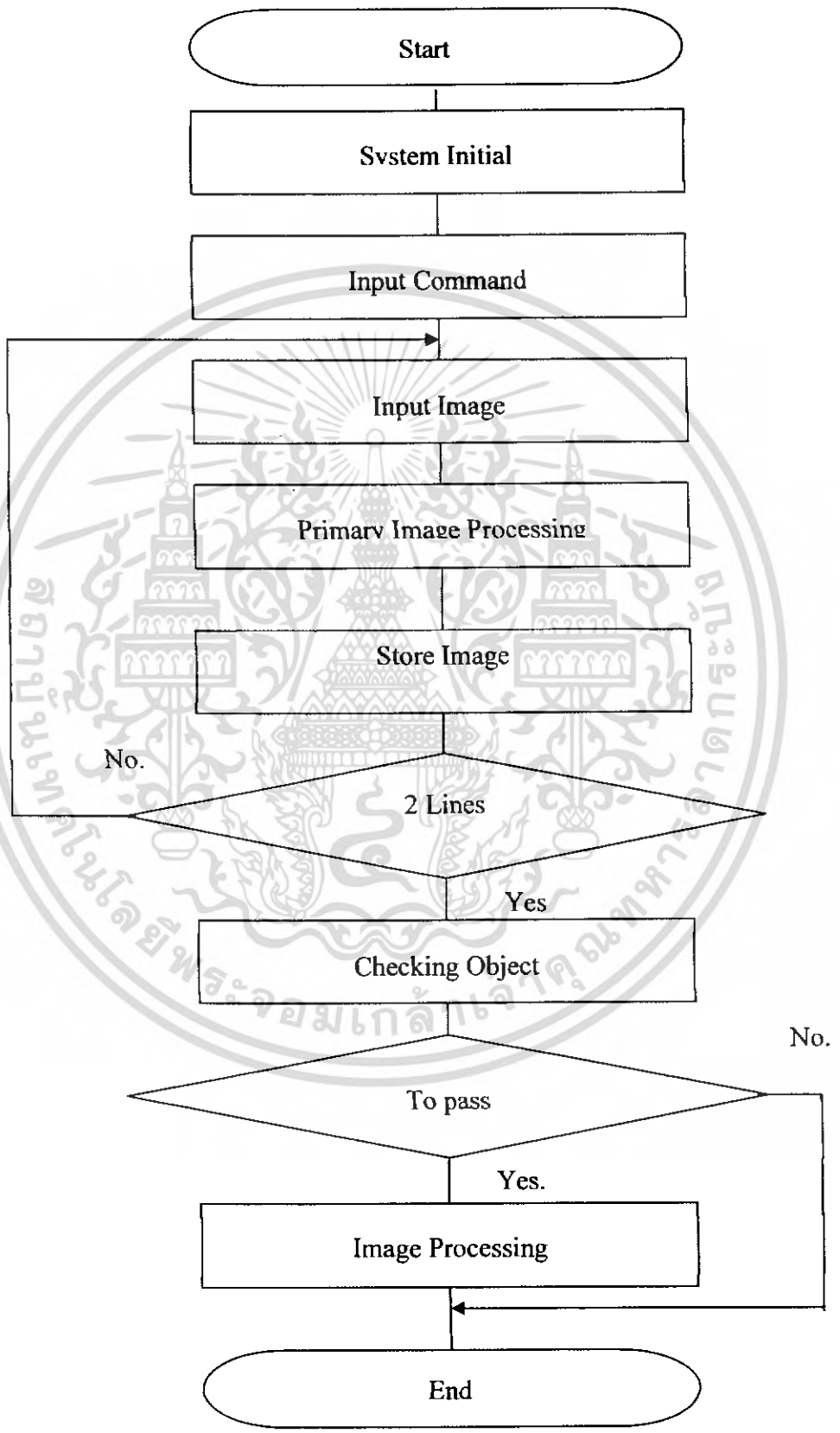
รูปที่ 2.4 แสดงเส้นเลเซอร์ที่ถูกส่งจากแหล่งกำเนิดไปกระทบวัตถุ



รูปที่ 2.5 แสดงโครงสร้างของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 หลักการของระบบโดยรวม



รูปที่ 2.7 แสดงหลักการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 หลักการของระบบ

2.2.1 System Intitial เป็นการทำงานของซอฟแวร์ คือจะติดต่อกับไมโครคอนโทรเลอร์ ที่ควบคุมรถทดสอบ

2.2.2 Input Command ไมโครคอนโทรเลอร์จะได้รับคำสั่งจากภายนอกให้รถทดสอบเคลื่อนที่ และส่งค่าความเร็วในการเคลื่อนที่ให้กับคอมพิวเตอร์แบบพกพา เพื่อใช้ในการประมวลผล

2.2.3 Input Image เป็นการรับภาพผ่านกล้องเว็บแคม ซึ่งถูกควบคุมด้วยซอฟแวร์ (Soft Ware) ภาพที่ได้จะเป็นแนวเส้นสแกนลำแสงเลเซอร์ และสิ่งแวดล้อมบนพื้นถนน ซึ่งจะมองเห็นเส้นลำแสงเลเซอร์ชัดเจนที่สุด แล้วนำภาพที่ได้ส่งต่อไปในส่วนถัดไป

2.2.4 Primary Image Processing เป็นการนำภาพที่ได้จาก Input Image ประมวลผลเพื่อเลือกเอาส่วนที่ใช้งานไว้ (ซึ่งต้องการแนวเส้นลำแสงเลเซอร์เท่านั้น) และกำจัดส่วนของภาพที่ไม่ต้องการออกไป เช่น สิ่งแวดล้อมบนพื้นถนน สัญญาณรบกวน หรือรายละเอียดของภาพที่ไม่ต้องการ

2.2.5 Store Image นำภาพแนวเส้นลำแสงที่ผ่านการทำ Primary Image Processing ในทุกๆ วงรอบของการสแกนซึ่งกำหนดจากความเร็วของรถยนต์ มารวมกันและนำไปสร้างเป็นภาพอย่างน้อย 2 แนวเส้นลำแสงเลเซอร์

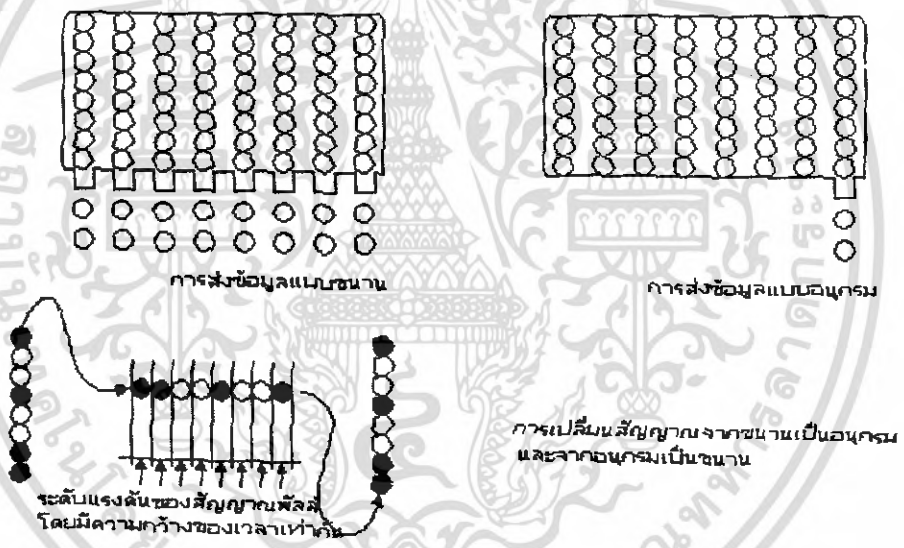
2.2.6 Chacking Object นำภาพ 3 มิติ ที่ผ่านการทำ Store Image มาตรวจสอบความสูง , ความกว้าง และพิจารณาว่ารถยนต์สามารถเคลื่อนที่ผ่านได้หรือไม่ หรือถ้าไม่ได้ ยังมีทางเลือกอื่นหรือไม่ ซึ่งจะหาจุดของแนวเส้นลำแสงเลเซอร์ที่สูงที่สุดที่รถยนต์ผ่านได้ ถ้าสูงเกินไปหรือไม่มีทางเลือกอื่นก็จะสั่งรถหยุดคลื่อนที่ หรือพิจารณาแล้วไม่มีสิ่งกีดขวางหรือมี แต่รถยนต์สามารถผ่านได้

2.2.7 Image Processing นำภาพ ที่ผ่านการทำ Store Image และพิจารณาแล้วรถยนต์สามารถเคลื่อนที่ผ่านได้ มาประกอบเป็นรูป 3 มิติ และเก็บข้อมูลของเส้นลำแสงเลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีการสื่อสารแบบอนุกรม

ข้อมูลในไมโครคอนโทรลเลอร์ที่ใช้งานนี้ จะเป็นข้อมูลที่มีความยาวขนาด 1 ไบต์ หรือ 8 บิต ซึ่งโดยปกติถ้าเราจะให้ส่งข้อมูลพร้อมๆกันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดงได้ดังรูป 3.1 จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนานจึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสาย และระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง



รูปที่ 3.1 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม

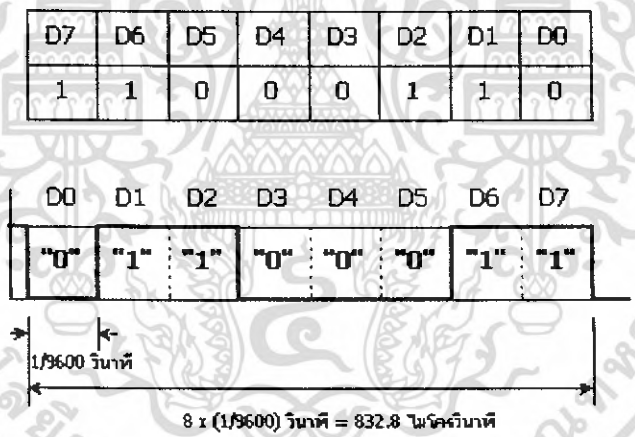
ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูลหรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง(TxD) 1 เส้น สายรับ(RxD) 1 เส้น และสายกราวด์ร่วม(Ground) 1 เส้น) นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ดังในรูป 1ข ถ้าหากต้องการส่งข้อมูลขนาด 8 บิต ก็จะทำการส่งข้อมูลออกไปทีละบิตเป็นลำดับไปจนกว่าจะครบจำนวนทั้ง 8 บิต ดังในรูป 3.1 จะแสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลจะถูกส่งไปตามสายสัญญาณที่ละบิตตามจังหวะเวลาที่กำหนด เป็นความกว้างของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐานของฝ่ายส่งและฝ่ายรับด้วย ในการรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ส่งมาทีละบิด จะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "1" หรือ "0" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม

3.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม เพื่อรับหรือส่งข้อมูล จะเป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวนบิดในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด(Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1.200, 2,400, 4,800 และ 9,600 บิตต่อวินาที ในรูป 3.2 ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ 8 x 104.1 หรือ 832.8 ไมโครวินาที



รูป 3.2 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

3.2 รูปแบบของการสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส เป็นวิธีการรับและส่งข้อมูลโดยไม่ต้องอาศัยสัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้นแสดงดังรูปที่ 3.3 ซึ่งประกอบด้วยกัน 4 ส่วนคือ

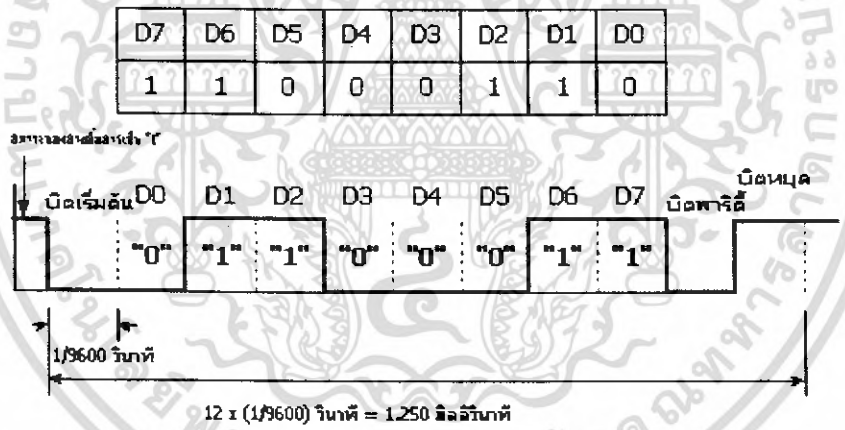
3.2.1 บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต จะเป็นละดับลอจิกตรงกันข้ามกับระดับลอจิกของสภาวะสายสื่อสาร ขณะที่ยังไม่มีการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 บิตข้อมูล (Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือบิต LSB ก่อนโดยข้อมูลที่จะส่งอาจจะมีขนาด 5 ,6, 7 หรือ 8 บิตก็ได้

3.2.3 บิตแสดงสภาวะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิตโดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1" โดยเลือกการส่งข้อมูลเป็นแบบพาริตี้คู่ หรือ พาริตี้คี่ ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตี้คู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะให้บิตพาริตี้นี้เป็น "1"เพื่อจะได้จำนวนเลข "1" เป็นคู่นั้นเอง ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น "1" รวมทั้งบิตพาริตี้ 1 บิต ถ้ามีค่า "1" เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง * สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตี้บิตก็ได้

3.2.4 บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีจำนวนมากกว่าหนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต



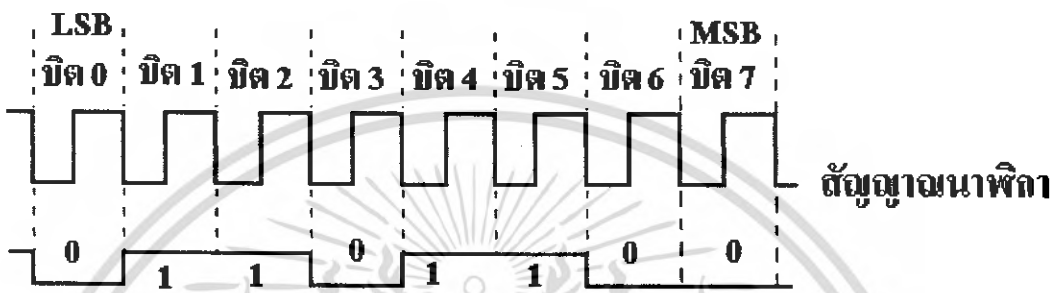
รูป 3.3 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น,บิตพาริตี้,บิตหยุด ด้วยความเร็ว 9600 บิตต่อวินาที

การสื่อสารแบบอนุกรมคือการสื่อสารระหว่างกันโดยใช้หลักการรับ-ส่งข้อมูล ครั้งละ 1 บิต แต่สามารถรับ – ส่งข้อมูลได้คราวละหลายๆ บิต จะแบ่งออกได้เป็น 2 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสื่อสารข้อมูลอนุกรมแบบซิงโครนัส

การสื่อสารข้อมูลอนุกรมแบบซิงโครนัส คือการรับ – ส่งข้อมูลโดยจะมีสัญญาณนาฬิการวมอยู่กับการรับ – ส่งสัญญาณด้วย การสื่อสารแบบนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อย 3 เส้น คือ สัญญาณนาฬิกา, สัญญาณข้อมูล และกราวด์



รูปที่ 3.4 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส

3.4 การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส

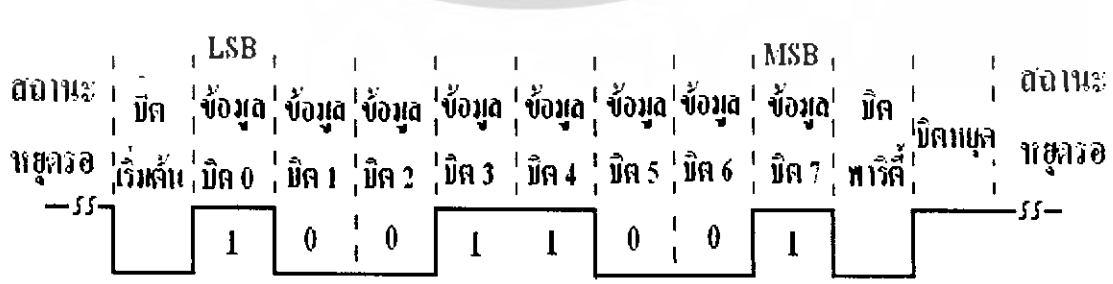
การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส คือ การรับ – ส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้การกำหนดค่าสัญญาณนาฬิการทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าภาครับและภาคส่งว่า อัตราการถ่ายถอดข้อมูล หรือ บอดเรต (Baudrate) มีหน่วยเป็น บิตต่อวินาที (Bit per Secoun : bps) รูปแบบของข้อมูลที่ใช้ในการรับ – ส่งแบบอะซิงโครนัส ประกอบด้วย 4 ส่วน คือ

3.4.1 บิตเริ่มต้น (Start Bit) มีขนาด 1 bit

3.4.2 บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 และ 8 บิต

3.4.3 บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มีก็ได้

3.4.4 บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต



รูปที่ 3.5 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขาสัญญาณข้อมูลจะมีสถานะลอจิก " 1 " ซึ่งเรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting State) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาสัญญาณข้อมูลมีลอจิก " 0 " ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิตร 5,6,7 และ 8 บิตก็ได้ (MSB) จากนั้นตามด้วย บิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตสุดท้ายคือบิตปิดท้าย ซึ่งจะให้ขาสัญญาณข้อมูลมีลอจิก " 1 " อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2.0 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (Odd) , แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก " 1 " ภายในข้อมูลที่ส่งไป 1 ไบต์ มีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย เช่น ต้องการส่งข้อมูล "10011001" จำนวนลอจิกที่เป็น " 1 " มีจำนวน 4 ตัว ซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าในบิตพาริตี้จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ค่าที่บิตพาริตี้จะต้องเป็น " 1 " เพื่อให้ข้อมูล 1 ไบท์ รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" รวมกันเป็นเลขคี่ ในตารางที่ 3.1 แสดงตัวอย่างของบิตพาริตี้ในการรับ – ส่งข้อมูลอนุกรม

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คดี้
00000000	0	1
00000001	1	0
00010110	1	0
10010110	0	1
11111110	1	0
11111111	0	1

ตารางที่ 3.1 แสดงตัวอย่างของบิตพาริตี้ในการรับ – ส่งข้อมูลอนุกรม

เพื่อให้การรับส่งข้อมูลแบบอนุกรมสามารถทำได้ที่ระยะไกลมากขึ้นระดับแรงดันจะถูกแปลงเป็นแรงดันที่สูงขึ้น โดยลอจิก " 0 " มีระดับแรงดัน +3V ถึง +12V ในขณะที่ลอจิก " 1 " มีระดับแรงดัน -3V ถึง -12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

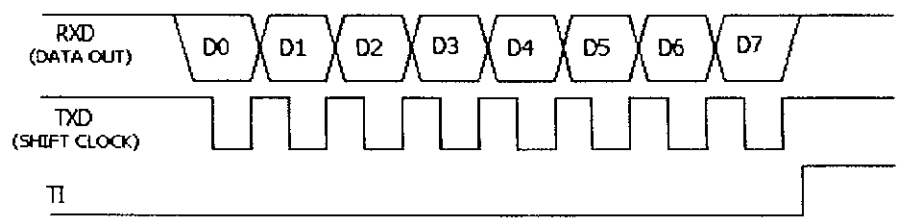
3.5 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะมีโครงสร้างเป็นแบบฟูลดูเพล็ก ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้ โดยจะมีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการ โดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต เพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้ว จึงจะทำการเซตบิตโดยกำหนดค่าของแฟลก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไบต์ต่อไปได้ การรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้น โดยการกำหนดค่าของบิต REN ที่อยู่ในในรีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ " 1 " หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอก ก็จะทำการเลื่อนข้อมูลไปโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ที่รีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ"1" ซึ่งส่งผลให้เกิดการอินเตอร์รัพท์โปรแกรมขึ้น

3.6 โหมดการการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

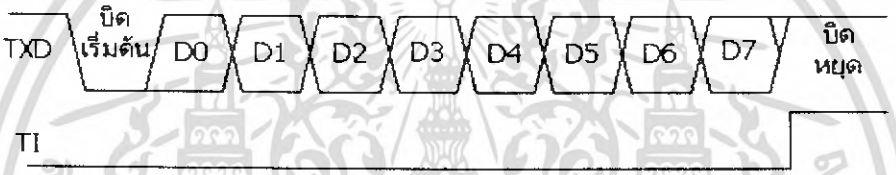
โหมด 0 : จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียว และจะไม่มีการส่งบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อน ข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิฟท์รีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุต ให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน เพราะไอซี ไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่นั้น มีจำนวนพอร์ตที่มากพอและมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



รูป 3.6 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

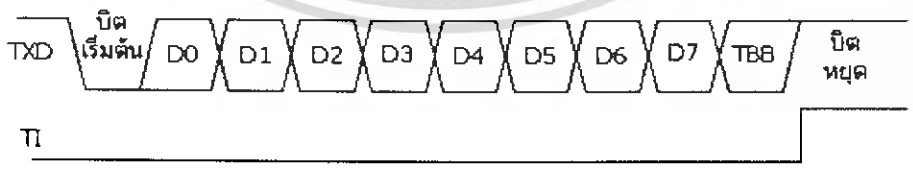
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 1 : จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ



รูป 3.7แสดงสัญญาณการส่งข้อมูลในโหมด 1

โหมด 2 : จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วย บิตแรกจะเป็นบิตเริ่มต้น (Start bit ค่า 0) , 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต (Stop bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit)



รูป 3.8 แสดงสัญญาณการส่งข้อมูลในโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 3 : เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วใน การรับและส่งข้อมูลได้ตามต้องการ

*** UART (Universal Asynchronous Receiver Transmitters) : เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่งและด้านรับ เป็นการส่งข้อมูลโดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป (Start bit ,Stop bit ,Parity bit) เพื่อให้การรับ และการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

การสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับและส่งข้อมูลทางอนุกรมมีอยู่ด้วยกัน 2 ระบบคือ

1) Single Processor System คือระบบการสื่อสารโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมต่อหากัน

2) Multiprocessors System คือระบบการสื่อสารแบบมัลติโปรเซสเซอร์ โดยใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นตัวแม่ (Master) และสามารถที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นตัวลูก (Slave)ได้อีกเป็นจำนวนหลายๆ ตัว

3.7 รีจิสเตอร์ที่ใช้ในการควบคุม และรับส่งข้อมูลของพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้งานในการติต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะประกอบด้วยรีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และรีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับ-ส่ง โดยรีจิสเตอร์แต่ละตัวจะมีหน้าที่ และการทำงานในแต่ละบิตดังต่อไปนี้

3.7.1 รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ (Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 (ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล(TB8) รายละเอียดของแต่ละบิตมีดังต่อไปนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

รูป 3.9 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0, SM1: (Serial port mode bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด

SM0	SM1	โหมด การทำงาน อัตรารับ-ส่ง
0	0	0 Shift register f osc /12
0	1	1 8 bit UART Variable
1	0	2 9 bit UART fosc/32 หรือ fosc/64
1	1	3 9 bit UART Variable

รูป 3.10 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน

SM2 : เป็นบิตที่ทำหน้าที่ควบคุมการทำงานและเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ แบบ Single Processor System หรือ Multi processors System โดยกำหนดให้

SM2 = 1 เป็นการเลือกแบบ Multi processors System คือระบบการสื่อสารแบบใช้ซีพียูหลายๆตัวร่วมกันทำงาน จะใช้งานในโหมด 2 หรือโหมด 3

SM2 = 0 เป็นการเลือกแบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานในโหมด 0 ต้องกำหนดให้ SM2 = 0)

ในกรณีที่เลือกให้ SM2 = 1 แบบ Multi processors System ถ้าข้อมูลที่รับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น "1" ทำให้แฟลกอินเตอร์รัพต์ทางด้านรับ จะถูกเซตให้เป็น 1 (RI = 1) แต่ถ้าข้อมูลในบิตที่ 9 รับเข้ามามีค่าเป็น "0" จะทำให้แฟลกอินเตอร์รัพต์ทางด้านรับเป็น 0 (RI = 0) การทำงาน ในโหมด 1 ถ้าให้ SM2 = 1 แฟลกอินเตอร์รัพต์ทางด้านรับ (แฟลก RI) จะไม่ถูกเซต หากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop bit)

REN : (Enable Serial Reception)เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์

1 = ให้มีการรับข้อมูล

0 = ไม่ให้มีการรับข้อมูล

TB8 : (Transmit bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RB8 : (Receive bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการรับข้อมูล โดยใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้จะเป็นค่าของ Stop Bit ที่รับเข้ามา สำหรับโหมด 0 จะไม่ใช้งานบิตนี้

TI : (Transmit Interrupt Flag)เป็นบิตที่ใช้งานในการอินเตอร์รัพท์ด้านส่งข้อมูล และจะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 (Shift register) หรือเมื่อเริ่มต้นส่ง Stop bit ในโหมด 1,2 หรือ 3 และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ ของการส่งข้อมูลเรียบร้อยแล้ว

RI : (Receive Interrupt Flag)เป็นบิตที่ใช้งานในการอินเตอร์รัพท์ทางด้านรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ในโหมด 0 (Shift register) และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัพท์ของการรับข้อมูลเรียบร้อยแล้ว หรืออาจกล่าวได้ว่าถ้าบิต RI ถูกเซต เมื่อใด หมายถึงข้อมูลได้เข้ามาเก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้ว สามารถที่จะอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

3.7.2 รีจิสเตอร์ SBUF (serial data buffer register) เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 99H และเข้าถึงข้อมูลแบบไบต์ได้อย่างเดียว จะทำหน้าที่รับและส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ในการอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะต้องอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งเป็นบัปเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้ามาได้จากภายนอก และในทำนองเดียวกันขณะที่ต้องการส่งข้อมูล เราก็จะนำเอาค่าข้อมูลที่ส่งออกไปไว้ที่ในรีจิสเตอร์ SBUFก่อน และหลังจากนั้นจึงจะส่งออกไป โดยจะใช้คำสั่งการโอนย้ายข้อมูลแบบไบต์เช่น MOV SBUF,#20H หรือ MOV SBUF,@R1 ก็ได้ การรับข้อมูลในโหมด 0 จะเริ่มต้นรับ เมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start bit เข้ามาที่ขา RXD

3.7.3 รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียวเท่านั้น โดยจะประกอบด้วยบิตดังต่อไปนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

รูป 3.11 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

PC0N.7 SMOD : ในกรณีที่ใช้ไทม์เมอร์ 1 เป็นตัวกำหนดอัตรารับส่ง (Baud rate) และหากกำหนดให้บิตนี้มีค่าเป็น "0" ในการใช้งานกับพอร์ตสื่อสารอนุกรมโหมด 1 ,2 และโหมด 3 ค่าอัตรารับส่ง (Baud rate) จะเพิ่มขึ้นเป็นสองเท่า รีจิสเตอร์ PCON ไม่สามารถอ้างตำแหน่งแบบบิตได้ แต่จะใช้คำสั่งทางลอจิกของการ OR เช่น ORL PCON,#80H จะเป็นการเซตบิตที่ 7 ของรีจิสเตอร์ PCON และการกำหนดให้บิตมีสถานะเป็น "0"หรือเคลียร์บิตจะใช้การ AND เช่น ANL PCON,#01111111B จะเป็นการเคลียร์บิตที่ 7 ของรีจิสเตอร์ PCON

3.8 มาตรฐานพอร์ตอนุกรมแบบ RS – 232

มาตรฐานพอร์ตอนุกรม แบบ RS – 232 เป็นมาตรฐานอุสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS – 232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อจะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดย สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่เรียกว่า EIA RS – 232 มาตรฐานนี้ใช้คอนเน็คเตอร์เป็นแบบ DB – 25 และ DB – 9 โดยกำหนดความยาวสูงสุดไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ – 3V ถึง – 12V แสดงว่ามีข้อมูล (Mask) และ +3V ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

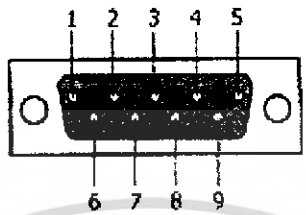
มาตรฐาน RS – 232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรปลายทาง (Data Circuit Tirminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ (Microcomputer) ซึ่งมีความสามารถในการสร้างข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS – 232 ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็คเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็คเตอร์ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้งานกันอยู่ทั่วไปเป็นแบบ DTE ส่วนคอนเน็คเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS – 232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาท์ โดยสามารถรับ – ส่งข้อมูลได้ที่ความยาว 20 เมตร

3.9 คอนเน็คเตอร์สำหรับพอร์ต RS – 232 และการเชื่อมต่อ

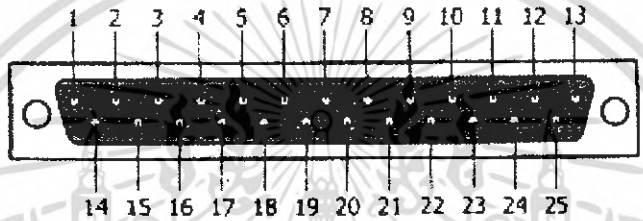
มาตรฐานการเชื่อมต่อแบบ RS – 232 จะต้องใช้คอนเน็คเตอร์แบบ DB – 25 ตัวผู้หรือ DB – 9 ตัวผู้ ซึ่งคอนเน็คเตอร์แบบ DB – 25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็คเตอร์แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62367

DB – 9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยสรุปรูปร่างและตำแหน่งขาในรูปที่ 3.11 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS – 232 ทั้งแบบ DB – 9 และ DB – 25



(ก) คอนเน็กเตอร์อนุกรม 9 ขาหรือDB-9 (มองจากด้านหลังของคอมพิวเตอร์)



(ข) คอนเน็กเตอร์อนุกรม 25 ขา หรือ แบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

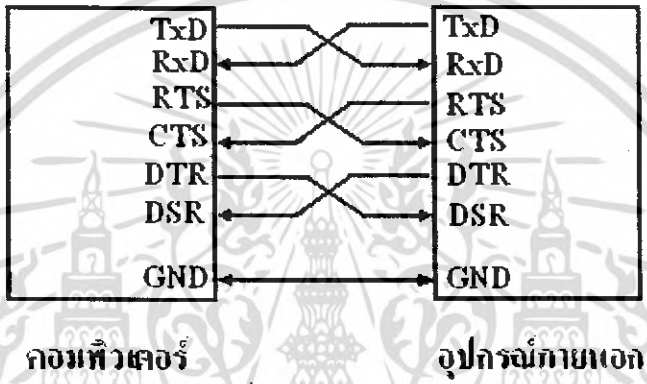
รูปที่ 3.12 การจัดขาของคอนเน็คเตอร์พอร์ตออนุกรมตามมาตรฐาน RS – 232 ทั้งแบบ DB – 9 และ DB – 25

คอนเน็กเตอร์ DB – 9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect:DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

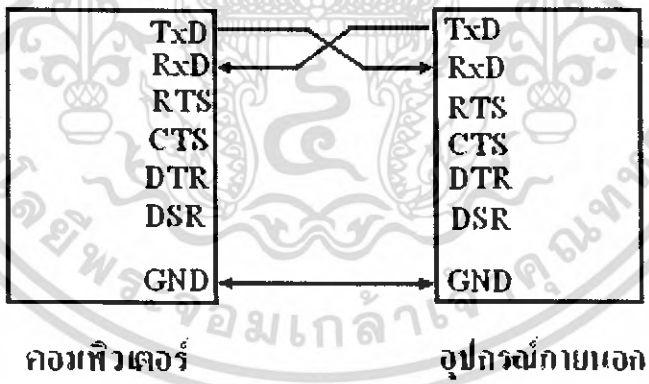
ตารางที่ 3.13 การจัดขาของคอนเน็คเตอร์พอร์ตออนุกรมตามมาตรฐาน RS – 232 ทั้งแบบ DB – 9 และ DB – 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงในรูปที่ 3.14 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ ในลักษณะต่างๆ ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 3.14 (ก) เป็นการเชื่อมต่อแบบ Null Modem หรือการเชื่อมโดยตรงไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเด็มรูปแบบ ส่วนในรูปที่ 3.14 (ข) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ NULL MODEM



(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS – 232 โดยใช้สัญญาณเพียงสามเส้น

รูปที่ 3.14 การเชื่อมต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 รายละเอียดหน้าที่การทำงานในแต่ละขอของพอร์ตอนุกรม RS – 232 มีดังนี้

3.10.1 Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : DCD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล สำหรับการใช้งานปกติ ขานี้จะไม่ถูกใช้งานมากนัก

3.10.2 Receive Dtat : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจีสเตอร์ บัฟเฟอร์

3.10.3 Transmitted Data : TD หรือ TxD ใช้ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

3.10.4 Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้ต้องเชื่อมต่อกับขอ DSR ของอุปกรณ์ปลายทาง และขอ DTR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขอ DSR ของคอมพิวเตอร์ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และจะต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

3.10.5 Signal Ground : GND เป็นกราวด์ของระบบ

3.10.6 Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกณ์ปลายทาง ซึ่งขา DSR นี้ จะเป็นขาสำหรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

3.10.7 Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่มีการเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและการส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

3.10.8 Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้น ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

3.10.9 Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ปกติในการสื่อสาร โดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรม มี 4 ตำแหน่ง คือ

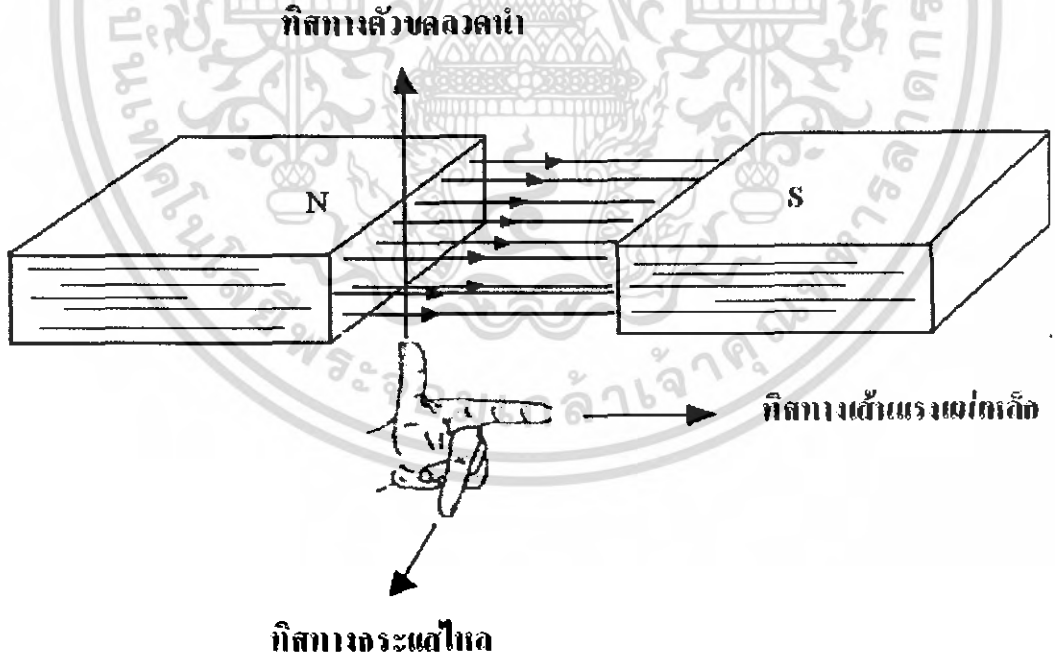
- COM 1 : 3F8H
- COM 2 : 2F8H
- COM 3 : 3E8H
- COM 4 : 2E8H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทฤษฎีพื้นฐานและหลักการทำงานของมอเตอร์กระแสตรง และหลักการทำงานของมอเตอร์กระแสตรงชนิดเฟืองทด

4.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

เมื่อมีลวดตัวนำหมุนตัดกับสนามแม่เหล็กจะทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำในลวดตัวนำนั้น และทิศทางการไหลของกระแสในลวดตัวนำจะไหลในทิศทางที่ต้านการหมุนของลวดตัวนำนั้น ถ้าทราบชนิดการไหลของกระแสไฟฟ้าก็จะพบความสัมพันธ์ ดังนี้คือ ถ้าทราบทิศทางของเส้นแรงแม่เหล็ก ทิศทางการไหลของกระแสในตัวนำ จะทำให้ทราบทิศทางการหมุนของขดลวดตัวนำได้ การค้นพบนี้เรียกว่ากฎมือซ้ายของเฟลมมิ่ง ความสัมพันธ์สามารถหาได้โดยการใช้มือซ้าย กางหัวแม่มือ นิ้วชี้และนิ้วกลาง ให้ตั้งฉากซึ่งกันและกัน ให้นิ้วหัวแม่มือชี้ทิศทางการเคลื่อนที่ของตัวนำ นิ้วชี้ชี้ทิศทางของเส้นแรงแม่เหล็กและนิ้วกลางชี้ทิศทางของกระแสไฟฟ้าที่ไหลในตัวนำ ดังรูปที่ 5.1 แสดงกฎมือซ้ายของเฟลมมิ่ง



รูปที่ 4.1 แสดงกฎมือซ้ายของเฟลมมิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันไฟฟ้าเหนี่ยวนำจะเกิดขึ้นในทิศทางตรงกันข้ามกับแรงดันไฟฟ้าที่จ่ายจากภายนอกเข้าไปในลวดตัวนำ สำหรับมอเตอร์แล้วจะเห็นว่าแรงดันไฟฟ้าเหนี่ยวนำที่เกิดขึ้นในลวดตัวนำ จะมีทิศทางตรงกันข้ามกับแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ แรงดันไฟฟ้าเหนี่ยวนำที่เกิดขึ้นนี้เรียกว่า แรงดันไฟฟ้าต่อต้านหรือแรงดันไฟฟ้าต้านกลับ (Counter e.m.f. or Back e.m.f.) ใช้สัญญาลักษณ์ Ec ซึ่งมันจะเกิดขึ้นในลวดอาร์เมเจอร์ของมอเตอร์ จึงมีค่าเท่ากับแรงดันไฟฟ้าที่จ่ายให้ขดลวดด้วยแรงดันไฟฟ้าต้านกลับ เขียนเป็นสมการได้คือ

$$IaRa = V - Eb$$

หรือ

$$Eb = V - IaRa$$

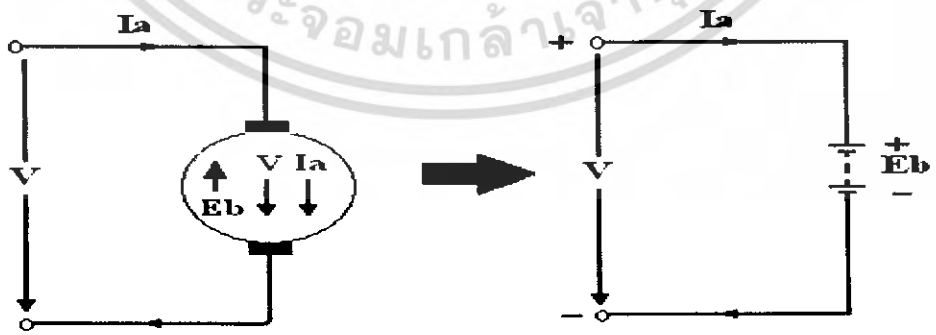
เมื่อ Eb = แรงดันไฟฟ้าต้านกลับ

V = แรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์

Ia = กระแสที่ไหลในอาร์เมเจอร์

Ra = ความต้านทานของขดลวดอาร์เมเจอร์

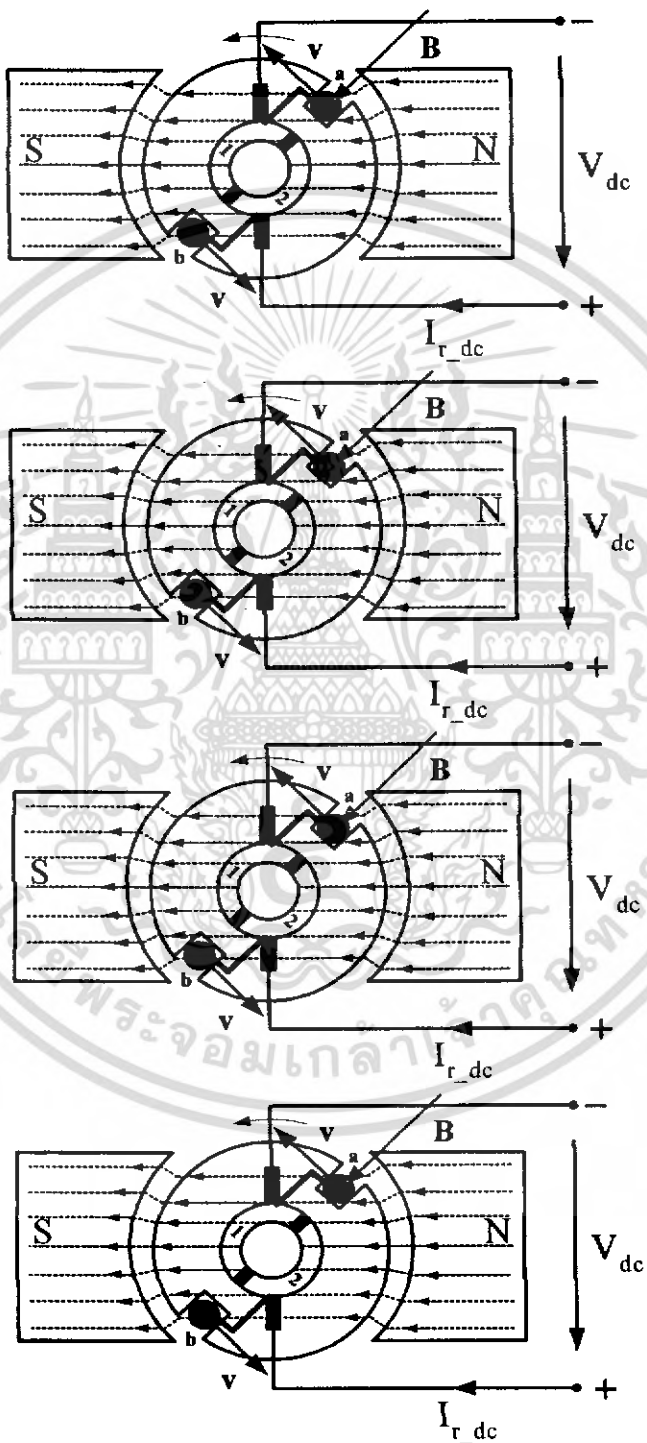
ค่าแรงดันไฟฟ้าต้านกลับนี้ จะมีค่าไม่เท่ากับแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ แรงดันไฟฟ้าต้านกลับที่เกิดขึ้นสามารถเขียนเปรียบเทียบได้เหมือนกับในอาร์เมเจอร์นั้น มีแหล่งจ่ายไฟฟ้ากระแสตรงซ่อนอยู่ และจ่ายไฟออกมาตรงกันข้ามกับแรงดันไฟที่จ่ายเข้าไป ดังรูปที่ 4.2 วงจรเทียบเคียงแรงดันไฟฟ้าต้านกลับ



รูปที่ 4.2 วงจรเทียบเคียงแรงดันไฟฟ้าต้านกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

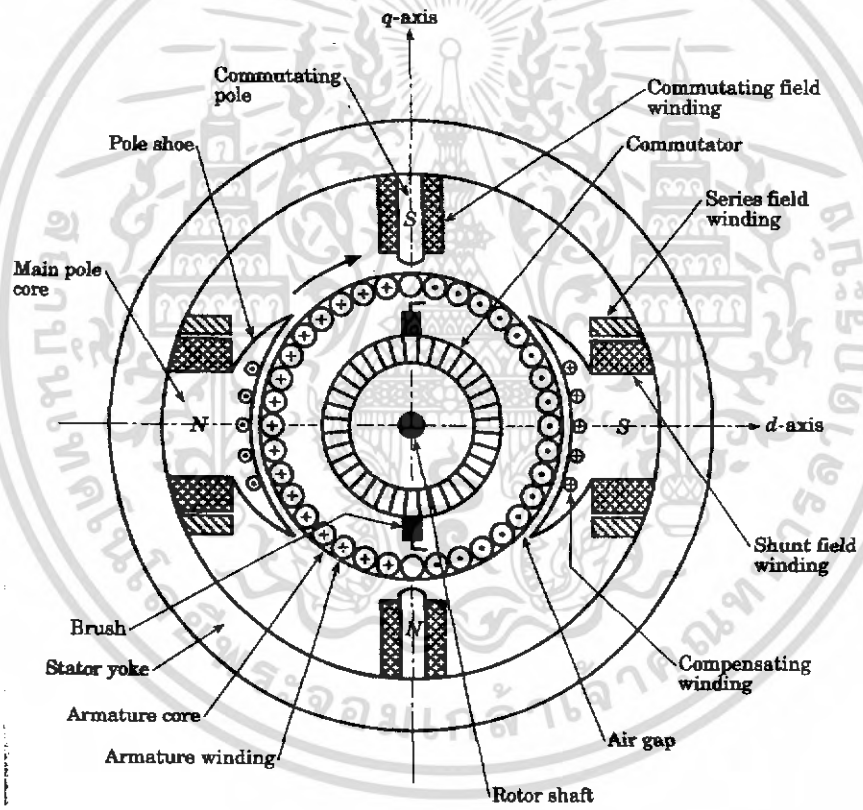
จากรูปเป็นมอเตอร์ไฟฟ้ากระแสตรงอย่างง่าย ซึ่งประกอบด้วยโครงลวดที่วางอยู่ระหว่างขั้วแม่เหล็กโดยปลายของขดลวดทั้งสองต่อเข้ากับคอมมิวเตเตอร์ด้านละซี่ ซึ่งจะมีแปรงถ่านต่อไว้และแปรงถ่านทั้งสองต่อเข้ากับแหล่งจ่ายไฟฟ้ากระแสตรงจากภายนอก



รูปที่ 4.3 แสดงมอเตอร์ไฟฟ้ากระแสตรงอย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าอาร์เมเจอร์ที่ใช้ขดลวดหลายชุด แต่ละชุดก็เกิดปฏิกิริยาเช่นเดียวกันกับขดลวด 1 ชุด ในขณะที่ตั้งฉากกับสนามแม่เหล็กจากขั้วแม่เหล็กซึ่งจะมีขั้วดังแสดงในรูป 4.4 ขั้วเหนือของสนามแม่เหล็กที่ขดลวดอาร์เมเจอร์จะดูดเข้าหาขั้วใต้ของขั้วแม่เหล็กจากขั้วแม่เหล็กหลัก แรงดึงดูดกันนี้ทำให้มีแรงผลักหมุนที่อาร์เมเจอร์ ทำให้หมุนไปในทิศทางตามเข็มนาฬิกา เกิดแรงบิดที่สม่ำเสมอราบเรียบเรียบไม่กระตุก เพราะมีขดลวดหลายๆ ชุดที่ติดกันพลัดการทำงานทีละชุดตามลำดับ การที่มีขดลวดหลายๆ ชุดเช่นนี้ จึงทำให้เกิดสนามแม่เหล็กลัพธ์ที่อาร์เมเจอร์ขึ้นค่าหนึ่ง ซึ่งเปรียบสเหมือนว่ามีค่าคงที่หรือสนามแม่เหล็กที่อาร์เมเจอร์จะอยู่ในตำแหน่งที่คงที่ในขณะที่อาร์เมเจอร์หมุนอยู่



รูปที่ 4.4 แสดงการเกิดสนามแม่เหล็กที่ขดลวดอาร์เมเจอร์

แรงที่กระทำบนขดลวดตัวนำต่อหนึ่งขดในขณะที่มีกระแสไหลผ่านตัวนำนั้น จะเป็นสัดส่วนโดยตรงกับจำนวนกระแสที่ไหล ความเข้มของสนามแม่เหล็กและความยาวของตัวนำในส่วนที่คัดผ่านสนามแม่เหล็ก แรงที่เกิดขึ้นนี้ใช้สัญญาลักษณ์ F ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F = BLI

หรือ F = แรงที่กระทำที่ขดลวดตัวนำมีหน่วยเป็นนิวตัน

B = ความหนาแน่นของเส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์ต่อตารางเมตร

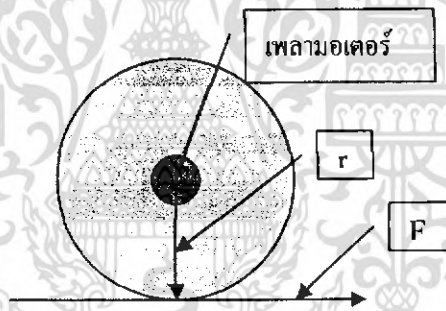
L = ความยาวของตัวนำในส่วนที่ผ่านสนามแม่เหล็กมีหน่วยเป็นเมตร

I = กระแสที่ไหลในขดลวดตัวนำมีหน่วยเป็นแอมแปร์

จากสมการจะเห็นว่า ถ้าสนามแม่เหล็กมีค่าคงที่ และความยาวของขดลวดตัวนำมีค่าคงที่ แรงผลักหรือแรงที่กระทำก็จะเป็นสัดส่วนโดยตรงกับกระแสไฟฟ้าที่ไหลในขดลวดตัวนำอย่างเดียว

4.2 แรงบิด (Torque) ในมอเตอร์กระแสตรง

เป็นการหมุนหรือการบิดของโมเมนต์ของแรงๆ หนึ่งที่กระทำรอบแกนอันหนึ่ง สามารถวัดได้จากผลคูณของแรงกับรัศมี ณ ที่แรงไปกระทำ



รูปที่ 4.5 แสดงการคำนวณแรงบิด

จากรูปพิจารณาวงล้อที่มีรัศมี r เมตร มีแรง F นิวตัน มากระทำกับวงล้อนี้ จึงทำให้เกิดการหมุนด้วยความเร็ว N รอบต่อนาที

จะได้แรงบิด T = F x r (นิวตัน – เมตร)

งานที่กระทำได้ในหนึ่งรอบจากนี้ = แรงคูณกับระยะทาง

= F x 2¶r (จูลส์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

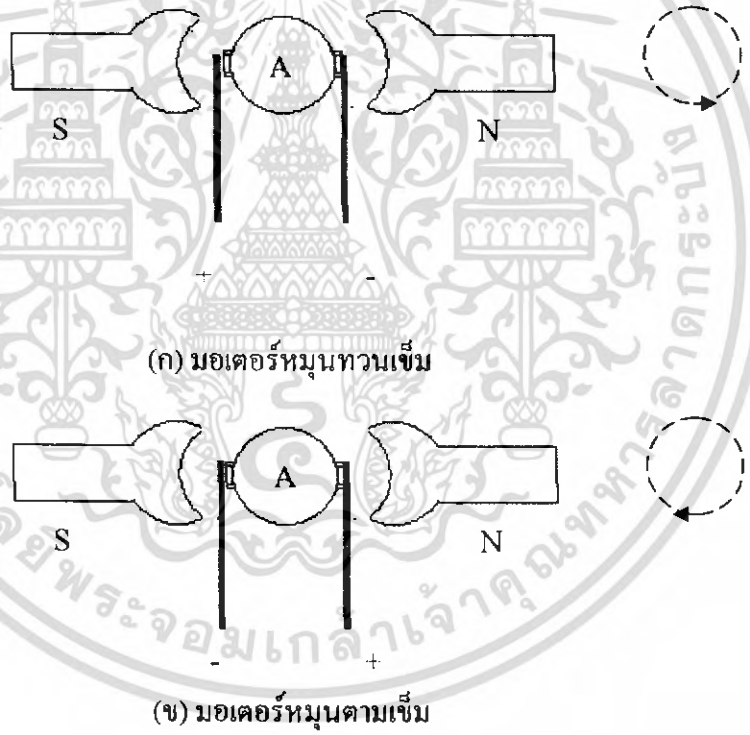
งานที่กระทำต่อวินาที W = F2¶rN ซึ่ง 2¶rN เป็นมุม θ เรเดียนต่อวินาที และ Fr เป็นแรงบิด

เพราะฉนั้น งานที่ทำต่อวินาที $W = T\theta$ (จูลส์)

4.3 วิธีการกลับทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรง

ทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรงนั้น ขึ้นอยู่กับทิศทางของทิศทางกระแสที่ไหลในอาร์เมเจอร์ โดยปกติมอเตอร์จะต้องหมุนในทิศทางใดทิศทางหนึ่ง ถ้าต้องการกลับทิศทางการหมุนสามารถทำโดยกลับขั้วต่อของอาร์เมเจอร์

ในการกลับทิศทางของมอเตอร์ นั้น จะไม่ทำให้ประสิทธิภาพ ความเร็ว แรงบิดและกระแสที่ไหลผ่านอาร์เมเจอร์เปลี่ยนแปลง



(ก) มอเตอร์หมุนทวนเข็ม

(ข) มอเตอร์หมุนตามเข็ม

รูปที่ 4.6 แสดงการกลับทิศทางการหมุนขอมอเตอร์ไฟฟ้ากระแส

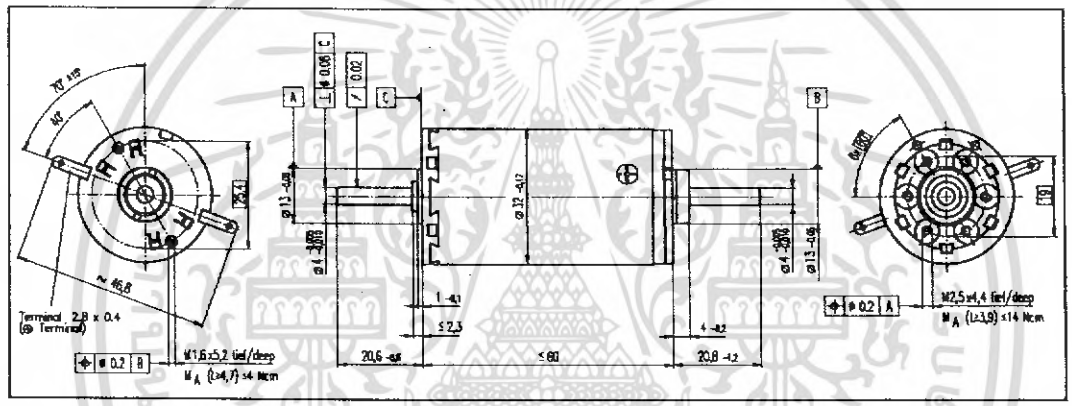
4.4 มอเตอร์ไฟฟ้ากระแสตรงชนิดทดเฟือง

มอเตอร์ไฟฟ้ากระแสตรงชนิดทดเฟือง ประกอบด้วย

- มอเตอร์ไฟฟ้ากระแสตรง
- เกียร์เฟืองทด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างไฟฟ้ากระแสตรงที่ใช้มีโครงสร้างเช่นเดียวกับมอตอร์ไฟฟ้ากระแสตรงธรรมดา แต่ในส่วนของเพลาขับของมอเตอร์จะต่อเข้าไปในชุดเกียร์เฟืองทด และมีเพลาขับที่ออกมาจากชุดเกียร์ โดยจะมีขนาดเพลาที่ใหญ่ขึ้น แต่มีความเร็วในการหมุนลดลงตามอัตราทดของชุดเกียร์เฟืองทด จุดประสงค์ของการนำมอเตอร์ไฟฟ้ากระแสตรงไปประกอบกับชุดเกียร์เป็นมอเตอร์ไฟฟ้ากระแสตรงชนิดเกียร์เฟืองทดนั้น ก็เพื่อต้องการแรงบิดที่เพิ่มขึ้นแต่มีขนาดเล็ก เหมาะสำหรับการนำไปใช้งาน แต่ก็มีข้อเสียคือ มอเตอร์ไฟฟ้ากระแสตรงชนิดทดเฟือง จะมีความเร็วในการหมุนช้าลง และเพลาขับที่เยื้องจากศูนย์กลางของมอเตอร์ ตามรูป 4.7 ลักษณะของมอเตอร์ไฟฟ้ากระแสตรงและเกียร์เฟืองทด

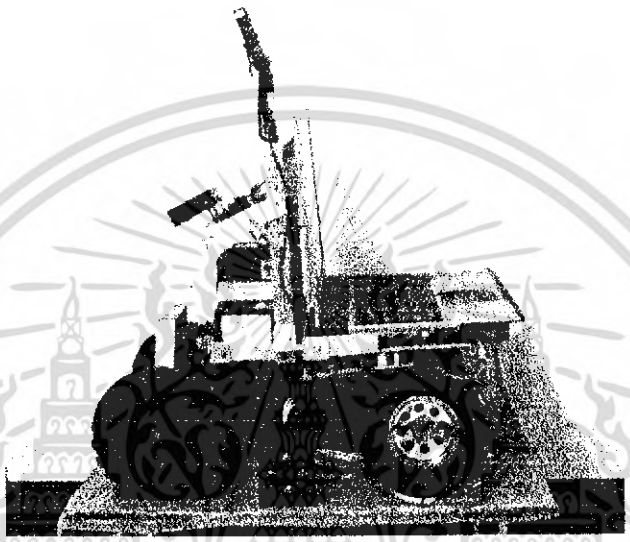


รูปที่ 4.7 ลักษณะของมอเตอร์ไฟฟ้ากระแสตรงและเกียร์เฟืองทด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
โครงสร้างและหลักการทำงานของชุดทดลองต้นแบบ

6.1 โครงสร้างและส่วนประกอบของระบบนำร่องรถยนต์



(ก) รูปด้านข้างของรถทดสอบ

(ข) รูปด้านหน้าของรถทดสอบ

รูปที่ 6.1 โครงสร้างและส่วนประกอบโดยรวมของระบบนำร่องรถยนต์

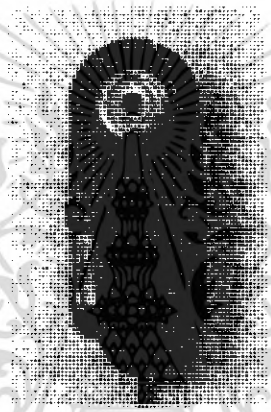
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 อุปกรณ์ที่ติดตั้งบนรถทดสอบ

6.2.1 แบตเตอร์รี่ มีจำนวน 3 ก้อน ขนาด 12V 7AH จำนวน 1 ก้อน ขนาด 12 V 1.2AH จำนวน 2 ก้อน (ต่ออนุกรม) เนื่องจากใช้ขับมอเตอร์ 24 V

6.2.2 คอมพิวเตอร์แบบพกพา มีขนาดจอ 13.5 นิ้ว น้ำหนัก 2.8 Kg รวมแบตเตอร์รี่ ซึ่งทำงานโดยใช้แบตเตอร์รี่ของเครื่อง

6.2.3 กล้องเว็บแคม



รูปที่ 6.2 กล้องเว็บแคมที่ใช้งาน

6.2.3.1 คุณสมบัติกล้องเว็บแคม

- สามารถถ่ายภาพสี และภาพนิ่งได้
- ให้ภาพที่มีความละเอียดสูง มากกว่า 640 x 480 Pixel ขนาด 32 Bit (แล้วแต่ตั้งค่าใช้งานในซอฟแวร์)
- Frame Rate มากกว่า 50 เฟรมต่อวินาที (แล้วแต่ตั้งค่าใช้งานในซอฟแวร์)
- ควบคุมการทำงานผ่านพอร์ต USB V2.0
- ใช้ไฟ DC ขนาด 5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3 การติดตั้งกล้องเว็บแคม

6.2.3.2 ความสามารถของกล้องเว็บแคม

1.) เก็บภาพเคลื่อนไหวในไมโครคอมพิวเตอร์ การใช้งานกล้องเว็บแคมในการเก็บภาพเคลื่อนไหวนั้น เป็นวิธีที่ง่ายและประหยัดวิธีหนึ่ง ในการเก็บภาพไว้ในไมโครคอมพิวเตอร์ ทำให้สามารถที่จะติดต่อสื่อสารกัน โดยแสดงภาพเคลื่อนไหวได้ และสามารถหลีกเลี่ยงปัญหาในขั้นตอนการติดตั้งฮาร์ดแวร์ ราคาของอุปกรณ์ที่แพง หรือวิธีการใช้งาน การใช้กล้องเว็บแคมจะช่วยให้เราสามารถที่จะ Capture ได้ง่าย และทำการดัดแปลงแก้ไขได้ด้วย

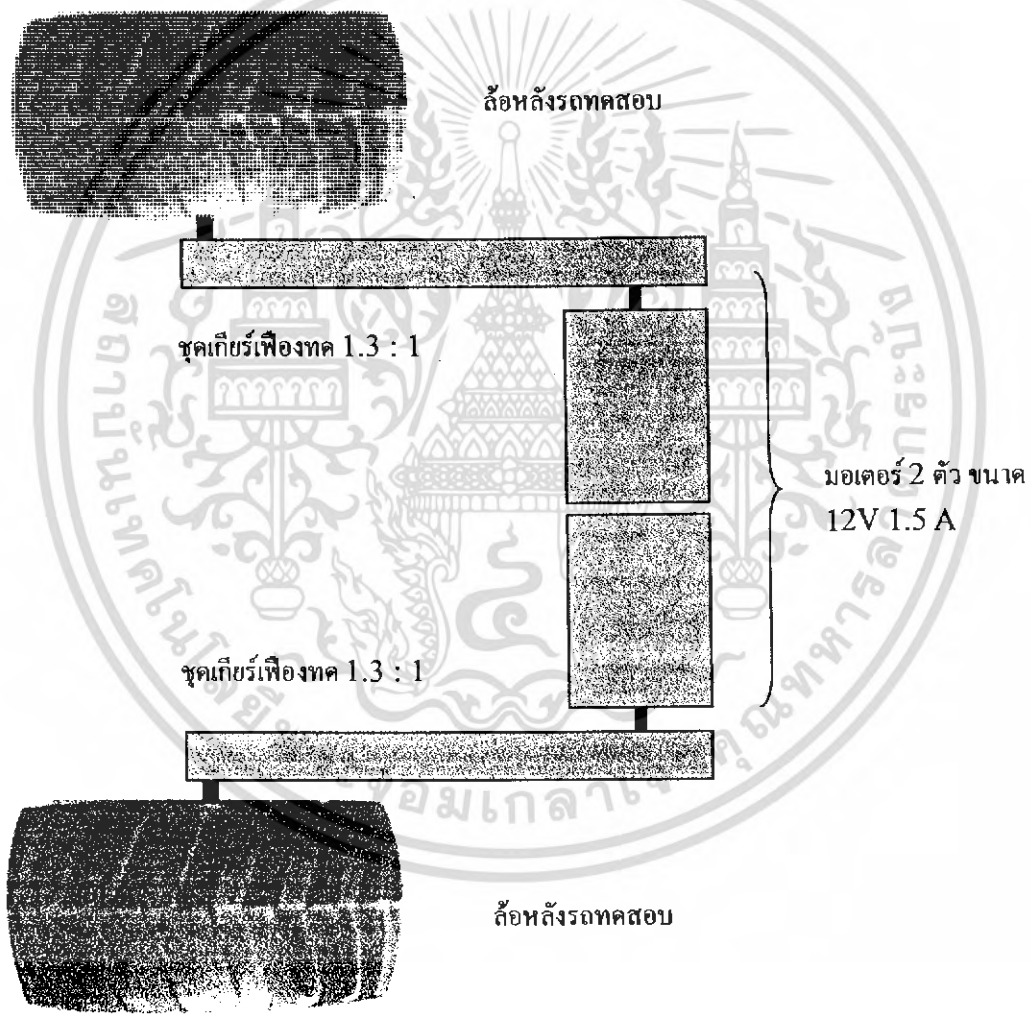
2.) เก็บภาพนิ่งในไมโครคอมพิวเตอร์ กล้องเว็บแคมมีซอร์ฟแวร์ที่ทำให้สามารถบันทึกภาพนิ่งให้อยู่ในรูปแบบ BMP, TIF หรือ JPG ซึ่งสามารถนำมาประมวลผลใน Word Processing , Page Layout , การ Present งาน หรือโปรแกรมกราฟฟิค ซึ่งภาพนิ่งเหล่านั้นสามารถนำไปใช้ในการวางตำแหน่ง, การใช้งานพิมพ์กราฟฟิค ที่ต้องการคุณภาพสูง เช่น จดหมาย หรือเอกสารสำคัญต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 ระบบส่งกำลังและระบบบังคับเลี้ยว

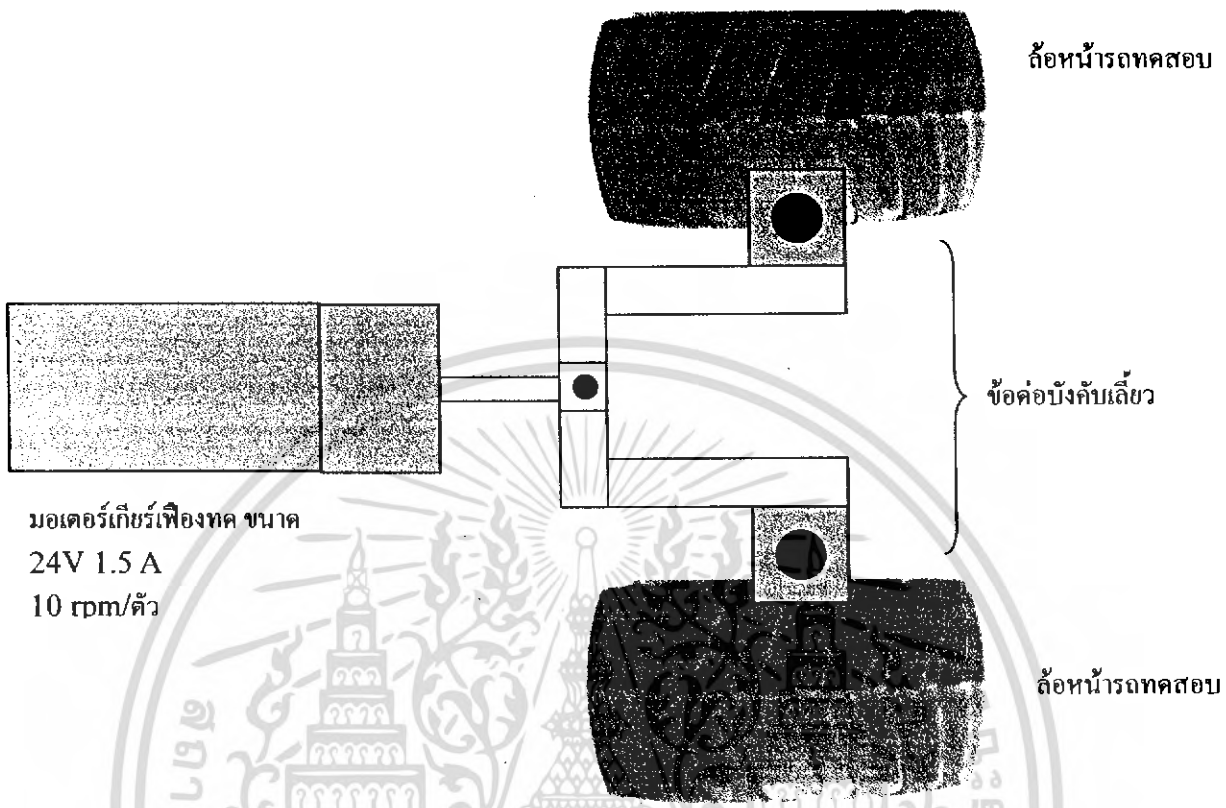
6.2.4.1 ระบบส่งกำลังในรถทดสอบ การเคลื่อนที่ไปข้างหน้าของรถยนต์ทดสอบ เกิดจากมอเตอร์ จำนวน 2 ตัว ทางด้านซ้ายและขวาของรถทดสอบ ขนาด 12V ตามรูปที่ 6.4 (ก) โดยจะผ่านระบบเฟืองทดลง 1.3 : 1.0 (จากการวัดและคำนวณ) ทำให้เกิดแรงบิดเพิ่มขึ้นและมีความเร็วต่ำลง

6.2.3.2 ระบบการบังคับเลี้ยวในรถทดสอบ เกิดจากมอเตอร์ จำนวน 1 ตัว ท้างด้านหน้าของรถทดสอบ ตามรูปที่ 6.4 (ข) ใช้มอเตอร์ ขนาด 24V 10 rpm ทำให้มีแรงบิดมากพอในการบังคับเลี้ยวที่พื้นที่ทดสอบต่างๆ



(ก) ระบบส่งกำลังในรถทดสอบ

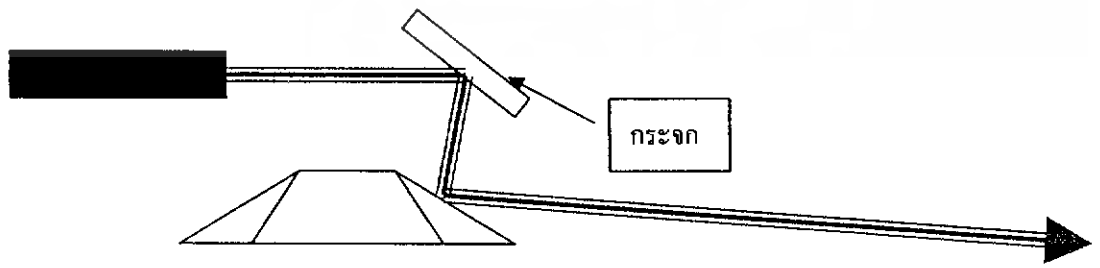
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) ระบบบังคับเลี้ยวในรถทดสอบ

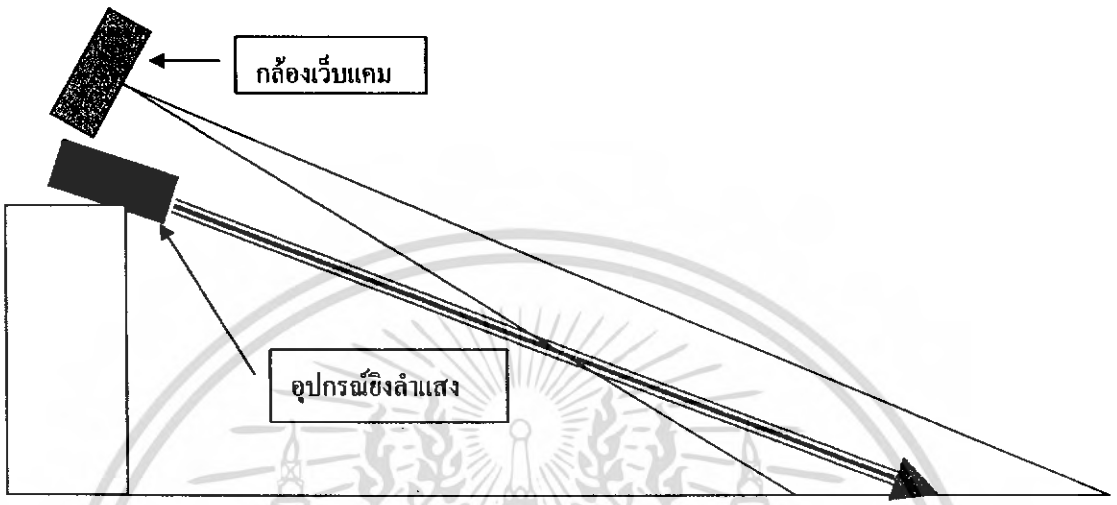
รูปที่ 6.4 โครงสร้างและส่วนประกอบการทำงานของรถทดสอบ

6.2.4 ระบบสร้างเส้นสแกนลำแสงเลเซอร์แบบเก่า (ภาคเรียนที่ 1) ประกอบจากมอเตอร์พัดลม ขนาด 1 นิ้ว 1500 rpm ยึดกับกระจกหกเหลี่ยม และมีอุปกรณ์ยิงลำแสงเลเซอร์ โดยประกอบขึ้นตามรูป 6.5



(ก) แสดงตำแหน่งการสร้างลำเส้นสแกนลำแสงเลเซอร์

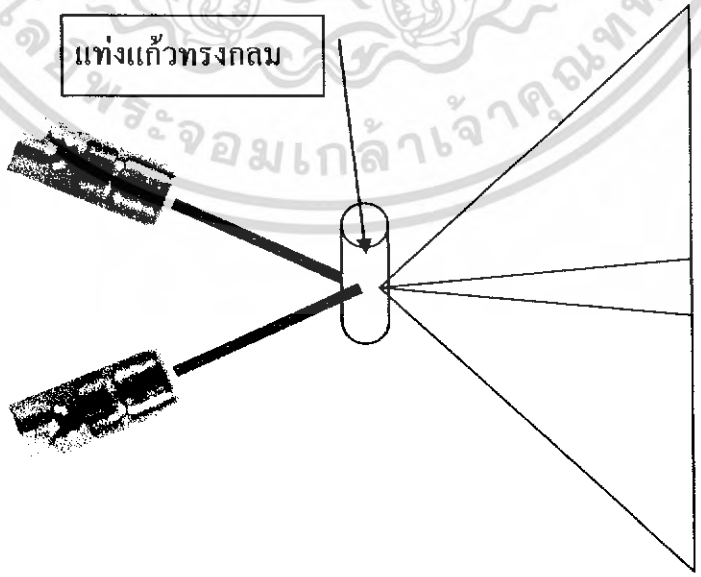
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม

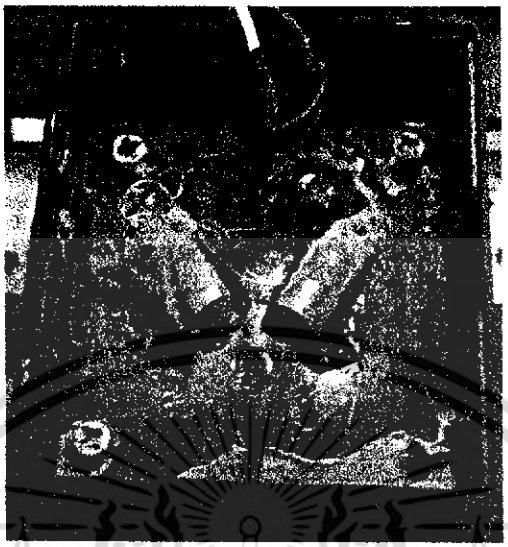
รูปที่ 6.5 โครงสร้างและส่วนประกอบของระบบสร้างเส้นสแกนลำแสงเลเซอร์แบบเก่า (ภาคเรียนที่ 1)

6.2.5 ระบบสร้างเส้นสแกนลำแสงเลเซอร์แบบใหม่ (ภาคเรียนที่ 2) เป็นอุปกรณ์ที่ใช้ติดตั้งกับรถทดสอบ ซึ่งประกอบจากแท่งแก้วใสขนาดเส้นผ่าศูนย์กลางประมาณ 1 ซม. และมีอุปกรณ์ยิงลำแสงเลเซอร์ จำนวน 2 ตัว โดยประกอบขึ้นตามรูป 6.6



(ก) แสดงตำแหน่งการสร้างลำเส้นสแกนลำแสงเลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) แสดงตำแหน่งการติดตั้งแหล่งกำเนิดลำแสงเลเซอร์และแท่งแก้ว

(ค) แสดงตำแหน่งแหล่งกำเนิดลำแสงเลเซอร์ และกล้องเว็บแคม

รูปที่ 6.6 โครงสร้างและส่วนประกอบของระบบสร้างเส้นสแกนสำแสงเลเซอร์แบบใหม่

(ภาคเรียนที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

****** ข้อดีและข้อเสีย อุปกรณ์สร้างเส้นสแกนสำแสงเลเซอร์แบบเก่าและแบบใหม่**

อุปกรณ์สร้างเส้นสแกนสำแสงเลเซอร์ที่สร้างในภาคเรียนที่ 1	ข้อเสีย • ความคมชัดน้อย • เส้นสแกนหนา (บีมกว้าง) ทำให้เกิดความผิดพลาดในการทำงานของโปรแกรมมากขึ้น • เปลืองพลังงาน ข้อดี • ความสว่างเท่ากันตลอดเส้น
อุปกรณ์สร้างเส้นสแกนสำแสงเลเซอร์ที่สร้างในภาคเรียนที่ 2	ข้อเสีย • ความสว่างของแสงไม่เท่ากันตลอดเส้น ข้อดี • มีความชัดแจนของแสงมาก • เส้นสแกนบาง (บีมแคบ) ลดความพิดพลาดของเส้นสแกน • ติดตั้งและปรับแต่งง่าย

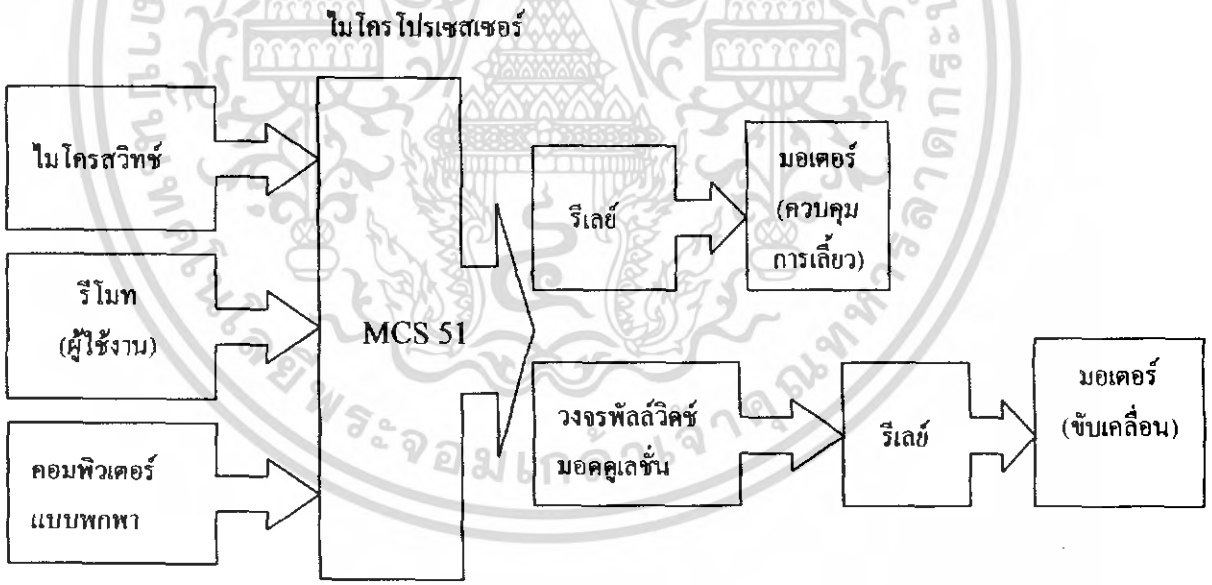
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
วงจรการทำงานและโปรแกรมการทำงานของชุดทดลองต้นแบบ

การทำงานของระบบนำร่องรถยนต์ด้วยแสงเลเซอร์ เกิดจากการทำงานร่วมกันระหว่างฮาร์แวร์และซอฟแวร์ ซึ่งฮาร์แวร์เป็นส่วนที่ติดตั้งบนรถยนต์และคอยควบคุมระบบของรถยนต์ทั้งหมด โดยได้รับคำสั่งจากรีโมทหรือคอมพิวเตอร์แบบพกพาผ่านการสื่อสารแบบ RS 232 ซึ่งจะเลือกใช้ระบบใดระหนึ่งเท่านั้น โดยผ่านการกดสวิทช์เลือกจากรถทดลอง

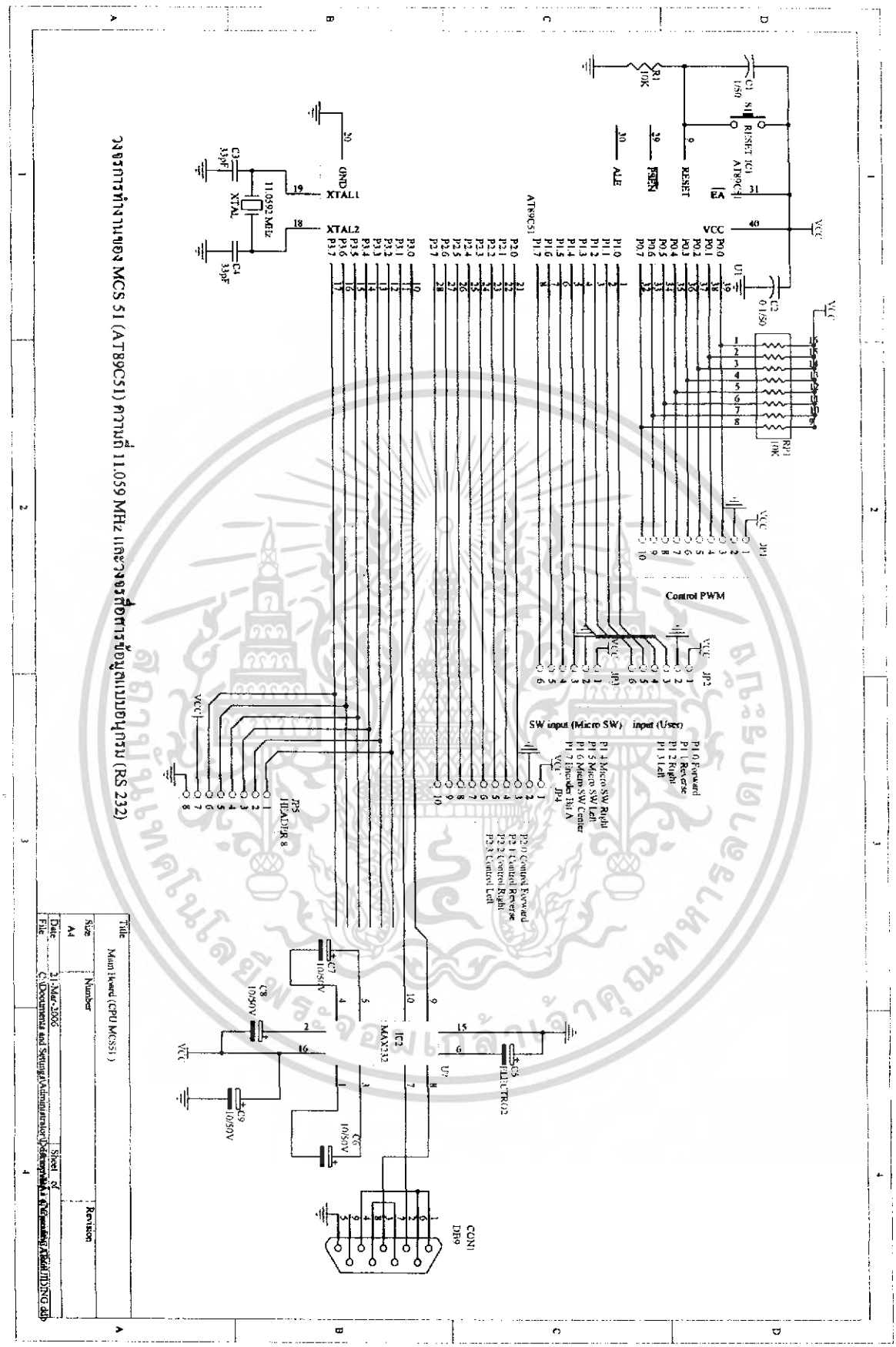
7.1 ไมโครโปรเซสเซอร์ MCS 51 (AT89C51)

ไมโครโปรเซสเซอร์เป็นอุปกรณ์สำคัญในการทำงานของรถทดสอบ โดยจะเป็นอุปกรณ์รับคำสั่งต่างๆ ตรวจสอบคำสั่งและสภาวะรถทดสอบให้สอดคล้องกับคำสั่งที่ได้รับ และควบคุมการทำงานของวงจรภาคอื่นให้สามารถทำงานได้อย่างถูกต้อง โดยมีโครงสร้างการทำงานโดยรวมดังรูปที่ 7.1 และมีวงการใช้งานตามรูปที่ 7.2



รูปที่ 7.1 บล็อกการทำงานของไมโครโปรเซสเซอร์โดยรวมของระบบทั้งหมด

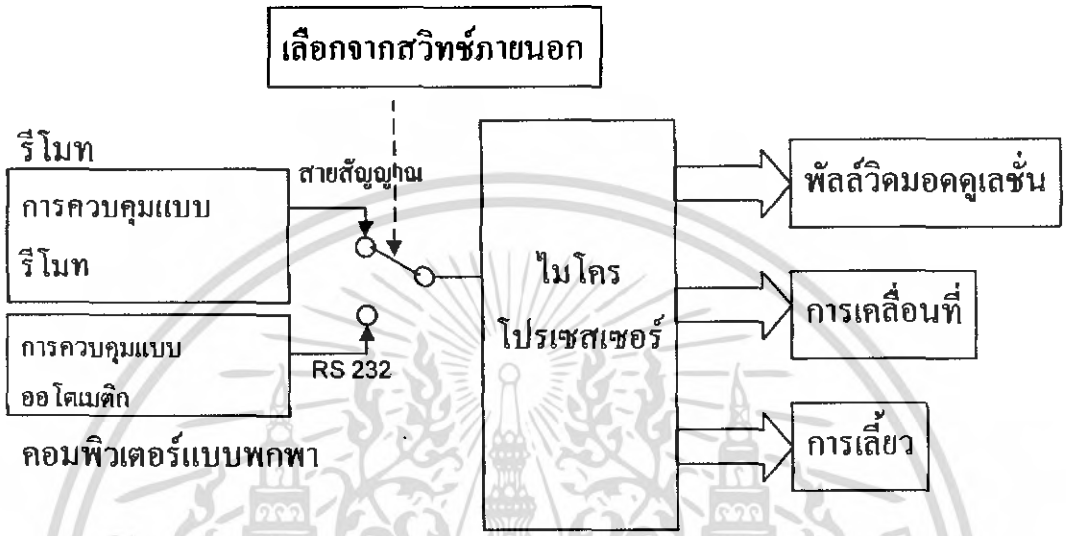
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 วงจรการใช้งานของไมโครโปรเซสเซอร์ MCS 51 (AT89C51)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของไมโครโปรเซสเซอร์แบ่งออกเป็น 2 รูปแบบ คือใช้การควบคุมแบบรีโมท จะสายในการควบคุม และการควบคุมแบบออโตเมติก จะใช้การสื่อสารแบบ RS 232 ตามรูปที่ 7.3



รูปที่ 7.3 แสดงการรับคำสั่งควบคุมของไมโครโปรเซสเซอร์

7.1.1 การควบคุมแบบรีโมท เป็นการใช้สวิทช์ จำนวน 4 ตัว เพื่อควบคุมให้รถทดสอบเคลื่อนที่เดินหน้า, ถอยหลัง, เลี้ยวซ้าย, เลี้ยวขวา และควบคุมความเร็ว สร้างขึ้นเพื่อใช้ทดลองการเคลื่อนที่ของรถทดสอบ การเก็บค่าข้อมูลต่างๆ ของรถทดสอบ และการทดลองทางซอฟแวร์บนเครื่องคอมพิวเตอร์แบบพกพา

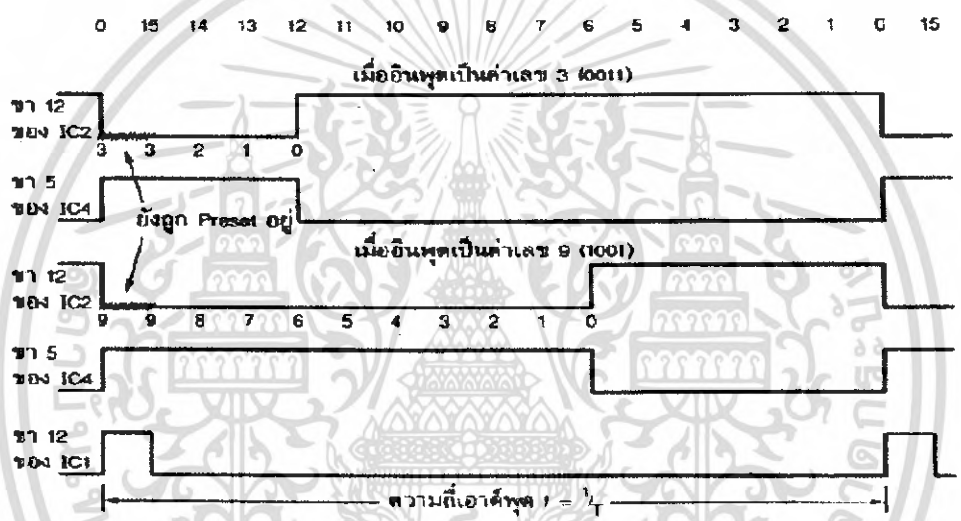
7.1.2 การควบคุมแบบออโตเมติก เป็นการรับคำสั่งจากคอมพิวเตอร์แบบพกพา ผ่านการสื่อสารแบบ RS 232 เพื่อควบคุมให้รถทดสอบเคลื่อนที่เดินหน้า, ถอยหลัง, เลี้ยวซ้าย, เลี้ยวขวา และควบคุมความเร็วสร้างขึ้นเพื่อใช้ควบคุมการเคลื่อนที่ของรถทดสอบเมื่อเจอสิ่งกีดขวางต่างๆ เป็นรูปแบบ

7.2 วงจรพัลส์วิดมอดดูเลชั่น (Pulse width Modulation)

เป็นวงจรควบคุมความเร็วของมอเตอร์โดยใช้พัลส์ กล่าวคือวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์ จำนวน 4 บิต (Binary) เพื่อใช้ควบคุมความเร็ว จำนวน 16 ความเร็ว โดยใช้หลักการของดิวตี้ไซเคิล เช่น ถ้าให้สัญญาณควบคุมเท่ากับ 0001b จะมีค่าเท่ากับ 1/16 ทำให้มอเตอร์เริ่มทำงานที่ความเร็วที่ 1 ถ้าให้สัญญาณควบคุมเท่ากับ 1000b จะมีค่าเท่ากับ 8/16 ทำให้มอเตอร์เริ่มทำงานที่ความเร็วที่ 8 หรือประมาณ 50% ของความเร็วสูงสุด ซึ่งความเร็วของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นอยู่กับความถี่ออสซิเลเตอร์ด้วย กล่าวคือมอเตอร์ประกอบจากความเหนี่ยวนำและความต้านทานภายใน เปรียบเสมือนวงจร LPF (Low Pass Filter) หากมีความถี่สูง จะทำให้มอเตอร์มีความเร็วและแรงในการหมุนลดน้อยลง ซึ่งจากวงจรจะใช้ IC555 สร้างความถี่ออสซิลเลเตอร์ ซึ่งสามารถคำนวณความถี่ได้คือ f = 1.44/((R1+2VR1)C1) และด้านเอาท์พุทของวงจรจรจะต่อกับ POWER MOSFET (IRF 5140) และส่งให้วงจรรีเลย์ เพื่อควบคุมทิศทางการเคลื่อนที่ การใช้วงพัลส์วิชมอดดูเลชั่นสามารถควบคุมความเร็วได้และควบคุมแรงบิดในการเคลื่อนที่ของรถทดทดสอบในสภาพถนนต่างๆ ได้อีกด้วย โดยมีไทม์มิ่งไดอะแกรมตามรูปที่ 7.4 และวงจรการใช้งานตามรูปที่ 7.5

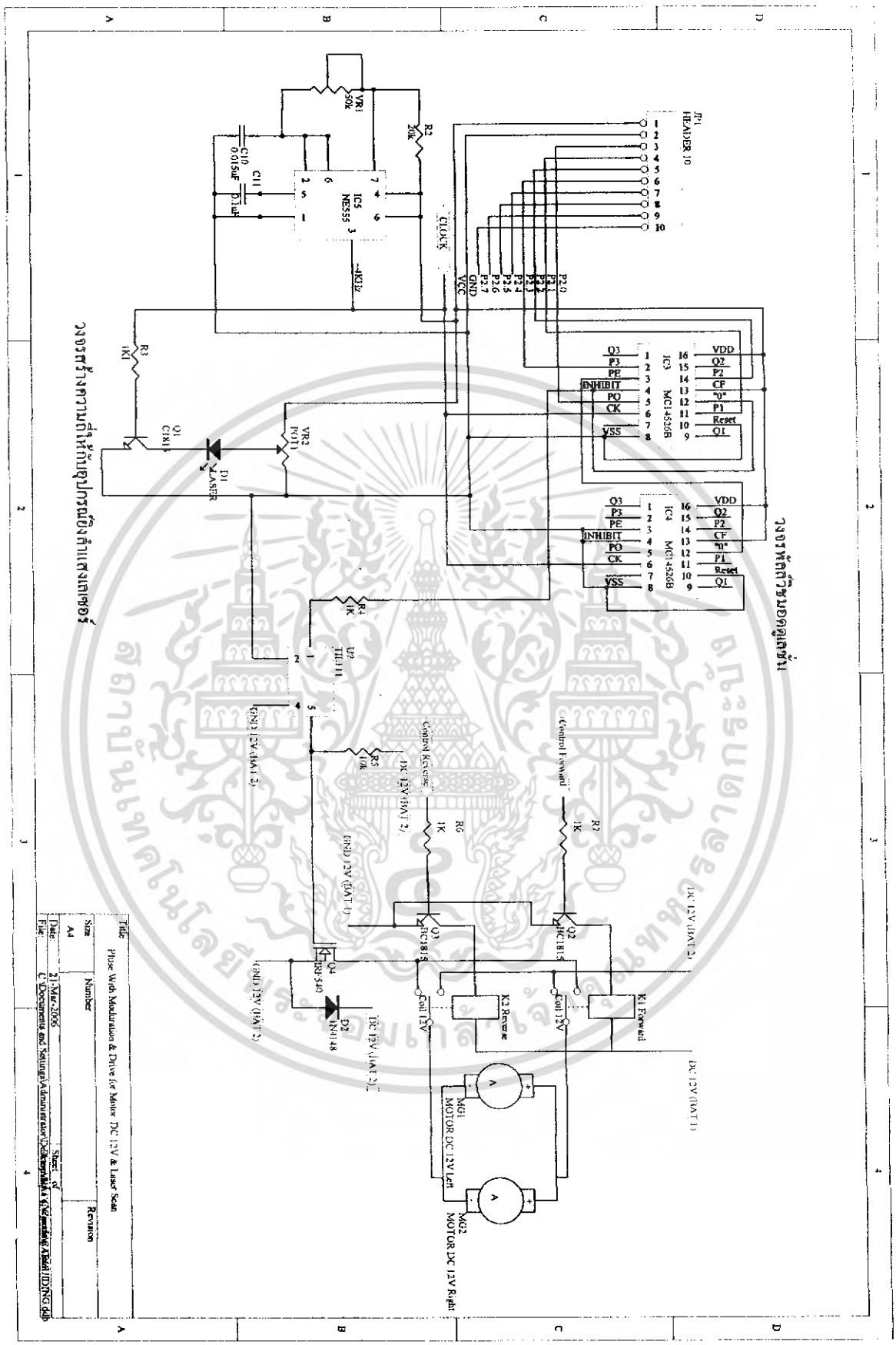


รูปที่ 7.4 แสดงการทำงานของวงจรพัลส์วิดมอดดูเลชั่นในการให้สัญญาณควบคุมแบบต่างๆ

7.3 การทำงานของวงจรสร้างลำแสงเลเซอร์

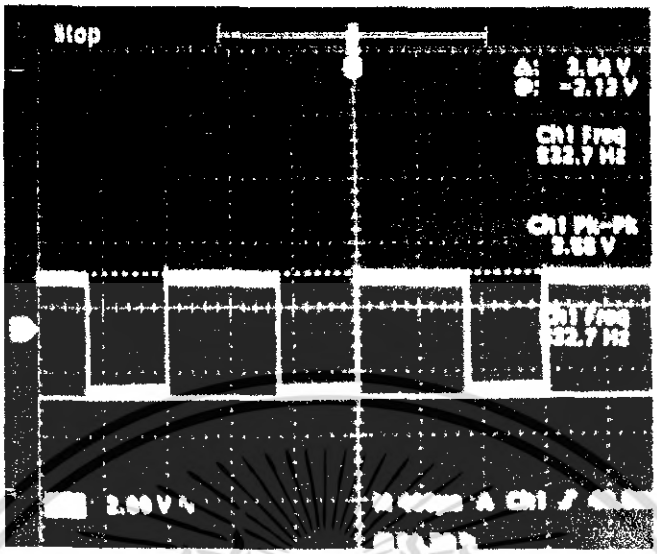
วงจรสร้างลำแสงเลเซอร์จะใช้ความถี่ออสซิเลเตอร์ ผ่านวงจรขยายแบบดอมมอนอิมิตเตอร์เพื่อใช้งานเป็นวงจรสวิชชิ่ง จะได้ความเข้มแสงที่คงที่และสูงสุด โดยมีวงจรการใช้งานตามรูปที่ 7.5 และสัญญาณที่อุปกรณ์ยิงแสงเลเซอร์ตามรูปที่ 7.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.5 แสดงวงจรการพัลส์วิดมอดดูเลชั่น วงจรรีเลย์ควบคุมการเคลื่อนที่ และวงจรสร้างลำแสงเลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6 สัญญาณที่อุปกรณ์ยิงแสงเลเซอร์ที่ใช้ออสซิเลเตอร์เดียวกับวงจรพัลส์วิดมอดดูเลชั่น

7.4 การทำงานของวงจรรีเลย์

วงจรรีเลย์เป็นวงจรเปลี่ยนทิศทางการหมุนของมอเตอร์ เพื่อให้มอเตอร์ทำงานตามคำสั่ง โดยได้รับสัญญาณจากไมโครโปรเซสเซอร์ จำนวน 4 บิต คือ บิตควบคุมการเคลื่อนที่ไปข้างหน้า, บิตควบคุมการเคลื่อนที่ไปข้างหลัง, บิตควบคุมการเคลื่อนที่ไปทางซ้าย และบิตควบคุมการเคลื่อนที่ไปขวา โดยแต่ละบิตทำงานเป็นอิสระ วงจรรีเลย์มี 2 วงจร คือควบคุมการบังคับเลี้ยว และควบคุมการเคลื่อนที่ ซึ่งความเร็วในการเคลื่อนที่ได้มาจากวงจรพัลส์วิดมอดดูเลชั่น ซึ่งมีวงจรการใช้งานตามรูปที่ 7.7

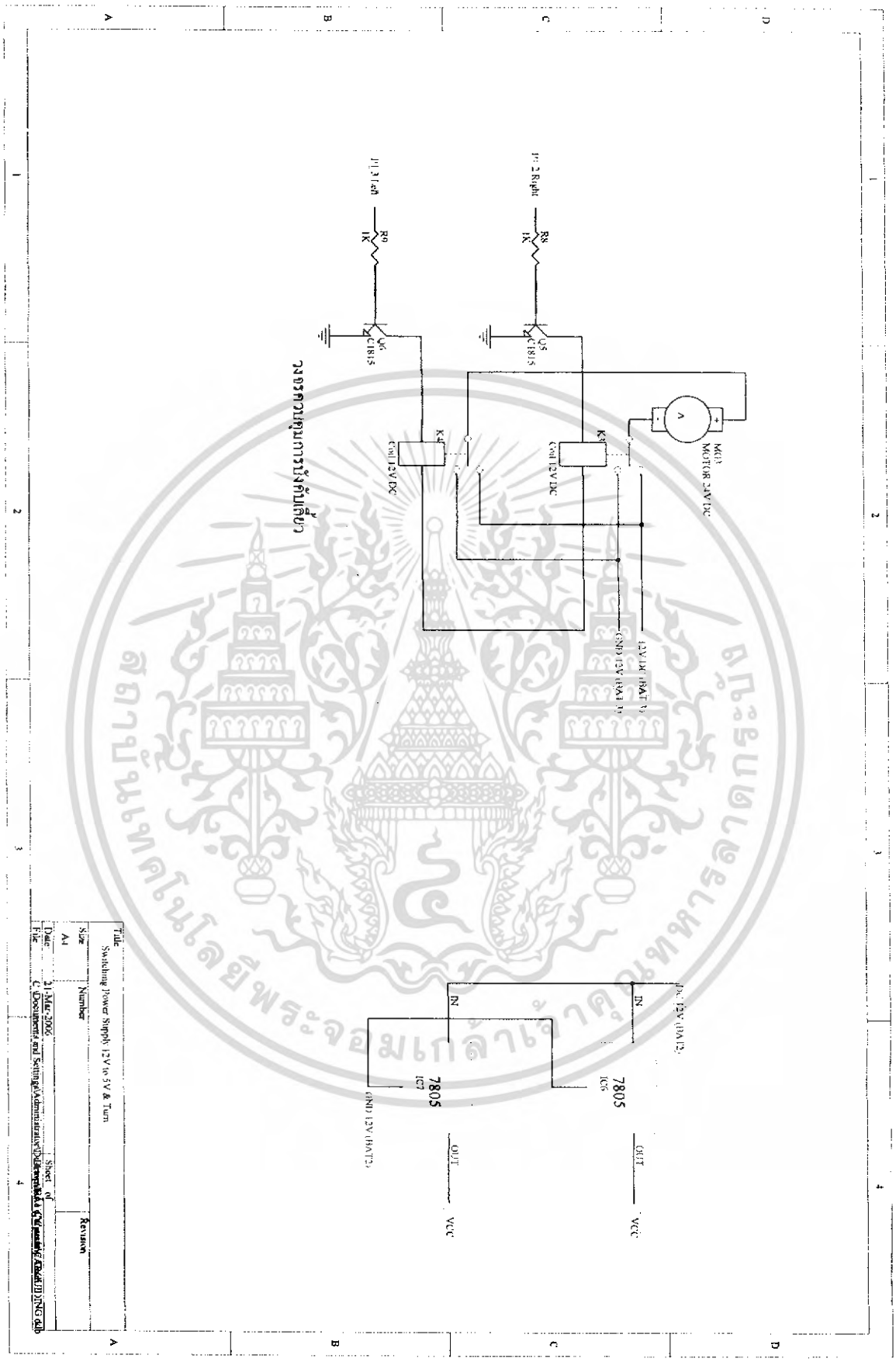
7.5 วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟขนาด 5 โวลท์ สร้างจาก IC7805 จำนวน 2 ตัว เพื่อใช้เลี้ยงวงจรต่างๆ โดยได้รับไฟอินพุทจาก BATTERY ขนาด 12 โวลท์ ซึ่งมีวงจรการใช้งานตามรูปที่ 7.7

7.6 วงจรอินพุทสวิทช์

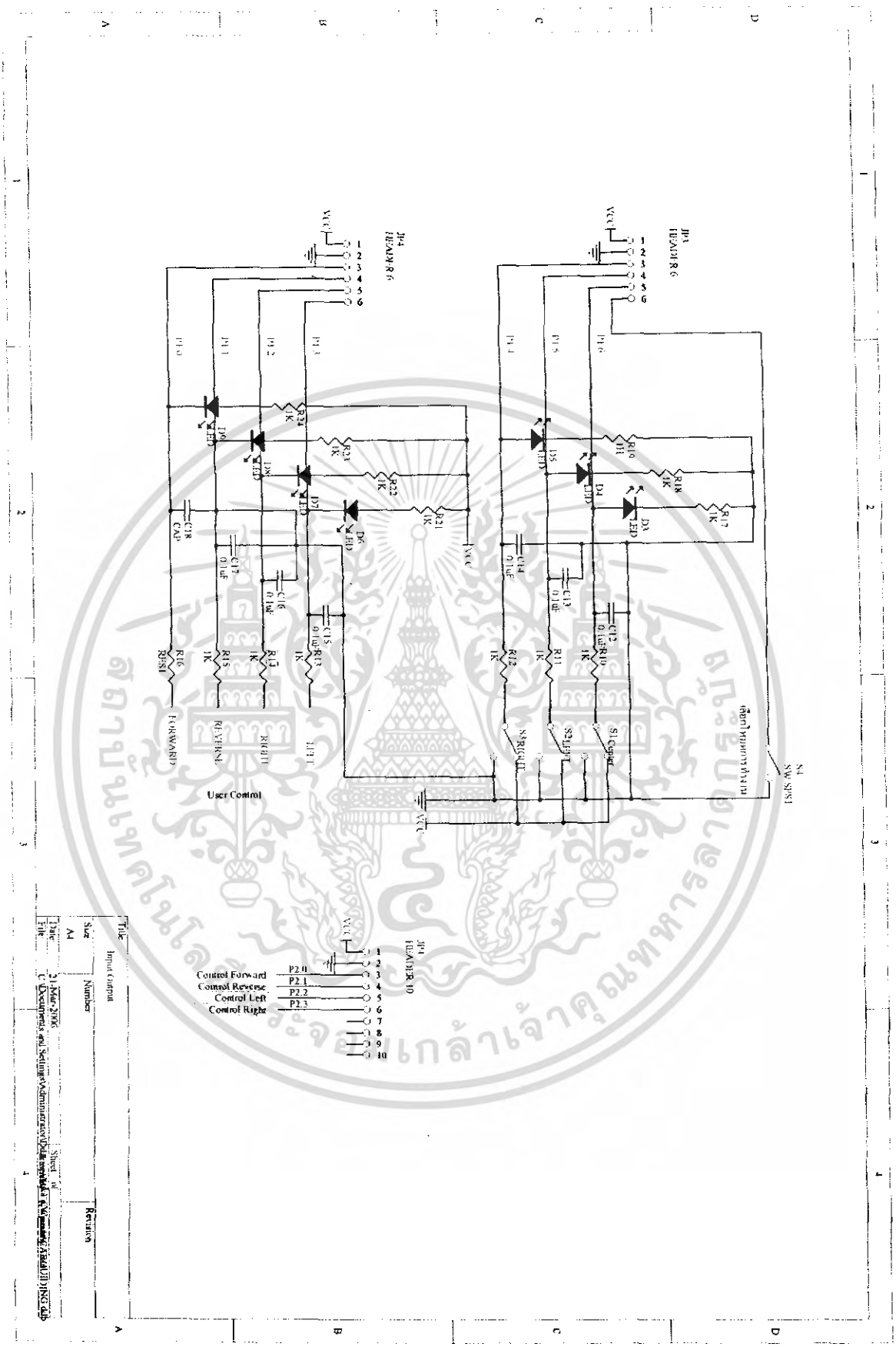
วงจรอินพุทสวิทช์เป็นวงจรอ่านค่าไมโครสวิทช์และสวิทช์จากชุดรีโมท โดยจะผ่านวงจร HFP (Hight Pass Filter) เพื่อลดสัญญาณรบกวนและแสดงผลผ่าน LED (Light Emitting Diode) ซึ่งมีวงจรการใช้งานตามรูปที่ 7.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.7 แสดงวงจรรีเลย์และวงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.8 แสดงวงจรอินพุทสวิทช์

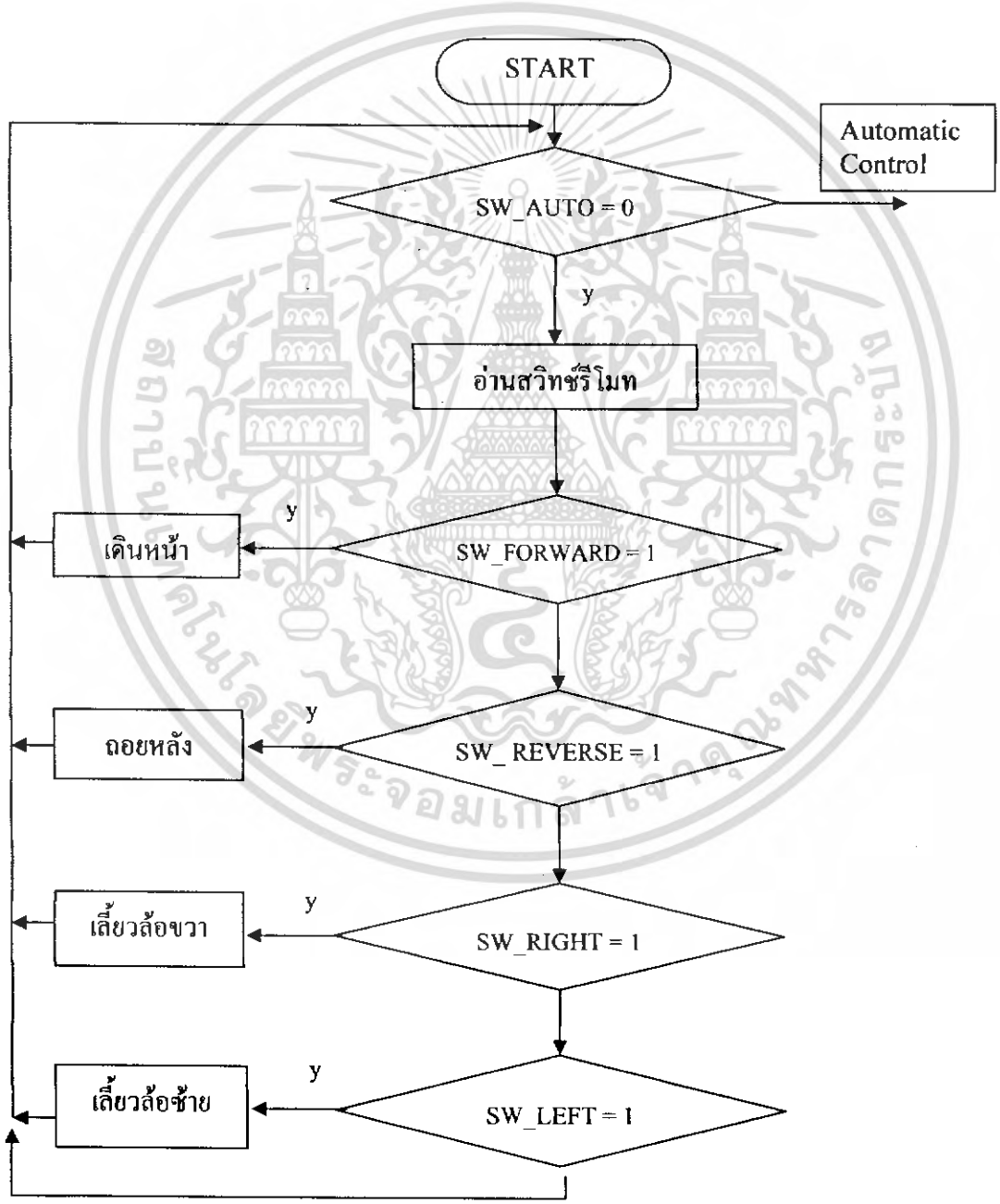
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7 โปรแกรมการทำงานของชุดทดลองต้นแบบ

โปรแกรมการทำงานของชุดทดลองต้นแบบใช้ภาษา SDCC ในการเขียน ซึ่งจะถูกบรรจุอยู่ในไมโครโปรเซสเซอร์ MCS 51 เพื่อควบคุมการทำงานของรถทดสอบ ตามคำสั่งที่ได้รับจากการควบคุมแบบรีโมทหรือการควบคุมแบบออโตเมติก

7.7.1 โปรแกรมการทำงานตามการควบคุมแบบรีโมท

มีหลักการทำงานตามแผนภาพตามรูปที่ 7.9



รูปที่ 7.9 แผนภาพแสดงการทำงานของโปรแกรมควบคุมแบบรีโมท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของโปรแกรมการควบคุมแบบรีโมท

จะทำการอ่านค่าอินพุทสวิทช์ที่เลืออกระบบการควบคุม ถ้าผู้ใช้งานต้องการให้เป็นระบบรีโมท สวิทช์ออโตเมติกจะมีลอจิกเป็น "0" จึงอ่านค่าสวิทช์รีโมทและเปรียบเทียบว่าได้รับคำสั่งอย่างไร

7.7.1.1 ได้รับคำสั่งเดินหน้า (สวิทช์เดินหน้าถูกกด) ไมโครโปรเซสเสซอร์จะให้วงจรพัลส์วิดมอดดูเลชั่นทำงานโดยส่งค่า 0001b หน่วงเวลา 200mS และส่ง 0010b หน่วงเวลา 300mS และให้วงจรรีเลย์ทำงาน รถทดสอบก็จะเคลื่อนที่ไปข้างหน้า จนสวิทช์เดินหน้าถูกปล่อย จะให้วงจรพัลส์วิดมอดดูเลชั่นหยุดทำงานโดยส่งค่า 0000b

7.7.1.2 ได้รับคำสั่งถอยหลัง (สวิทช์ถอยหลังถูกกด) ไมโครโปรเซสเสซอร์จะให้วงจรพัลส์วิดมอดดูเลชั่นทำงานโดยส่งค่า 0001b หน่วงเวลา 200mS และส่ง 0010b หน่วงเวลา 300mS และให้วงจรรีเลย์ทำงาน รถทดสอบก็จะเคลื่อนที่ไปข้างหลัง จนสวิทช์ถอยหลังถูกปล่อย จะให้วงจรพัลส์วิดมอดดูเลชั่นหยุดทำงานโดยส่งค่า 0000b

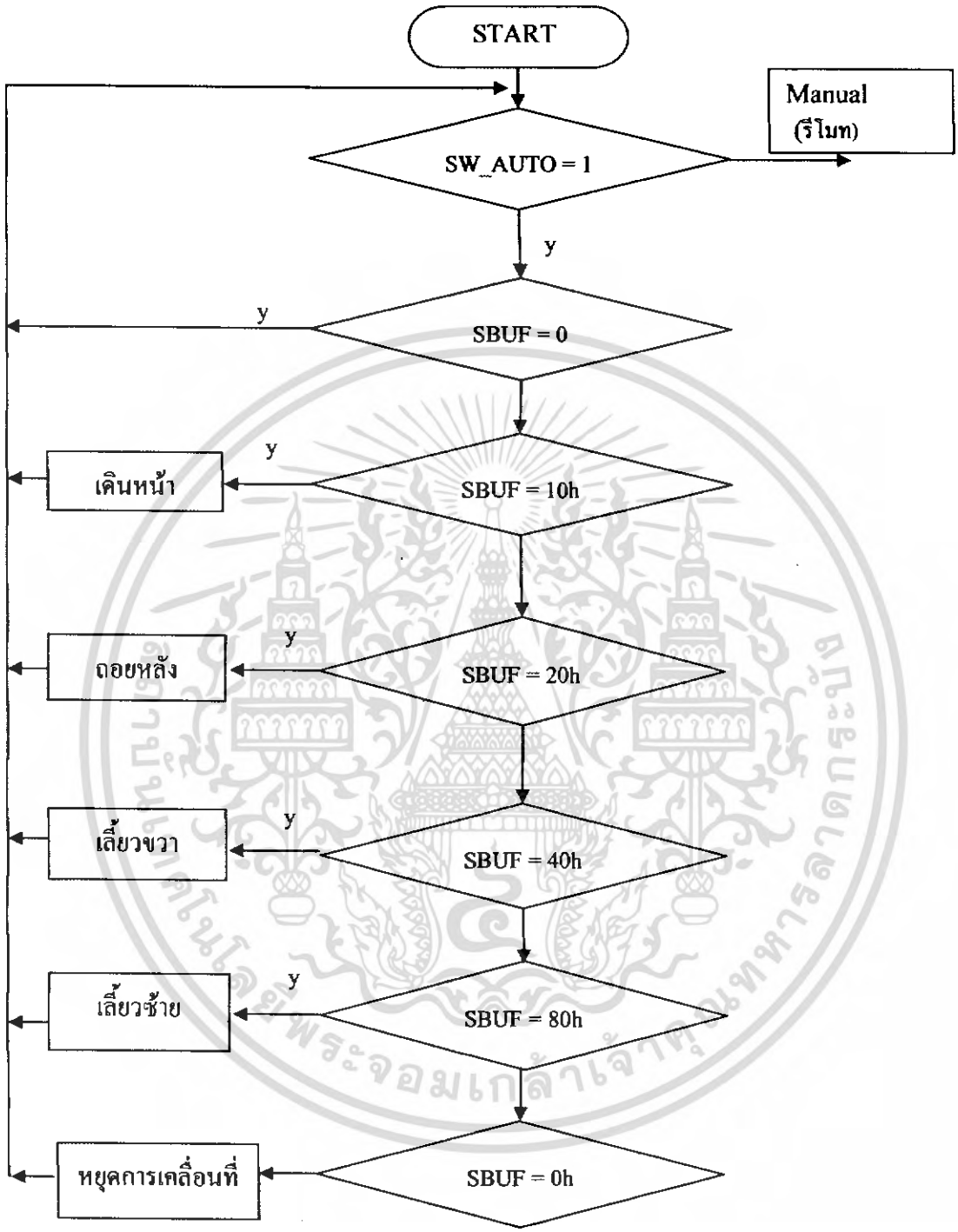
7.7.1.3 ได้รับคำสั่งเลี้ยวขวา (สวิทช์เลี้ยวขวาถูกกด) ให้วงจรรีเลย์ทำงานเลี้ยวขวา ล้อหน้าของรถทดสอบก็จะเลี้ยวขวาจนไมโครสวิทช์ที่ติดตั้งที่ล้อด้านขวาทำงาน และเมื่อสวิทช์เลี้ยวขวาถูกปล่อยจะสั่งให้เลี้ยวขวาจนกว่าล้อหน้าตรงไปข้างหน้า

7.7.1.4 ได้รับคำสั่งเลี้ยวซ้าย (สวิทช์เลี้ยวซ้ายถูกกด) ให้วงจรรีเลย์ทำงานเลี้ยวซ้าย ล้อหน้าของรถทดสอบก็จะเลี้ยวซ้ายจนไมโครสวิทช์ที่ติดตั้งที่ล้อด้านซ้ายทำงาน และเมื่อสวิทช์เลี้ยวซ้ายถูกปล่อยจะสั่งให้เลี้ยวขวาจนกว่าล้อหน้าตรงไปข้างหน้า

7.7.2 โปรแกรมการทำงานตามการควบคุมแบบออโตเมติก

จะรับสัญญาณควบคุมผ่านการสื่อสารแบบ RS 232 จากคอมพิวเตอร์แบบพกพา โดยรับสัญญาณผ่าน IC MAX232 เข้าสู่ไมโครโปรเซสเซอร์ ซึ่งในโปรแกรมจะใช้การอินเตอร์รัพท์หมายเลข 4 ในการรับข้อมูลจึงทำให้การควบคุมรถทดสอบเป็นไปอย่างต่อเนื่อง ซึ่งจะใช้อัตราการส่ง (Bit per Secound) 9600 bps ,ข้อมูล (Data Bits) 8 bit, เงื่อนไข (Parity) ไม่มี และบิตการหยุด (Stop Bit) 1 bit ซึ่งข้อมูลที่ถูกส่งมาจะอยู่ในรีจิสเตอร์ Serial Data Buffer (SBUF) ในโปรแกรมจะนำข้อมูลที่ได้มาเปรียบเทียบเป็นคำสั่งต่างๆ มีหลักการทำงานตามแผนภาพตามรูปที่ 7.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.10 แผนภาพแสดงการทำงานของโปรแกรมควบคุมแบบออโตเมติก

หลักการทำงานของโปรแกรมการควบคุมแบบรีโมท

จะทำการอ่านค่าอินพุทสวิทช์ที่เลืออกระบบการควบคุม ถ้าผู้ใช้งานต้องการให้เป็นระบบออโตเมติกสวิทช์ออโตเมติกจะมีลอจิกเป็น "1" จึงอ่านค่าในรีจิสเตอร์ SBUF และเปรียบเทียบว่าได้รับคำสั่งอย่างไร ซึ่งมีการทำงานคล้ายการควบคุมแบบรีโมท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7.2.1 เมื่อ SBUF = 10h คือคำสั่งเดินหน้า ไมโครโปรเซสเซอร์จะให้วงจรพัลส์วิดมอดดูเลชั่นทำงานโดยส่งค่า 0001b หน่วงเวลา 200mS และส่ง 0010b หน่วงเวลา 300mS และให้วงจรรีเลย์ทำงาน รถทดสอบก็จะเคลื่อนที่ไปข้างหน้า จน SBUF = 0h จะให้วงจรพัลส์วิดมอดดูเลชั่นหยุดทำงานโดยส่งค่า 0000b

7.7.2.2 เมื่อ SBUF = 20h คือคำสั่งถอยหลัง ไมโครโปรเซสเซอร์จะให้วงจรพัลส์วิดมอดดูเลชั่นทำงานโดยส่งค่า 0001b หน่วงเวลา 200mS และส่ง 0010b หน่วงเวลา 300mS และให้วงจรรีเลย์ทำงาน รถทดสอบก็จะเคลื่อนที่ไปข้างหลัง จน SBUF = 0h จะให้วงจรพัลส์วิดมอดดูเลชั่นหยุดทำงานโดยส่งค่า 0000b

7.7.2.3 เมื่อ SBUF = 30h คือคำสั่งเลี้ยวขวา ให้วงจรรีเลย์ทำงานเลี้ยวขวา ล้อหน้าของรถทดสอบก็จะเลี้ยวขวาจนไมโครสวิทช์ที่ติดตั้งที่ล้อด้านขวาทำงาน จน SBUF = 0h จะสั่งให้เลี้ยวขวาจนกว่าล้อหน้าตรงไปข้างหน้า

7.7.2.4 เมื่อ SBUF = 40h คือคำสั่งเลี้ยวซ้าย ให้วงจรรีเลย์ทำงานเลี้ยวซ้าย ล้อหน้าของรถทดสอบก็จะเลี้ยวซ้ายจนไมโครสวิทช์ที่ติดตั้งที่ล้อด้านซ้ายทำงาน จน SBUF = 0h จะสั่งให้เลี้ยวขวาจนกว่าล้อหน้าตรงไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
คณิตศาสตร์ในการสร้างภาพ

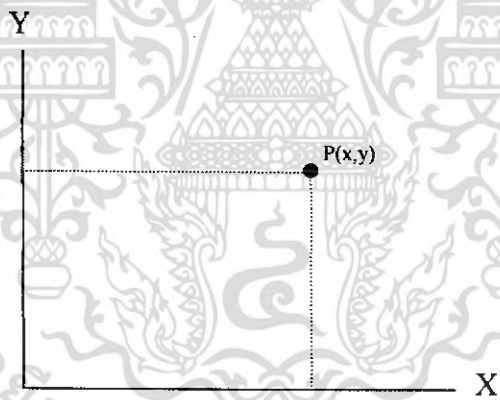
8.1 ทฤษฎี กราฟิก 2 มิติ และกราฟิก 3 มิติ

8.1.1 กราฟิก 2 มิติ

ระบบที่เป็นพื้นฐานของระบบคอมพิวเตอร์กราฟิก คือ ระบบกราฟิก 2 มิติ เพราะภาพที่ได้จากระบบกราฟิก 2 มิตินี้ สามารถนำไปใช้ประโยชน์ได้มากมาย เช่น รูปกราฟต่างๆในระบบกราฟิก 2 มิตินี้ ตำแหน่งของข้อมูลต่างๆจะถูกแทนที่ได้โดยตัวแปร 2 ตัว คือ (x,y) เมื่อ x คือระยะทางในแนวนอนจากจุดกำเนิด (origin) ใดๆ และ y คือระยะทางในแนวตั้งจากจุดกำเนิดเดียวกัน สิ่งที่สำคัญในระบบกราฟิก 2 มิติ คือ จุด (point) เส้นตรง (line) และรูปหลายเหลี่ยม (polygon) เนื่องจากสามารถที่จะนำสิ่งเหล่านี้ไปประกอบเพื่อให้เกิดเป็นภาพขึ้นมาตามความต้องกา

8.1.2 จุด

จุดในระบบกราฟิก 2 มิติจะถูกแสดงโดยพิกัด (Coordinate) โดยค่า 2 ค่าด้วยกัน คือ

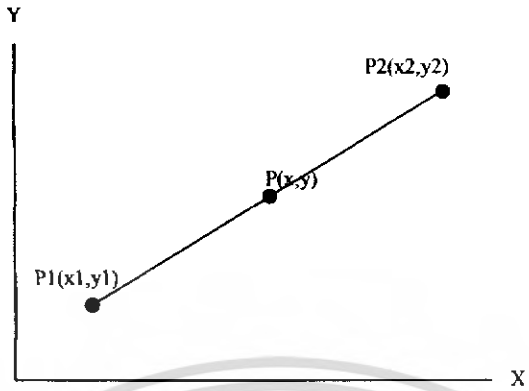


รูปที่ 8.1 จุดในระบบกราฟิก 2 มิติ

8.1.3 เส้นตรง

เส้นตรงในระบบกราฟิก 2 มิตินั้นเกิดจากระยะทางระหว่างจุด 2 จุด ดังรูปที่ 8.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 เส้นตรงในระบบกราฟิก 2 มิติ

สมการของเส้นตรงสามารถอธิบายได้ดังนี้ ถ้าจุด P(x,y) เป็นจุดใดๆบนเส้นตรงที่เกิดจากจุด $P_1(x_1,y_1)$ และ $P_2(x_2,y_2)$ ใดๆแล้ว สมการของเส้นตรงคือ

$$y = mx+b \qquad (1)$$

โดยที่ $$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - mx_1$$

สมการที่ 1 เรียกว่าสมการเส้นตรงแบบ Slope-intercept

m = ความชัน (Slope)

b = จุดตัดแกน y ของสมการเส้นตรง ณ ตำแหน่งที่ x มีค่าเป็นศูนย์ (0,b)

8.1.4 รูปหลายเหลี่ยม

รูปหลายเหลี่ยม คือ รูปที่เกิดจากการรวมของจุดและเส้นตรงที่เชื่อมติดกันที่บริเวณปลายทั้งสองข้าง รูปหลายเหลี่ยมที่เกิดขึ้นนั้นจะมีลักษณะเป็นภาพปิด (Close figure) เรียกด้านที่ประกอบกันของรูปหลายเหลี่ยมว่าด้าน (side) หรือขอบ (edges) ของรูป ส่วนจุดปลายสุดของแต่ละด้านของรูปหลายเหลี่ยมจะเรียกว่าจุดรวม (vertices)

8.1.5 การเปลี่ยนแปลงตำแหน่ง

ในระบบคอมพิวเตอร์กราฟิกนั้นในบางครั้งมีความจำเป็นที่จะต้องทำการเปลี่ยนแปลงลักษณะของภาพเพื่อให้สามารถนำไปใช้ประโยชน์ในด้านอื่น หรือ เพื่อความสะดวกแก่ผู้ใช้งาน โดยไม่ต้องสร้างภาพขึ้นมาใหม่ เช่น การขยายภาพ การลดขนาดภาพ หรือการเปลี่ยนทิศทางของภาพเดิม แต่จะนำสมการทางคณิตศาสตร์บางอย่างมาช่วยในการเปลี่ยนแปลงลักษณะของภาพจำลอง

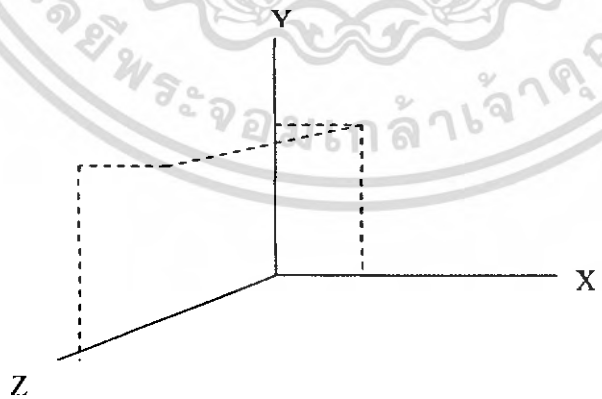
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 กราฟิก 3 มิติ

ภาพกราฟิกบางชนิดจำเป็นต้องใช้ระบบ 2 มิติ เพื่อการแสดงผลเช่น ภาพของกราฟต่างๆ แผนที่ หรือแม้กระทั่งภาพที่ถูกสร้างจากจิตรกรต่างๆ ก็เป็นภาพกราฟิกในระบบ 2 มิติ แต่ในงานบางอย่างก็จำเป็นที่จะต้องใช้ภาพในระบบ 3 มิติ ในการแสดงผลเพื่อให้เกิดความง่ายในการที่จะทำให้เข้าใจในภาพนั้น เช่น ภาพของโครงสร้างต่างๆซึ่งถ้าเป็นภาพจำลองที่สร้างขึ้นมาในระบบ 3 มิติจะสามารถทำให้ผู้ใช้สามารถมองเห็นภาพรวมในจุดมอง (Viewpoint) ที่ต่างๆกัน

8.2.1 จุดและระนาบ

สิ่งที่ง่ายที่สุดในการที่จะนำมาใช้ในการอธิบายระบบ 3 มิติคือ จุด เนื่องจากในระบบ 2 มิติ ได้อธิบายจุดโดยที่บอกตำแหน่งของจุดโดยอาศัย ค่า 2 ค่า แต่ในระบบ 3 มิตินี้จะต้องเพิ่มแกนอีก 1 แกนเพื่อที่จะสามารถนำมาใช้บอกตำแหน่งของจุดในระบบ 3 มิติ ซึ่งจะทำให้ตัวเลขที่ใช้ในการบอกตำแหน่งของจุดมี 3 ค่าด้วยกัน คือ ค่าแรกจะแทนค่าความสูง (height) ของจุด ส่วนค้าที่ 2 และ 3 จะแทนค่าความกว้าง (width) และความลึก (depth) ตามลำดับ โดยที่ทั้ง 3 แกนที่ให้ในการอธิบายตำแหน่งในระบบ 3 มิติ จะมีทิศทางที่ตั้งฉากซึ่งกันและกัน ในระบบ 2 มิตินั้นใช้ค่าในแนวแกน X และแกน Y แทนค่าตำแหน่งในด้านความกว้างและความยาว ส่วนในระบบ 3 มิตินั้นจะใช้แนวแกน Z แทนค่าของตำแหน่งในด้านความลึก ส่วนในแนวแกน X และแกน Y นั้นเหมือนกับในระบบ 2 มิติทุกประการ ดังนั้นการกำหนดตำแหน่งของจุดในระบบ 3 มิติจะสามารถกำหนดได้โดยตัวแปร 3 ตัว คือ P(x,y,z) ดังแสดงในรูปที่ 8.3



รูปที่ 8.3 รูปแสดงตำแหน่งของจุดในระบบ 3 มิติ

และสมการเส้นตรงในระบบ 3 มิติ คือ

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1} \quad (2)$$

ส่วนสมการของระนาบในระบบ 3 มิติคือ

$$Ax + By + Cz + D = 0 \quad (3)$$

เมื่อ A,B,C และ D คือค่าคงที่หรือ

$$x + B_1 y + C_1 z + D_1 = 0 \quad (4)$$

โดยที่ $B_1 = \frac{B}{A}$, $C_1 = \frac{C}{A}$, และ $D_1 = \frac{D}{A}$

และระยะทางระหว่างจุด(x,y,z) ใดๆกับระนาบถูกกำหนดโดย

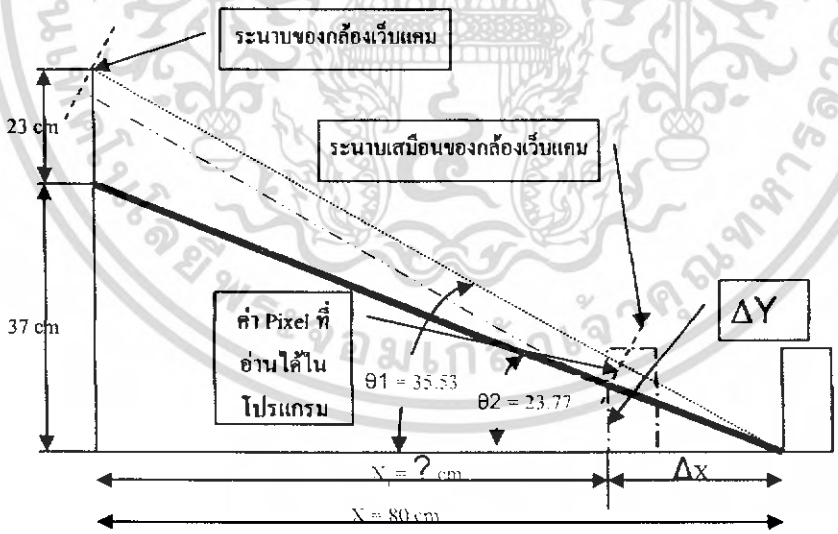
$$L = |A_2 x + B_2 y + C_2 z + D_2| \quad (5)$$

โดยที่ $A_2 = \frac{A}{d}$, $B_2 = \frac{B}{d}$, $C_2 = \frac{C}{d}$ และ $D_2 = \frac{D}{d}$

8.3 การคำนวณหาความสัมพันธ์ของพิกเซล (Pixel) ระยะต่างๆ

8.3.1 การคำนวณหาค่าระยะความสูงของวัตถุจากพิกเซล

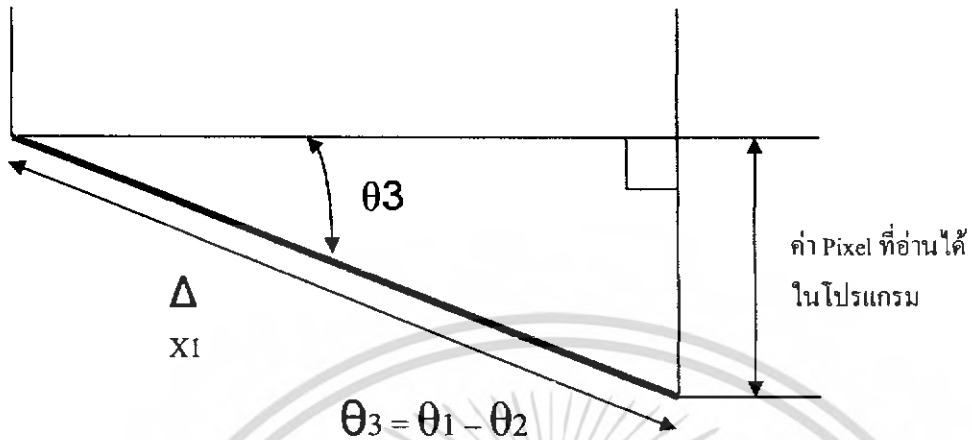
การหาระยะความสูงของวัตถุจากพิกเซล ใช้หลักการของตรีโกณมิติ ซึ่งสามารถคำนวณได้โดยมีรูปแบบ ดังนี้



รูปที่ 8.4 รูปแสดงการหาระยะความสูงของวัตถุจากพิกเซล

เมื่อนำกล้องเว็บแคมมาถ่ายภาพตามรูป จะเกิดระนาบเสมือนเกิดขึ้นกับวัตถุ เป็นแนวขนานกับแนวเส้นกึ่งกลางของกล้องเว็บแคม (เส้นประสีแดง) ดังรูป เมื่อรู้ค่ามุมต่างๆ ก็สามารถคำนวณหาค่าคงที่ (Constant) ที่นำไปคูณกับพิกเซล ซึ่งจะได้ค่าจริงเป็นเซนติเมตร การหาค่าคงที่นี้ต้องได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$\theta_3 = \theta_1 - \theta_2$

$= 11.76$

รูปที่ 8.5 แสดงการหาระยะ $\Delta X1$

$\theta_3 = \theta_1 - \theta_2$

$\Delta X1 = \frac{\text{Pixel x Constant}}{\text{Sine } \theta_3}$

$\Delta Y = \frac{\Delta X1}{\text{Sine } \theta_2}$

$= \frac{\text{Pixel x Constant x Sine } \theta_2}{\text{Sine } \theta_3}$

8.3.2 การคำนวณหาค่าระยะความห่างของวัตถุจากพิกเซล

การหาระยะความห่างของวัตถุจากพิกเซล ใช้หลักการของตรีโกณมิติเช่นกัน ซึ่งสามารถคำนวณได้ โดยมีรูปแบบ ดังนี้

จาก

$\Delta Y = \frac{\text{Pixel x Constant x Sine } \theta_2}{\text{Sine } \theta_3}$

$\text{Tan } \theta_2 = \Delta Y / \Delta X$

$\Delta X = \frac{\Delta Y}{\text{Tan } \theta_2}$

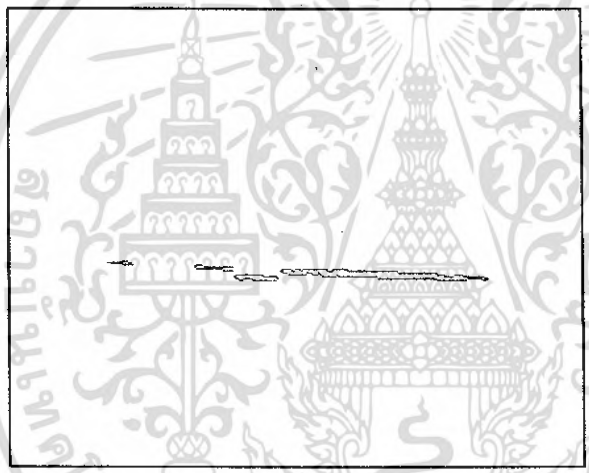
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉะนั้น

$X1 = X - \Delta X$

8.3.3 การคำนวณหาค่าคงที่

การคำนวณหาค่าคงที่นี้มีความสำคัญมาก เนื่องจากมีความสัมพันธ์กับการหาระยะต่างๆ ซึ่งหากมีค่าไม่ถูกต้องอาจทำให้มีความผิดพลาดในการคำนวณ สามารถหาโดยการแทนค่าสูตร ΔY ย้อนกลับ ซึ่งเราต้องทราบค่าจริงเสียก่อน (จากการทดลอง) โดยการวัดค่าความสูงที่แน่นอน ค่าคงที่นี้จะเกี่ยวพันกับระนาบกล้องเว็บแคมด้วย กล่าวคือ ระนาบต้องตั้งฉากกับทิศทางการถ่ายภาพ (เส้นระนาบของเส้นต้องอยู่กลางกล้องเว็บแคม) จึงจะได้ค่าที่ถูกต้องเท่านั้น ตัวอย่างการนำไปใช้งานดังรูปที่ 8.6

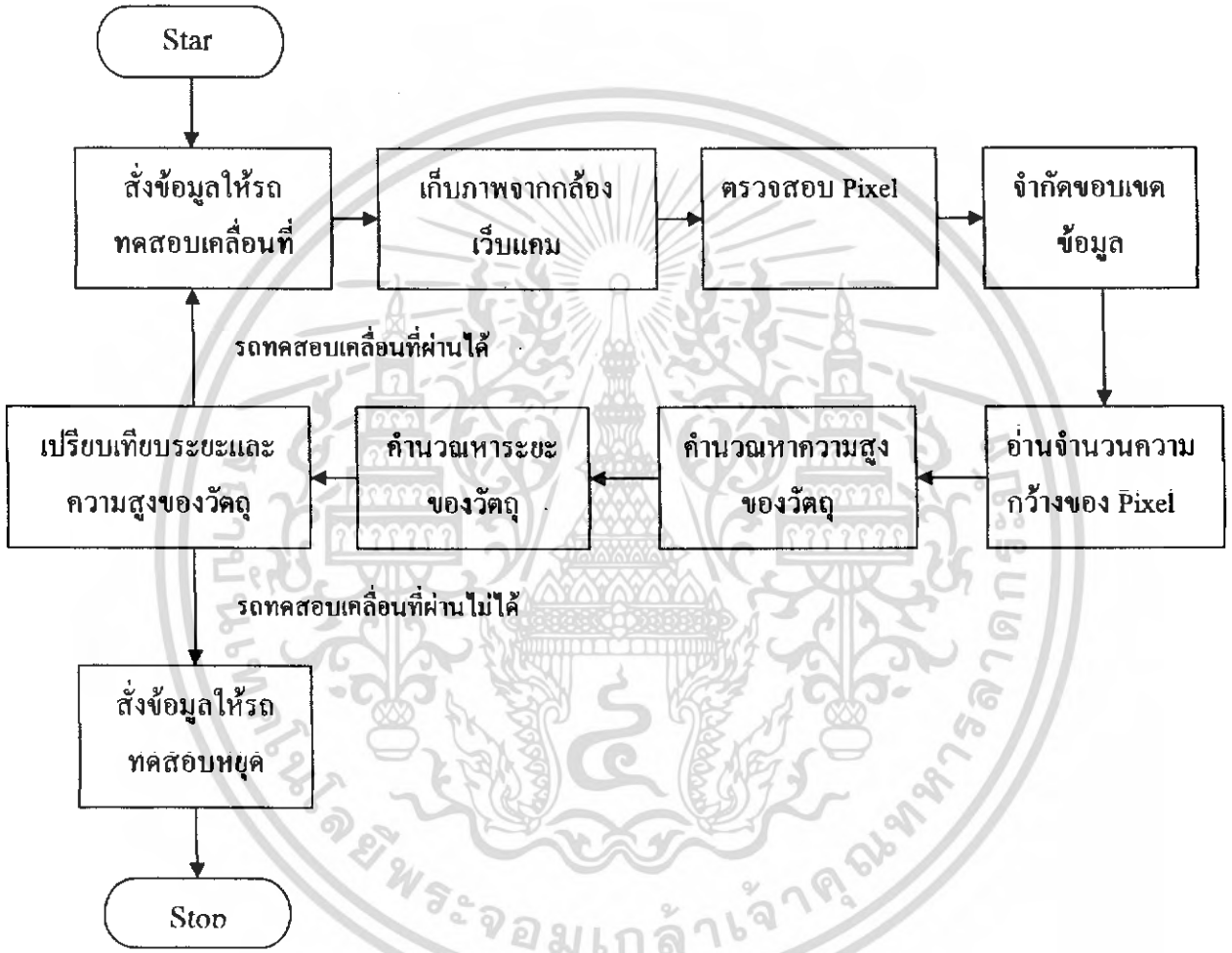


รูปที่ 8.6 แสดงการอ่านค่าพิกเซลเพื่อหาระยะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
โปรแกรมควบคุมการทำงานของคอมพิวเตอร์แบบพกพา

9.1 โครงสร้างการทำงานของโปรแกรมระบบนำร่องรถยนต์ ด้วยแสงเลเซอร์



รูปที่ 9.1 บล็อกไดอะแกรมแสดงการทำงานของโปรแกรมระบบนำร่องรถยนต์

9.1.1 ส่งข้อมูลให้รถเคลื่อนที่

คือขั้นตอนที่คอมพิวเตอร์แบบพกพาส่งข้อมูลให้ไมโครโปรเซสเซอร์ (MCS 51) ผ่านการสื่อสารแบบ RS 232 สั่งให้รถเดินหน้า โดยส่งข้อมูล 01h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.1.2 เก็บภาพจากกล้องเว็บแคม

คือขั้นตอนการรับภาพจากกล้องเว็บแคมมาเก็บไว้ในหน่วยความจำ เพื่อนำไปทำกระบวนการประเมินผลต่อไป

9.1.3 ตรวจสอบ Pixel

คือขั้นตอนการกำจัด Pixel สิ่งแวดล้อมที่เป็นสีอื่นๆ ออก และคงไว้เพียง Pixel สีแดง ที่ได้จากเส้นสแกนเลเซอร์เท่านั้น

สูตรการคำนวณ : ((R>G)&&(R>B)&&((R-G>dG)||(R-B>dB)))

R : องค์ประกอบสีแดง (0 – 255)

G : องค์ประกอบสีเขียว (0 – 255)

B : องค์ประกอบสีน้ำเงิน (0 – 255)

dG : ค่าคงที่ในการตรวจสอบองค์ประกอบสีแดงกับองค์ประกอบสีเขียว

dB : ค่าคงที่ในการตรวจสอบองค์ประกอบสีแดงกับองค์ประกอบสีน้ำเงิน

9.1.4 จำกัดขอบเขตข้อมูล

คือขั้นตอนการการกำจัด Pixel รบกวนที่บริเวณอื่น ที่ไม่ต้องการเพื่อการตรวจสอบค่า Pixel ให้มีความถูกต้องมากขึ้น โดยจะจำกัดบริเวณของของภาพเท่านั้น และหาค่าความสูงสุดและต่ำสุดของ Pixel

9.1.5 อ่านจำนวนความกว้างของ Pixel

คือขั้นตอนการประมวลผลหาจำนวนความกว้างของ Pixel ซึ่งเกิดจากเส้นสแกนเลเซอร์ตัดกับวัตถุ ทำให้เกิดความห่างระหว่างเส้นที่ตัดวัตถุกับแนวเส้นปกติ

9.1.6 คำนวณหาค่าความสูงของวัตถุ

คือขั้นตอนการแทนค่าในสูตร

$$\text{ความสูงของวัตถุ} = \frac{\text{Pixel x Constant x Sine } \theta_2}{\text{Sine } \theta_3}$$

*** ดูรูปที่ 8.4 และ 8.5 ในบทที่ 8 ประกอบ

9.1.7 คำนวณหาค่าระยะของวัตถุ

$$\Delta X = \frac{\Delta Y}{\text{Tan } \theta_2}$$

$$\text{ระยะห่างวัตถุ} = \text{ระยะเส้นสแกนในแนวปกติ} - \left\{ \frac{\text{Pixel x Constant x Sine } \theta_2}{\text{Sine } \theta_3} \right\} \Big/ \text{Tan } \theta_2$$

*** ดูรูปที่ 8.4 และ 8.5 ในบทที่ 8 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.1.8 เปรียบเทียบระยะและความสูงของวัตถุ

คือขั้นตอนการเปรียบเทียบค่าที่คำนวณได้จากโปรแกรม กับค่าคงที่ที่ตั้งไว้ หากสามารถให้รถเคลื่อนที่ผ่านได้ จะสั่งให้รถทดสอบเคลื่อนที่ต่อไป หากรถทดสอบเคลื่อนที่ผ่านไม่ได้ ก็จะสั่งให้รถหยุดและแสดง Alarm บนหน้าจอคอมพิวเตอร์

9.1.9 สั่งให้รถทดสอบหยุด

คือขั้นตอนต่อจากการเปรียบเทียบระยะและความสูงของวัตถุแล้ว ปรากฏว่ารถทดสอบไม่สามารถผ่านได้ คอมพิวเตอร์แบบพกพาส่งข้อมูลให้ไมโครโปรเซสเซอร์ (MCS 51) ผ่านการสื่อสารแบบ RS 232 ให้รถหยุด โดยส่งข้อมูล 0h

9.2 หลักการทำงานของโปรแกรมระบบนำร่องรถยนต์ด้วยแสงเลเซอร์บนคอมพิวเตอร์แบบพกพา

ในการสร้างโปรแกรมใช้งานต่างๆ จะใช้โปรแกรม Visual C++.Net แบบ Win32 ในการสร้าง

9.2.1 โปรแกรมเก็บภาพจากกล้องเว็บแคม

คือขั้นตอนการรับภาพจากกล้องเว็บแคมที่เป็นภาพนิ่ง โดยข้อมูลที่จะสามารถกำหนดขนาดของภาพ ความเข้มแสง และความสว่างได้ และนำมาเก็บไว้ในหน่วยความจำ

9.2.2 โปรแกรมตรวจสอบ Pixel

คือขั้นตอนการตรวจสอบค่าสีแดง โดยจะดึงข้อมูลในหน่วยความจำแต่ละจุดออกมา เพื่อหาค่าองค์ประกอบ โดยตรวจสอบทีละ Pixel คือทดสอบความห่างขององค์ประกอบ Pixel คือ ((R>G)&&(R>B)&&((R-G>dG)||(R-B>dB))) ถ้าเป็นจริง จะกำหนดให้ Pixel นั้นเป็นสีแดง (R = 255, G=0 และ B = 0) กับ และถ้าเป็นเท็จ จะให้ Pixel นั้น เป็นสีขาว (R = 255, G=255 และ B = 255) ซึ่งหากกำหนดให้ภาพมีความละเอียดมากและมีความสว่างมาก ก็จะทำให้โปรแกรมมีความล่าช้า

9.2.3 โปรแกรมการจำกัดขอบเขตข้อมูล

คือขั้นตอนการการกำจัด Pixel นอกบริเวณตรวจสอบ (ช่วงความกว้างของล้อรถทดสอบ) และตัดส่วนด้านบนและล่างออก ให้เป็นสีขาว (R = 255, G = 255, B = 255) เพื่อกำจัดสีของ Pixel รบกวน และหา Pixel ที่อยู่ในแนวแกน X ที่มีค่าสูงสุด คือ X1 มีค่าต่ำสุด คือ X2 หากการตั้งกล้องเว็บแคมไม่ได้ระนาบตามรูปที่ 8.4 และ 8.5 ในบทที่ 8 จะทำให้สูญเสียข้อมูลของภาพ และทำให้เกิดความผิดพลาดในการคำนวณได้

9.2.4 โปรแกรมคำนวณหาค่าความสูงและระยะของวัตถุ

คืออ่านค่าความกว้างของ Pixel ที่เกิดจากระยะห่างระหว่างเส้นปกติและเส้นที่ตัดกับวัตถุ (X1 – X2) เป็นจำนวนของ Pixel และคำนวณหาค่าของแนวระนาบกล้องที่เป็นค่าจริง (ซม.) แล้ว

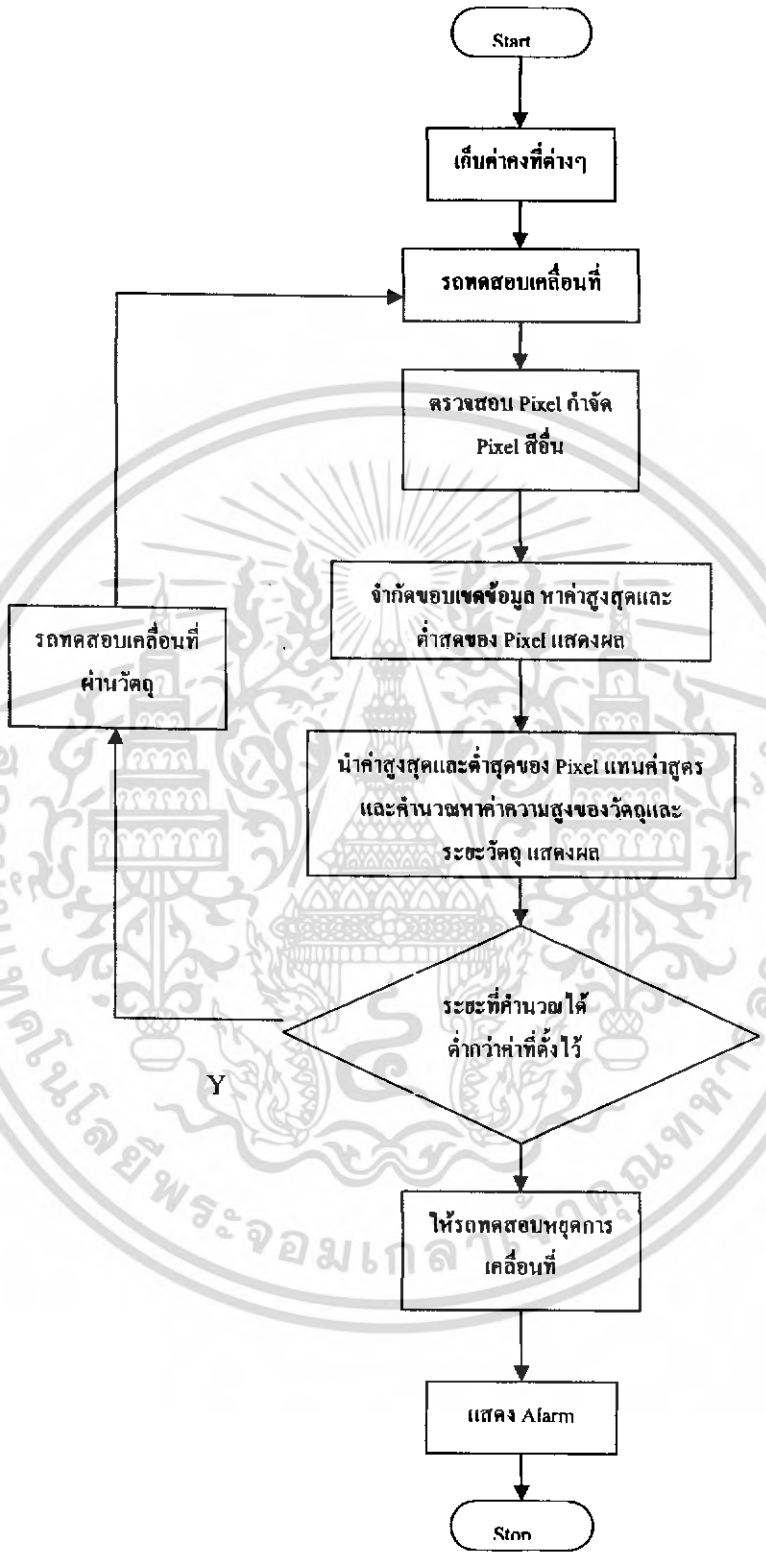
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำไปเข้าสูตร จะได้ค่าความสูงออกมาเป็นเซนติเมตร (ค่าจริงของวัตถุ) นำค่าความสูงที่ได้นำไปเข้าสูตรหาระยะห่างของวัตถุออกมาเป็นเซนติเมตร (ค่าจริงของวัตถุ)

9.2.5 โปรแกรมเปรียบเทียบ

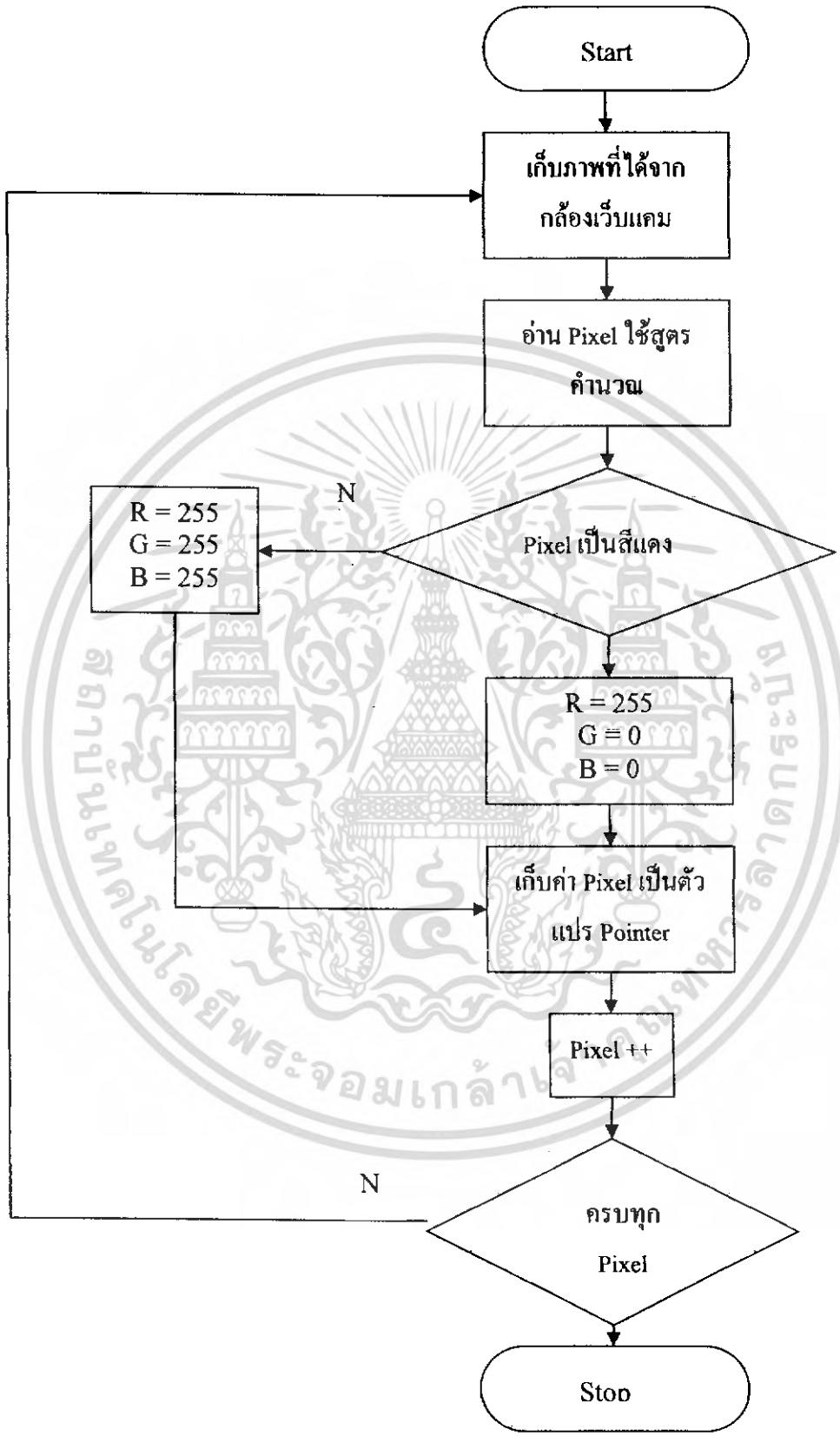
คือการนำค่าความสูงของวัตถุและระยะของวัตถุเปรียบเทียบการค่าคงที่ (ค่าคงที่ของรถทดสอบ) หากค่าความสูงที่คำนวณได้มีค่าน้อยกว่าค่าคงที่ที่ตั้งไว้ก็จะออกจากโปรแกรม หากมีค่ามากกว่าจะส่งข้อมูลให้ไมโครโปรเซสเซอร์หยุดการเคลื่อนที่ของรถทดสอบทันที และแสดง Alarm ทันที รอจนกว่าได้รับคำสั่งอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



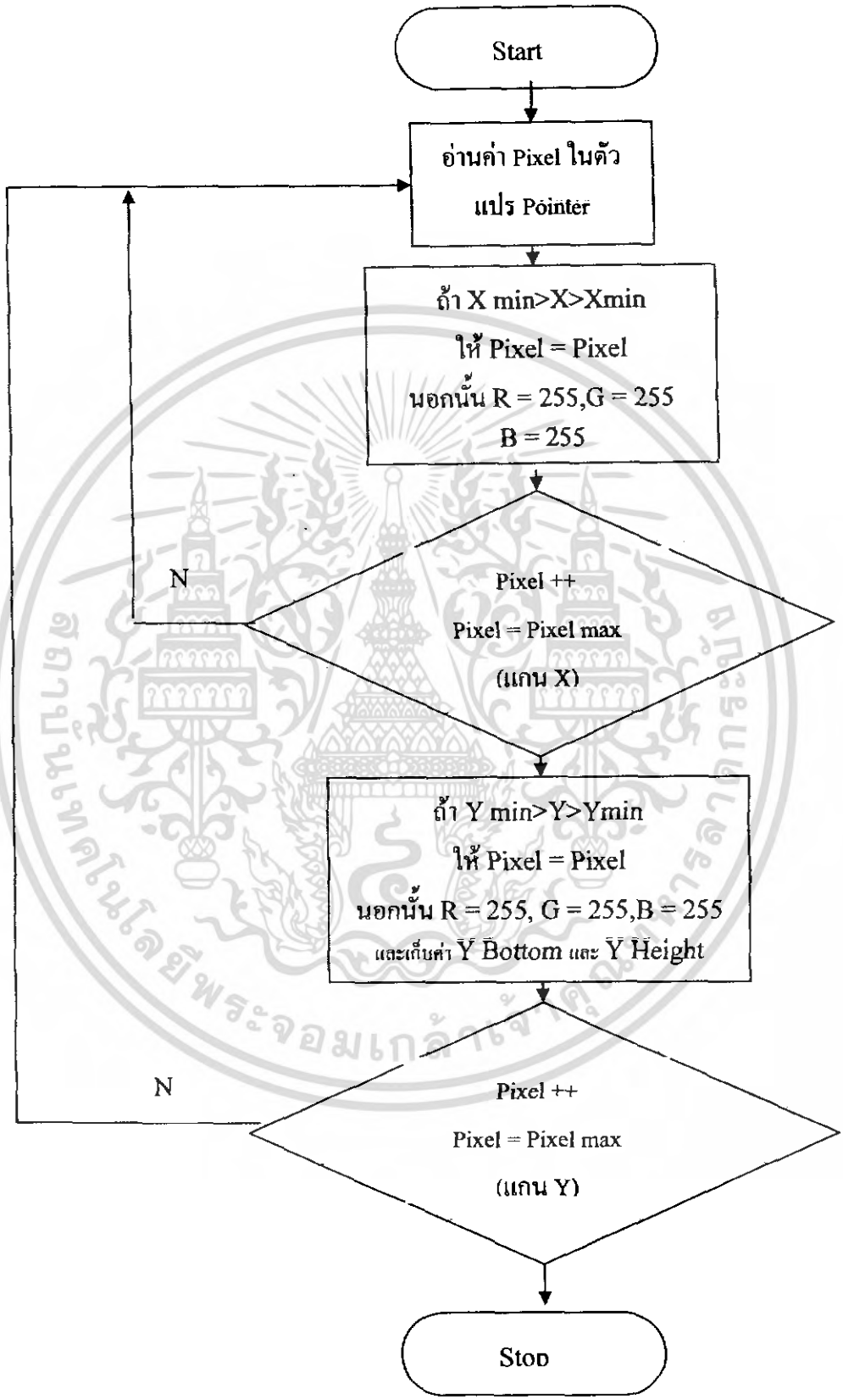
รูปที่ 9.2 แผนภาพแสดงโครงสร้างโดยรวมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



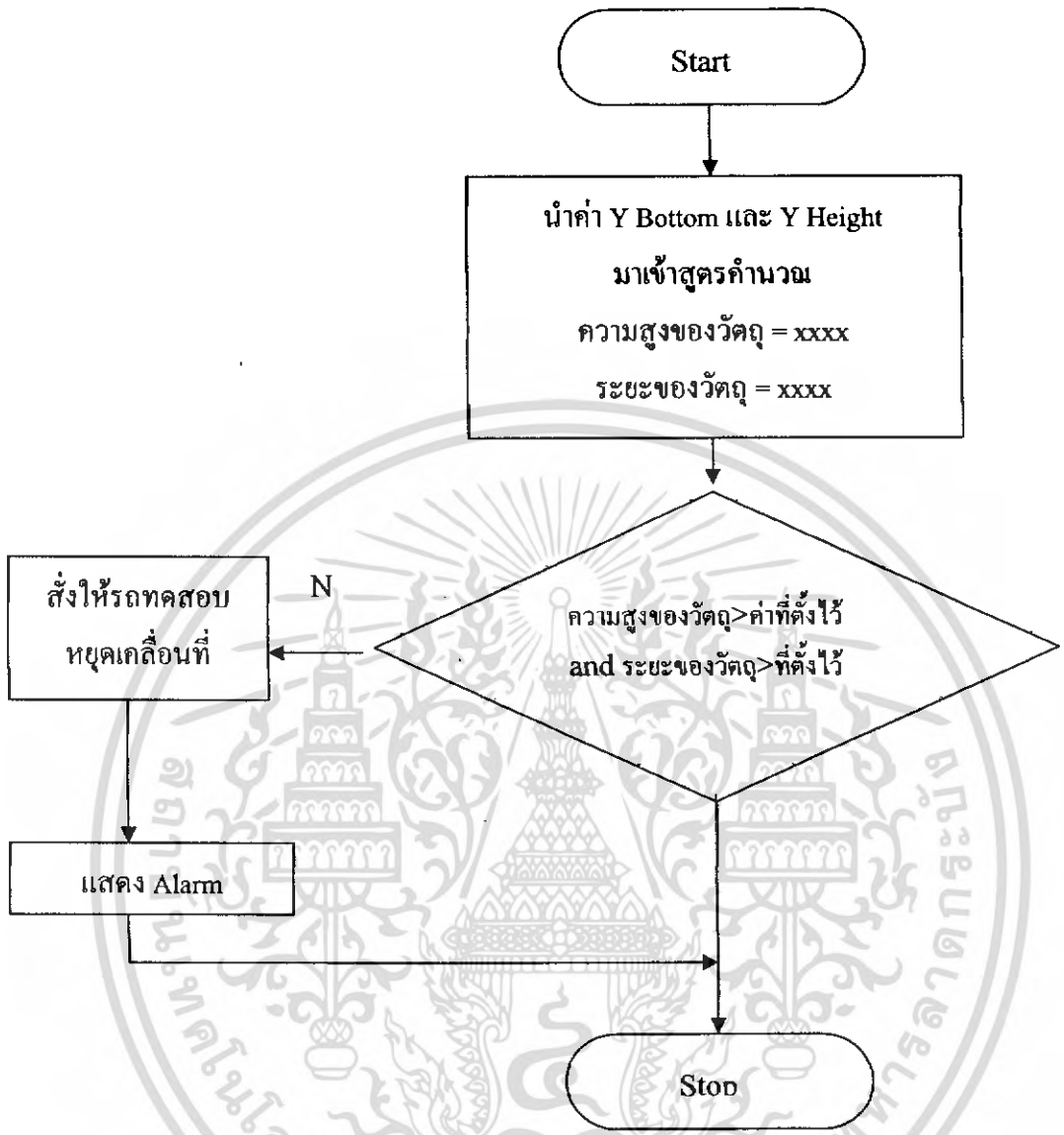
รูปที่ 9.3 แผนภาพแสดงโปรแกรมตรวจสอบ Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.4 แผนภาพแสดงโปรแกรมการจำกัดขอบเขตข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.5 แผนภาพแสดงโปรแกรมคำนวณหาค่าความสูงและระยะของวัตถุและโปรแกรมเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
การทดลอง

10.1 การเตรียมอุปกรณ์ในการทดลองการสร้างเส้นลำแสงเลเซอร์สแกนและเก็บภาพที่ได้จากกล้องเว็บแคม

10.1.1 จัดมุมของอุปกรณ์สร้างสร้างเส้นสแกนลำแสงเลเซอร์ให้มีค่าระยะห่างเท่ากับ 80 ซม.

10.1.2 เชื่อมต่อสายสัญญาณต่างๆ คือ เส้นของกล้องเว็บแคม, สายสัญญาณ RS 232

10.1.3 จัดมุมกล้องเว็บแคมให้เกิดเส้นเลเซอร์เส้นปกติกึ่งกลางหน้าจอ

10.1.4 ปิดสวิทช์ S2 ให้ทำงานเป็นแบบรีโมท

10.2 วิธีการทดลองอุปกรณ์สร้างเส้นสแกนลำแสงเลเซอร์

10.2.1 ใช้อุปกรณ์สร้างเส้นสแกนลำแสงเลเซอร์ที่สร้างขึ้นในภาคเรียนที่ 1

รูปที่ 10.1 เส้นสแกนเลเซอร์ของอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 1

10.2.2 เป็นอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 2

รูปที่ 10.2 เส้นสแกนเลเซอร์ของอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

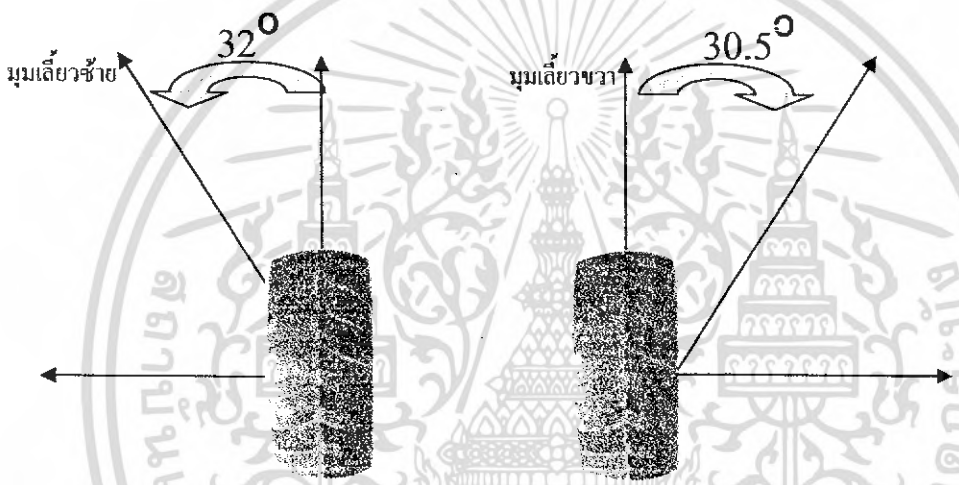
10.2.3 ผลการทดลอง

เส้นสแกนของอุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 2 จะมีความกว้างของเส้นเลเซอร์น้อยกว่า และมีความเข้มของแสงมากกว่า แต่มีความเข้มไม่เท่ากันตลอดแนวเส้นเลเซอร์ เมื่อนำมาทดสอบจะได้ผลดีกว่า อุปกรณ์ที่สร้างขึ้นในภาคเรียนที่ 2 ดังรูปที่ 10.1 และรูปที่ 10.2

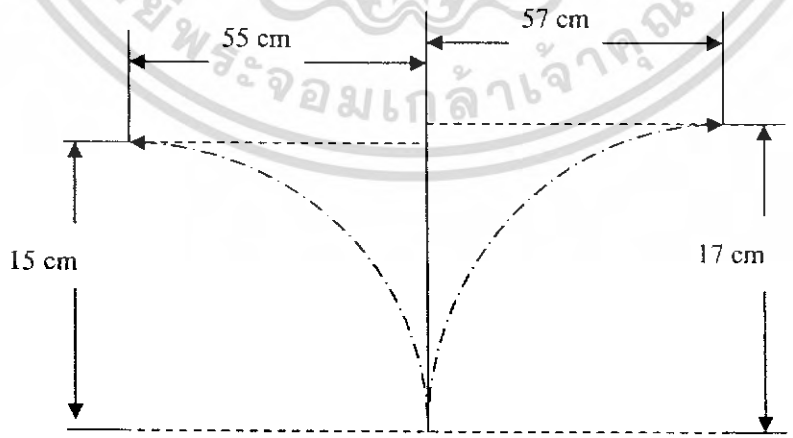
10.3 วิธีการทดลองการหามุมเลี้ยวของรถทดสอบ

10.3.1 ปิดสวิทช์ S2 ใช้งานการควบคุมแบบรีโมท

10.3.2 สั่งรถทดสอบให้เลี้ยวขวาและซ้าย



รูปที่ 10.3 แสดงมุมเลี้ยวของล้อรถทดสอบทั้งสองล้อ



รูปที่ 10.4 แสดงวงเลี้ยวของรถทดสอบทั้งขวาและซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

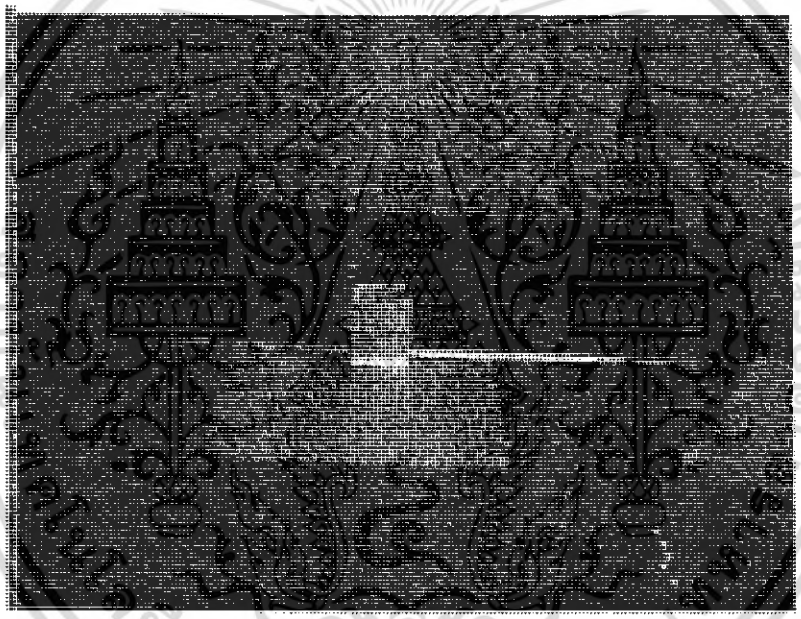
10.3.3 ผลการทดลอง

จากการหาค่ามุมเลี้ยวของรถทดสอบจะได้มุมที่ไม่เท่ากัน เนื่องจากการติดตั้งไมโครสวิทช์จำกัดมุมเลี้ยว มีระยะการทำงานที่ไม่เท่ากัน ตามรูปที่ 10.3 และเมื่อให้รถเคลื่อนที่จะได้วงเลี้ยวของทางด้านซ้ายน้อยกว่าทางด้านขวา ตามรูปที่ 10.4

10.4 วิธีการทดลองโปรแกรมระบบนำร่องรถยนต์ด้วยแสงเลเซอร์บนคอมพิวเตอร์แบบพกพา

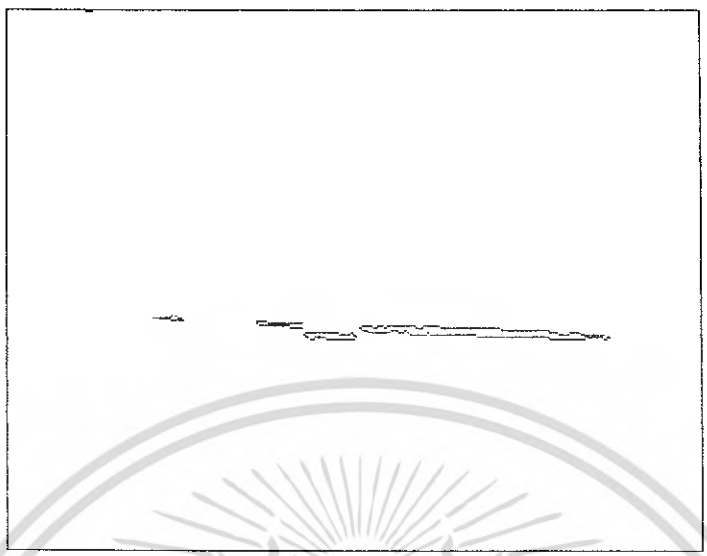
10.4.1 ปิดสวิทช์ S2 ใช้งานการควบคุมแบบรีโมท

10.4.2 นำวัตถุทดสอบความสูง 10 ซม. ไปตั้งกีดขวางเส้นทางของรถทดสอบ

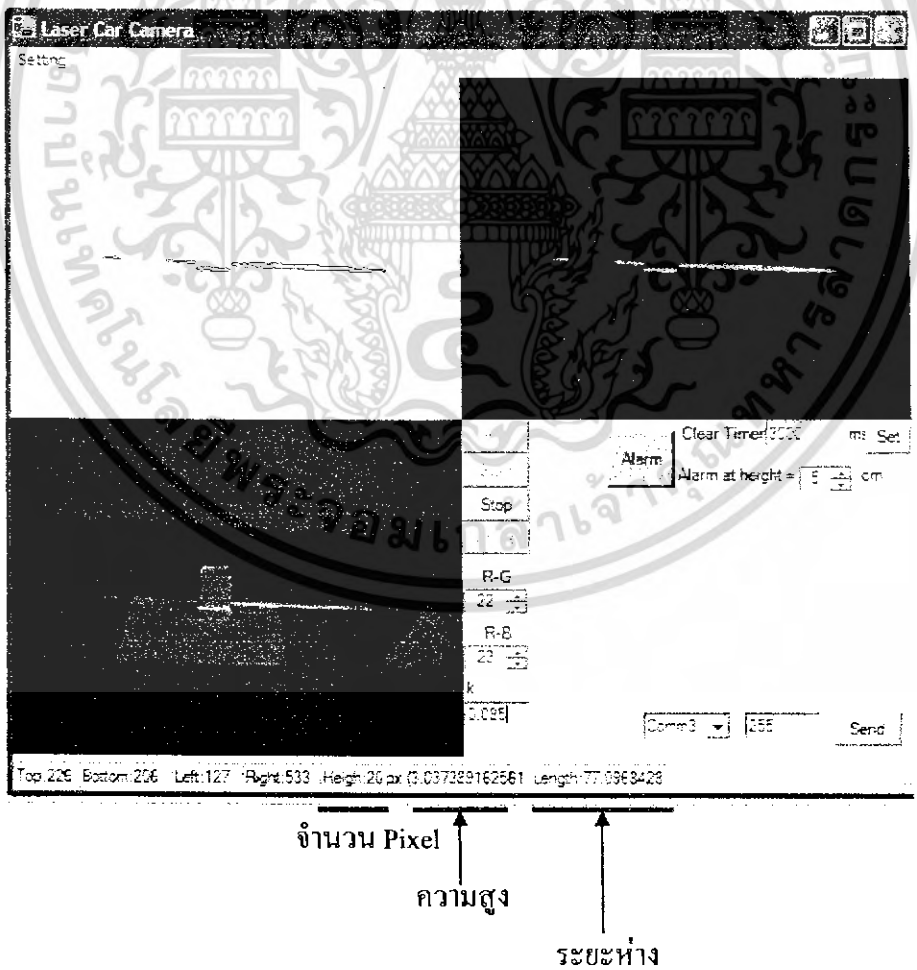


รูปที่ 10.5 แสดงรูปที่ผ่านโปรแกรมเก็บภาพจากกล้องเว็บแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

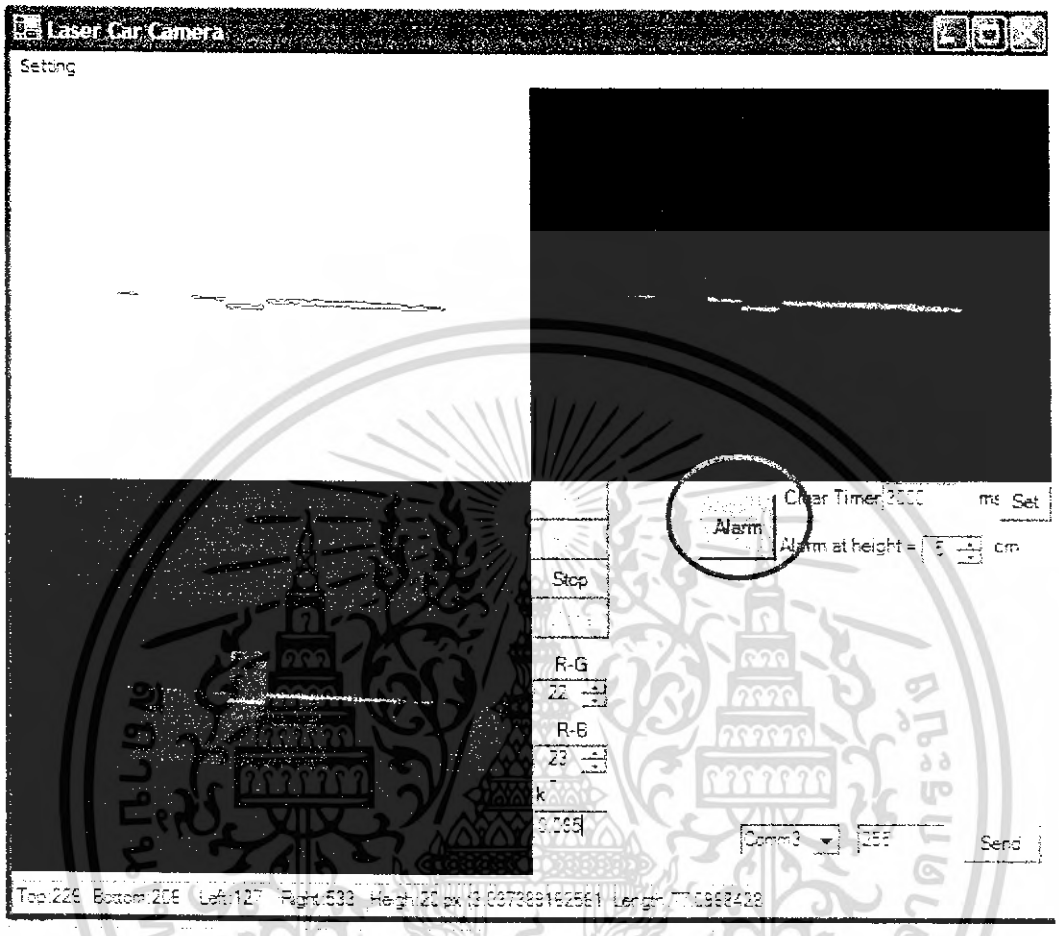


รูปที่ 10.6 แสดงรูปที่ผ่านโปรแกรมตรวจสอบ Pixel และ โปรแกรมการจำกัดขอบเขตข้อมูล



รูปที่ 10.7 แสดงรูปการแสดงผลค่าที่ได้จากการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

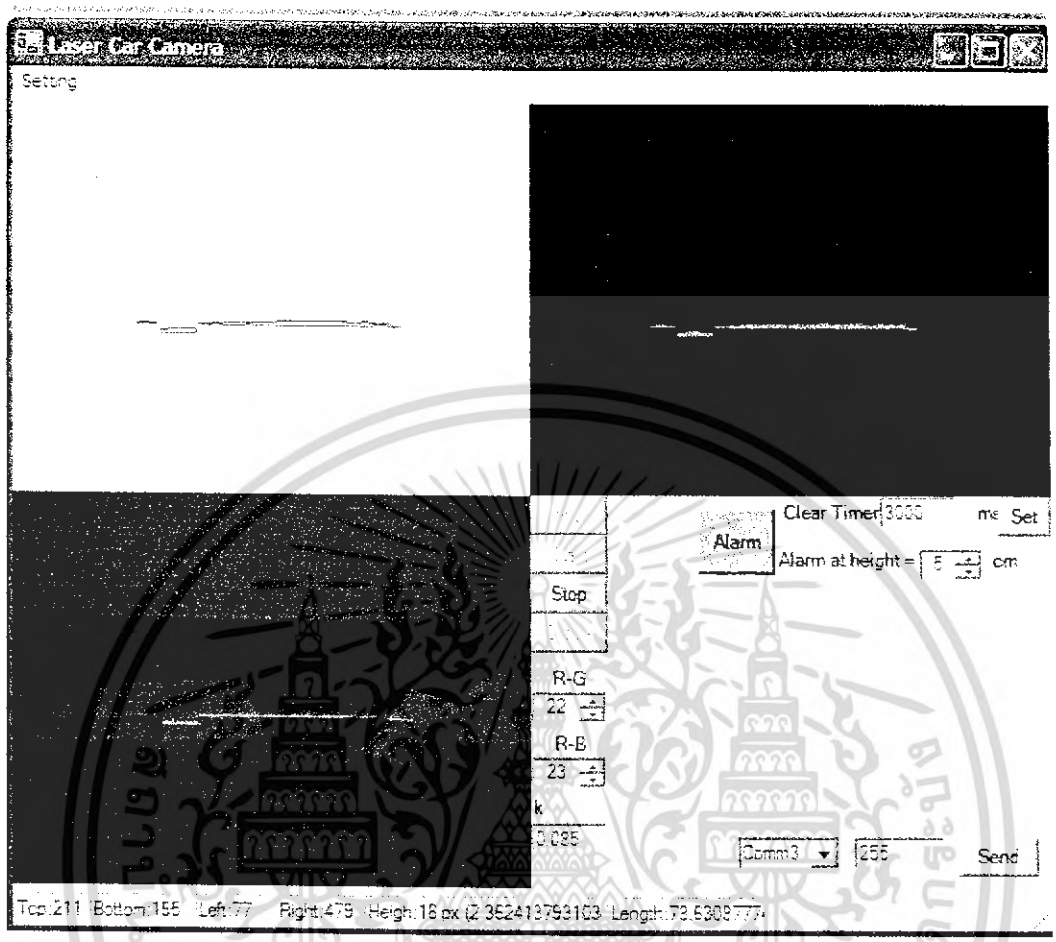


รูปที่ 10.8 แสดงรูปที่ผ่านโปรแกรมเปรียบเทียบแล้วมีค่าน้อยกว่าค่าที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

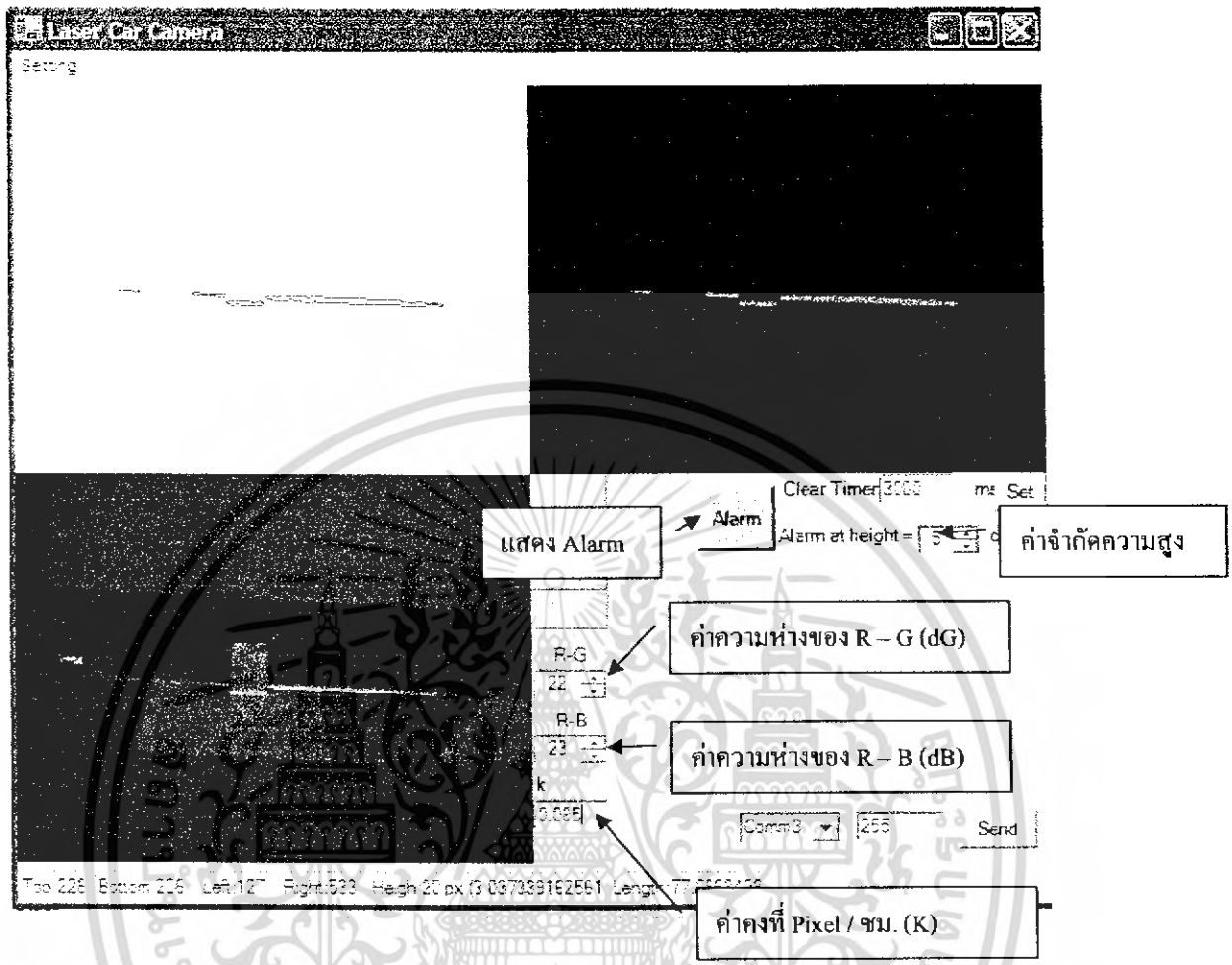
รูปที่ 10.9 แสดงรูปที่ผ่านโปรแกรมเปรียบเทียบแล้วมีค่ามากกว่าค่าที่ตั้งไว้และแสดง Alram และหยุดการเคลื่อนที่ของรถทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.10 แสดงผลของโปรแกรมเมื่อใช้วัตถุทดสอบที่มีความสูง 3 ซม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.11 แสดงรูปค่าคงที่ต่างๆ ที่ใช้ในการทดลอง

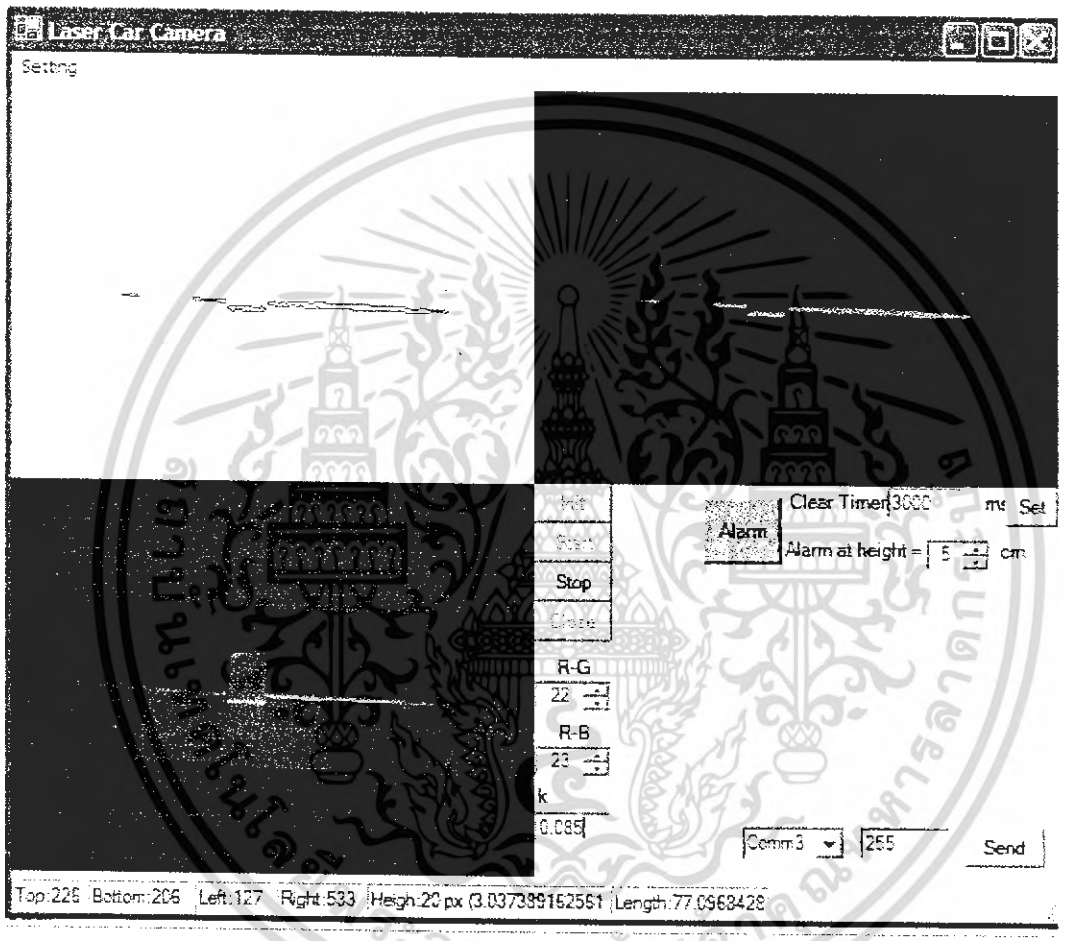
10.4.3 <u>ผลการทดลอง</u>

จากการทดลองจะเก็บภาพจากกล้องเว็บแคม ผ่านโปรแกรมตรวจสอบ Pixel และ โปรแกรมการจำกัดขอบเขตข้อมูล โปรแกรมเปรียบเทียบ ซึ่งจะได้ความสูงและระยะของวัตถุ ตามรูปที่ 10.5 – 10.8 หากวัตถุมีความสูงมากกว่าค่าทั้งตั้งไว้จะแสดง Alarm ดังรูปที่ 10.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11
สรุปและวิจารณ์ผลการทดลอง

11.1 สรุปและวิจารณ์ผลการทดลอง



รูปที่ 11.1 แสดงผลการทดลองเมื่อใช้วัตถุทดสอบที่มีความสูง 10 ซม.

<u>สรุปผลการทดลอง</u>

จากการทดลองปรับให้เส้นเลเซอร์ตัดวัตถุที่ความสูง 3 ซม. ที่ระยะ 75 ซม. โปรแกรมคำนวณค่าความสูงของวัตถุได้ประมาณ 3.03 ซม. ระยะของวัตถุประมาณ 77.09 ซม. จากการทดลองพบว่ามีค่าความผิดพลาดของโปรแกรมในการหาค่าความสูงและระยะวัตถุ เนื่องจากระนาบของกล้องเว็บเคมจะแปรผันตรงกับค่าที่คำนวณ เมื่อตั้งระนาบของกล้องผิดพลาดจะทำให้เกิดความผิดพลาดในการคำนวณค่าต่างๆ จากการทดลองค่าที่คำนวณได้ใกล้เคียงกับค่าจริงในการทดลอบ โดยมีความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดพลาดในการคำนวณความสูงของวัตถุประมาณ 1.23% และมีความผิดพลาดในการคำนวณระยะวัตถุประมาณ 2.78% ประมาณ

วิจารณ์ผลการทดลอง

ในชุดของรถทดสอบยังมีความสั่นสะเทือนในการเคลื่อนที่ของรถทดสอบ ทำให้เกิดความผิดพลาดขึ้น ประกอบกับการตั้งระนาบของกล้องเว็บแคมยังไม่ถูกต้อง

11.2 สิ่งที่ดำเนินการในภาคเรียนที่ 1

11.2.1 สร้างโครงของรถทดสอบ

- ติดตั้งอุปกรณ์ยิงลำแสงเลเซอร์ (แบบเก่า)
- ติดตั้งมอเตอร์ขับเคลื่อนและมอเตอร์บังคับเลี้ยว
- ติดตั้งกล้องเว็บแคม
- ติดตั้งไมโครสวิทช์ต่างๆ

11.2.2 ทดสอบรถทดสอบ

- สร้างวงจรพัลส์วิชมอดดูเลชั่น เพื่อทดสอบความเร็ว
- ทดลองติดตั้งและรับสัญญาณภาพจากกล้องเว็บแคม
- ทดสอบสร้างเส้นสแกนเลเซอร์

11.3 สิ่งที่ดำเนินการในภาคเรียนที่ 2

11.3.1 ส่วนของฮาร์ดแวร์

- แก้ไขอุปกรณ์ยิงลำแสงเลเซอร์
- สร้างวงจรควบคุมการเคลื่อนที่และการบังคับเลี้ยว
- ติดตั้งอุปกรณ์ทั้งหมดลงบนรถทดสอบและทดสอบการวิ่งของรถ

11.3.2 ส่วนของซอร์ฟแวร์

- สร้างโปรแกรมควบคุมของไมโครโปรเซสเซอร์ MCS 51
- สร้างโปรแกรมการทำงานของระบบนำร่องรถยนต์บนเครื่องคอมพิวเตอร์แบบพกพา
- ทดลองและทดสอบระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.4 ข้อจำกัดของโครงงาน

11.4.1 ความเร็วในการเคลื่อนที่ของรถทดสอบ โครงงานนี้หากให้รถทดสอบมีความเร็วในการเคลื่อนที่เพิ่มขึ้น จำทำให้โปรแกรมไม่สามารถอ่านค่าและคำนวณค่าต่างๆ ได้ทัน ทำให้เกิดความผิดพลาดได้

11.4.2 วัตถุที่มีความบาง หากวัตถุทดสอบมีความบางมาก จะทำให้โปรแกรมไม่สามารถตรวจสอบระยะ Pixel ได้ เปรียบเสมือนไม่มีวัตถุกีดขวางอยู่ทำให้เกิดความผิดพลาด

11.4.3 พื้นที่ทดสอบมีความสว่างมาก หากพื้นที่ทดสอบมีความสว่างมาก จะทำให้โปรแกรมมองไม่เห็นเส้นสแกนเลเซอร์ โปรแกรมก็จะหยุดทำงาน ทำให้เกิดความผิดพลาด

11.4.4 พื้นที่ทดสอบต้องไม่มีสีแดงใกล้เคียงกับเส้นสแกนเลเซอร์ ซึ่งอาจทำให้โปรแกรมทำงานผิดพลาดได้

11.5 แนวทางการปรับปรุงโครงงาน

11.5.1 ปรับปรุงการสั่นสะเทือนของรถทดสอบ ถ้ารถทดสอบมีความสั่นสะเทือนน้อย จะทำให้เกิดความผิดพลาดน้อยลง

11.5.2 ความเร็วของรถทดสอบควรมีความสัมพันธ์กับเวลาในการทำงานเป็นวงรอบของโปรแกรม

11.5.3 ควรติดตั้งให้เกิดแนวระนาบของกล้องเว็บแคมที่แน่นอนเพื่อให้จะได้สร้างโปรแกรมที่สมบูรณ์

11.5.4 เพิ่มความเร็วในการประมวลผลของโปรแกรมบนเครื่องคอมพิวเตอร์แบบพกพา ซึ่งจะทำให้รถทดสอบสามารถวิ่งได้เร็วยิ่งขึ้น

11.5.5 สร้างโปรแกรมของไมโครโปรเซสเซอร์และโปรแกรมบนเครื่องคอมพิวเตอร์แบบพกพาให้สามารถสื่อสารกันได้สองทางเพื่อความเป็นระบบออโตเมติกมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. จิรพงศ์ บัวเลิศ, " เครื่องเก็บข้อมูลภาพ 3 มิติ ", ปริญญานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปี 2546, หน้า 1 - 22

2. อรรถพล บุญยะโภคา, " เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์อนุกรม " หน้า 1 – 22

3. ชินภัทร นันทจิวากรชัย ," เอกสารประกอบการสอนวิชา ไมโครคอนโทรเลอร์ แอบปริเคชั่น" , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , หน้า 1 – 50

4. วรพจน์ กรแก้ววัฒนกุล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS – 51" หน้า 1 – 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก ก

```
1: /*---------------------------------------------------------------------------*/
2: // Program      : Car Guiding Systems
3: // Description  : All systems
4: // Filename     : MainPro
5: // C compiler   : MIDE 51 V6.1
6: /*---------------------------------------------------------------------------*/
7: #include<8051.h>        // Header include register of P89C51RD2
8: /*---------------------------------------------------------------------------*/
9: unsigned char data_series = 0;
10: sbit at 0x90 m_forward; //p1.0
11: sbit at 0x91 m_reverse; //p1.1
12: sbit at 0x92 m_right;   //p1.2
13: sbit at 0x93 m_left;    //p1.3
14: sbit at 0xa0 forward; //p2.0
15: sbit at 0xa1 reverse; //p2.1
16: sbit at 0xa2 left;    //p2.2
17: sbit at 0xa3 right;   //p2.3
18: sbit at 0xa4 select;  //p2.4
19: sbit at 0xa5 s_cen;   //p2.5
20: sbit at 0xa6 s_right; //p2.6
21: sbit at 0xa7 s_left;  //p2.7
22: sfr  at 0x80 P0; //control Speed
23:
24: /*****************************************************************************/
25: /***************************** Function Delay time ***************************/
26: /*****************************************************************************/
27: void delay(unsigned long time)
28: {
29:         do
30:         {
31:          time--;
32:         }while(time>0);
33: }
34: /*****************************************************************************/
35: /************************** Sevice Interrupt Serial Port *********************/
36: /*****************************************************************************/
37: void sevice_serial() interrupt 4
38: {
39:         if(RI)
40:         {
41:                 RI = 0;
42:                 data_series = SBUF;
43:         }
44:         if(TI)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
45:         {
46:                 TI = 0;
47:         }
48: }
49: /*****************************************************************************/
50: /********************************* Main loop *********************************/
51: /*****************************************************************************/
52: void main(void)
53: { unsigned char moving=0x0,turn=0x0,fn,fn_1,gear;
54:   unsigned char forward_1,reverse_1,right_1,left_1;
55:   TMOD = 0x21;
56:   SCON = 0x50;
57:   TH1 = 0xFA;
58:   TL1 = 0xFA;
59:   RI = 0;
60:   TI = 0;
61:   EA = 1;
62:   ES = 1;
63:   TR1 = 1;
64:   m_forward=0;
65:   m_reverse=0;
66:   m_right=0;
67:   m_left=0;
68: /*****************************************************************************
69: **********/
70: /****************************** MANUAL Function
71: ******************************************/
72: /*****************************************************************************
73: **********/
74:   while(1)
75:   {
76:    while(select==1)
77:     {
78:       if (forward==0)
79:        {
80:          m_forward=1;
81:          moving=1;
82:          P0=0x01;
83:          delay(2000);
84:          P0=0x02;
85:          delay(3000);
86:        }
87:       if (forward==1)
88:        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 89:          P0=0x01;
 90:          delay(6000);
 91:          P0=0x00;
 92:          m_forward=0;
 93:          moving=0;
 94:        }
 95:       if (reverse==0)
 96:        {
 97:          m_reverse=1;
 98:          moving=1;
 99:          P0=0x01;
100:          delay(2000);
101:          P0=0x02;
102:          delay(3000);
103:        }
104:       if (reverse==1)
105:        {
106:          P0=0x01;
107:          delay(1000);
108:          P0=0x00;
109:          delay(1000);
110:          m_reverse=0;
111:          moving=0;
112:        }
113:       if ((right==1)&&(turn==0))
114:        {
115:          m_right=1;
116:          turn=1;
117:        }
118:
119:       if ((right==0)&&(turn==1))
120:        {
121:          m_left=1;
122:        }
123:
124:       if ((right==1)&&(s_right==1))
125:        {
126:          m_right=0;
127:          turn=1;
128:        }
129:       if ((left==1)&&(turn==0))
130:        {
131:          m_left=1;
132:          turn=2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
133:         }
134:        if ((left==0)&&(turn==2))
135:         {
136:           m_right=1;
137:         }
138:        if ((left==1)&&(s_left==1))
139:         {
140:           m_left=0;
141:           turn=2;
142:         }
143:        if ((s_cen==1)&&((turn==1)||(turn==2)))
144:         {
145:           m_right=0;
146:           m_left=0;
147:           turn=0;
148:         }
149:       }
150:
151: /*****************************************************************************
152: **********/
153: /******************************** AUTO Function
154: *****************************************/
155: /*****************************************************************************
156: **********/
157:     while(select==0)
158:       {
159:        fn=data_series;
160:         {
161:          fn_1=fn&0xf0;
162:          switch(fn_1)
163:            {
164:              case 0x00  : {
165:                           forward_1=0;
166:                           reverse_1=0;
167:                           right_1=0;
168:                           left_1=0;
169:                          }break;
170:              case 0x10  : {
171:                           forward_1=1;
172:                           reverse_1=0;
173:                           right_1=0;
174:                           left_1=0;
175:                          }break;
176:              case 0x20  : {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
177:                        forward_1=0;
178:                        reverse_1=1;
179:                        right_1=0;
180:                        left_1=0;
181:                        }break;
182:            case 0x30  : {
183:                        forward_1=0;
184:                        reverse_1=0;
185:                        right_1=1;
186:                        left_1=0;
187:                        }break;
188:            case 0x40  : {
189:                        forward_1=0;
190:                        reverse_1=0;
191:                        right_1=0;
192:                        left_1=1;
193:                        }break;
194:          }
195:         if (forward_1==1) //Forward = select Gear
196:          {
197:            m_forward=1;
198:            m_reverse=0;
199:            moving=1;
200:            P0=0x01;
201:            delay(2000);
202:            P0=0x0;
203:            delay(3000);
204:          }
205:         if (forward_1==0) //Stop Forward
206:          {
207:           if (moving==1)
208:           {
209:            P0=0x0;
210:            delay(6000);
211:            m_forward=0;
212:            moving=0;
213:           }
214:          }
215:         if (reverse_1==1) //Reverse = Gear 1
216:          {
217:           m_forward=0;
218:           m_reverse=1;
219:           moving=2;
220:           P0=0x01;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
221:           delay(2000);
222:           P0=0x0;
223:           delay(3000);
224:          }
225:         if (reverse_1==0)  //Stop Reverse
226:          {
227:           P0=0x0;
228:           delay(6000);
229:           m_reverse=0;
230:           moving=0;
231:          }
232:
233:          if (left_1==1)  //Control Left
234:          {m_forward=0;
235:           m_reverse=0;
236:           while(s_left==0)
237:            {
238:             m_right=0;
239:             m_left=1;
240:             delay(300);
241:            }
242:           if (s_left==1)
243:            {
244:             m_right=0;
245:             m_left=0;
246:            }
247:           turn=2;
248:          }
249:         if (right_1==1)   //Control Right
250:          {m_forward=0;
251:           m_reverse=0;
252:           while(s_right==0)
253:            {
254:             m_left=0;
255:             m_right=1;
256:             delay(300);
257:            }
258:           if (s_right==1)
259:            {
260:             m_right=0;
261:             m_left=0;
262:            }
263:           turn=1;
264:          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
265:         if ((right_1==0)&&(turn==1)) //Notcontrol Right
266:          { m_forward=0;
267:            m_reverse=0;
268:            while(s_cen==0)
269:             {
270:              m_right=0;
271:              m_left=1;
272:             }
273:            if (s_cen==1)
274:             {
275:              m_right=0;
276:              m_left=0;
277:              turn=0;
278:             }
279:          }
280:         if ((left_1==0)&&(turn==2))
281:          { m_forward=0;
282:            m_reverse=0;
283:            while(s_cen==0)
284:             {
285:              m_right=1;
286:              m_left=0;
287:             }
288:            if (s_cen==1)
289:             {
290:              m_right=0;
291:              m_left=0;
292:              turn=0;
293:             }
294:           }
295:         }
296:      }
297:     }
298:    }
299:
300:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก ข

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 1

```
#pragma once


namespace camcar00
{
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Threading;

    /// <summary>
    /// Summary for Form1
    ///
    /// WARNING: If you change the name of this class, you will need to ↙
    change the
    ///          'Resource File Name' property for the managed resource ↙
    compiler tool
    ///          associated with all .resx files this class depends on. ↙
     Otherwise,
    ///          the designers will not be able to interact properly ↙
    with localized
    ///          resources associated with this form.
    /// </summary>
    public __gc class Form1 : public System::Windows::Forms::Form
    {
    private:
        int imageHandle;

        bool running;
        bool clearing;
        bool drawing;
        Thread *cameraThread;

        Bitmap *lineBitmap;
        Bitmap *objectBitmap;

    private: System::Windows::Forms::MainMenu *  mainMenu1;
    private: System::Windows::Forms::MenuItem *  mnSetting;
    private: System::Windows::Forms::MenuItem *  mnSettingVideoFormat;
    private: System::Windows::Forms::MenuItem *  mnSettingVideoSource;
    private: AxInterop::VIDEOOCXLib::AxVideoOCX *  ShowLine;
    private: AxInterop::VIDEOOCXLib::AxVideoOCX *  ShowObject;

    private: System::Windows::Forms::Label *  label1;

    private: System::Windows::Forms::Label *  label2;
    private: System::Windows::Forms::NumericUpDown *  nudRG;
    private: System::Windows::Forms::NumericUpDown *  nudRB;
    private: System::Windows::Forms::TextBox *  tb;
    private: AxInterop::MSCommLib::AxMSComm *  Comm;
    private: System::Windows::Forms::Button *  button1;
    private: System::Windows::Forms::ComboBox *  cbCommPort;
    private: System::Windows::Forms::Timer *  timer1;
    private: System::Windows::Forms::StatusBar *  sb;


    private: System::Windows::Forms::Label *  label3;
    private: System::Windows::Forms::TextBox *  tbInterval;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรแจ็ก\camcar00ข\Form1.h 2

```
    private: System::Windows::Forms::StatusBarPanel *  sbp1;
    private: System::Windows::Forms::StatusBarPanel *  sbp2;
    private: System::Windows::Forms::StatusBarPanel *  sbp3;
    private: System::Windows::Forms::StatusBarPanel *  sbp4;
    private: System::Windows::Forms::Button *  button2;
    private: System::Windows::Forms::StatusBarPanel *  sbp5;
    private: System::Windows::Forms::StatusBarPanel *  sbp6;
    private: System::Windows::Forms::TextBox *  tbK;
    private: System::Windows::Forms::Label *  label4;

    private: System::Windows::Forms::Label *  label5;

    private: System::Windows::Forms::Button *  btAlarm;
    private: System::Windows::Forms::NumericUpDown *  nudHA;


    private: System::Windows::Forms::MenuItem *  mnSettingVideoDriver;




    public:
        Form1(void)
            : imageHandle(0)
            ,running(false)
            ,cameraThread(NULL)
            ,clearing(true)
            ,drawing(false)
        {
            InitializeComponent();
        }

    protected:
        void Dispose(Boolean disposing)
        {
            if(running)
            {
                running = false;
                cameraThread->Join();
                Camera->Stop();
                Camera->Close();
                Camera->ReleaseImageHandle(imageHandle);
            }
            if (disposing && components)
            {
                components->Dispose();
            }
            __super::Dispose(disposing);
        }



    private: System::Windows::Forms::Button *  btInit;
    private: System::Windows::Forms::Button *  btStart;
    private: System::Windows::Forms::Button *  btStop;
    private: System::Windows::Forms::Button *  btClose;
    private: AxInterop::VIDEOOCXLib::AxVideoOCX *  Camera;
    private: System::ComponentModel::IContainer *  components;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 3

```
    private:
        /// <summary>
        /// Required designer variable.
        /// </summary>


        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        void InitializeComponent(void)...
    private: System::Void btInit_Click(System::Object *  sender, System: ↙
    :EventArgs *  e)
             {
                if(imageHandle)
                    return;

                if(!Camera->Init())
                {
                    MessageBox::Show(Camera->GetLastErrorString(),        ↙
    "VideoOCX Error");
                    return;
                }

                imageHandle = Camera->GetColorImageHandle();

                Camera->SetPreview(true);

                btStart->Enabled = true;

                btInit->Enabled = false;
                btClose->Enabled = true;
             }

private: System::Void btStart_Click(System::Object *  sender, System::   ↙
    EventArgs *  e)
         {
             if(!imageHandle)
                 return;

            if(!Camera->Start())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX  ↙
    Error");
                return;
            }

            running = true;
            cameraThread = new Thread(new ThreadStart(this,               ↙
    CaptureThread));
            cameraThread->Start();

            btStart->Enabled = false;

            btStop->Enabled = true;
            btClose->Enabled = false;
        }

private: System::Void btStop_Click(System::Object *  sender, System::    ↙
    EventArgs *  e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            running = false;
            cameraThread->Join();

            if(!Camera->Stop())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX Error");
                return;
            }

            btStop->Enabled = false;

            btStart->Enabled = true;
            btClose->Enabled = true;
        }

private: System::Void btClose_Click(System::Object *  sender, System::EventArgs *  e)
        {
            if(!imageHandle)
                return;

            if(!Camera->Close())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX Error");
                return;
            }

            Camera->ReleaseImageHandle(imageHandle);
            imageHandle = 0;

            btClose->Enabled = false;

            btInit->Enabled = true;
            btStart->Enabled = false;
        }

private: System::Void CaptureThread(void)
        {
            int width = Camera->GetImageWidth(imageHandle);
            int height = Camera->GetImageHeight(imageHandle);
            int size = Camera->GetImageDataSize(imageHandle);

            unsigned char *  imagePtr = (unsigned char *)Camera->GetDataPointer(imageHandle);

            Camera->SetPreview(false);

            Camera->CtlCapture(imageHandle);
            Camera->AVISaveFrameInit("temp.avi",imageHandle,false);
            Camera->AVISaveFrameAdd(imageHandle);
            Camera->AVISaveFrameClose();


            ShowLine->Mode = 1;
            ShowLine->InitFromFile("temp.avi");
            int lineHandle = ShowLine->GetColorImageHandle();
            unsigned char *  linePtr = (unsigned char *)ShowLine->GetDataPointer(lineHandle);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 5

```
        ShowObject->Mode = 1;
        ShowObject->InitFromFile("temp.avi");
        int objectHandle = ShowObject->GetColorImageHandle();
        unsigned char *  objectPtr = (unsigned char *)ShowObject->
GetDataPointer(objectHandle);

        int * tops = new int[width];
        int * bottoms = new int[width];

        while(running)
         {
             if(!imageHandle)
                 continue;


             int top=0;
             int bottom=height;

             int highest = 0;
             int highestcolumn = 0;
             int left = width*75/100;
             int right = width*25/200;

             Camera->CtlCapture(imageHandle);

            unsigned char * ptr = imagePtr;
            unsigned char * lptr = linePtr;
            unsigned char * optr = objectPtr;

            for(int i = 0; i < width; i++)
            {
                tops[i] = 0;
                bottoms[i] = height;
            }
            //memset(tops,0,width);
            //memset(bottoms,height,width);

            for(int j = 0; j < height; j++)
             {
                 for(int i = 0; i < width; i++)
                 {
                     unsigned char *r = ptr+2;
                     unsigned char *g = ptr+1;
                     unsigned char *b = ptr;

                     if((*r > *g && *r > *b) && ((*r - *g > nudRG->
Value) || (*r - *b > nudRB->Value)))
                     {
                         if(tops[i] < j)
                             tops[i] = j;
                         if(bottoms[i] > j)
                             bottoms[i] = j;

                         if(top < j)
                             top = j;
                         if(bottom > j)
                             bottom = j;

                     }
                     else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 6

```
                    {
                    }

                    ptr+=3;
                }
            }

            for(i = 0; i < width; i++)
            {
                if(highest < top - bottoms[i])
                {
                    highest = top - bottoms[i];
                    highestcolumn = i;
                }
                if(i < width-2)
                {
                    if((top - bottoms[i]) - (top - bottoms[i+1]) > 5 ↙
 && right < i && tops[i] != height)
                    {
                        right = i;
                    }

                    if((top - bottoms[i+1]) - (top - bottoms[i]) > 5 ↙
 && left > i && tops[i] != height)
                    {
                        left = i;
                    }

                }
            }

            if(left == width)
                left = 0;
            if(right == 0)
                right = width;

            sbp1->Text = String::Format("Top:{0}",top.ToString());
            sbp2->Text = String::Format("Bottom:{0}",bottom.        ↙
ToString());
            sbp3->Text = String::Format("Left:{0}",left.ToString()) ↙
;
            sbp4->Text = String::Format("Right:{0}",right.ToString  ↙
());

            double k = Convert::ToDouble(tbK->Text);
            double hcm = ((double)(highest-2)*k/0.203)*0.403;
            double lcm = 84-(hcm/0.44);

            if(hcm > Convert::ToDouble(nudHA->Value))
            {
                btAlarm->BackColor = Color::Red;
            }
            else
            {
                btAlarm->BackColor = Color::LightGreen;
            }

            sbp5->Text = String::Format("Heigh:{0} px ({1:0.00} cm) ↙
",highest.ToString(),hcm.ToString());
            sbp6->Text = String::Format("Length:{0:0.00} cm",lcm.   ↙
ToString());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00๒\Form1.h 7

```
                ptr = imagePtr;
                lptr = linePtr;
                optr = objectPtr;

                for(j = 0; j < height; j++)
                 {
                     for(int i = 0; i < width; i++)
                     {
                        unsigned char *r = ptr+2;
                        unsigned char *g = ptr+1;
                        unsigned char *b = ptr;
                        unsigned char *lr = lptr+2;
                        unsigned char *lg = lptr+1;
                        unsigned char *lb = lptr;
                        unsigned char *or = optr+2;
                        unsigned char *og = optr+1;
                        unsigned char *ob = optr;

                        if((left !=0 || right != width) && !drawing)
                        {
                            if(i == width-1 && j == height-1)
                                drawing = true;
                            *or = 0;
                            *og = 0;
                            *ob = 0;
                        }

                        if(clearing && !drawing)
                        {
                            *or = 0;
                            *og = 0;
                            *ob = 0;
                        }

                        if((*r > *g && *r > *b) && ((*r - *g > nudRG->
    Value) || (*r - *b > nudRB->Value)))
                        {
                            *lr = 255;
                            *lg = 200;
                            *lb = 200;
                        }
                        else
                        {
                            *lr = 255;
                            *lg = 255;
                            *lb = 255;
                        }

                        if((*r > *g && *r > *b) && ((*r - *g > nudRG->
    Value) || (*r - *b > nudRB->Value)))// && i >= left && i <= right)
                        {
/*                          if(*or <= 255-16)
                                *or += 16;
                            if(*og <= 255-16)
                                *og += 16;
                            if(*ob <= 255-16)
                                *ob += 16;
)*/
                            *or = 255;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }

                    if(j == top)
                    {
                        *lr = 200;
                        *lg = 200;
                        *lb = 200;
                    }
                    if(j == bottom)
                    {
                        *lr = 200;
                        *lg = 200;
                        *lb = 200;
                    }
                    if(i == highestcolumn)
                    {
                        *lr = 200;
                        *lg = 200;
                        *lb = 200;
                    }
                    if(i == right)
                    {
                        *lr = 200;
                        *lg = 200;
                        *lb = 200;
                    }
                    if(i == left)
                    {
                        *lr = 200;
                        *lg = 200;
                        *lb = 200;
                    }

                    if(j == tops[i] && j != 0)
                    {
                        *lr = 0;
                        *lg = 0;
                        *lb = 0;
                    }
                    if(j == bottoms[i] && j != 0)
                    {
                        *lr = 0;
                        *lg = 0;
                        *lb = 0;
                    }

                    ptr+=3;
                    lptr+=3;
                    optr+=3;
                }

            }
            clearing = false;

            Camera->CtlShow(imageHandle);
            ShowLine->CtlShow(lineHandle);
            ShowObject->CtlShow(objectHandle);
        }

        delete tops;
        delete bottoms;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 9

```
            Camera->SetPreview(true);

            ShowLine->ReleaseImageHandle(lineHandle);
            ShowObject->ReleaseImageHandle(objectHandle);
            ShowLine->Close();
            ShowObject->Close();

        }

private: System::Void mnSettingVideoFormat_Click(System::Object *
    sender, System::EventArgs *  e)
        {
            if(!imageHandle)
                return;
            if(running)
                return;

            if(!Camera->ShowFormatDlg())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX
    Error");
                return;
            }

            btClose_Click(sender,e);
            btInit_Click(sender,e);

        }

private: System::Void mnSettingVideoSource_Click(System::Object *
    sender, System::EventArgs *  e)
        {
            if(!imageHandle)
                return;
            if(running)
                return;

            if(!Camera->ShowSourceDlg())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX
    Error");
                return;
            }
        }

private: System::Void mnSettingVideoDriver_Click(System::Object *
    sender, System::EventArgs *  e)
        {
            if(!Camera->ShowDriverDlg())
            {
                MessageBox::Show(Camera->GetLastErrorString(),"VideoOCX
    Error");
                return;
            }
        }

private: System::Void button1_Click(System::Object * sender, System::
    EventArgs *  e)
        {
            char ch = Convert::ToByte(tb->Text)%255;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ก\camcar00ข\Form1.h 10

```
            String *temp = Convert::ToString(&ch);

            Comm->PortOpen = true;
            Comm->Output = temp;
            Comm->PortOpen = false;
        }


private: System::Void cbCommPort_SelectedIndexChanged(System::Object *
    sender, System::EventArgs *  e)
        {
            Comm->CommPort = cbCommPort->SelectedIndex+1;
        }

private: System::Void timer1_Tick(System::Object *  sender, System::
    EventArgs *  e)
        {
            clearing = true;
            drawing = false;
        }

private: System::Void button2_Click(System::Object *  sender, System::
    EventArgs *  e)
        {
            timer1->Interval = Convert::ToInt32(tbInterval->Text);
        }

private: System::Void tbK_TextChanged(System::Object *  sender, System::
    EventArgs *  e)
        {
        }

};
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C:\Documents and Settings\Administrator\Desktop\โปรเจ็ค\camcar00ข\stdafx.h 1

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
#pragma once


#define WIN32_LEAN_AND_MEAN      // Exclude rarely-used stuff from
    Windows headers
// C RunTime Header Files
#include <stdlib.h>
#include <malloc.h>
#include <memory.h>
#include <tchar.h>

// TODO: reference additional headers your program requires here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้